

Florian Schuchter

Methoden des maschinellen Lernens in der Funktionsentwicklung von hybriden Fahrzeuggetrieben am Beispiel des Doppelkupplungsgetriebes

Machine Learning Methods in Function Development of Hybrid Transmissions by the Example of the Dual Clutch Transmission

Band 180

Systeme ■ Methoden ■ Prozesse

Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers
Univ.-Prof. Dr.-Ing. S. Matthiesen
(Hrsg.)

Forschungsberichte



Florian Schuchter

Methoden des maschinellen Lernens in der Funktionsentwicklung von hybriden Fahrzeuggetrieben am Beispiel des Doppelkupplungsgetriebes

Machine Learning Methods in Function Development of Hybrid Transmissions by the Example of the Dual Clutch Transmission

Band 180

Herausgeber Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers
Univ.-Prof. Dr.-Ing. S. Matthiesen

| | |
|-----------|---|
| Copyright | IPEK • Institut für Produktentwicklung, 2024 Karlsruher Institut für Technologie (KIT) Die Forschungsuniversität in der Helmholtz-Gemeinschaft Alle Rechte vorbehalten |
| Druck | Stolzenberger Druck und Werbung GmbH & Co. KG, Leimen 06224-7697915 |
| ISSN | 1615-8113 |

Methoden des maschinellen Lernens in der Funktionsentwicklung von hybriden Fahrzeuggetrieben am Beispiel des Doppelkupplungsgetriebes

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Florian Schuchter

| | |
|-----------------------------|--|
| Tag der mündlichen Prüfung: | 11.12.2024 |
| Hauptreferent | Univ. Prof. Dr.-Ing. Dr. h. c. Albert Albers |
| Korreferent | Univ. Prof. Dr.-Ing. Hans-Christian Reuss |

Vorwort der Herausgeber

Wissen ist einer der entscheidenden Faktoren in den Volkswirtschaften unserer Zeit. Der Unternehmenserfolg wird mehr denn je davon abhängen, wie schnell ein Unternehmen neues Wissen aufnehmen, zugänglich machen und verwerten kann. Die Aufgabe eines Universitätsinstitutes ist es, hier einen wesentlichen Beitrag zu leisten. In den Forschungsarbeiten wird ständig Wissen generiert. Dieses kann aber nur wirksam und für die Gemeinschaft nutzbar werden, wenn es in geeigneter Form kommuniziert wird. Diese Schriftenreihe dient seit mehr als 20 Jahren als eine Plattform zum Transfer und macht damit das Wissenspotenzial aus aktuellen Forschungsarbeiten am IPEK - Institut für Produktentwicklung Karlsruhe* am Karlsruher Institut für Technologie (KIT) verfügbar. Die Forschung des IPEK ist dabei strukturiert in die Kategorien Systeme, Methoden und Prozesse, um so der Komplexität heutiger Produktentwicklung ganzheitlich gerecht zu werden. Erst die Verknüpfung dieser drei Kategorien ermöglicht die Synthese innovativer Systeme durch Nutzung neuester Methoden und Prozesse. Gleichzeitig werden durch die Systemsynthese die erforschten neuen Methoden und Prozesse validiert und deren Mehrwert für die Praxis abgesichert. Dieses Forschungskonzept prägt nicht nur das IPEK-Leitbild, sondern auch den Charakter dieser Schriftenreihe, da immer alle drei Kategorien und deren Wechselwirkungen berücksichtigt werden. Jeder Band setzt hier individuelle Schwerpunkte und adressiert dabei folgende Forschungsgebiete des IPEK:

- das Entwicklungs- und Innovationsmanagement,
- die Entwicklungs- und Konstruktionsmethodik,
- der Leichtbau von der Ebene des ganzen Systems bis hinunter zur Optimierung des Bauteils,
- die Validierung technischer Systeme auch unter Berücksichtigung der NVH Aspekte (Noise, Vibration, Harshness) mit dem Fokus auf Schwingungen und Akustik an Komponenten und in den Gesamtsystemen sowie deren subjektiver Beurteilung durch den Menschen,
- die Antriebssystemtechnik mit den Schwerpunkten komplette Antriebslösungen für Fahrzeuge und Maschinen,
- das Design, die Tribologie und Erprobung von Kupplungen und Bremsen sowie
- die Gerätetechnik mit dem Schwerpunkt auf Power-Tools.

Die Forschungsberichte stellen Ergebnisse unserer Forschung sowohl anderen Wissenschaftlern als auch den Unternehmen zu Verfügung, um damit die Produktentwicklung in allen ihren Facetten mit innovativen Impulsen zu optimieren.

Albert Albers und Sven Matthiesen

* Eh.: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau, Universität Karlsruhe (TH)

Vorwort zu Band 180

Die individuelle Mobilität ist eine zentrale Errungenschaft unserer Gesellschaft und hat zu deren Entwicklung, auch aus historischer Sicht, immens beigetragen. Durch die Möglichkeiten der individuellen Mobilität können Gesellschaften die Arbeitsteilung, Versorgungsteilung und auch die kulturelle Interaktion ganz wesentlich positiv beeinflussen. Unter den heutigen Randbedingungen der Umweltherausforderungen, insbesondere unter dem Stichwort CO₂ und Klimaerwärmung, bedeutet es aber, dass wir die technischen Lösungen für die individuelle Mobilität in den nächsten Jahren drastisch unter dem Gesichtspunkt der CO₂ Emission weiter verbessern müssen. Auf diesem Weg wurden in den letzten Jahren schon erhebliche Fortschritte erreicht. Ein zentraler Punkt ist hierbei die Technologie, da es nicht genau eine Lösung für alle Fälle gibt. Die wesentlichen Antriebskonzepte und Architekturen, die verfolgt werden, sind für den PKW-Bereich der batterieelektrische Antrieb – BEV, der elektrische Antrieb mit einer Brennstoffzelle – BZV, die Plug-In-Hybridlösungen aus Kombinationen von Verbrennungsmotor und elektromotorischem Antrieb, sowie die hoch optimierten klassischen Verbrennungsmotor-Lösungen mit Benzin- oder Dieselmotor. Betrachtet man die sehr heterogenen Randbedingungen zum Betrieb von Fahrzeugen in einem globalen Maßstab, so wird klar, dass es hier nicht „eine“ Lösung geben kann. In den hoch entwickelten Ländern – und dort insbesondere im urbanen Bereich – bieten sich die batterieelektrischen Fahrzeuge an. Voraussetzung ist dabei natürlich, dass der genutzte Strom klimaneutral erzeugt wird. Für größere Fahrzeuge und solche im Langstreckenbetrieb führt die dann notwendige Reichweite zu sehr großen und vor allen Dingen schweren Batterien. Hier sind als Alternative das Elektrofahrzeug mit Brennstoffzelle, sowie der Plug-In-Hybrid als Lösung ins Auge zu fassen. Insbesondere bei den Plug-In-Hybrid-Fahrzeugen hat sich in den letzten Jahren eine erhebliche Weiterentwicklung gezeigt. Während anfänglich elektrische Reichweiten von meist weniger als 30 Kilometern möglich waren, werden in den modernen Architekturen typischerweise 80, 100 oder gar 120 Kilometer elektrische Reichweite erreicht. Diese Entwicklung ist absolut sinnvoll, da dann im urbanen Bereich, oder auch im Kurzstreckenumfeld, rein elektrisch gefahren werden kann und für die längeren Strecken dann der optimierte Verbrennungsmotor mit seinem deutlich im Leistungsgewicht besseren Energiespeicher, nämlich dem Kraftstofftank, genutzt werden kann. Die Nutzung des Verbrennungsmotors muss dann aber spätestens mittelfristig natürlich mit klimaneutral erzeugten Kraftstoffen erfolgen! Die modernen Plug-In-Antriebssysteme stellen komplexe mechatronische Systeme dar, die sich mittlerweile sogar hin zu cyberphysischen Systemen entwickeln, da man über den „Over the Air“-Zugriff und die Vernetzung mit der Infrastruktur den Antriebsstrang in ein Informationsnetzwerk zum Verkehrssystem einbindet. Eine solche Antriebsarchitektur beinhaltet typischerweise immer auch ein mechanisches, mehrgängiges Getriebe, das sowohl auf den Betrieb mit dem Verbrennungsmotor als auch gegebenenfalls auf den Betrieb mit dem Elektromotor abgestimmt werden muss. Dieser Vorgang, der mit Begriff Funktionsentwicklung beschrieben wird, ist aber äußerst komplex,

da eine sehr feingranulare Abstimmung der verschiedenen Parameter der Antriebsarchitektur notwendig ist und gleichzeitig der Abgleich der Funktionen mit den Umweltrandbedingungen, die mit Sensorik erfasst werden, geleistet werden muss. Die Untersuchung dieses Prozesses der Funktionsentwicklung von hybriden Fahrzeugantrieben stellt sich in seiner Arbeit Herr Dr.-Ing. Florian Schuchter mit dem Ziel, durch die Nutzung von Methoden des maschinellen Lernens den Entwicklungsprozess für die Funktionsentwicklung zu unterstützen, sicherer zu machen und auch zu beschleunigen. Mit den von Herrn Dr.-Ing. Florian Schuchter in seiner Arbeit erforschten und als Methoden beschriebenen Lösungen hat er einen wichtigen wissenschaftlichen und auch praxisrelevanten Beitrag für die Anwendung des maschinellen Lernens in einem hoch aktuellen und anspruchsvollen Themenfeld der Fahrzeugentwicklung geleistet.

September, 2024

Albert Albers

Kurzfassung

In dieser Arbeit werden drei Vorgehensmodelle für den Einsatz der zentralen Methoden des maschinellen Lernens - überwachtes Lernen, nicht überwachtes Lernen und verstärkendes Lernen - in der Funktionsentwicklung von hybriden Fahrzeuggetrieben am Beispiel des Doppelkupplungsgetriebes erarbeitet.

Zunächst werden auf Basis von Experteninterviews und einer Systemanalyse die Herausforderungen in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben abgeleitet. Die Herausforderungen umfassen die Fahrfunktionen Motorwiederstart, hybrides Anfahren sowie die hohe Komplexität des Antriebsstrangs, die zu einer komplexen Auslegung der Steuerung und Regelung sowie der Modellierung von virtuellen Sensoren führt.

Für die Methodenentwicklung werden die drei zentralen Methoden des maschinellen Lernens an den Kontext der Funktionsentwicklung paralleler hybrider Fahrzeuggetriebe adaptiert und daraus drei konkrete Vorgehensmodelle abgeleitet. Alle drei Methoden bieten die Vorteile der Automatisierung, Skalierbarkeit und einfachen Implementierung bei gleichzeitig hoher Funktionalität durch vordefinierte Bibliotheken der Programmiersprache Python.

Die drei Methoden werden jeweils an einer praxisrelevanten Entwicklungsaufgabe der Mercedes-Benz Group AG evaluiert. Mit Hilfe der Methode überwachtes Lernen werden verschiedene Kupplungstemperaturmodelle eines Hybridgetriebes auf Basis von Fahrzeugmessdaten modelliert. Die Abbildungsgüte wird gegenüber den derzeit verwendeten Modellen um bis 15,5 % zu 22,5 % verbessert bei nur 30 % bzw. 61 % der Modellparameter. Die Methode des verstärkenden Lernens wird auf den Kupplungsfüllvorgang angewendet. Die Zielgröße Druckeinbruch konnte mit einer parameterfreien Ansteuerung um bis zu 86 % bei gleicher Füllzeit im Vergleich zur Ansteuerung eines Entwickelnden reduziert werden. Die Methode des nicht überwachten Lernens identifiziert aus 278 detektierten Motorstarts vier Motorstart-Cluster und charakterisiert diese über Kennzahlen. Darüber hinaus identifiziert die Methode Verbesserungspotentiale der Applikation und kann somit Entwicklungspotentiale aufzeigen.

Um die Anwendbarkeit der entwickelten Methoden in der Praxis zu untersuchen, werden drei Studien in der Forschungsumgebung durchgeführt. Zwei Methoden werden anhand einer Befragungsstudie evaluiert, für die Methode des überwachten Lernens wird eine Anwenderstudie mit einem Software-Tool vorgenommen. Alle drei Studien bestätigen das Methodenverständnis und die Methodenakzeptanz und zeigen einzelne Verbesserungspotentiale auf.

Abstract

In this work, three process models for the application of central machine learning methods - supervised learning, unsupervised learning and reinforcement learning - are developed in the function development of hybrid vehicle transmissions using the example of the dual clutch transmission.

First, the challenges in the function development of parallel hybrid vehicle transmissions are derived based on a literature review and a system analysis. The challenges can be related to the driving functions engine restart, hybrid start-up as well as the high complexity of the powertrain, which leads to a complex design of the control system and modelling of sensor models.

For method development, the already existing machine learning methods from the state of the art are supplemented with concrete procedure models and a description of the selection of complex algorithms and optimization procedures for implementing the methods. All three methods presented can be described via the extended ZHO triple of product development. Furthermore, they offer the common advantages of automation, scaling and ease of implementation while providing high functionality through predefined libraries of the Python programming language.

The three methods are evaluated using a practical development example from Mercedes-Benz Group AG. With the help of the supervised learning method, different clutch temperature models of a hybrid transmission are modelled based on vehicle measurement data. The mapping quality is improved by up to 15,5 % to 22,5 % with 30 % to 61 % of the model parameters compared to currently used models. The reinforcement learning method is applied to the clutch filling process. The target value of pressure drop could be reduced by up to 86 % with parameter-free control while maintaining the same filling time compared to the control of a developer. The unsupervised learning method successfully identifies four motor start classes from 278 detected motor starts and characterizes them via key figures. Furthermore, the method identifies potential for improvement of the application and can thus show development potentials.

To investigate the applicability of the developed methods in practice, three studies are carried out in the research environment. Two methods are evaluated based on a survey study, for the method of supervised learning a user study is carried out with a software tool. All three studies confirm the understanding and acceptance of methods and show individual potential for improvement.

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Promotion am IPEK – Institut für Produktentwicklung (IPEK) am Karlsruher Institut für Technologie (KIT) und wurde bei der Mercedes-Benz Group AG innerhalb der Entwicklungsabteilung „Funktionsentwicklung und Applikation Front-Antriebe“ im Entwicklungszentrum in Sindelfingen durchgeführt.

Zunächst möchte ich mich bei meinem Doktorvater Herrn Univ.-Prof. Dr.-Ing. Dr. h.c. Albert Albers für die Anregungen, Impulse und Diskussionen zu meiner Arbeit bedanken. Vor allem bedanke ich mich für die immer positive, wertschätzende und stets professionelle Zusammenarbeit.

Mein Dank gilt auch Herrn Prof. Dr.-Ing. Hans-Jürgen Reuss für die Übernahme des Korreferats und dem damit verbundenen Interesse an meiner Forschungsarbeit.

Danken möchte ich dem Forschungs-Team Antriebssystemtechnik des IPEK, allem voran Katharina Bause und Adrian Braumandl für Anregungen und Diskussion meiner Arbeit. Natürlich danke ich auch dem gesamten IPEK für die tolle Arbeitsatmosphäre und die rege Diskussion und Austausch in den Forschungsgesprächen und bei den IPEK Think Days.

Für die Initiierung des Themas der Arbeit und die fachliche Betreuung von Seiten der Mercedes-Benz Group AG danke ich ganz besonders Herrn Dr. Anton Rink, sowie Dr. Oliver Amendt für die weitere Übernahme der Betreuung. Von eurer jahrzehntelangen Erfahrung in der Getriebeentwicklung konnte ich immer profitieren.

Für die fachliche Diskussion an meiner Arbeit danke ich Thomas Krais und Philipp Schrapel, sowie alle weiteren Teamkollegen, die immer ein offenes Ohr hatten und starkes Interesse an meiner Arbeit gezeigt haben. Für die fachliche Diskussion und Unterstützung im Themengebiet der Künstlichen Intelligenz bei der Mercedes-Group AG gilt mein besonderer Dank Stefan Altenburg, Katja Deuschl, Dr. Gerd Gaiselmann, Thomas Doster und Stefan Studer.

Darüber hinaus danke ich meiner Familie, vor allem meinen Eltern für die Unterstützung während meines gesamten Studiums und meiner Zeit als Doktorand. Mein größter Dank gilt meiner Partnerin Melanie. Du hast mich durch alle Höhen und Tiefen der Arbeit begleitet, auf mich Rücksicht genommen und hattest immer ein offenes Ohr für mich. Dafür bin ich dir sehr dankbar.

September, 2024

Florian Schuchter

„The best way to predict the future is to invent it“
Allan Kay, Informatiker.

Inhaltsverzeichnis

| | |
|---|--------------|
| Kurzfassung..... | i |
| Abstract..... | iii |
| Abbildungsverzeichnis..... | xiii |
| Tabellenverzeichnis..... | xix |
| Abkürzungsverzeichnis..... | xxi |
| Formelzeichen..... | xxiii |
| 1 Einleitung und Motivation | 1 |
| 2 Grundlagen und Stand der Forschung..... | 3 |
| 2.1 Paralleler Vollhybrid-Antrieb für Kraftfahrzeuge..... | 3 |
| 2.1.1 Fahrfunktionen und Topologie | 3 |
| 2.1.2 Nasslaufende Lamellenkupplung | 7 |
| 2.1.3 Steuerung und Regelung von Automatikgetrieben..... | 9 |
| 2.1.4 Auslegung von Steuer- und Regelvorgängen | 15 |
| 2.1.5 Modellbildung | 16 |
| 2.1.6 Methoden zur Datenanalyse..... | 18 |
| 2.1.7 Zwischenfazit..... | 19 |
| 2.2 Relevante Prozesse und Methoden der KaSPro | 19 |
| 2.2.1 Modell der SGE- Systemgenerationsentwicklung nach Albers..... | 20 |
| 2.2.2 Verifikation und Validierung im Produktentstehungsprozess | 22 |
| 2.2.3 IPEK-X-in-the-Loop-Ansatz (IPEK-XiL-Ansatz)..... | 25 |
| 2.2.4 Contact und Channel Ansatz (C&C ² -Ansatz) | 28 |
| 2.2.5 Zwischenfazit..... | 29 |
| 2.3 Maschinelles Lernen (ML) | 30 |
| 2.3.1 Überwachtes Lernen..... | 31 |
| 2.3.2 Nicht überwachtes Lernen | 39 |
| 2.3.3 Verstärkendes Lernen | 45 |
| 2.3.4 Zwischenfazit..... | 49 |
| 2.4 Einsatz von maschinellem Lernen in Fahrzeuggetrieben..... | 49 |
| 2.4.1 Konventionelle Fahrzeuggetriebe | 50 |
| 2.4.2 Hybride Fahrzeuggetriebe | 52 |
| 2.4.3 Zwischenfazit..... | 54 |
| 3 Forschungsbedarf und Zielsetzung | 55 |
| 3.1 Forschungsbedarf | 55 |
| 3.2 Zielsetzung..... | 56 |

| | | |
|----------|--|-----------|
| 4 | Forschungsvorgehen..... | 58 |
| 4.1 | Forschungsmethode | 58 |
| 4.2 | Empirische Methoden | 59 |
| 4.3 | Forschungsumgebung: Mercedes-Benz Group AG | 61 |
| 4.4 | Vorgehensweise und Forschungsdesign | 62 |
| 5 | Herausforderungen in der Getriebefunktionsentwicklung von parallelen hybriden Fahrzeuggetrieben | 64 |
| 5.1 | Experteninterviews..... | 65 |
| 5.2 | Systemanalyse | 71 |
| 5.2.1 | Fahrfunktionen | 71 |
| 5.2.2 | Getriebesystem | 74 |
| 5.2.3 | Kupplungsansteuerung | 77 |
| 5.3 | Zwischenfazit | 83 |
| 6 | Methoden des maschinellen Lernens im Kontext der Getriebefunktionsentwicklung..... | 85 |
| 6.1 | Bewertung der existierenden Methodennutzung in Fahrzeuggetrieben | 86 |
| 6.2 | Übergreifende Zielsysteme der Methoden | 87 |
| 6.3 | Modellierung von virtuellen Sensoren mit Künstlichen Neuronalen Netzen | 91 |
| 6.3.1 | Zielsystem..... | 92 |
| 6.3.2 | Datengenerierung | 94 |
| 6.3.3 | Definition der Modelleingangsgrößen | 95 |
| 6.3.4 | Datenaufbereitung | 96 |
| 6.3.5 | Modellerstellung | 98 |
| 6.3.6 | Erklärbarkeitsansätze..... | 101 |
| 6.3.7 | Modellvalidierung | 102 |
| 6.3.8 | Anwendung und Übertragbarkeit..... | 102 |
| 6.4 | Steuerungs-Optimierung mit verstärkendem Lernen..... | 103 |
| 6.4.1 | Optimierungsziel | 104 |
| 6.4.2 | Definition Aktions- und Zustandsraum..... | 105 |
| 6.4.3 | Definition Belohnungsfunktion..... | 105 |
| 6.4.4 | Definition Trainingsumgebung..... | 106 |
| 6.4.5 | Training..... | 108 |
| 6.4.6 | Validierung..... | 109 |
| 6.4.7 | Anwendung und Übertragbarkeit..... | 109 |
| 6.5 | Messdatenanalyse mit Cluster-Algorithmen | 111 |
| 6.5.1 | Testergebnisse | 112 |
| 6.5.2 | Datenspezifikation und -aufbereitung | 113 |
| 6.5.3 | Ereigniserkennung | 114 |
| 6.5.4 | Kennzahldefinition..... | 114 |

| | | |
|----------|--|------------|
| 6.5.5 | Korrelationsanalyse | 115 |
| 6.5.6 | Clustering | 116 |
| 6.5.7 | Visualisierung | 117 |
| 6.5.8 | Verifikation und Validierung | 117 |
| 6.5.9 | Anwendung und Übertragbarkeit | 118 |
| 6.6 | Zwischenfazit..... | 119 |
| 7 | Anwendung der Methoden | 122 |
| 7.1 | Kupplungstemperaturmodell..... | 123 |
| 7.1.1 | Zielsystem | 126 |
| 7.1.2 | Datengenerierung..... | 128 |
| 7.1.3 | Definition der Modelleingangsgrößen | 129 |
| 7.1.4 | Datenaufbereitung | 131 |
| 7.1.5 | Modellerstellung | 132 |
| 7.1.6 | Erklärbarkeitsansätze | 134 |
| 7.1.7 | Modellvalidierung..... | 138 |
| 7.1.8 | Anwendung auf Trennkupplung K0..... | 140 |
| 7.1.9 | Anwendung auf Kupplung K1 konventionelles Getriebe | 141 |
| 7.2 | Kupplungsfüllung..... | 143 |
| 7.2.1 | Optimierungsziel..... | 145 |
| 7.2.2 | Definition Aktions- und Zustandsraum | 145 |
| 7.2.3 | Definition Belohnungsfunktion und Algorithmus | 146 |
| 7.2.4 | Definition Trainingsumgebung | 146 |
| 7.2.5 | Training | 148 |
| 7.2.6 | Validierung..... | 149 |
| 7.3 | Hybrider Motorwiederstart | 155 |
| 7.3.1 | Testergebnisse | 156 |
| 7.3.2 | Datenspezifikation und -aufbereitung..... | 157 |
| 7.3.3 | Ereigniserkennung..... | 158 |
| 7.3.4 | Kennzahldefinition | 158 |
| 7.3.5 | Korrelationsanalyse | 159 |
| 7.3.6 | Clustering | 161 |
| 7.3.7 | Visualisierung | 163 |
| 7.3.8 | Verifikation und Validierung | 164 |
| 7.4 | Zwischenfazit..... | 167 |
| 8 | Studien zur Anwendbarkeit..... | 172 |
| 8.1 | Design der Befragungsstudie | 172 |
| 8.2 | Ergebnisse der Befragungsstudie..... | 173 |
| 8.2.1 | Steuerungs-Optimierung mit verstärkendem Lernen..... | 174 |
| 8.2.2 | Messdatenanalyse mit Cluster-Algorithmen..... | 175 |
| 8.3 | Design der Anwenderstudie..... | 176 |

| | | |
|---|--|-------------|
| 8.4 | Ergebnisse der Anwenderstudie | 179 |
| 8.5 | Zwischenfazit | 184 |
| 9 | Zusammenfassung und Ausblick | 186 |
| 9.1 | Zusammenfassung | 186 |
| 9.2 | Ausblick | 189 |
| Literaturverzeichnis | | I |
| Veröffentlichungen, die unter Mitautorenschaft des Autors / der Autorin dieser Forschungsarbeit entstanden sind: | | XV |
| Glossar | | XVII |
| Anhang | | XIX |
| | Literaturliste ML in hybriden Fahrzeuggetrieben | XIX |
| | Leitfaden Experteninterview | XXIV |
| | Fragebogen Steuerungs-Optimierung | XXV |
| | Fragebogen Datenanalyse | XXVI |
| | Ergebnisse Fragebogen Steuerungs-Optimierung | XXVII |
| | Ergebnisse Fragebogen Datenanalyse | XXVIII |
| | Aufgabenstellung und Anleitung Anwenderstudie Modellierung | XXIX |
| | Fragebogen Anwenderstudie Modellierung | XLIV |

Abbildungsverzeichnis

| | | |
|-----------------|---|----|
| Abbildung 2.1: | Konzepte paralleler Hybridantriebe (Eigene Darstellung nach Reif et al., 2012). | 5 |
| Abbildung 2.2: | Komponenten des hybriden Antriebsstrang am Beispiel der Mercedes-Benz A-Klasse (Maisch, Hörz & Jesser, 2019). | 6 |
| Abbildung 2.3: | Nasslaufende Lamellenkupplung mit variablem Drehschwingungsdämpfer (Naunheimer et al., 2019, S. 431). .. | 7 |
| Abbildung 2.4: | Softwarearchitektur eines Doppelkupplungsgetriebes (Fischer et al., 2016, S. 221). | 11 |
| Abbildung 2.5: | Allgemeine Darstellung Steuerung und Regelung | 12 |
| Abbildung 2.6: | Steuer- und Regelstrecke Kupplungsansteuerung | 14 |
| Abbildung 2.7: | Zustandsmaschine der Kupplungsansteuerung (Fischer et al., 2016, S. 244). | 14 |
| Abbildung 2.8: | Methoden zur theoretischen und experimentellen Modellbildung, (Eigene Darstellung nach Isermann, 2007). | 17 |
| Abbildung 2.9: | Das Referenzsystem im Modell der Systemgenerationsentwicklung (SGE) (Albers et al., 2019). Der Tesla Roadster übernimmt das Chassis des Lotus Elise (Übernahmevariation, CV), die Batteriezellen aus dem Referenzprodukt Laptop wurden in einer neuen Konfiguration integriert (AV). | 21 |
| Abbildung 2.10: | Design und Validierung im Produktentstehungsprozess (Albers, Behrendt et al., 2016, S. 545). | 23 |
| Abbildung 2.11: | Korrelation zwischen Produkt-Kreation und Validierung (Eigene Darstellung nach Mandel et al., 2020). | 25 |
| Abbildung 2.12: | IPEK-XiL-Framework..... | 26 |
| Abbildung 2.13: | Beschreibungsmodell Koppelsystem (Pinner, 2017, S. 105). .. | 28 |
| Abbildung 2.14: | Übersicht über den C&C ² -Ansatz und seine Elemente (Tröster, Klotz, Rapp, Ott & Albers, 2021). | 29 |
| Abbildung 2.15: | Einordnung maschinelles Lernen im Kontext der Künstlichen Intelligenz (Eigene Darstellung nach Russell & Norvig, 2002). | 31 |
| Abbildung 2.16: | Einfaches mathematisches Modell eines Neurons (Russell & Norvig, 2002, S. 846). | 32 |

| | | |
|-----------------|---|----|
| Abbildung 2.17: | Häufig verwendete Aktivierungsfunktionen | 33 |
| Abbildung 2.18: | Graphische Darstellung von Trainings-, Validierungs- und Testdatensatz | 34 |
| Abbildung 2.19: | Graphische Darstellung der LIME-Analyse (Eigene Darstellung nach Ribeiro, Singh & Guestrin, 2016). Zur Erklärung der Vorhersage für den Eingangsvektor X werden aus dem Black-Box-Modell anhand zusätzlicher um X liegender Vektoren (Stichproben) Vorhersagen erzeugt. Basierend auf diesen Vorhersagen und dem Eingangsvektor X erzeugt der LIME-Algorithmus dann ein lineares Ersatzmodell. | 37 |
| Abbildung 2.20: | Vorgehensmodell Maschinelles Lernen aufgezeigt anhand des CRISP-ML (Q) Modells (Eigene Darstellung nach Studer et al., 2021). | 38 |
| Abbildung 2.21: | Beispiele für 2-dimensionale Clusters-Srukturen mit verschiedenen Eigenschaften (Ester & Sander, 2013, S. 45). | 39 |
| Abbildung 2.22: | Anwendung von Cluster-Algorithmen der Scikit-Learn Bibliothek auf verschiedene zweidimensionale Cluster (Eigene Darstellung nach Pedregosa et al., 2022). | 40 |
| Abbildung 2.23: | Anwendung des DBSCAN Cluster-Algorithmus auf zweidimensionale Datenpunkte (Schubert et al., 2017). A ist ein Kernpunkt, Punkt B und C sind Grenzpunkte, N ist ein Rauschpunkt. | 41 |
| Abbildung 2.24: | Pseudocode des originalen sequentiellen DBSCAN-Algorithmus (Schubert et al., 2017). | 42 |
| Abbildung 2.25: | Auswertung einer Korrelations-Matrix für 6 Variablen | 43 |
| Abbildung 2.26: | Veranschaulichung des a) KDD-Prozess (Fayyad et al., 1996, S. 41) und b) CRISP-DM für Data Mining (Wirth & Hipp, 2000, S. 5). | 44 |
| Abbildung 2.27: | Schematische Darstellung des verstärkenden Lernens (Eigene Darstellung nach Sutton & Barto, 2018). | 45 |
| Abbildung 2.28: | Actor-Critic Architektur (Eigene Darstellung nach Sutton & Barto, 1998). Der Critic bewertet die Aktionen des Actors (Strategie) mit einem Fehler (TD-Error), der die Differenz zwischen erwarteter Belohnung (Value) und tatsächlicher Belohnung darstellt. | 47 |
| Abbildung 4.1: | Projekttypen im DRM-Modell (Eigene Darstellung nach Blessing & Chakrabarti, 2009). | 59 |

| | | |
|----------------|---|-----|
| Abbildung 4.2: | Methodensteckbrief – Interview (Eigene Darstellung nach Marxen, 2014). | 60 |
| Abbildung 4.3: | Methodensteckbrief – Fragebogen (Eigene Darstellung nach Marxen, 2014). | 60 |
| Abbildung 4.4: | Forschungsvorgehen der Arbeit in Anlehnung an die DRM (Blessing & Chakrabarti, 2009)..... | 62 |
| Abbildung 5.1: | Vorgehen der ersten Deskriptiven Studie | 65 |
| Abbildung 5.2: | Ergebnisse der Experteninterviews, Häufigkeit der Nennung in Prozent pro Kategorie | 67 |
| Abbildung 5.3: | Bewertungskriterien der Fahrfunktionen | 72 |
| Abbildung 5.4: | Prinzipskizze eines hybriden DKG in P2-Anordnung mit Ansteuerelementen | 75 |
| Abbildung 5.5: | Hauptkomponenten des Getriebes der Systemgeneration <i>Gn{hybrid}</i> (Maisch et al., 2019). | 75 |
| Abbildung 5.6: | Softwarearchitektur eines hybriden Doppelkupplungsgetriebes | 77 |
| Abbildung 5.7: | Steuer- und Regelkreis Kupplungsansteuerung | 78 |
| Abbildung 5.8: | Identifikation von Einflussfaktoren auf die Kupplungsansteuerung anhand des C&C ² -Modells der Kupplungsansteuerung und dem zugehörigen Wirknetz | 79 |
| Abbildung 6.1: | Vorgehen der Präskriptiven Studie | 85 |
| Abbildung 6.2: | Ist- und Soll-Zustand für Modelle auf Steuergeräten, Auslegung Steuerung und Regelung und Messdatenanalyse | 88 |
| Abbildung 6.3: | Vorgehen Modellierung von virtuellen Sensoren mit KNN | 92 |
| Abbildung 6.4: | Datengenerierung anhand von Fahrmanövern und Testumgebungen, 1) nach (Ott, 2024)..... | 93 |
| Abbildung 6.5: | Vorgehen Modellierung von virtuellen Sensoren mit KNN, vereinfachte Darstellung, Modellerstellung hervorgehoben | 98 |
| Abbildung 6.6: | Darstellung drei verschiedener Abbruchkriterien des Modelltrainings anhand des Validierungsfehlers | 99 |
| Abbildung 6.7: | Erweitertes IPEK-XiL-Framework mit dem optionalen System Optimierer (Eigene Darstellung nach Schröter, 2013). | 103 |
| Abbildung 6.8: | Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen | 104 |

| | | |
|-----------------|--|-----|
| Abbildung 6.9: | Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen, vereinfachte Darstellung, Definition Trainingsumgebung hervorgehoben | 107 |
| Abbildung 6.10: | Verknüpfung der rein virtuellen Trainingsumgebung und des Agenten anhand des IPEK-XiL-Frameworks am Beispiel eines Gangwechsels | 108 |
| Abbildung 6.11: | Korrelation zwischen Produkt-Kreation und Validierung (Eigene Darstellung nach Mandel et al., 2020). | 111 |
| Abbildung 6.12: | Vorgehen Messdatenanalyse mit Cluster-Algorithmen | 112 |
| Abbildung 6.13: | Vorgehen Messdatenanalyse mit Cluster-Algorithmen, Ausschnitt und vereinfachte Darstellung, Korrelationsanalyse hervorgehoben | 115 |
| Abbildung 6.14: | Clustering am Beispiel des DBSCAN Algorithmus | 116 |
| Abbildung 6.15: | Veranschaulichung der Visualisierung | 117 |
| Abbildung 6.16: | Datenanalyse im Kontext der SGE-Systemgenerationsentwicklung nach Albers. Die Erkenntnisse aus der Analyse von Feld- und Entwicklungsdaten können für verschiedene Systemgenerationen genutzt werden. | 119 |
| Abbildung 7.1: | Einordnung der Entwicklungsaufgaben in den Entwicklungsprozess von drei verschiedenen Systemgenerationen..... | 122 |
| Abbildung 7.2: | a) Kupplungsquerschnitt (Eigene Darstellung nach Naunheimer et al., 2019, b) Bohrungsquerschnitt Temperatursensor in Stahllamelle c) Temperatursensor und Stahllamelle | 124 |
| Abbildung 7.3: | Vorgehen Modellierung von virtuellen Sensoren mit KNN angewendet auf Kupplungstemperaturmodelle | 126 |
| Abbildung 7.4: | Normierter Temperaturverlauf des Testdatensatzes mit zugehörigen Lastfällen. Es werden für den Lastfall Anfahren für verschiedene Steigungen je vier Fahrzustände durchfahren, aufgebaut aus der Kombination hybrider Fahrbetrieb (H), elektrischer Fahrbetrieb (E) und vorwärts und rückwärts. | 129 |
| Abbildung 7.5: | Eingangs- und Ausgangsgrößen des KNN | 131 |
| Abbildung 7.6: | Ergebnisse der Hyperparameteroptimierung. Die Netzwerkarchitektur hat nur geringen Einfluss auf den Modellfehler, mit zunehmender Anzahl an Zeitschritten verringert sich der Modellfehler deutlich. | 134 |

| | | |
|-----------------|--|-----|
| Abbildung 7.7: | Ausschnitt der Sensitivitätsanalyse der Differenzdrehzahl. Die vorhergesagte Temperatur verhält sich proportional zur Differenzdrehzahl, dies entspricht ungefähr dem Verhalten des Referenzmodells. | 135 |
| Abbildung 7.8: | Ausschnitt der Sensitivitätsanalyse des KNN für alle fünf Eingangsgrößen. Das Modellverhalten ist bei einer Variation der Eingangsgrößen um bis zu 20 % nachvollziehbar. | 136 |
| Abbildung 7.9: | Die LIME-Analyse zeigt den Einfluss der Modelleingangsgrößen auf die Kupplungstemperatur für jeden Zeitschritt. Rot bedeutet, dass eine Zunahme der Größe zu einem Ansteigen der Temperatur führt, dunkles Blau entspricht analog einer Reduzierung der Temperatur. | 137 |
| Abbildung 7.10: | Vergleich der Kupplungstemperaturmodelle zu realen Messdaten für Fahrmanöver „Anfahren 30 %“ und „Anfahren 20 %“. Das KNN erzielt eine höhere Modellgüte wie das Referenzmodell. | 139 |
| Abbildung 7.11: | Vergleich der Kupplungstemperaturmodelle zu realen Messdaten. Das KNN erzielt eine höhere Modellgüte wie das Referenzmodell. | 140 |
| Abbildung 7.12: | Vergleich der Kupplungstemperaturmodelle zu realen Messdaten für das Fahrmanöver „30 % Anfahren“. Das KNN erzielt eine höhere Modellgüte als das Referenzmodell. | 142 |
| Abbildung 7.13: | Schematische Darstellung Lamellenkupplung (Eigene Darstellung nach Pinte, Depraetere, Symens, Swevers & Sas, 2010). | 143 |
| Abbildung 7.14: | Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen angewendet auf eine Kupplungsfüllung | 144 |
| Abbildung 7.15: | Virtuelle Validierungsumgebung der Kupplungsfüllung im Trainingsprozess anhand des IPEK-XiL-Ansatzes | 147 |
| Abbildung 7.16: | Hydrauliks Schaltplan der Getriebehydraulik | 148 |
| Abbildung 7.17: | Verlauf der mittleren Belohnung über Test-Episoden während des Trainingsvorgangs für den Systemdruck p_{sys} | 149 |
| Abbildung 7.18: | Parametrisierung des Ventilstroms | 150 |
| Abbildung 7.19: | Vergleich der Referenzansteuerung, Verstärkendes Lernen und Genetischer Algorithmus für eine Füllung der Kupplung K2 bei einem Systemdruck p_{sys} in der Trainingsumgebung | 151 |

| | | |
|-----------------|--|-----|
| Abbildung 7.20: | Fahrzeugmessung des Systemdrucks und der Füllzeit zur Validierung der Optimierungsergebnisse des Verstärkenden Lernens und des Genetischen Algorithmus | 153 |
| Abbildung 7.21: | Vorgehen Messdatenanalyse mit Cluster-Algorithmen, angewendet auf die Funktion Motorzustart | 156 |
| Abbildung 7.22: | Schematische Darstellung des Antriebsstrangs und relevante Messsignale..... | 157 |
| Abbildung 7.23: | Geschwindigkeit- und Lastverteilung des untersuchten Datensatzes..... | 157 |
| Abbildung 7.24: | Zeitlicher Ablauf des Motorwiederstarts dargestellt über Drehzahl von EM und VM, sowie des Gaspedalwinkels..... | 159 |
| Abbildung 7.25: | Korrelations-Matrix der zu untersuchenden Kennzahlen..... | 160 |
| Abbildung 7.26: | Darstellung von vier identifizierten Cluster des Motorzustarts über Geschwindigkeit, Gaspedalwinkel beim Zustart sowie Zustartzeit..... | 162 |
| Abbildung 7.27: | Vergleich der Motorzustart-Cluster in einer graphischen Übersicht | 163 |
| Abbildung 7.28: | Kennfeld von VM und EM für die vier identifizierten Motorzustart-Cluster | 164 |
| Abbildung 7.29: | Analyse des Motorstart-Cluster „langsamer Zustart“. Anhand des Clustering kann eine Verzögerung des Motorstarts erkannt werden..... | 166 |
| Abbildung 7.30: | Durch die Verwendung eines KNN im Vergleich zum Referenzmodell verringert sich der Modellfehler für drei verschiedene Kupplungen und Antriebsstränge um bis zu 22,5% | 168 |
| Abbildung 7.31: | Ansteuerungen der Kupplungsfüllung im Vergleich über die Zielgrößen Systemdruckeinbruch und Füllzeit | 170 |
| Abbildung 8.1: | Ausschnitt aus dem Software-Tool der Anwenderstudie und Beschreibung der Abschnitte | 178 |
| Abbildung 8.2 | Kenntnisstand beider Untersuchungsgruppen | 179 |
| Abbildung 8.3 | Ergebnisse Fragebogen zur Anwendbarkeit, beide Gruppen im Vergleich | 180 |

Tabellenverzeichnis

Tabelle 2.1 Metriken zur Bewertung der Modellgüte (Naser & Alavi, 2021)18

Tabelle 5.1 Ursachen, Folgen und abgeleitete Erkenntnisse der
Experteninterviews70

Tabelle 5.2 Bewertung der Fahrfunktionen72

Tabelle 6.1 Cluster-Algorithmen und Auswahlkriterien.....116

Tabelle 7.1 Fahrmanöver Trainingsdatensatz127

Tabelle 7.2 Hyperparameterraum133

Tabelle 7.3 Fahrmanöver Trainingsdatensatz K1 konventionelles Getriebe.....141

Tabelle 7.4 Vergleich Simulationsergebnisse Ansteuerung Kupplungsfüllung152

Tabelle 7.5 Vergleich Validierungsergebnisse Ansteuerung Kupplungsfüllung ...154

Tabelle 7.6. Hyperparameterbestimmung DBSCAN161

Abkürzungsverzeichnis

| | |
|--------------------------|---|
| Adam | Adaptive Momentum Estimation |
| ASIL | Automotive Safety Integrity Level |
| AV | Ausprägungsvariation |
| C | Connector |
| CN | Convolutional Network |
| CRISP | Cross Industry Standard Process |
| C&C ² -Ansatz | Contact and Channel Ansatz |
| DBSCAN | Density Based Spatial Clustering of Applications with Noise |
| DRM | Design Research Methodology |
| DKG | Doppelkupplungsgetriebe |
| EM | Elektromotor |
| CNN | faltende neuronale Netze |
| FF | Forschungsfrage |
| FNN | Vorwärts gerichtete neuronale Netze |
| HARA | hazard analysis and risk assessment |
| IPEK | Institut für Produktenwicklung |
| IPEK-XiL | IPEK-X-in-the-loop |
| KaSPro | Karlsruher Schule der Produktentwicklung |
| KDD | Knowledge Discovery in Databases |
| KI | Künstliche Intelligenz |
| KNN | Künstliches neuronales Netz |
| LIME | Local Interpretable Model-Agnostic Explanations |
| LSS | Leitstützstruktur |
| LSTM | Long Short-Term Memory Cells |
| MDP | Markov-Decision-Process |
| MFR | Mehrfachrückschaltung |
| ML | Maschinelles Lernen |

| | |
|-------|-----------------------------------|
| MSE | mean squared error |
| MPR | Modellprädiktive Regelung |
| NVB | Nieder-Volt-Batterie |
| NVH | Noise Vibration Harshness |
| PE | Produktentwicklung |
| PV | Prinzipvariation |
| RNN | Rekurrentes neuronales Netz |
| RMSE | root mean square error |
| RSE | Referenzsystemelement |
| SGE | Systemgenerationsentwicklung |
| SOTIF | Sicherheit der Sollfunktion |
| SAC | Soft-Actor-Critic |
| TCN | Zeitlich faltende neuronale Netze |
| ÜV | Übernahmevariation |
| VM | Verbrennungsmotor |
| WFP | Wirkflächenpaar |

Formelzeichen

| | |
|-------------------|---|
| a_i | Eingangsvektor Neuron |
| a_t | Aktion |
| AS_n | Ausprägungsvariation |
| c | spezifische Wärmespeicherfähigkeit |
| c_i | Gewichtung Belohnungsfunktion |
| $c_{Ventil_{K2}}$ | Ventilstrom Kupplung K2 |
| g | Aktivierungsfunktion |
| i_{EM} | Übersetzung zwischen EM und Getriebeeingang |
| m | Kupplungsmasse |
| $minPts$ | minimale Punkte für ein Cluster |
| n_{EM} | Drehzahl E-Maschine |
| n_{in} | Getriebeeingangsdrehzahl |
| n_{VM} | Drehzahl Verbrennungsmotor |
| p | Anpressdruck |
| p_{An} | Anlegedruck |
| p_{eff} | effektiver Druckanteil |
| p_{K2} | Druck Kupplung K2 |
| pos_{K2} | Position Kupplung K2 |
| PS_n | Prinzipvariation |
| p_{sys} | Systemdruck |
| $p_{sys_{ist}}$ | Ist-Systemdruck |
| $p_{sys_{max}}$ | maximaler Systemdruckeinbruch |
| $p_{sys_{soll}}$ | Soll-Systemdruck |
| r | Belohnung |

| | |
|---------------|------------------------------------|
| r_m | Mittlerer Radius |
| s_t | Zustand |
| s_x | Standardabweichung |
| s_{xy} | Kovarianz |
| $t_{Füll}$ | Füllzeit |
| $t_{Füllmax}$ | maximale Füllzeit |
| t_r | Rutschzeit |
| t_{start} | Motorzustartzeit |
| v | Fahrzeuggeschwindigkeit |
| w_{10} | Winkelgeschwindigkeit |
| $w_{i,j}$ | Gewichtsvektors |
| z | Anzahl Reiblamellen |
| A | Aktionsraum |
| A_K | Kolbenfläche |
| A_W | am Wärmeübergang beteiligte Fläche |
| F_N | Normalkraft |
| G_i | Sytemgeneration |
| G_S^{-1} | inverses Streckenmodell |
| G_s | Regelstrecke |
| H | Zeithorizont |
| J | erwartete zurückgegebene Belohnung |
| Max | maximaler absoluter Betragsfehler |
| M_{des} | Soll-Drehmoment |
| $Mean$ | mittlerer absoluter Betragsfehler |
| M_{EM} | Drehmoment E-Maschine |
| M_K | Kupplungsmoment |
| MSE | mean-squared-error |

| | |
|---------------|---------------------------------------|
| M_{VM} | Drehmoment Verbrennungsmotor |
| P | Bestrafung/Belohnungsterm |
| P_R | Reibeistung |
| Q | Reibenergie |
| R | Bestimmtheitsmaß |
| R_i | Referenzsystem |
| $RMSE$ | root-mean-squared-error |
| $R_{pearson}$ | Korrelationskoeffizient nach Pearson |
| S | Zustandsraum |
| T_i | gemessener Temperatur |
| T_k | Lamellentemperatur |
| $T_{KNN,i}$ | Modelltemperatur |
| $T_{öl}$ | Ölsumpftemperatur |
| US_n | Übernahmevariation |
| Z | Zielgröße |
| Z_{norm} | normierte Zielgröße |
| \mathbb{E} | Erwartete Belohnung |
| \mathcal{H} | Entropie-Term |
| \mathbb{P} | probabilistische Übertragungsfunktion |
| \dot{Q} | abgeführte Wärmeleistung |
| $q_{öl}$ | Ölvolumenstrom |
| $Q(s, a)$ | Aktionswert-Funktion |
| $u(t)$ | Stellgröße |
| $V(s)$ | Zustandswert-Funktion |
| $w(t)$ | Führungsgröße |
| $w_i(t)$ | Zeitabhängige Winkelgeschwindigkeit |
| $x(t)$ | Regelgröße |

| | |
|-------------------|-------------------------------------|
| $Z(t)$ | Zielgröße pro Zeitschritt |
| $z(t)$ | Störung |
| α_K | Wärmeübergangskennzahl |
| α | Temperaturparameter |
| Δn | Differenzdrehzahl |
| Δn_{EM} | Drehzahldifferenz EM |
| ΔT | Temperaturänderung |
| Δw | Differenzwinkelgeschwindigkeit |
| ∇n_{VM} | Gradient Drehzahl Verbrennungsmotor |
| ϵ_{Max} | maximaler relativer Betragsfehler |
| ϵ_{Mean} | mittlerer relativer Betragsfehler |
| ε | Parameter der Dichte für DBSCAN |
| μ | Reibungszahl |
| π | Strategie |

1 Einleitung und Motivation

Bereits im Jahre 1956 wurde der Begriff der künstlichen Intelligenz (KI) zum ersten Mal auf der Dartmouth Conference in Hanover, New Hampshire eingeführt (Moor, 2006). KI ist somit keine neue Erfindung des 21. Jahrhunderts. Erst die Steigerung der Transistorendichte und damit auch der Rechenleistung von Mikroprozessoren über die letzten 50 Jahre um den Faktor 25.000.000¹, einer jährlich steigenden generierten Datenmenge von 64,3 Zettabyte im Jahr 2020 (Tenzer) sowie die Verfügbarkeit von OpenSource Bibliotheken (Abadi et al., 2016) haben den erfolgreichen Einsatz von KI ermöglicht.

Mit maschinellem Lernen (ML), einem Teilgebiet der KI, wurden in Anwendungen wie Sprach- und Bilderkennung in den letzten Jahren bahnbrechende Ergebnisse erzielt. Medizinische Diagnosesysteme erkennen Brustkrebs früher und besser als Ärzte mit jahrelanger Erfahrung (McKinney et al., 2020). Auch in strategischen Spielen wie dem koreanischen Brettspiel Go wurden die besten menschlichen Spieler der Welt mehrfach durch KI geschlagen (Silver et al., 2016). Die weiter steigende Rechenleistung, die Entwicklung Cloud-Computing sowie Unmengen an Sensordaten durch Internet-of-Things und Industrie 4.0 treiben den Einsatz und Nutzen von KI exponentiell voran (Kreutzer & Sirrenberg, 2019, S. 73–105). ML bietet somit ein großes Potential, vielfältige und komplexe Probleme zu lösen.

Gleichzeitig findet in der Automobilindustrie eine der größten Transformationen in der Geschichte statt (Proff, 2020). Neben batterieelektrischen Fahrzeugen sind auch Hybridfahrzeuge entscheidend, um die CO₂-Flottenziele zu erreichen. Hybridfahrzeuge sollen sich dabei in ihrem Fahrverhalten nicht von konventionellen Antrieben unterscheiden. Für das Fahrverhalten bzw. die Fahrbarkeit ist das Getriebe und die Getriebe-Software hauptverantwortlich. Diese steuert und regelt alle Abläufe im Getriebe und enthält Überwachungs- und Diagnosefunktionen, um die Fahrzeugsicherheit zu gewährleisten. Die Entwicklung von Fahrfunktionen der Getriebe-Software wird übergreifend als Funktionsentwicklung bezeichnet. Durch die Elektrifizierung ergeben sich weitere komplexe Fahrfunktionen, wie der Motorwiederstart des Verbrennungsmotors während des rein elektrischen

¹ Reduzierung der Strukturbreite von Mikroprozessoren von 10 µm im Jahr 1971 auf 2 nm im Jahr 2021 (Lemieux und Lewis (2004); Pollie (2021))

Fahrbetriebs. Außerdem steigt die Komplexität des Gesamtgetriebes durch die Integration einer E-Maschine und weiteren Kupplungen deutlich an (vgl. Kapitel 2.1). Hohe Kundenanforderungen an Komfort, Dynamik und Verbrauch (Braun, Künnemann, Rich, Mißler-Behr & Woll, 2020, S. 144–161) sind weitere Randbedingungen, die die Entwicklung erschweren. Es bedarf neuer Methoden, um diese Herausforderungen zu bewältigen (Bock, Sippl, Siegl & German, 2019).

Der Einsatz von ML-Methoden scheint auf den ersten Blick hierfür geeignet zu sein. Diese Arbeit untersucht deshalb, wie ML-Methoden gewinnbringend in der Funktionsentwicklung hybrider Fahrzeuggetriebe eingesetzt werden können, um Produktentwicklungs-Teams (PE-Teams) in ihrer Arbeit zu unterstützen. Nach der Identifikation der Herausforderungen in diesem Feld werden die bestehenden ML-Methoden an den Kontext der Funktionsentwicklung hybrider Fahrzeuggetriebe adaptiert. Eine Methodenbeschreibung soll PE-Teams in die Lage versetzen, ML-Methoden auf weitere Anwendungsfälle im Antriebsstrang anzuwenden.

Die Forschungskooperation zwischen IPEK - Institut für Produktentwicklung und dem Entwicklungsbereich von Hybridfahrzeuggetrieben der Mercedes-Benz Group AG bietet die große Chance die entwickelten Methoden am Beispiel eines hybriden Doppelkupplungsgetriebes (DKG) praxisnah anzuwenden und zu evaluieren. Die Forschungskooperation liefert somit für die KaSPro - Karlsruher Schule der Produktentwicklung wichtige Erkenntnisse zur Weiterentwicklung und Validierung der eigenen Modelle und Methoden (Albers, Burkardt & Duser, 2006).

Nach dieser Einleitung werden in **Kapitel 2** die Grundlagen von parallelen Hybridantrieben, relevante Prozesse und Methoden der KaSPro und Grundlagen von ML beschrieben. Außerdem werden der Stand der Forschung zum Einsatz von ML-Methoden in Fahrzeuggetrieben untersucht und eine Forschungslücke abgeleitet. Basierend auf dem Stand der Forschung werden in **Kapitel 3** Forschungsbedarf und Zielsetzung der Arbeit erläutert. **Kapitel 4** stellt das Forschungsvorgehen anhand der Forschungsmethoden und der Forschungsumgebung dar. Um ML-Methoden gezielt einsetzen zu können, werden in **Kapitel 5** die Herausforderungen in der Funktionsentwicklung von Hybridfahrzeuggetrieben am Beispiel des DKG abgeleitet. Anschließend werden in **Kapitel 6** die ML-Methoden im Kontext der Funktionsentwicklung von hybriden DKG für die Produktentwicklung nutzbar gemacht. Der Fokus liegt dabei auf den Adaptionen, die durch die Verwendung in der Antriebsstrangentwicklung gemacht werden. Daraufaufgehend werden die entwickelten Methoden in **Kapitel 7** auf aktuelle Entwicklungsaufgaben angewendet und in **Kapitel 8** in Studien hinsichtlich ihrer Anwendbarkeit evaluiert. **Kapitel 9** fasst die Ergebnisse der Arbeit zusammen und diskutiert weitere Untersuchungen, die an dieser Arbeit anknüpfen können.

2 Grundlagen und Stand der Forschung

In diesem Kapitel werden die Grundlagen und Definitionen der in dieser Arbeit betrachteten Themenfelder sowie der aktuelle Stand der Forschung zusammengetragen.

Da sich diese Forschungsarbeit mit parallelen Hybridantrieben beschäftigt, werden zunächst deren mechanischer Aufbau und die Regelung und Steuerung von Getrieben vorgestellt. In Kapitel 2.2 wird auf die relevanten Prozesse und Methoden der KaSPro – Karlsruher Schule der Produktentwicklung eingegangen und dabei die in dieser Arbeit genutzten Methoden IPEK-X-in-the-loop-Ansatz (IPEK-XiL) und Contact and Channel Ansatz (C&C²-Ansatz) erläutert. Anschließend werden die Methoden des maschinellen Lernens vorgestellt, die in dieser Arbeit Anwendung finden. Das vorletzte Kapitel stellt den Stand der Forschung des Einsatzes von ML-Methoden in Fahrzeuggetrieben dar.

2.1 Paralleler Vollhybrid-Antrieb für Kraftfahrzeuge

Nachfolgend werden die Fahrfunktionen und Topologie paralleler Vollhybrid-Antriebe beschrieben. Anschließend werden die Steuerung und Regelung von Fahrzeuggetrieben erläutert, wobei vertieft auf die Kupplungsansteuerung eingegangen wird. Abschließend wird der Stand der Forschung der Auslegung der Steuerung und Regelung, der Modellbildung und der Datenanalyse in Fahrzeuggetrieben dargestellt.

2.1.1 Fahrfunktionen und Topologie

Ein Hybridantrieb beinhaltet mindestens zwei fahrzeugeigene Energiewandler und Energiespeicher für den Antrieb des Fahrzeugs (ECE R83). Aktuell auf dem Markt befindliche Hybridfahrzeuge besitzen die Energiequellen Kraftstoff und elektrische Energiespeicher. In dieser Arbeit werden Hybridfahrzeuge mit Kraftstoff und Batterie als Energiespeicher und Verbrennungsmotor (VM) und E-Maschine (EM) als Energiewandler betrachtet.

Je nach Anordnung der Komponenten VM, EM, Generator, Batterie und Getriebe lassen sich Hybridantriebe in serielle, parallele und leistungsverzweigte Hybridantriebe einteilen (Hofmann, 2014, S. 24).

Bei parallelen Hybridantrieben ist ein direkter mechanischer Durchtrieb vom VM bis zum Rad möglich. Die elektrischen Antriebskomponenten liegen parallel vor und können dadurch zu- oder abgeschaltet werden. Somit sind ein elektrischer, konventioneller und gemischter Fahrbetrieb möglich. Die Leistungen von VM und EM können über drei Möglichkeiten miteinander verbunden werden. Bei der Drehzahladdition werden die Drehzahlen der beiden Kraftmaschinen über ein Planetengetriebe gekoppelt. Bei der Momentenaddition sind die beiden Kraftmaschinen über eine direkte Kopplung oder Stirnradstufe verbunden. Bei der Zugkraftaddition werden die beiden Zugkräfte durch Antreiben unterschiedlicher Achsen addiert. Der große Vorteil dieser Antriebsart liegt darin, dass drei verschiedene Fahrbetriebe möglich sind, wodurch der jeweils effizienteste gewählt werden kann. Außerdem wird nur eine EM benötigt. Nachteilig ist, dass der VM nicht mehr stationär betrieben werden kann.

Fahrfunktionen. Der Parallel-Hybrid ermöglicht die Funktionen rein elektrisches Fahren, rekuperatives Bremsen, Lastpunktverschiebung und Boosten. Dadurch ist der Parallel-Hybrid in Stadtzyklen sehr effizient.

Ist es nach der Antriebsstrangkonfiguration möglich den VM durch eine Kupplung vom Antriebsstrang zu trennen, so ist rein elektrisches Fahren darstellbar.

Der generatorische Betrieb der EM ermöglicht es, die kinetische Energie des Fahrzeugs bei Bremsvorgängen in elektrische Energie umzuwandeln. Dieser Vorgang wird regeneratives oder rekuperatives Bremsen genannt. Dabei wird das Verzögerungsmoment auf die mechanische Bremse und den Generator als Bremse aufgeteilt.

Durch die EM wird es möglich, zusätzlich zu den Standphasen während den Bremsphasen des Fahrzeugs den VM auszuschalten und somit weitere Kraftstoffeinsparungen zu realisieren. Bedingung dafür ist, dass elektrische Nebenaggregate wie Klimakompressoren und Hydraulikpumpen vorhanden sind (Reif, Noreikat & Borgeest, 2012, S. 17–25).

Weitere Kraftstoffeinsparungen sind durch die Lastpunktverschiebung des VM möglich. Die EM arbeitet dabei wieder als Generator und erhöht die Last des VM, was dessen Betriebspunkt in einen Bereich höheren Wirkungsgrades verschiebt. Gleichzeitig wird die Batterie geladen. Eine geeignete Betriebsstrategie zwischen Laden der Batterie durch die Lastpunktverschiebung und dem elektrischen Fahren

muss vorhanden sein, um die Effizienz zu maximieren. Hierbei müssen auch die Verluste durch Generator, Batterie und EM genau betrachtet werden.

Wird während der verbrennungsmotorischen Fahrt ein zusätzliches Beschleunigungsmoment benötigt, so kann die EM dieses bereitstellen und den VM unterstützen. Dieses Boosten kann auch genutzt werden, um Drehmomentenschwäche, sowie Defizite bei der Effizienz und Emissionen auszugleichen.

Zusätzlich ist auch eine Einteilung der Hybride nach dem Hybridisierungsgrad in Mikro-, Mild-, und Vollhybrid möglich. Mikro-Hybride können begrenzt Energie rekuperieren, boosten sowie die Start-Stopp-Funktion erfüllen, jedoch meist nicht rein elektrisch fahren. Mild-Hybride erfüllen die gleichen Funktionalitäten, sind aber in der Lage kurze Strecken rein elektrisch zurückzulegen und die Lastpunktverschiebung durchzuführen. Voll-Hybride erfüllen alle beschriebenen Funktionen des hybriden Antriebs und können Strecken von wenigen Kilometern zurücklegen. Der Plug-In-Hybrid ist ein Vollhybrid, der eine große Batterie (10-20 kWh) besitzt und über eine externe Energiequelle (Plug, deutsch: Stecker) geladen werden kann. Dadurch kann dieser Antrieb größere Strecken rein elektrisch zurücklegen. (Reif et al., 2012, S. 67–69)

Topologie. Für die Anordnung der EM in einem parallelen Hybridantrieb gibt es vier Konzepte (vgl. Abbildung 2.1).

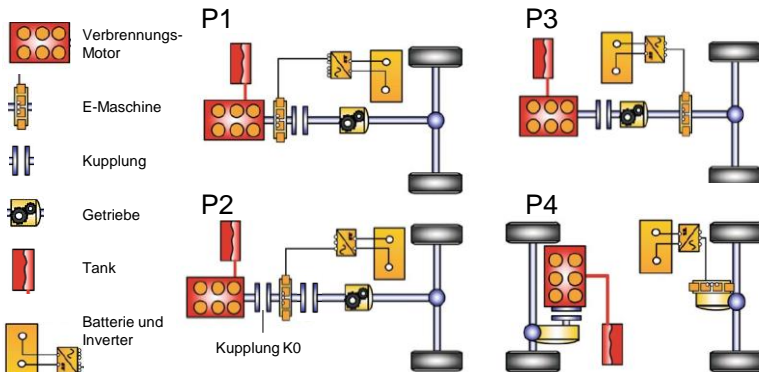


Abbildung 2.1: Konzepte paralleler Hybridantriebe (Eigene Darstellung nach Reif et al., 2012).

Die Anordnungen P1 bis P4 verschieben jeweils die EM näher an die Antriebsachse. Beim P1-Hybrid sitzt die EM direkt am VM und ist mit der Kurbelwelle fest

verbunden. Der Vorteil dieser Konfiguration liegt in der einfachen Integrierbarkeit der EM in den Antriebsstrang. Nachteilig sind die hohen Schleppverluste des VM bei Rekuperation über die EM und dass ein elektrischer Fahrbetrieb nicht möglich ist.

Beim Parallel-2-Hybrid sitzt die EM am Getriebeeingang. Durch das Abtrennen des VM über die Trennkupplung K0 ist rein elektrisches Fahren und Rekuperation ohne Motorschleppverluste möglich.

Beim Parallel-3-Hybrid sitzt die EM am Getriebeausgang. Dadurch ist es möglich während Getriebschaltungen die Zugkraft zu erhalten. Außerdem ist rein elektrisches Fahren einfach möglich und die EM ist mit wenig Aufwand im Antriebsstrang integrierbar.

Beim Parallel-4-Hybrid treiben die EM und der VM jeweils eine eigene Achse an. Dies entspricht dem Konzept einer Zugkraftaddition. Die Integration der EM in den Antriebsstrang und rein elektrisches Fahren ist einfach möglich. Außerdem kann ein Allradantrieb dargestellt werden. Durch dieses Konzept ist jedoch keine Start-Stopp-Funktion und Lastpunktverschiebung möglich (Reif et al., 2012, S. 32–36).

Der in dieser Arbeit betrachtete hybride Antriebsstrang ist ein elektrifiziertes Doppelkupplungsgetriebe in einer P2-Anordnung. Abbildung 2.2 gibt eine Übersicht über den kompletten Antriebsstrang samt aller Komponenten.

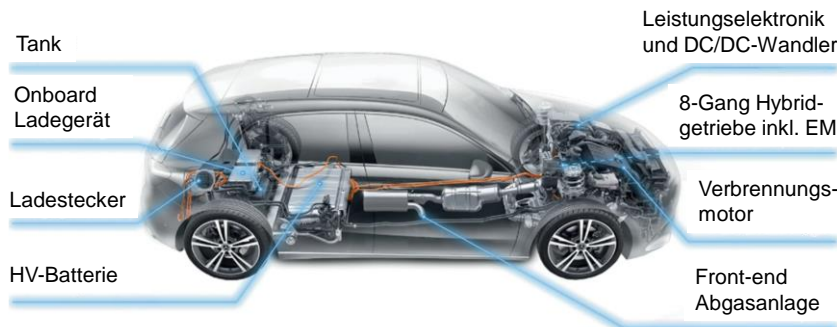


Abbildung 2.2: Komponenten des hybriden Antriebsstrang am Beispiel der Mercedes-Benz A-Klasse (Maisch, Hörz & Jesser, 2019).

Ein Doppelkupplungsgetriebe besteht aus zwei Teilgetrieben mit jeweils allen geraden und ungeraden Gängen. Über zwei Kupplungen K1 und K2 kann jeweils ein Teilgetriebe kraftschlüssig mit dem Antrieb verbunden werden. Doppelkupplungsgetriebe besitzen die Vorteile von sehr geringen Schaltzeiten und

Schaltungen können ohne Zugkraftunterbrechung durchgeführt werden (Naunheimer, Bertsche, Ryborz, Novak & Fellows, 2019, S. 177–180). Das in dieser Arbeit zugrundeliegende Referenzgetriebe wird in Kapitel 5.2.2 im Detail beschrieben. Die Kupplungen K0, K1 und K2 sind als nasslaufende Lamellenkupplungen ausgeführt. Deren Aufbau und Berechnungsgrundlagen werden nachfolgend erläutert.

2.1.2 Nasslaufende Lamellenkupplung

Nasslaufende Lamellenkupplungen bestehen aus Stahllamellen und Belagslamellen, die abwechselnd angeordnet und jeweils mit dem Gehäuse (Außenlamelle) oder der Welle (Innenlamelle) verbunden sind (Albers et al., 2018, S. 291). Diese Kupplungen werden von einem Kühlöl in einem ödichten Gehäuse durchströmt (vgl. Abbildung 2.3) (Naunheimer et al., 2019, S. 430) .

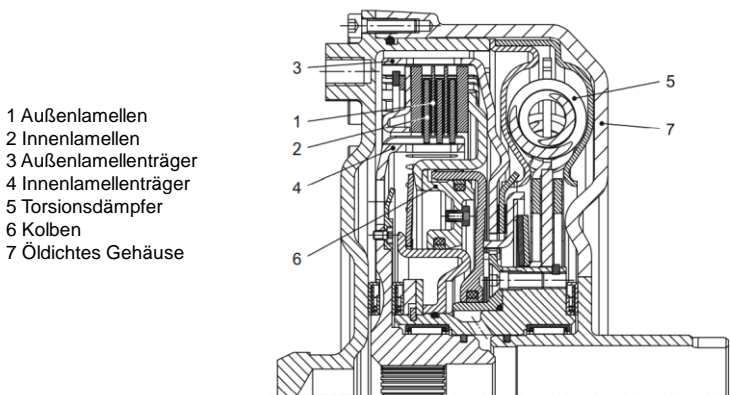


Abbildung 2.3: Nasslaufende Lamellenkupplung mit variablem Drehschwingungsdämpfer (Naunheimer et al., 2019, S. 431).

Durch die aktive Kühlung der Lamellen können hohe Wärmemengen in den Kühlölkreislauf abgegeben werden. Dadurch besitzen sie auch eine gute Dauerhaltbarkeit bei geringem Verschleiß und eine gute Regelbarkeit. Durch die axiale Anordnung mehrerer Lamellen besitzen sie außerdem ein geringes Massenträgheitsmoment bezogen auf das übertragbare Drehmoment. Allerdings sorgt das Laufen in Öl für Schleppverluste und einen höheren technischen Aufwand bei der Ölversorgung im Vergleich zu trockenlaufenden Kupplungen (Naunheimer et al., 2019, S. 429–435).

Das Reibmoment während der Schlupfphase berechnet sich unter der Annahme einer homogenen Flächenpressung und einer konstanten Reibungszahl μ , aus dem Produkt von Reibungszahl, der Normalkraft F_N , dem mittleren Radius r_m und der Anzahl an Reibflächen z . Es ist anzumerken, dass eine homogene Flächenpressung im realen Reibsystem nicht auftritt und hier als vereinfachender Ansatz herangezogen wird.

$$M_K = \mu r_m F_N z \quad 2.1$$

Die Reibungszahl μ ist ein Funktional von beispielsweise Ort, Zeit, Geschwindigkeit, Flächenpressung oder Temperatur (vgl. Glossar) (Albers et al., 2018, 310–311).

Die während der Schlupfphase eingebrachte Reibleistung P_R berechnet sich aus dem übertragenen Reibmoment M_K des Kupplungssystems und Differenz der zeitabhängigen Winkelgeschwindigkeiten $\omega_1(t)$ und $\omega_2(t)$ am Kupplungseingang und -ausgang.

$$P_R = M_K (\omega_2(t) - \omega_1(t)) \quad 2.2$$

Die Reibenergie Q während einer Schlupfphase entspricht dem Integral der Reibleistung P_R über die zeitlichen Grenzen t_0 und t (Albers et al., 2018, S. 324–327).

$$Q = M_K \int_{t_0}^t (\omega_2(t) - \omega_1(t)) dt \quad 2.3$$

Für die in der Kupplung gespeicherte Wärmemenge sind die Kupplungsmasse m , die spezifische Wärmespeicherfähigkeit des Werkstoffs c und die Temperaturzunahme ΔT entscheidend.

$$Q = mc\Delta T \quad 2.4$$

Die abgeführte Wärmeleistung \dot{Q} an das Kühllöl aufgrund von Konvektion berechnet sich aus der Differenz der Lamellentemperatur T_k und der Temperatur des vorbeiströmenden Öls $T_{öl}$, der Wärmeübergangskennzahl α_K und der an dem Wärmeübergang beteiligten Fläche A_W . Die Wärmeübergangskennzahl α_K ist von

der Relativgeschwindigkeit zwischen Öl und Lamelle abhängig (Albers et al., 2018, S. 324–332).

$$\dot{Q} = \alpha_K A_W (T_K - T_{\text{öl}}) \quad 2.5$$

Die Betätigung von Doppelkupplungen kann elektrohydraulisch oder elektromechanisch durchgeführt werden. Auf die Kupplungsansteuerung wird im nachfolgenden Kapitel tiefer eingegangen.

Zusätzlich weist der Drehmomentenverlauf einer Kupplung ein Hysterese-Verhalten auf (Fischer, Küçükay, Jürgens & Pollak, 2016, S. 243; Kirchner, 2007, S. 287). Dieses unterteilt sich im Falle einer hydraulischen Ansteuerung in einen mechanischen Anteil, Kupplungsdruck zu Drehmoment, sowie einen hydraulischen Anteil, Ventilstrom zu Kupplungsdruck. Der mechanische Anteil entsteht durch Veränderung der Reibungszahl aufgrund tribologischer Einflüsse in der Reibpaarung sowie Reibverlusten im Ausrücksystem (Klotzbach & Henrichfreise, 2002), den Mitnehmerverzahnungen der Lamellen und in den Federn für die Belagsrückstellung (Zeng, Xue & Xiao, 2019). Der hydraulische Anteil entsteht über das Hydraulikventil durch Verluste auf Grund von Reibung in den Dichtflächen, dem Schließelement des Ventils und der Ventiltfeder und der magnetischen Hystereseanteile des Ventils selbst (Cavina, Olivi, Corti, Poggio & Marcigliano, 2012; Jung, Choi, Ko, Kim & Lee, 2019).

2.1.3 Steuerung und Regelung von Automatikgetrieben

Um eine Automatisierung des Getriebes zu ermöglichen, müssen die Aufgaben, die der Fahrer bei einem manuellen Schaltgetriebe selbst durchführt, wie der Schaltvorgang oder das Anfahren, über zusätzliche Komponenten und eine Steuerung und Regelung ersetzt werden. Das Herz der elektronischen Steuerung ist ein Mikrocontroller, der einen Rechnerkern sowie Peripherie zum Auslesen der Sensoren und Ansteuerung der Aktuatoren enthält. Die Software, die auf diesem Mikrocontroller läuft, wird als Embedded Software bezeichnet und muss sehr sparsam mit Ressourcen wie Rechenleistung und Speicher umgehen und die Ausführung der Software in Echtzeit garantieren. Der große Vorteil einer Software liegt daran, dass zwar hohe Einmalkosten in der Entwicklung entstehen, jedoch im Gegensatz zur Hardware-Komponente keinerlei Stückkosten anfallen. Durch eine intelligente und optimierte Software können Hardware und Stückkosten pro Getriebe reduziert werden. Zum Beispiel können durchdachte Regelalgorithmen der Software

die Schwächen von günstigeren Hydraulikventilen ausgleichen (Fischer et al., 2016, S. 219–222).

Zu den Aufgaben einer Getriebesoftware zählt (Fischer et al., 2016, S. 221–222):

- Vorverarbeitung der Sensorsignale
- Auswertung der Fahrereingaben (Wählhebel, Gas, Bremse)
- Bestimmung der Fahrsituation und des Fahrganges
- Berechnungen zur Steuerung der Fahrzustände (z.B. Anfahren, Konstantfahrt oder Schalten), auch Antriebsstrangmanagement genannt
- Kupplungs- und Gangschaltsteuerung bzw. Kupplungs- und Bremsensteuerung, Fernsteuerung des VM (während Schaltungen)
- Ansteuerung der Aktuatoren (z.B. Hydraulikventile)
- Selbstdiagnose (On-Board-Diagnose)
- Sicherheitssoftware zur Vermeidung kritischer Fahrsituationen (z. B. Anfahren in die falsche Fahrtrichtung)

Der in dieser Arbeit verwendete Begriff der Getriebefunktionen beinhaltet dabei die genannten Aufgaben.

Die Softwarearchitektur definiert die Bausteine der Software und beschreibt die Schnittstellen zwischen diesen Komponenten und zur Außenwelt. Abbildung 2.4 zeigt beispielhaft die Softwarearchitektur eines Doppelkupplungsgetriebes, die wichtigsten Softwaremodule, sowie die Kommunikationssignale zwischen den Modulen.

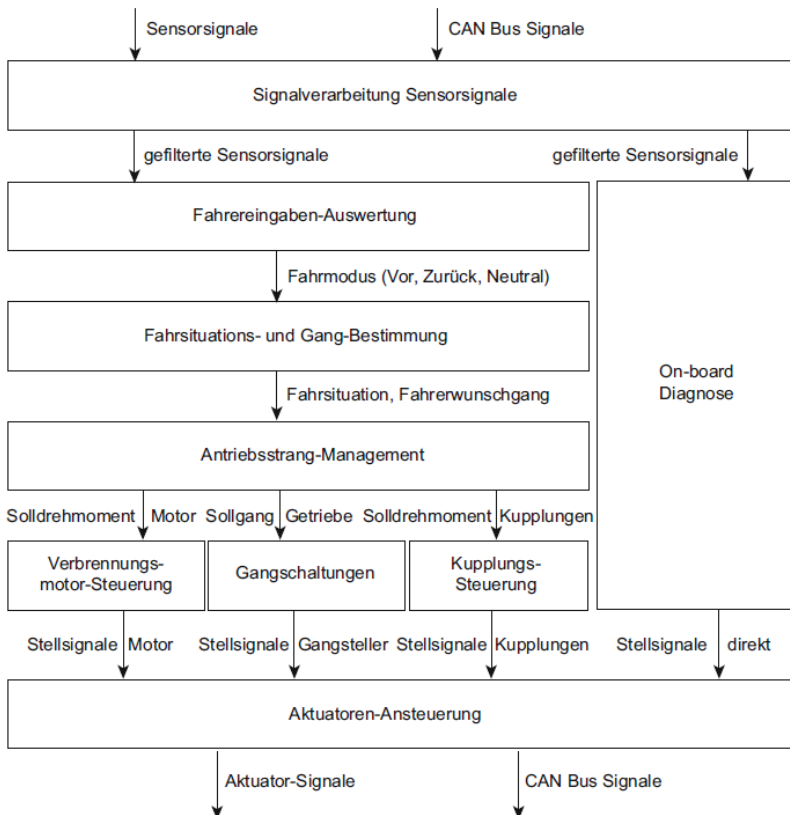


Abbildung 2.4: Softwarearchitektur eines Doppelkupplungsgetriebes (Fischer et al., 2016, S. 221).

Die wichtigsten Elemente der Steuerung und Regelung eines Automatikgetriebes sind in Abbildung 2.5 dargestellt und werden nachfolgend anhand des Beispiels Drehzahlregelung während einer Schaltung und Kupplungsfüllung erläutert.

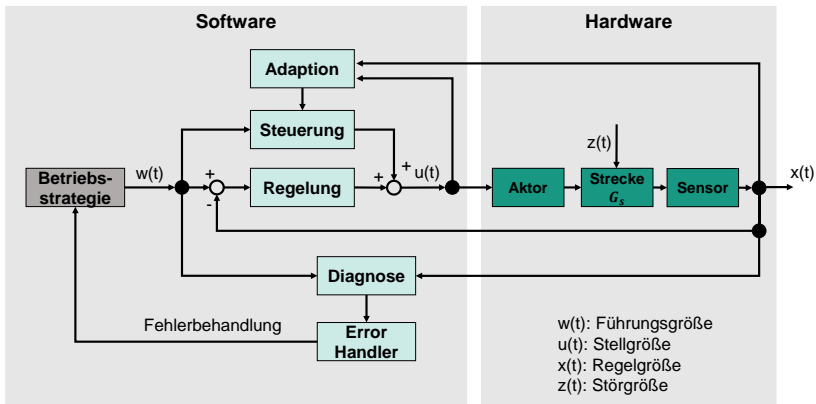


Abbildung 2.5: Allgemeine Darstellung Steuerung und Regelung

Die Betriebsstrategie bestimmt die Abläufe im Getriebe. Sie gibt eine Führungsgröße aus, die von der Steuerung und Regelung umgesetzt werden soll.

Für die Drehzahlregelung während einer Schaltung stellt die Betriebsstrategie eine Soll-Drehzahl als Führungsgröße $w(t)$. Die Differenz aus der Soll-Drehzahl und der rückgeführten gemessenen Ist-Drehzahl als Regelgröße $x(t)$ ist die Eingangsgröße der Regelung. Dieses Signal wird von der Regelung in einen Kupplungsdruck als Stellgröße $u(t)$ gewandelt. Der Kupplungsdruck führt zu einer Änderung der Ist-Drehzahl (Regelgröße $x(t)$). Grundlegend bei der Regelung ist die Rückführung der Regelgröße. Dadurch können Störungen in der Strecke wie eine ungenaue Berechnung der Kupplungsmomente aufgrund unterschiedlicher Streckenzeiten oder Temperaturänderung ausgeglichen werden. Über Regler-Parameter ist das Regel-Verhalten einstellbar. Der Nachteil von Reglern ist, dass die Rückführung für sehr dynamische Vorgänge zu langsam ist, um eine Regelung zu ermöglichen. Außerdem muss die Regelgröße $x(t)$ über einen Sensor gemessen werden.

Bei einer Kupplungsfüllung werden die Kupplungslamellen aus der durch die Rückstellkraft der Feder bedingten offenen Position in Kontakt gebracht und es wird möglich ein Drehmoment zu übertragen. Bei der Betrachtung der Füllung als Steuerungsvorgang wird der Kupplungsdruck als Führungsgröße $w(t)$ anhand eines inversen Streckenmodells G_s^{-1} in den Ventilstrom als Stellgröße $u(t)$ umgerechnet, die weiter an die Hardware gegeben wird. Der Ventilstrom führt zu einer Änderung des Kupplungsdrucks als Regelgröße $x(t)$. Für eine optimale Steuerung wird ein sehr genaues inverses Streckenmodell benötigt, da aufgrund der fehlenden

Rückführung der Regelgröße keine Kompensation von unbekannten Störungen $z(t)$ in der Strecke G_s und Modellungenauigkeiten ausgeglichen werden können. Die Vorteile der Steuerung sind, dass sehr dynamische Vorgänge gesteuert werden können und keine Messung der Regelgröße notwendig ist. Von Nachteil ist der bereits angesprochene fehlende Ausgleich von unbekannten Störgrößen (Geering, 2013, S. 6).

„Eine Adaption wird implementiert, um interne und externe Störeinflüsse mit großer Zeitkonstante, wie äußere Temperaturänderungen, Produktionsstreuung und Alterungseffekte, über eine Vorsteuerung zu minimieren“ (Fischer et al., 2016, S. 238).

Adaptionen gleichen Streuungen des Herstellungsprozesses (initiale Adaption) und Streuungen aufgrund von Verschleiß und Alterung über Lebensdauer (Adaption über Lebensdauer) aus. Dabei werden Parameter der Getriebesoftware an die aktuellen Bedingungen angepasst und stellen damit ein gleichbleibendes Systemverhalten sicher. Initial wird die Charakteristik der Magnet-Regelventile Strom zu Druck adaptiert. Über Lebensdauer werden dann z.B. die Parameter für die in der Getriebesoftware hinterlegte Reibungszahl (Reif, 2007, S. 198–201) und der Haftpunkt der Kupplung eingestellt. Geeignete Adaptionsgrenzen und Adaptionsgeschwindigkeiten werden dabei anhand aufwendiger Messungen an Prüfständen und in Fahrzeugen ermittelt. (Fischer et al., 2016, S. 238–239)

Diagnose-Funktionen überwachen die Signale der Software und melden bei Fehlverhalten des Getriebes einen Fehler. Ein Ausfall des Eingangswellendrehzahlsensors kann zum Beispiel über redundante Drehzahlsensoren wie Raddrehzahlsensoren festgestellt werden. Der Error-Handler gibt dann auf Grundlage des Fehlers eine Fehlerbehandlung an die Betriebsstrategie weiter.

Eine Kernaufgabe der Getriebesoftware ist die Kupplungsansteuerung. Diese ist für den Schaltablauf sowie das Anfahren verantwortlich und wird deshalb nachfolgend im Detail vorgestellt.

Kupplungsansteuerung. Abbildung 2.6 zeigt die Steuer- und Regelstrecke der Kupplungsansteuerung. Das Steuergerät liefert einen Solldruck an die Kupplungsaktuatorik. Diese wird standardmäßig über ein Magnet-Regel-Ventil mit einer Strom-Druck-Kennlinie umgesetzt, wobei der Druck über einen PID-Druckregler eingestellt wird. Im Normalfall wird zur Regelung der Druck im Kolben über einen Drucksensor direkt gemessen. Die Verzögerung zwischen Kupplungsdruck zu Kupplungsmoment liegt im ein- bis zweistelligen Millisekundenbereich und kann deshalb vernachlässigt werden.

Wird eine elektrisch betriebene und damit motordrehzahlunabhängige Hydraulikpumpe eingesetzt, so wird deren Soll-Strom über das Getriebesteuergerät gestellt.

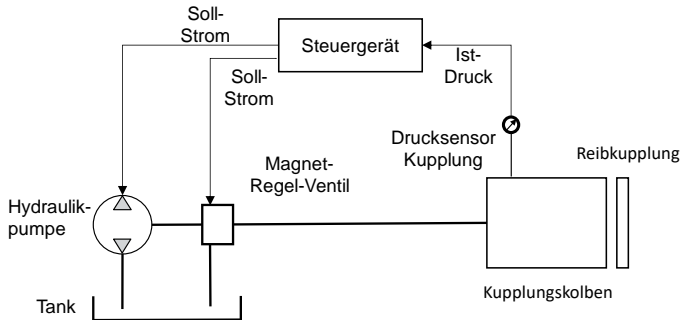


Abbildung 2.6: Steuer- und Regelstrecke Kupplungsansteuerung

Die übergeordnete Steuerung der Kupplung ist eine Zustandsmaschine mit den stationären Zuständen „Momentenvorgabe“ und „Ausgerückt“ sowie den transienten Zuständen „Anlegen“ und „Öffnen“ (vgl. Abbildung 2.7).

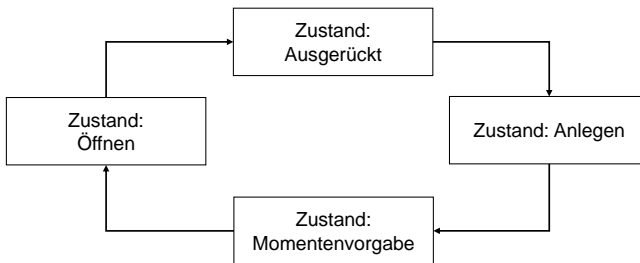


Abbildung 2.7: Zustandsmaschine der Kupplungsansteuerung (Fischer et al., 2016, S. 244).

Im Zustand Momentenvorgabe liegen die Kupplungslamellen an und es wird ein definiertes Drehmoment übertragen. Die anliegende Normalkraft der Kupplung kann aus dem Produkt des Kupplungsbetätigungsdruck p und der Kolbenfläche A_K berechnet werden. Eingesetzt in Gleichung 2.1 ergibt sich Folgendes für das Kupplungsmoment:

$$M_K = \mu r_m p A_K z \quad 2.6$$

Diese Gleichung kann dann nach dem effektiven Druckanteil p_{eff} in Abhängigkeit des geforderten Solldrehmoments M_{des} umgeformt werden. Wird zusätzlich der Anlegedruck p_0 addiert, so ergibt sich die Gleichung für den Solldruck p_{des} nach:

$$p_{des} = \frac{M_{des}}{A z r_m \mu} + p_0 \quad 2.7$$

Im Zustand „Anlegen“ findet die Kupplungsfüllung statt. Die Kupplungsfüllung ermöglicht damit den Übergang vom Zustand „Ausgerückt“ in den Zustand „Momentenvorgabe“. Im Zustand „Ausgerückt“ ist die Kupplung geöffnet. Um das Eindringen von Luft in die Aktuatorik zu verhindern, wird die Kupplung mit einem Minimaldruck von wenigen 100 mbar angesteuert. Dieser Druck ist deutlich kleiner als der zur Überwindung der Vorspannung benötigte Druck p_0 , weshalb die Kupplung sicher geöffnet ist. Im Zustand „Öffnen“ wird die Kupplung vom Zustand „Momentenvorgabe“ in den Zustand „Ausgerückt“ überführt. Dabei wird die Kupplung ab einem gestellten Drehmoment von 0 Nm entleert. (Fischer et al., 2016, 242–246)

2.1.4 Auslegung von Steuer- und Regelvorgängen

In der Getriebesoftware werden für die verschiedenen Regler hauptsächlich Standard-Regler mit Proportional (P), Integral (I) und Differential (D) Anteil verwendet. Für die Bestimmung dieser Parameter gibt es Einstellregeln z.B. nach Ziegler-Nichols (Ziegler & Nichols, 1942, S. 759–768), Ersatzkenngrößen der Übergangsfunktion der Regelstrecke oder die Methode des symmetrischen Optimums (Schmidt, 1991, S. 260–269). Als weitere Verfahren existieren Mehrzieloptimierungen. Diese optimieren mehrere Kostenfunktionen gleichzeitig. Im Normalfall kann keine alleinige optimale Lösung ermittelt werden, sondern eine Schar an gleichwertigen Lösungen, die eine Pareto-Front bilden. Zu den Verfahren der Mehrzieloptimierung zählen das Verfahren der gewichteten Summen, Normal Boundary Intersection (NBI), Normalized Normal Constraint (NNC) und Downhill-Simplex-Verfahren (DS) (Marler & Arora, 2004).

In der Getriebeentwicklung werden die Parameter der PID-Regler häufig anhand von Erfahrungswerten eingestellt oder durch Berechnungen in Abhängigkeit anderen Größen in der Software bestimmt.

Neben PID-Reglern können grundsätzlich auch modellbasierte Ansätze eingesetzt werden. Diese werden in zwei Arten unterteilt. Kompensationsregler kompensieren die Regelstrecke vollständig. Die genau definierte Übertragungsfunktion der Regelstrecke muss dafür bekannt sein (Isermann, 2008). Modellprädiktive Regler reagieren nicht wie ein PID-Regler auf einen aktuellen Zustand des Systems, sondern bestimmen anhand eines Modells das zukünftige Systemverhalten und leiten daraus eine Regelgröße ab, um das zukünftige Systemverhalten zu verbessern (Völz, 2016). Diese Ansätze benötigen jedoch ein sehr genaues Modell des Regelsystems. Dies führt zu Reglern mit hohen Rechenzeiten, die nur für langsame Regelungen eingesetzt werden können oder zu stark vereinfachten Reglern, die aber wiederum an Genauigkeit verlieren.

2.1.5 Modellbildung

Modelle werden in der Getriebesoftware für inverse Streckenmodelle für Steuerungen und virtuelle Sensoren benötigt. Zusätzlich werden für die Entwicklung der Getriebesoftware Modelle des realen Systems genutzt. Auf diese soll in dieser Arbeit aber nicht eingegangen werden, da der Fokus auf der Getriebesoftware liegt.

Streckenmodelle rechnen im Rahmen einer Steuerung eine Führungsgröße in eine Stellgröße um. Streckenmodelle haben hohe Anforderungen an ihre Genauigkeit, um eine optimale Ansteuerung zu ermöglichen. Ungenauigkeiten können dabei zu Komforteinbußen führen.

Um eine Rückmeldung aus dem realen System zu erhalten, werden Sensoren verwendet. Diese messen Drehzahlen, Drücke, Temperaturen oder Positionen. Aufgrund von geringem Bauraum und Kosten können nicht alle Größen über Sensoren gemessen werden. Deshalb werden virtuelle Sensoren benötigt, welche aus verschiedenen Größen in der Software eine Sensorersatzgröße berechnen.

Nach Isermann (2007) lassen sich die Methoden der Modellbildung in die theoretische (White-Box) und die experimentelle Modellbildung (Black-Box) einteilen (vgl. Abbildung 2.8).

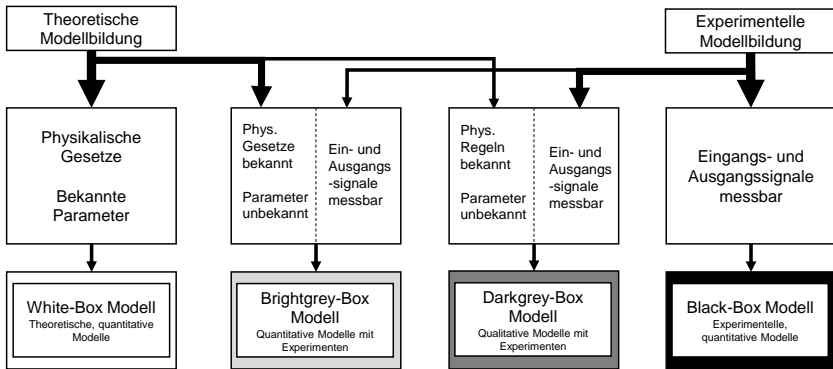


Abbildung 2.8: Methoden zur theoretischen und experimentellen Modellbildung, (Eigene Darstellung nach Isermann, 2007).

Bei der theoretischen Modellbildung wird das Modell anhand mathematisch formulierter Naturgesetze aufgestellt. Als Grundgleichungen werden konstitutive, phänomenologische Gleichungen, sowie Bilanz-, Entropiebilanz- und Schaltungsgleichungen eingesetzt. Der Modellbildungsprozess beginnt mit vereinfachten Annahmen, um die Modellbildung erst möglich zu machen. Die Parameter des Modells sind bekannt. Diese Modelle sind häufig umfangreich und komplex, so dass sie für weitere Anwendungen zum Beispiel mit Linearisierung oder Reduktion der Modellordnung vereinfacht werden. (Isermann, 2007, S. 47–48)

Bei der experimentellen Modellbildung werden Modelle rein auf Basis von Messdaten aus Experimenten bestimmt. Die Struktur des Systems ist dabei unbekannt (Black-Box). Mit Identifikationsmethoden werden mathematische Modelle gebildet, um den Zusammenhang zwischen Modelleingangs- und Modellausgangsgrößen darzustellen. Polynome, künstliche neuronale Netze (vgl. Kapitel 2.3.1) oder Fuzzy-Logiken finden Anwendung. (Isermann, 2007, S. 49)

Zwischen der theoretischen und experimentellen Modellbildung existieren Zwischenstufen. Beim Brightgrey-Box Modell ist die Modellstruktur aus physikalischen Gesetzen bekannt, die Parameter werden aber experimentell durch Parameteridentifikation bestimmt. Dieser Ansatz wird auch als semiphysikalisches Modell bezeichnet. Beim Darkgrey-Box Modell ist nur eine grobe Modellstruktur bekannt. Die Modellparameter werden mit Identifikationsmethoden bestimmt. (Isermann, 2007, S. 47–51)

Zur Bewertung der Modellgüte existieren verschiedene Metriken, die in Tabelle 2.1 dargestellt sind.

Tabelle 2.1 Metriken zur Bewertung der Modellgüte (Naser & Alavi, 2021)

| Kenngroße | Symbol | Formel |
|-----------------------------------|-------------------|--|
| maximaler absoluter Betragsfehler | Max | $\max_{i=1...K} y^{(i)} - \hat{y}^{(i)} $ |
| mittlerer absoluter Betragsfehler | $Mean$ | $\frac{1}{K} \sum_{i=1}^K y^{(i)} - \hat{y}^{(i)} $ |
| maximaler relativer Betragsfehler | ϵ_{Max} | $\max_{i=1...K} \left \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \right $ |
| mittlerer relativer Betragsfehler | ϵ_{Mean} | $\frac{1}{K} \sum_{i=1}^K \left \frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}} \right $ |
| mean-squared-error (MSE) | MSE | $\frac{1}{K} \sum_{i=1}^K (y^{(i)} - \hat{y}^{(i)})^2$ |
| root-mean-squared-error (RMSE) | $RMSE$ | $\frac{1}{K} \sqrt{\sum_{i=1}^K (y^{(i)} - \hat{y}^{(i)})^2}$ |
| Bestimmtheitsmaß | R | $\frac{\sum_{i=1}^K (y^{(i)} - \hat{y})^2 - \sum_{i=1}^K (y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^K (y^{(i)} - \hat{y})^2}$ |

In der Getriebeentwicklung wird der Ansatz der theoretischen Modellbildung in Form von Brightgrey-Box Modellen verfolgt. Parameter werden dabei aufwendig am Prüfstand mit Experimenten ermittelt. Da Modelle auch die Anforderungen der Steuergeräte hinsichtlich Echtzeitfähigkeit und geringem Speicher erfüllen müssen, können nur einfache Modelle auf dem Steuergerät hinterlegt werden, was wiederum die Modellgenauigkeit einschränkt (Fischer et al., 2016, S. 217–220).

2.1.6 Methoden zur Datenanalyse

In der Getriebefunktionsentwicklung von Fahrzeugen entstehen durch die Validierungsaktivitäten große Mengen an Simulations-, Prüfstands- und Fahrzeugmessdaten. Diese werden größtenteils mit statistischen Methoden ausgewertet. Allein durch Dauerlaufdaten entstehen sehr große Datenmengen. Der gesamte Datenumfang kann mit herkömmlichen Analysemethoden nicht mehr beherrscht werden. Die zielgerichtete Auswertung von Messdaten kann genutzt werden, um Problemursachen zu identifizieren und Handlungsempfehlungen für die Entwicklung abzuleiten (Kistner, Sanzenbacher, Munier & Fietkau, 2019).

2.1.7 Zwischenfazit

In diesem Kapitel wurden die Topologie und wichtige mechanische Komponenten des Hybridantriebsstrang, speziell für den parallelen Vollhybridantrieb beschrieben. Anschließend wurden die Steuerung und Regelung von Automatikgetrieben anhand der Softwarearchitektur und wichtigsten Elementen dargestellt. Die Kupplungsansteuerung als eine der zentralen Funktionen der Getriebesoftware wurde im Detail erläutert.

Die Auslegung der Steuerung und Regelung wird mit Systemverständnis und Erfahrung ausgeführt. Im Stand der Forschung existieren Regeln für die Regler-Auslegung, sowie heuristische Optimierungsansätze. Modelle in der Getriebesoftware für inverse Streckenmodelle oder virtuelle Sensoren werden nach der theoretischen Modellbildung bestimmt. Parameter werden aufwendig an Prüfständen über Messungen ermittelt. Die Datenanalyse von Messdaten findet häufig über statistische Methoden statt. Diese können aber nur eine begrenzte Menge an Informationen aus den verfügbaren Messdaten gewinnen.

2.2 Relevante Prozesse und Methoden der KaSPro

Die für diese Arbeit relevanten Prozesse und Methoden der KaSPro – Karlsruher Schule der Produktentwicklung (Albers et al., 2006) werden in diesem Kapitel vorgestellt.

Zunächst wird die SGE - Systemgenerationsentwicklung nach Albers beschrieben. Anschließend werden die Begriffe Validierung und Verifikation (V&V) im Produktentstehungsprozess definiert und der IPEK-XiL-Ansatz erläutert. Abschließend wird der C&C²-Ansatz vorgestellt.

Die Getriebe-Hardware- sowie Software-Entwicklung lässt sich anhand des Beschreibungsmodells der SGE - Produktgenerationsentwicklung beschreiben. Die SGE ermöglicht außerdem die formalisierte Datenerfassung in Produktentwicklungsprozessen und datengesteuerte Ansätze zur Produktoptimierung wie bspw. Entwicklungs- und Lebenszyklusdaten.

Der IPEK-XiL-Ansatz ermöglicht die Validierung eines Teilsystems in Zusammenspiel mit dem Gesamtsystem und weiteren interagierenden Systemen, wobei die Teilsysteme in unterschiedlicher Ausprägung (physisch, virtuell, gemischt) vorliegen können. Gleichzeitig ist die Validierung auf unterschiedlichen

Systemebenen durchführbar. Somit ermöglicht der Ansatz eine effiziente und effektive Validierung und wird deshalb in dieser Arbeit angewendet.

Der C&C²-Ansatz ist ein modellbasierter Ansatz zur Beschreibung von Gestalt- und Funktionszusammenhängen eines technischen Systems und wird in dieser Arbeit zur Analyse des betrachteten Systems verwendet.

2.2.1 Modell der SGE- Systemgenerationsentwicklung nach Albers

Um Innovationen zu ermöglichen, ist es notwendig, vorhandene Lösungen von Vorgängern, Wettbewerbern und branchenfremden Quellen als Referenzen zu nutzen – insbesondere für Produkte, die keinen direkten Vorgänger haben. Das Modell der SGE – Systemgenerationsentwicklung beschreibt diese Zusammenhänge und basiert auf den folgenden zwei Hypothesen (Albers & Rapp, 2021):

Erstens: Die Entwicklung eines neuen Systems basiert auf einem Referenzsystem, welches aus Elementen existierender oder geplanter sozio-technischer Systeme und den zugehörigen Dokumentationen besteht. Unter anderem kann das Referenzsystem frühere Produkte eines Unternehmens, Produkte von Wettbewerbern, entsprechende Teilmodelle von Produkten und Produktdokumentationen umfassen. Das Referenzsystem wird von den Entwicklern eines neuen Systems definiert, indem ausgewählte Referenzen als Grundlage und Ausgangspunkt für die Entwicklung ihres eigenen Systems verwendet werden. Das Referenzsystem wird während des Entwicklungsprozesses kontinuierlich erweitert, um neu aufkommende Fragestellungen zu adressieren. Die Gestaltung der Produkte erfolgt durch die Variation der RSE (Albers et al., 2019; Albers, Ebertz et al., 2020; Albers & Rapp, 2021). Auch Antriebsstränge und Softwarestände von Fahrzeuggetrieben werden auf Basis von Referenzsystemen entwickelt.

Zweitens: Basierend auf dem Referenzsystem wird ein neues System durch die Aktivitäten Übernahme-, Ausprägungs- und Prinzipvariation entwickelt und gestaltet (Albers, Bursac & Wintergerst, 2015; Albers, Hirschter et al., 2020).

Die drei Variationsarten werden wie folgt definiert (Albers, Hirschter et al., 2020):

- **Übernahmevariation (ÜV):** Ein RSE wird in die neue Systemgeneration übernommen, wobei das Innere des Elements als "Blackbox" betrachtet wird und auf Basis der Anforderungen der Systemintegration und Randbedingungen der Schnittstellen angepasst wird.

- **Ausprägungsvariation (AV):** Die Verbindung der RSE wird in der neuen Systemgeneration beibehalten. Das Lösungsprinzip bleibt im Vergleich zum Referenzsystem unverändert. Jedoch werden die Attribute des RSE variiert.
- **Prinzipvariation (PV):** RSE und ihre Verbindungen werden so verändert, dass Elemente und Verbindungen entfernt oder hinzugefügt werden. Dadurch wird ein neues Lösungsprinzip realisiert, was im Vergleich zum Referenzsystem neu ist.

Das Entstehen einer neuen Systemgeneration auf Grundlage des zugrunde liegenden Referenzsystems R_i wird durch den Operator V beschrieben, der alle Variationen enthält, durch die ein neues System entwickelt wird. Gleichung 2.8 beschreibt diesen Zusammenhang.

$$R_n \xrightarrow{V} G_n = US_n \cup AS_n \cup PS_n \quad 2.8$$

Dabei steht G_i für eine Systemgeneration. G_n ist die Systemgeneration, die derzeit in Entwicklung ist und der Markteinführung am nächsten steht. US_n sind Teilsysteme, die innerhalb von G_n als Übernahmevariation entwickelt werden, AS_n bzw. PS_n sind Teilsysteme, die durch Ausprägungsvariation bzw. Prinzipvariation entwickelt werden (Albers, Bursac & Wintergerst, 2015; Albers et al., 2019; Albers, Rapp et al., 2020).

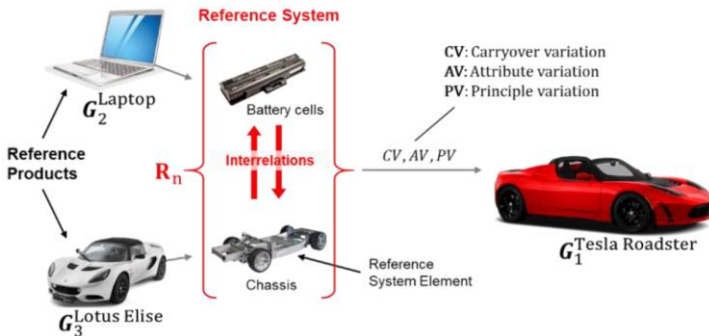


Abbildung 2.9: Das Referenzsystem im Modell der Systemgenerationsentwicklung (SGE) (Albers et al., 2019). Der Tesla Roadster übernimmt das Chassis des Lotus Elise (Übernahmevariation, CV), die Batteriezellen aus dem Referenzprodukt Laptop wurden in einer neuen Konfiguration integriert (AV).

Um zusätzliche Informationen über eine Systemgeneration zu berücksichtigen, kann die Nomenklatur durch weitere Indizes erweitert werden, wie zum Beispiel über den vorgesehenen Kunden oder Markt (Albers, Fahl et al., 2020) .

$$G_i^{\{Produktlinie, Kunde, Nutzer, \dots\}}$$

2.9

Unterschiedliche Inkremente, Iterationen oder Reifegrade in der Entwicklung einer neuen Systemgeneration entstehen ebenfalls durch die drei Variationsarten. Diese Inkremente stellen somit eine Generation innerhalb einer Systemgeneration dar und werden deshalb als Entwicklungsgenerationen E bezeichnet (Albers, Haug et al., 2016). In der Getriebeentwicklung können Software-Sprints und Hardware-Varianten als Entwicklungsgenerationen verstanden werden.

Die SGE ermöglicht außerdem die formalisierte Datenerfassung in Produktentwicklungsprozessen und datengesteuerte Ansätze zur Produktoptimierung wie bspw. durch Entwicklungs- und Lebenszyklus- oder Konkurrenzprodukt Daten.

2.2.2 Verifikation und Validierung im Produktentstehungsprozess

Nach Albers (2010) stellt die Validierung die zentrale Aktivität im Produktentstehungsprozess dar. Die Validierung soll frühzeitig und kontinuierlich erfolgen, denn späte Änderungen im Produktentstehungsprozess sind oftmals nur mit großem Aufwand verbunden möglich (Albers, Behrendt, Klingler & Matros, 2016, S. 543). Von einer Entwicklungsphase zur nächsten steigen dabei die Kosten um Faktor zehn, was durch die Zehnerregel (Rule of Ten) beschrieben wird (Ehrlenspiel, 2009, S. 140).

Albers beschreibt die Produktentwicklung auf Basis des ZHO-Tripels von Ropohl (Zielsystem, Handlungssystem, Objektsystem) (Ropohl 1975). Das Zielsystem enthält die geplanten Eigenschaften und Randbedingungen eines Produkts, jedoch nicht die Lösung als solche. Das Zielsystem wird nach der initialen Beschreibung im Laufe der Produktentwicklung kontinuierlich erweitert und angepasst. Das Objektsystem enthält alle Dokumente und Artefakte, die als Teillösung während des Produktentstehungsprozesses anfallen, sowie das endgültige Produkt. Das Handlungssystem ist ein sozio-technisches System, welches das Ziel- und Objektsystem generiert und alle für die Realisierung einer Produktentstehung notwendigen Ressourcen, Aktivitäten, Methoden und Prozesse enthält.

Das Ziel- und das Objektsystem werden in der Produktentwicklung iterativ entwickelt und unterliegen deshalb einer gewissen Dynamik. Um diesen iterativen Charakter in der Produktentwicklung abzubilden, ergänzen Albers, Ebel und Lohmeyer (2012) das ursprüngliche ZHO-Modell um die Entwicklungsaktivitäten Analyse und Synthese zum Erweiterten ZHO-Modell. Die Validierung und Kreation sind in diesem Modell zentrale Aktivitäten. Die Validierung beinhaltet dabei die Bewertung von Elementen des Objektsystems, die Objektivierung der Stakeholder-Erwartungen und den Vergleich der Elemente des Objekt- und Zielsystems (Verifikation). Abbildung 2.10 stellt das erweiterte ZHO-Tripel dar. (Albers, Behrendt et al., 2016)

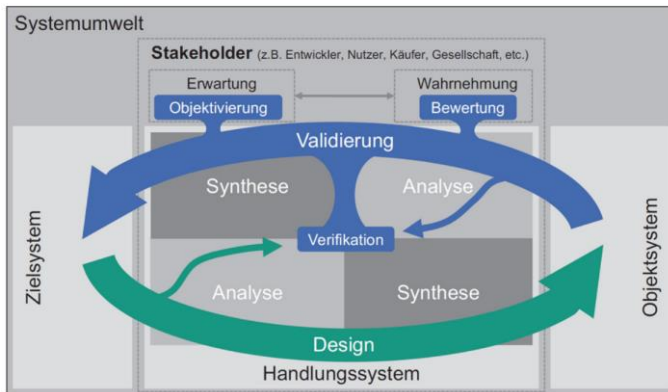


Abbildung 2.10: Design und Validierung im Produktentstehungsprozess (Albers, Behrendt et al., 2016, S. 545).

Nach der VDI-Richtlinie 2206 versteht man unter Verifikation „die Überprüfung [...], ob eine Realisierung (z. B. ein Software-Programm) mit der Spezifikation (in diesem Fall mit der Algorithmenbeschreibung) übereinstimmt“ (VDI 2206).

Unter Validierung ist „[...] die Prüfung zu verstehen, ob das Produkt für seinen Einsatzzweck geeignet ist bzw. den gewünschten Wert erzielt. Hier geht die Erwartungshaltung des Fachexperten und des Anwenders ein“ (VDI 2206).

Folglich stellt die Verifikation den reinen Abgleich zwischen dem Ziel- und dem Objektsystem dar, um dessen Konformität zu beurteilen. Bei der Validierung müssen Objekt- und Zielsystem mit den eigentlichen Bedarfen und zukünftigen Anwendungsfällen verglichen werden. Dabei generiert die Validierung durch die Analyse des Objektsystems neues Wissen und das Zielsystem wird in einem Syntheseschritt dementsprechend fortlaufend um diese Erkenntnisse erweitert und angepasst. Das erweiterte ZHO-Modell (Albers & Braun, 2011) verdeutlicht diesen

kontinuierlichen Wechsel von Kreation und Validierung als geschlossenen Kreislauf (Albers, Behrendt et al., 2016).

Die Validierung besteht aus den drei grundlegenden Aktivitäten Bewertung, Objektivierung und Verifikation (Albers, Matros, Behrendt & Jetzinger, 2015).

Ein Test wird nach Ebel (2015) wie folgt definiert:

„Ein Test ermittelt Systemeigenschaften eines [zu untersuchenden] Systems [. . .] und liefert Erkenntnisse über das System, insbesondere ob das System zuvor definierte Ziele, Anforderungen oder Hypothesen ganz, teilweise oder nicht erfüllt. Ein Test umfasst stets einen Testfall, eine Testumgebung und eine Testinterpretation“ (Ebel, 2015, S. 136).

Ein Testfall ist ein repräsentatives Modell eines Kollektivs von Anwendungsfällen. Er definiert die Start- und Randbedingungen, die Eingangsgrößen eines Systems sowie das erwartete Systemverhalten (Albers, Behrendt et al., 2016, S. 554). Die Testumgebung beinhaltet alle physischen und virtuellen Modelle bzw. Originale, die notwendig sind, um einen oder mehrere Testfälle durchzuführen und das erwartete Systemverhalten zu erfassen (Ebel, 2015, S. 136).

Die Testinterpretation „überführt das in einem durchgeführten Testfall erfasste Systemverhalten in eine oder mehrere zugrundeliegende Systemeigenschaften und liefert Erkenntnisse über das System und den Test, insbesondere ob zuvor definierte Ziele, Anforderungen oder Hypothesen ganz, teilweise oder nicht erfüllt werden“ (Ebel, 2015, S. 136). Eine Testinterpretation muss immer in Abhängigkeit der Testumgebung durchgeführt werden, da z.B. Verkürzungseigenschaften oder Messfehler berücksichtigt werden.

In Abbildung 2.11 werden Test, Testfall, Testumgebung und Testinterpretation mit Validierungsbedarfen und -zielen verknüpft. Aus dem Produktprofil werden Validierungsbedarfe und daraus wiederum Validierungsziele abgeleitet. Aus den Validierungszielen werden Testfälle definiert. Aus der Durchführung der Testfälle in Testumgebungen resultieren Testergebnisse, die wiederum interpretiert werden können. Der direkte Abgleich der Testergebnisse mit den Validierungszielen stellt die Verifikation dar, der Abgleich der interpretierten Testergebnisse mit den Validierungsbedarfen die Validierung. Es muss überprüft werden, ob die Validierungsziele und -bedarfe angepasst werden müssen oder ob weitere oder andere Validierungsziele und -bedarfe an die Testumgebung gestellt werden müssen, um die identifizierte Wissenslücke schließen zu können. (Albers, Behrendt et al., 2016, S. 554–556; Mandel et al., 2020)

Um den Validierungsaufwand möglichst gering zu halten, wird die gesamte Bandbreite der späteren Anwendung eines Produktes in einem möglichst zeiteffizienten Testfall abgebildet. Ein Testfall ist somit immer ein Modell eines Teils der späteren Anwendung. Es muss deshalb ein optimaler Kompromiss zwischen Validierungsaufwand und Sicherheit gefunden werden. (Albers, Behrendt et al., 2016, S. 554–556)

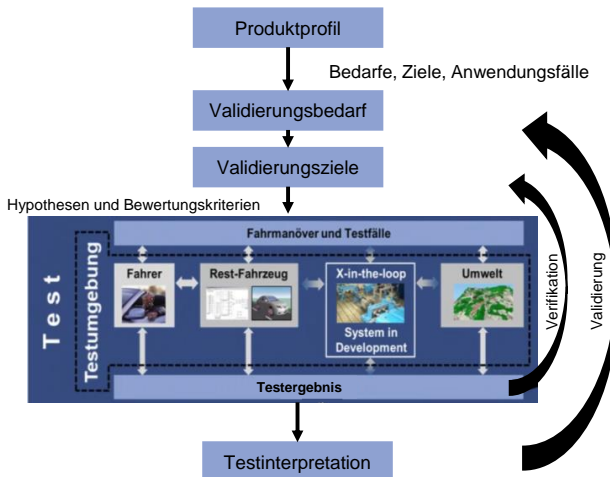


Abbildung 2.11: Korrelation zwischen Produkt-Kreation und Validierung (Eigene Darstellung nach Mandel et al., 2020).

2.2.3 IPEK-X-in-the-Loop-Ansatz (IPEK-XiL-Ansatz)

Der IPEK-XiL-Ansatz nach Albers und Düser (2010) beschreibt das grundlegende Verständnis, ein Teilsystem im Zusammenspiel mit dem Gesamtsystem, der Umgebung und anderen interagierenden Systemen (Fahrer) zu betrachten und damit zu validieren.

„Der IPEK-X-in-the-Loop-(IPEK-XiL)-Ansatz ... greift [die Ansätze des Model-in-the-Loop (MiL), Software-in-the-loop (SiL) und Hardware-in-the-Loop (HiL)] auf, integriert die jeweiligen Vorteile und erweitert diese konsequent um die Belange der Mechanik bzw. Mechatronik sowie der Entwickler aus unterschiedlichen Fachdisziplinen“ (Albers, Behrendt et al., 2016, S. 559).

Diese mit dem zu untersuchenden System interagierenden Systeme, beispielsweise Umwelt, Fahrer und Rest-Fahrzeug, werden als *Connected Systems* bezeichnet. Wie diese interagierenden Systeme integriert werden (virtuell oder physisch), muss je nach Validierungsziel von Fall zu Fall entschieden werden. Dabei können sich unterschiedliche Kombinationen aus rein virtuellen, gemischt physisch-virtuellen und rein physischen Validierungsformen ergeben. Diese verschiedenen Systeme, die Testcharakteristika und die Systemebenen der Subsysteme bilden das IPEK-XiL-Framework (vgl. Abbildung 2.12).

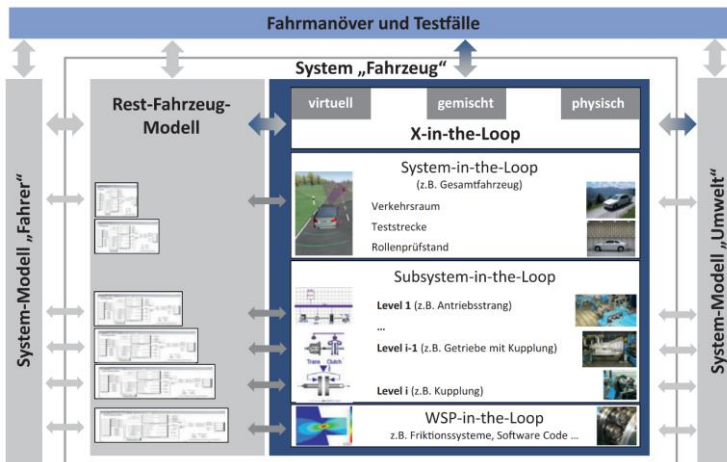


Abbildung 2.12: IPEK-XiL-Framework (Albers, Behrendt et al., 2016).

Die zu untersuchende Einheit „X“ wird als *System-in-Development* (SiD) bezeichnet und ist das Teilsystem, das sich in Entwicklung befindet bzw. welches ein Entwickelnder zu verantworten hat. Das SiD kann dabei in unterschiedlichen Systemebenen vorliegen. Im Kontext der Antriebsstrangentwicklung reicht dies von der Untersuchung von Wirkflächenpaaren (z.B. Frictionssysteme), über Teilsysteme (z.B. Getriebe) bis hin zu Gesamtsystemen (z.B. Gesamtfahrzeug) (vgl. Abbildung 2.12).

Wird ein Teilsystem nicht entwickelt, sondern die Validierungsaktivitäten zielen darauf ab, das Teilsystem zum Erkenntnisgewinn zu analysieren, so wird dieses als *System-under-Investigation* (Sul) bezeichnet. Das Sul kann dabei ein Teilsystem des SiD, das gesamte SiD, oder auch ein Nachbarsystem sein (Albers, Behrendt et al., 2016).

Um Teilsysteme über die verschiedenen Domänen einer Validierungskonfiguration hinweg zu verbinden, werden spezielle Systeme benötigt. Die hierfür eingesetzten Systeme werden als Koppelsysteme bezeichnet, die nach Pinner (2017) wie folgt definiert werden:

„Modelle und Koppelsysteme sind Teil von Validierungsumgebungen. Modelle (virtuelle, physische oder gemischte) repräsentieren relevantes Systemverhalten bei der Durchführung von Validierungsaktivitäten. Koppelsysteme können notwendig sein, um Modelle miteinander zu verbinden. Wenn der Output eines Modells als Input für ein weiteres Modell dient, sind Koppelsysteme all jene Systeme, die zwischen den Modellen angeordnet sind“ (Pinner, 2017, S. 101).

Koppelsysteme verbinden somit Modelle miteinander und ermöglichen Interaktion zwischen diesen. Diese Verbindung wird als Koppelfunktion bezeichnet. Modelle und deren Funktion werden nicht verändert, sondern es wird lediglich eine Kompatibilität gewährleistet (Pinner, 2017, S. 102). Ein in dieser Arbeit vorgestellter Ansatz zur Modellierung von virtuellen Sensoren (s. Kapitel 6.3) kann genutzt werden, um Koppelsysteme zwischen physischen Eingangsgrößen und einer virtuellen Größe darzustellen. Ein Koppelsystem wandelt in einem in mehreren Ebenen unterteilten Prozess virtuelle Eingangsgrößen in physische Ausgangsgrößen und umgekehrt (vgl. Abbildung 2.13). Diese Ebenen haben Einfluss auf den Signalfuss und können somit zu einem gewünschten oder ungewünschten Modellverhalten führen. Das Koppelsystem wird dabei keiner physischen oder virtuellen Domäne des IPEK-XiL Frameworks zugeordnet und ist lediglich für die Transformation der Signale zuständig. Eine detaillierte Beschreibung kann im Forschungsbericht von Pinner (2017, S. 104-110) nachgelesen werden.

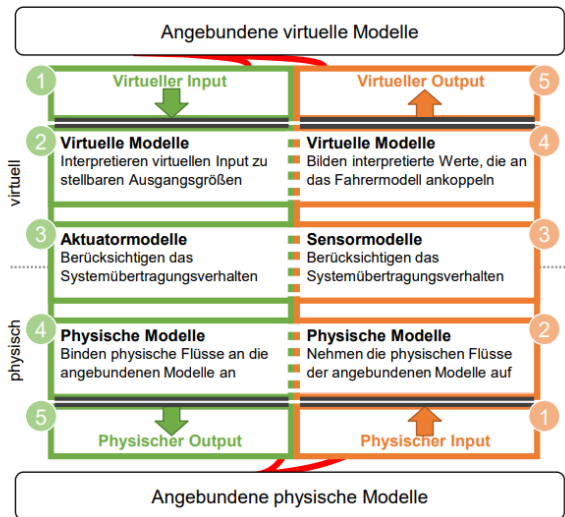


Abbildung 2.13: Beschreibungsmodell Koppelsystem (Pinner, 2017, S. 105).

2.2.4 Contact und Channel Ansatz (C&C²-Ansatz)

Der C&C²-Ansatz nach Albers, Alink, Matthiesen & Thau (2008) ist ein modellbasierter Ansatz zur Beschreibung von Gestalt- und Funktionszusammenhängen eines technischen Systems. Er ermöglicht die durchgängige Modellierung technischer Systeme auf beliebigen Detaillierungs- und Abstraktionsebenen und unterstützt den Analyse- und Syntheseprozess.

Die Beschreibung der Funktionen technischer Systeme kann nach Albers und Wintergerst (2014) mit folgenden Kernelementen erfolgen:

- Wirkflächenpaare (WFP) werden aus zwei Wirkflächen gebildet, die dauerhaft oder gelegentlich über beliebig geformte Oberflächen fester Körper oder generalisierten Grenzflächen von Flüssigkeiten, Gasen oder Feldern in Wechselwirkung stehen und am Energie-, Stoff- und/oder Informationsaustausch beteiligt sind.
- Leitstützstrukturen (LSS) sind Volumina von festen Körpern, Flüssigkeiten, Gasen oder felddurchsetzten Räumen, die genau zwei Wirkflächenpaare

verbinden und eine Leitung von Stoff, Energie und/oder Information zwischen diesen ermöglichen.

- Connectoren (C) sind Modellierungselemente, die die Wirkung und die Zustandseigenschaften der Umgebung darstellen, die für die Funktion eines Systems relevant sind.

Das Wirknetz bezeichnet die Summe aller Kernelemente WFP, LSS und C innerhalb des Gestaltungsraums.

Abbildung 2.14 fasst die Kernelemente sowie Grundhypothesen des C&C²-Ansatzes zusammen und zeigt die Modellierung am Beispiel eines Zahnradpaares, das die Funktion „Drehmoment übertragen“ umsetzt.

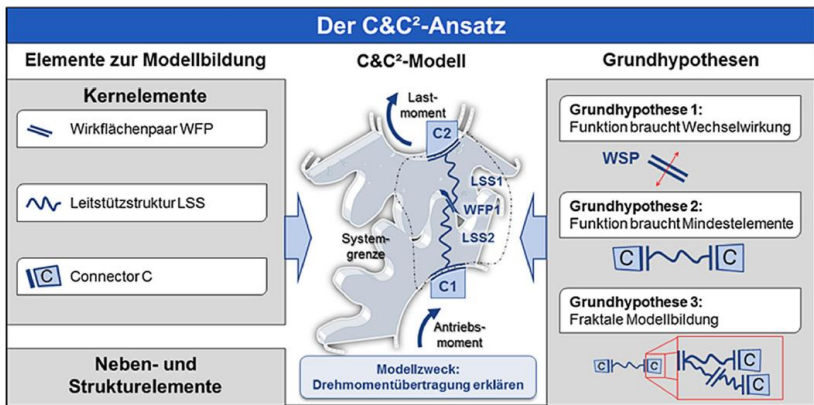


Abbildung 2.14: Übersicht über den C&C²-Ansatz und seine Elemente (Tröster, Klotz, Rapp, Ott & Albers, 2021).

Die graphische Darstellung der Funktions- und Gestaltzusammenhänge ermöglicht außerdem ein gemeinsames Verständnis innerhalb des PE-Teams.

Der C&C²-Ansatzes wird in dieser Arbeit zur Analyse der Kupplungsansteuerung genutzt, um ein Verständnis der Wirkzusammenhänge zu schaffen und Einflussfaktoren auf die Ansteuerung abzuleiten.

2.2.5 Zwischenfazit

Das Beschreibungsmodell der SGE - Systemgenerationsentwicklung nach Albers setzt auf den zwei Hypothesen auf, dass die Entwicklung eines neuen Systems auf

einem Referenzsystem basiert, welches aus Elementen existierender oder geplanter sozio-technischer Systeme besteht und deren Entwicklung durch die gezielte Kombination der Aktivitäten Übernahme-, Ausprägungs- und Prinzipvariation erfolgt. Dieser Ansatz lässt sich auf die Entwicklung der Steuerung und Regelung von Fahrzeuggetrieben anwenden.

Die Validierung technischer Systeme ist die zentrale Aktivität im Produktentstehungsprozess. Die Begriffe Validierung, Verifikation sowie das erweiterte ZHO-Tripel werden erläutert. Außerdem werden in diesem Zusammenhang die Begriffe Test, Testfall, Testumgebung und Testinterpretation definiert, da die in Kapitel 6.5 entwickelte Methode hierauf zurückgreift.

Der IPEK-XiL-Ansatz beschreibt das grundlegende Verständnis, ein Teilsystem im Zusammenspiel mit dem Gesamtsystem, der Umgebung und anderen interagierenden Systemen (Fahrer) zu betrachten und damit zu validieren. Somit ermöglicht der Ansatz eine durchgängige Validierung auf unterschiedlichen Systemebenen und damit eine effiziente und effektive Validierung. Koppelsysteme, auf die in Kapitel 6.3 Bezug genommen wird, verbinden als Teil von Validierungsumgebungen Modelle miteinander und ermöglichen Interaktion zwischen diesen.

Der C&C²-Ansatz ermöglicht die Beschreibung von Gestalt- und Funktionszusammenhängen eines technischen Systems und wird in Kapitel 5.2.3 zur Systemanalyse herangezogen.

2.3 Maschinelles Lernen (ML)

In diesem Kapitel werden die ML-Methoden vorgestellt und deren mathematische Grundlagen erläutert. Diese bilden die Ausgangsbasis, auf welcher später die ML-Methoden im Kontext der Funktionsentwicklung paralleler hybrider Fahrzeuggetriebe adaptiert werden.

ML ist ein Teilgebiet der Künstlichen Intelligenz (vgl. Abbildung 2.15).

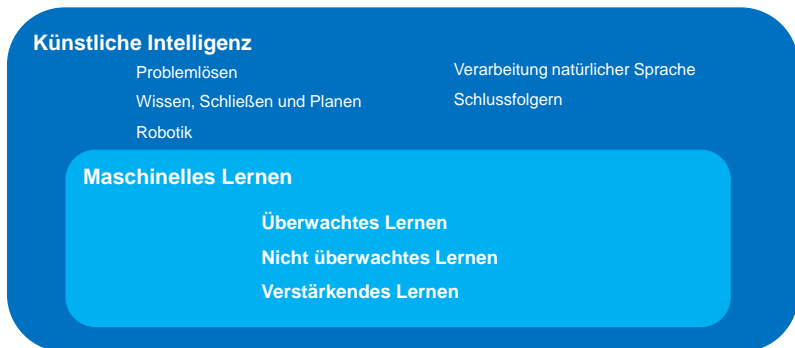


Abbildung 2.15: Einordnung maschinelles Lernen im Kontext der Künstlichen Intelligenz (Eigene Darstellung nach Russell & Norvig, 2012).

Maschinelles Lernen besteht aus den drei Lernverfahren:

- Überwachtes Lernen (Supervised Learning),
- Nicht überwachtes Lernen (Unsupervised Learning),
- Und verstärkendes Lernen (Reinforcement Learning).

Beim überwachten Lernen wird anhand von Eingangs- und Ausgangsdaten eine Abbildungsfunktion zwischen diesen erlernt. Beim nicht überwachten Lernen werden Muster bzw. Klassen in Daten erkannt. Beim verstärkendem Lernen lernt ein Agent eine Strategie anhand von Verstärkungen (Belohnungen) oder Bestrafungen. Diese Lernverfahren werden im Folgenden detailliert vorgestellt.

2.3.1 Überwachtes Lernen

Nach Russel & Norvig lautet die Aufgabe des überwachten Lernens:

„Für eine Trainingsmenge mit N Beispielen von Eingabe/Ausgabe-Paaren $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ in der jedes y_j durch eine unbekannte Funktion $y = f(x)$ generiert wurde, ist eine Funktion h zu entdecken, die die wahre Funktion f annähert“ (Russell & Norvig, 2012, S. 811).

Für diese Aufgabe existieren verschiedene Methoden:

- Lineare Regression
- Entscheidungsbäume

- Support-Vector Machines
- Gauß-Prozessmodelle
- Künstliche neuronale Netze

Da in dieser Arbeit künstliche neuronale Netze (KNN) zur Anwendung kommen (vgl. Kapitel 7.1), wird diese Methode nachfolgend anhand Russell und Norvig (2012, S. 845-856) vorgestellt.

KNN sind der Struktur des menschlichen Gehirns nachempfunden und bestehen aus einzelnen Neuronen.

Künstliches Neuron. Abbildung 2.16 zeigt eine einfache mathematische Darstellung eines Neurons nach McCulloch und Pitts (1943).

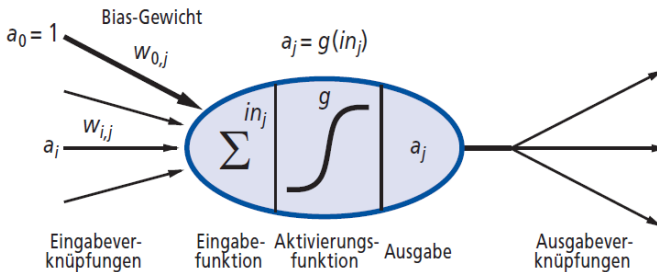


Abbildung 2.16: Einfaches mathematisches Modell eines Neurons (Russell & Norvig, 2012, S. 846).

Die Aktivierung eines Neurons berechnet sich aus dem Gewichtsvektor $w_{i,j}$ und dem Eingangsvektor a_i , auf die die Aktivierungsfunktion g angewendet wird. Vereinfacht ausgedrückt gibt ein Neuron eine Ausgabe weiter, wenn eine lineare Kombination von Eingangsgrößen einen Schwellwert übersteigt.

$$a_j = g(in_j) = g(\sum_{i=0}^n w_{i,j} a_i) \quad 2.10$$

Die Verwendung von nichtlinearen Aktivierungsfunktionen ermöglicht die Abbildung von nichtlinearen Zusammenhängen durch ein KNN. Häufig verwendete Aktivierungsfunktionen sind die Sigmoidfunktion, der Tangenshyperbolicus und Rectified Linear Unit, die in Abbildung 2.17 dargestellt sind. Dabei besitzen logistische Aktivierungsfunktionen den Vorteil der Differenzierbarkeit.

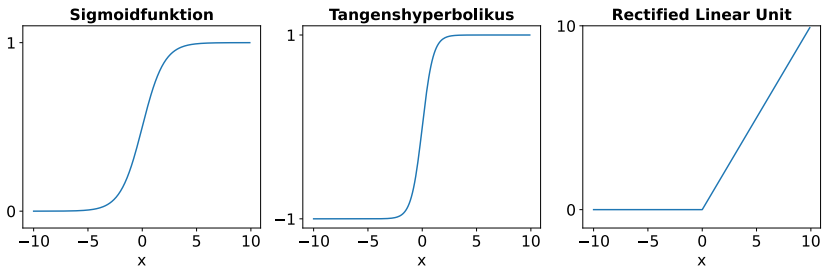


Abbildung 2.17: Häufig verwendete Aktivierungsfunktionen

Künstliche neuronale Netz (KNN). Bereits McCulloch und Pitts (1943) beschreiben, dass sich jede gewünschte Funktionalität abbilden lässt, wenn sehr viele Neuronen in Netzen beliebiger Tiefe miteinander verbunden werden.

Diese Annahme führte zur Entwicklung von KNN. Diese bestehen aus Schichten von Neuronen, die über gerichtete Verknüpfungen, auch Kanten genannt, miteinander verbunden sind. Jede Kante besitzt eine Gewichtung und aus allen Gewichten setzt sich der Gewichtsvektor $w_{i,j}$ zusammen.

Die erste Schicht wird als Eingangsschicht, die letzte Schicht als Ausgangsschicht bezeichnet. Die Anzahl der Neuronen dieser Schichten entspricht jeweils der Anzahl an Ein- bzw. Ausgangsgrößen, die abgebildet werden.

Training. Während des Trainingsvorgangs wird der Fehler zwischen Trainingsdaten und Vorhersage des KNN minimiert. Der Gesamtdatensatz wird in drei einzelne Datensätze unterteilt: Trainings-, Validierungs- und Testdatensatz (vgl. Abbildung 2.18). Der Trainingsdatensatz wird für das Training verwendet. Um ein Überanpassen des Netzes auf den Trainingsdatensatz zu verhindern, wird dieses regelmäßig mit dem Validierungsdatensatz geprüft. Mit diesem werden auch die Hyperparameter des Netzes optimiert. Nach dem abgeschlossenen Training und der Hyperparameteroptimierung wird die Leistungsfähigkeit mit einem für das Netz noch unbekannten Testdatensatz bestimmt (J. M. Zhang, Harman, Ma & Liu, 2020).

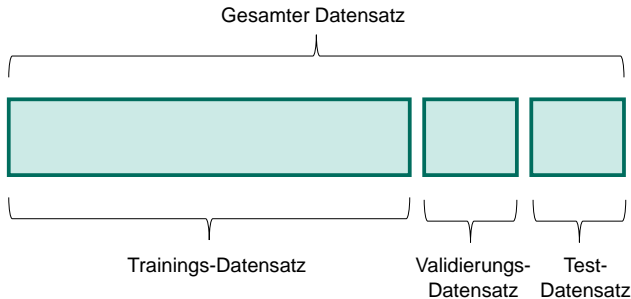


Abbildung 2.18: Graphische Darstellung von Trainings-, Validierungs- und Testdatensatz

Hier ist es wichtig anzumerken, dass sich der Begriff Validierungsdatsatz von dem Validierungsbegriff nach Albers (vgl. Kapitel 2.2.1) unterscheidet.

Albers beschreibt die „kontinuierliche Validierung im Entwicklungsprozess [als] ein steter Abgleich zwischen den Zielen und Zwecken des Produktes sowie dem jeweils erreichten Entwicklungsstand“ (Albers, Behrendt et al., 2016, S. 541).

Validierung im Kontext von KNN beschreibt die Berechnung des Fehlers (in regelmäßigen Abständen des Trainings) auf einen Validierungsdatsatz. Es kann somit noch keine Aussage über die Zweckerfüllung getroffen werden.

Das KNN beginnt das Training mit randomisierten Netzwerkparametern. Diese Parameter werden während des Trainings regelmäßig aktualisiert. Der Fehler wird zwischen den Netzwerkausgängen und den Ausgangsdaten des Datensatzes bestimmt. Unter Verwendung eines Backpropagation-Ansatzes wird der Ausgabefehler dann auf die Parameter in den verborgenen Schichten zurückgerechnet. Die Parameter mit dem größten Einfluss auf den Fehler werden dann so angepasst, dass der Fehler minimiert wird. Ein häufig verwendetes Optimierungsverfahren ist das Adaptive Momentum Estimation (Adam)-Verfahren (Kingma & Ba, 2015, S. 1-3), dessen Verwendung auch im Rahmen eines Vergleichs verschiedener Optimierungsverfahren empfohlen wird (Ruder, 2016). Das vollständige Durchlaufen des kompletten Trainingsdatensatzes wird als Trainingsepoche bezeichnet. Üblicherweise wird über viele Epochen trainiert.

Neben den Netzwerkparametern, wie z.B. den Gewichten $w_{i,j}$, gibt es noch Hyperparameter, wie z.B. die Netzwerkarchitektur oder die Stapelgröße.

Die Stapelgröße teilt den Trainingsdatensatz in einzelne Stapel auf, welche dann in einer Trainingsepoche zum Training genutzt werden. Das Zerlegen des Gesamtdatensatzes in mehrere Teile minimiert den benötigten Speicher beim Trainieren und beschleunigt das Training. Eine zu kleine Stapelgröße kann zu Ungenauigkeiten des Modells führen.

KNN können in unterschiedlichen Netzwerkstrukturen vorliegen:

Vorwärts gerichtete neuronale Netze (FNN). Diese Netzwerke besitzen Verknüpfungen der Neuronen nur in eine Richtung und bilden somit einen gerichteten azyklischen Graphen. Ein solches Netz gibt nur die Funktion seiner aktuellen Eingabe aus.

Rekurrente neuronale Netze (RNN). Dagegen führt ein rekurrentes neuronales Netz seine Ausgaben wieder als Eingaben in das Netz zurück. Das bedeutet, dass nicht nur der aktuelle Zeitschritt zur Berechnung der Ausgabe herangezogen wird, sondern auch Zeitschritte der Vergangenheit. Die Sequenzlänge beschreibt dabei die Anzahl dieser vergangenen Zeitschritte. Durch rekurrente Netze wird die Abbildung dynamischer und zeitabhängiger Prozesse möglich. Bildet ein rekurrentes Netz eine hohe Anzahl an zurückliegender Zeitschritte ab, so resultiert dies in einer hohen Anzahl an Schichten, die bei dem Backpropagation-Verfahren berücksichtigt werden müssen. Dies führt jedoch zu dem Problem, dass der Fehler nicht mehr eindeutig zugeordnet werden kann. Die Verwendung von Long-Short-Term-Memory Cells (LSTM) kann dieses Problem lösen.

Long-Short-Term-Memory (LSTM). Ein LSTM-Neuron (Hochreiter & Schmidhuber, 1997) ist eine Neuronenart, die vergangene Eingänge in internen Zuständen ablegen kann. Ein LSTM ermöglicht somit "Erinnerung" an frühere Zeitschritte, was einem Kurzzeitgedächtnis entspricht (daher auch die Namensgebung). Dafür besitzt das LSTM drei sogenannte Tore:

- Forget Gate: Bestimmt den Einfluss des vorherigen Zustands, der vergessen wird.
- Input Gate: Bestimmt den Einfluss eines neuen Zustands.
- Output Gate: Bestimmt den Einfluss des Zustands für den Ausgang der Zelle.

Zeitlich faltende neuronale Netze (TCN). Diese Netzwerke nutzen als Grundlage eindimensionale faltende neuronale Netze (CNN), die für eine Eingangssequenz der Eingangsgrößen eine Ausgangssequenz vorhersagen. CNN werden in der Bild- und Audioverarbeitung verwendet und können aufgrund einer Faltung eine sehr große

Anzahl an Eingangsgrößen verarbeiten. TCN zeigen im Vergleich zu rekurrenten neuronalen Netzen bessere Ergebnisse für maschinelle Übersetzung und Text zu Sprach-Systemen. Die konkrete Architektur wird von Bai, Kolter und Koltun (2018) beschrieben.

Methoden zur Auswahl von Modelleingangsgrößen. Zur Auswahl von Modelleingangsgrößen gibt es zwei Hauptverfahren: Filter-Methoden und Wrapper-Methoden. Filter-Methoden nutzen Korrelationsverfahren, um aus allen potentiellen Eingangsgrößen die für eine optimale Vorhersage relevanten Größen zu bestimmen. Filter-Methoden sind auf hochdimensionale Datensätze einfach skalierbar. Nachteil ist, dass Filter-Methoden vor der Modellauswahl durchgeführt werden und damit der Einfluss der Modellauswahl auf die Eingangsgrößenauswahl nicht berücksichtigt wird. Wrapper-Methoden lösen dieses Problem und testen alle Kombinationen aus Eingangsgrößen und Modellen. Von Nachteil ist der hohe Rechenaufwand (H. Liu, Motoda, Setiono & Zhao, 2010).

Skalierungsmethoden. Da die Modelleingangsgrößen unterschiedliche Größenordnungen besitzen, werden die Daten skaliert. Die zwei Hauptverfahren sind Standardisierung und Normalisierung. Bei der Standardisierung wird der Mittelwert einer Größe auf 0 gesetzt, die Standardabweichung auf 1. Bei der Normalisierung werden die Werte zwischen 0 und 1 skaliert. Die Min-Max-Normalisierung normalisiert jede Variable des Datensatzes auf den Bereich zwischen Minimum und Maximum. Die Quantil-Normalisierung normalisiert jede Variable des Datensatzes auf den Interquartilabstand (z.B. Abstand zwischen 25. und 75. Anteil). Dadurch wird der Einfluss von Ausreißern reduziert (Duboue, 2020, S. 35–41).

Hyperparameteroptimierung. Für die Hyperparameteroptimierung von ML-Modellen existiert eine Vielzahl von Verfahren. Nachfolgend werden die am häufigsten verwendeten Verfahren beschrieben. Bei der *Grid-Search* wird der Parameterraum vollständig abgesucht und jede Hyperparameterkombination getestet. Hierfür muss der Nutzende Vorwissen über den Parameterraum besitzen. Die *Random-Search* führt eine zufällige Suche über den Parameterraum durch. Die Suche endet, wenn eine bestimmte Anzahl an Versuchen oder eine bestimmte Genauigkeit erreicht wird. *Random-Search* erbringt bessere Ergebnisse, wenn ein Parameter ungleichmäßig verteilt ist. Die *Bayes'sche Optimierung* ist eine globale Optimierungsmethode bei der ein probabilistisches Modell der Zielfunktion erstellt und dieses anhand weiterer Datenpunkte iterativ verbessert wird. *Populationsbasierte Ansätze* nutzen genetische Algorithmen, um die optimalen Modellparameter zu identifizieren. (Yu & Zhu, 2020)

Local Interpretable Model-Agnostic Explanations (LIME). Die LIME-Analyse schafft ein lokales lineares Ersatzmodell des Ausgangsmodells zu einem festgelegten Eingangsvektor X , um den Einfluss der Eingangsgrößen auf die Vorhersage erklärbar zu machen. Abbildung 2.19 veranschaulicht dieses Vorgehen graphisch. Die blaue und die pinke Fläche repräsentieren das zu erklärende Black-Box Modell. Zur Erklärung der Vorhersage für den Eingangsvektor X werden aus dem Black-Box-Modell anhand zusätzlicher um X liegender Vektoren (Stichproben) Vorhersagen erzeugt. Basierend auf diesen Vorhersagen und dem Eingangsvektor X erzeugt der LIME-Algorithmus dann ein lineares Ersatzmodell. Die Gewichtungsfaktoren des linearen Ersatzmodells ermöglichen es anschließend den Einfluss der Eingangsgrößen auf die Vorhersage für den Eingangsvektor X zu erklären.

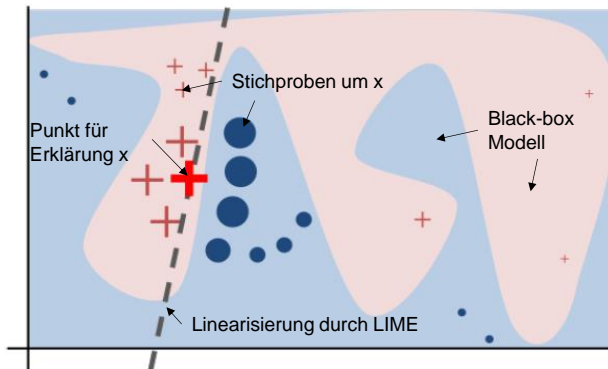


Abbildung 2.19: Graphische Darstellung der LIME-Analyse (Eigene Darstellung nach Ribeiro, Singh & Guestrin, 2016). Zur Erklärung der Vorhersage für den Eingangsvektor X werden aus dem Black-Box-Modell anhand zusätzlicher um X liegender Vektoren (Stichproben) Vorhersagen erzeugt. Basierend auf diesen Vorhersagen und dem Eingangsvektor X erzeugt der LIME-Algorithmus dann ein lineares Ersatzmodell.

Der konkrete Algorithmus kann in der Veröffentlichung von Ribeiro, Singh und Guestrin (2016) nachgelesen werden.

Vorgehensmodelle zur Anwendung von ML-Modellen. Das CRISP-ML(Q) Modell (Studer et al., 2021) erweitert bestehende Vorgehensmodelle, wie das CRISP-DM für Datenanalysen (vgl. Kapitel 2.3.2) oder Modelle für das Software Engineering (Amershi et al., 2019), um den konkreten Bezug zu ML-Modellen und

den Fokus auf Qualitätssicherung (vgl. Abbildung 2.20). Es kombiniert dabei bestehendes Wissen aus der Literatur und praxisnahe Erfahrungen.

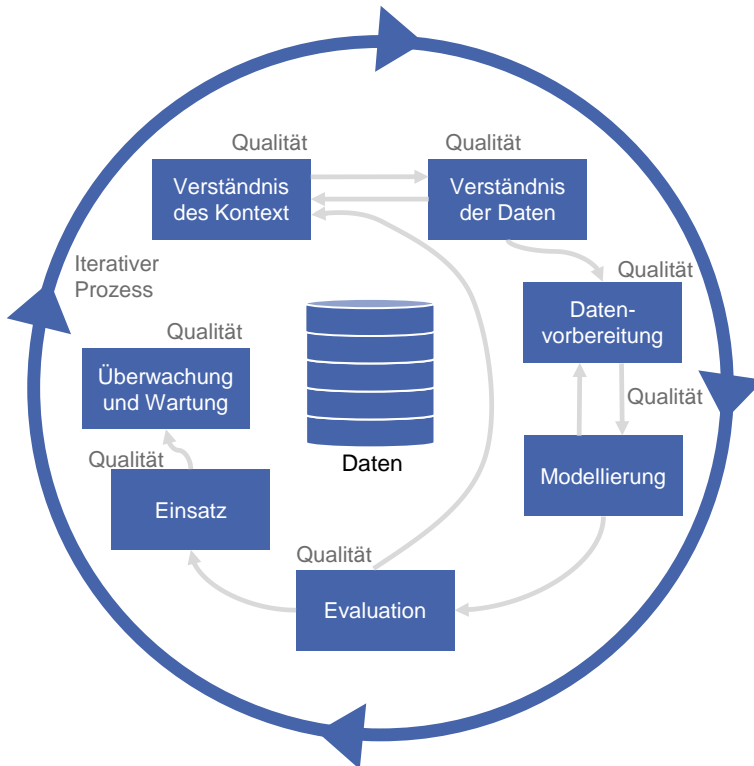


Abbildung 2.20: Vorgehensmodell Maschinelles Lernen aufgezeigt anhand des CRISP-ML (Q) Modells (Eigene Darstellung nach Studer et al., 2021).

Im ersten Schritt wird gleichzeitig ein Verständnis für den Kontext und die Daten geschaffen. Nach der Datenvorbereitung folgen die Modellierung und Evaluation der Modelle. Anschließend werden die Modelle in den Einsatz gebracht und ihre Funktionalität überwacht und entsprechend gewartet.

Dieser Ansatz ist bewusst generisch formuliert, um einen industrie- und anwendungsübergreifenden Einsatz zu ermöglichen. CRISP-ML (Q) bildet den Ausgangspunkt für die Methodenentwicklung in Kapitel 6.

2.3.2 Nicht überwachtes Lernen

In diesem Kapitel wird ein Überblick über das Themengebiet des nicht überwachten Lernens gegeben, wobei speziell auf Cluster-Algorithmen eingegangen wird. Der Cluster-Algorithmus DBSCAN wird im Detail vorgestellt.

Das Ziel von Cluster-Verfahren ist es, „Daten (semi-automatisch) so in Kategorien, Klassen oder Gruppen (Cluster) einzuteilen, daß Objekte im gleichen Cluster möglichst ähnlich und Objekte aus verschiedenen Clustern möglichst unähnlich zueinander sind“ (Ester & Sander, 2013, S. 45).

Bei der Auswahl und Anwendung von Cluster-Algorithmen muss beachtet werden, dass Cluster unterschiedliche Größe, Formen und Dichten haben können. Abbildung 2.21 gibt einen Überblick über Clusterstrukturen unterschiedlicher Eigenschaften.



Abbildung 2.21: Beispiele für 2-dimensionale Clusters-Strukturen mit verschiedenen Eigenschaften (Ester & Sander, 2013, S. 45).

Cluster-Verfahren können in partitionierende, dichte-basierte und hierarchische Verfahren eingeordnet werden.

Partitionierende Verfahren. Diese zerlegen Datenmengen in k Cluster, die mindestens ein Objekt enthalten und jedes Objekt zu genau einem Cluster zuzuordnen ist. Die Punkte müssen dafür in einem n -dimensionalen euklidischen Vektorraum vorliegen. Für diese Verfahren wird die Anzahl der Cluster vorgegeben. Ist keine optimale Clusteranzahl bekannt, kann für $k = 2, \dots, n-1$ ein Clustering vorgenommen werden und die Clustergüte wird anschließend mit dem Silhouetten-Koeffizient bewertet. Das am häufigsten verwendete partitionierende Verfahren ist der k Means-Algorithmus. Der Nachteil dieser Verfahren ist, dass sie Cluster heterogener Form, heterogener Größen und unterschiedlicher Punktdichte nicht klassifizieren können.

Dichte-basierte Verfahren. Um dieser Problematik entgegenzuwirken, existieren dichte-basierte Verfahren. Hierbei werden Cluster im n -dimensionalen Raum aus Objekten identifiziert, die dicht beieinanderliegen, getrennt durch Gebiete, in denen

die Objekte weniger dicht liegen. Ein Beispiel für ein dichte-basiertes Verfahren ist der DBSCAN-Algorithmus. Dichte-basierte Verfahren haben Probleme mit hierarchischen Clustern, stark unterschiedlicher Dichte in verschiedenen Raumbereichen und den Nachteil keine klare Trennung von Cluster und Rauschen zu haben.

Hierarchische Verfahren. Um hierarchische Strukturen erkennen zu können, wurden hierarchische Verfahren entwickelt. Diese erzeugen eine hierarchische Repräsentation der Daten, aus der man eine Clusterstruktur ableiten kann. Ein Beispiel für ein hierarchisches Verfahren ist der Ward-Algorithmus, der eine Unterkategorie des Agglomerative Clustering darstellt.

Darüber hinaus existieren auch Kombinationen dieser Verfahren wie bspw. dichte-basierte hierarchische Verfahren. Ein Beispiel für ein solches Verfahren ist der OPTICS-Algorithmus (Ester & Sander, 2013, S. 45–85).

Die unterschiedliche Cluster-Bildung der verschiedenen Verfahren angewendet auf jeweils sechs verschiedene zweidimensionale Cluster-Probleme ist in Abbildung 2.22 dargestellt (Pedregosa et al., 2011). Der Vorteil von dichte-basierten Ansätzen im Vergleich zu partitionierenden Verfahren Cluster stark unterschiedlicher Formen zu erkennen, zeigt sich in der Abbildung.

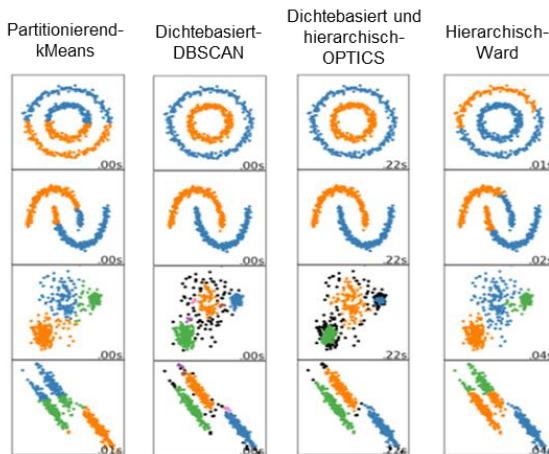


Abbildung 2.22: Anwendung von Cluster-Algorithmen der Scikit-Learn Bibliothek auf verschiedene zweidimensionale Cluster (Eigene Darstellung nach Pedregosa et al.).

In der späteren Entwicklungsaufgabe (vgl. Kapitel 7.3) wird der dichtebasierte Cluster-Algorithmus DBSCAN genutzt, weshalb dieser nachfolgend genauer beschrieben wird.

Density Based Spatial Clustering of Applications with Noise (DBSCAN). Der DBSCAN-Cluster-Algorithmus wurde bereits 1996 vorgestellt und ist ein dichtebasierter Cluster-Algorithmus (Ester, Kriegel, Sander & Xu, 1996). Das von DBSCAN eingeführte Modell bestimmt die Cluster-Dichte anhand der Anzahl der Nachbarn minPts innerhalb des Radius ϵ um einen Punkt. Einem Cluster werden Objekte (einschließlich des Abfragepunkts) mit mehr als minPts Nachbarn innerhalb dieses Radius zugeordnet. Der DBSCAN-Cluster-Algorithmus identifiziert nun die Bereiche, die diese Mindestdichte erfüllen und die durch Bereiche geringerer Dichte getrennt sind. Punkte eines Clusters, die mit ihrem Radius nur einen Punkt des Clusters einschließen, werden Grenzpunkte genannt. Punkte, die nicht innerhalb dieser Dichtedefinition liegen, gehören keinem Cluster an und werden als Rauschen betrachtet. Abbildung 2.23 veranschaulicht das Konzept von DBSCAN. Der minPts -Parameter ist 4 und der ϵ -Radius wird durch die Kreise angezeigt. N ist ein Rauschpunkt, A ist ein Kernpunkt und die Punkte B und C sind Grenzpunkte. Pfeile zeigen die direkte Zugänglichkeit der Dichte an. Die Punkte B und C gehören zum gleichen Cluster wie A, da beide von A aus erreichbar sind. Der Punkt N wird nicht erreicht und wird daher als Rauschpunkt angesehen (Schubert, Sander, Ester, Kriegel & Xu, 2017).

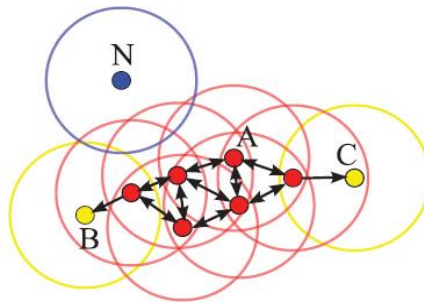


Abbildung 2.23: Anwendung des DBSCAN Cluster-Algorithmus auf zweidimensionale Datenpunkte (Schubert et al., 2017). A ist ein Kernpunkt, Punkt B und C sind Grenzpunkte, N ist ein Rauschpunkt.

Der Pseudocode des DBSCAN-Algorithmus ist in Abbildung 2.24 dargestellt. Zunächst wird die Datengrundlage linear nach noch nicht verarbeiteten Objekten durchsucht. Nicht-Kernpunkte werden dem Rauschen zugewiesen, und wenn ein

Kernpunkt entdeckt wird, werden seine Nachbarn iterativ erweitert und dem Cluster hinzugefügt (Schubert et al., 2017).

ALGORITHM 1: Pseudocode of Original Sequential DBSCAN Algorithm

```

Input: DB: Database
Input:  $\epsilon$ : Radius
Input: minPts: Density threshold
Input: dist: Distance function
Data: label: Point labels, initially undefined
1  foreach point  $p$  in database DB do                                // Iterate over every point
2      if label( $p$ )  $\neq$  undefined then continue                      // Skip processed points
3      Neighbors  $N \leftarrow \text{RANGEQUERY}(DB, \text{dist}, p, \epsilon)$     // Find initial neighbors
4      if  $|N| < \text{minPts}$  then                                         // Non-core points are noise
5          label( $p$ )  $\leftarrow$  Noise
6          continue
7       $c \leftarrow$  next cluster label                                // Start a new cluster
8      label( $p$ )  $\leftarrow c$ 
9      Seed set  $S \leftarrow N \setminus \{p\}$                         // Expand neighborhood
10     foreach  $q$  in  $S$  do
11         if label( $q$ ) = Noise then label( $q$ )  $\leftarrow c$ 
12         if label( $q$ )  $\neq$  undefined then continue
13         Neighbors  $N \leftarrow \text{RANGEQUERY}(DB, \text{dist}, q, \epsilon)$ 
14         label( $q$ )  $\leftarrow c$ 
15         if  $|N| < \text{minPts}$  then continue                          // Core-point check
16          $S \leftarrow S \cup N$ 

```

Abbildung 2.24: Pseudocode des originalen sequentiellen DBSCAN-Algorithmus (Schubert et al., 2017).

Zur Bestimmung von minPts und ϵ gibt es Empfehlungen aus der Literatur. Nach Ester et al. (1996) ist der minimale minPts-Parameter als $2 * \text{dim}$ definiert, kann aber auch größer gewählt werden. Der Parameter ϵ hat einen wesentlichen Einfluss auf die Größe der Cluster und deren Anzahl. Wird er zu klein gewählt, werden in den Daten keine Cluster identifiziert (hoher Rauschanteil). Wird er zu groß gewählt, dann ist es möglich, dass einzelne Cluster zu groß werden und alle Daten in ein Cluster fallen. Nach Schubert et al. (2017) sollte der Rauschanteil zwischen einem Prozent und 30 % liegen und das größte Cluster maximal 20 % bis 50 % der Datenpunkte enthalten. Anhand dieser Werte kann eine Optimierung von ϵ durchgeführt werden.

Korrelationsanalyse. In einer Korrelationsanalyse werden lineare Abhängigkeiten von Größen gegeneinander bestimmt. Diese können in einer Korrelationsmatrix graphisch dargestellt werden. Für jeden Matriceintrag wird dabei der Korrelationskoeffizient R_{pearson} nach Pearson (1896) bestimmt. Dieser ist der Quotient aus der Kovarianz zweier Größen x und y s_{xy} und dem Produkt der Standardabweichungen s_x und s_y (Kosfeld, Eckey & Türc, 2016).

$$R_{\text{Pearson}} = \frac{s_{xy}}{s_x s_y}$$

2.11

Dadurch ergibt sich eine symmetrische Matrix (vgl. Abbildung 2.25). Je näher der Korrelationskoeffizient an dem Wert 1 liegt, desto größer ist die Linearität. Der Wert -1 bedeutet eine umgekehrte Linearität. Liegt ein Koeffizient von 0 vor, so existiert keine Abhängigkeit der Größen.

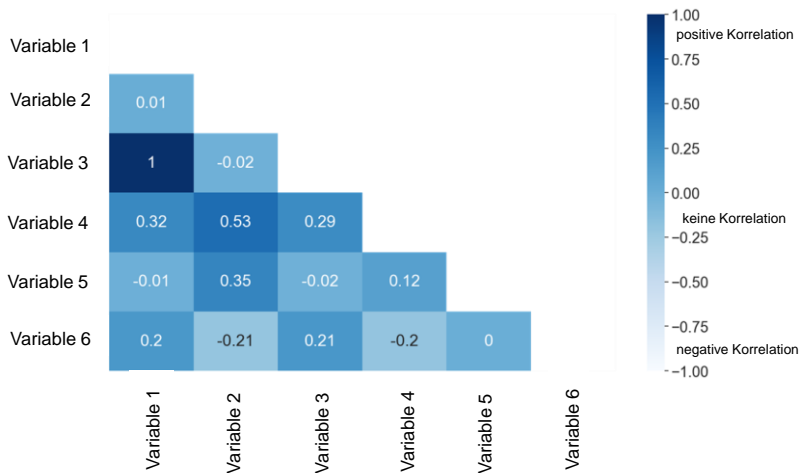


Abbildung 2.25: Auswertung einer Korrelations-Matrix für 6 Variablen

Vorgehensmodelle Datenanalyse. Zur Datenanalyse gibt es den Prozess der Knowledge Discovery in Databases (KDD) (Fayyad, Piatetsky-Shapiro & Smyth, 1996) oder das CRISP-DM (Chapman, Clinton, Khabaza, Reinartz & Wirth, 1999) (vgl. Abbildung 2.26). Im Gegensatz zu Vorgehensmodellen wie CRISP-ML(Q) (vgl. Kapitel 2.3.1) gehen diese nicht auf ML spezifische Aufgaben ein, sondern setzen den Fokus auf Data Mining. Das Ziel von Data Mining oder auch Knowledge-Discovery ist die systematische Generierung von Wissen aus Daten.

Der KDD-Prozess besteht aus neun Teilschritten: Verständnis der Domäne und Ziel des Prozesses, Erstellung des Datensatzes, Datenaufbereitung und Datenvorverarbeitung, Datenreduktion und Projektion, Zuordnung der Ziele zu einer Data-Mining Methode, Auswahl des Data-Mining-Ansatzes, Data-Mining,

Interpretation der erkannten Muster und Nutzen des erzeugten Wissens (vgl. Abbildung 2.26a)).

Das CRISP-DM besitzt eine hierarchische Struktur auf vier Abstraktionsebenen: Phasen, Generische Aufgaben, Spezielle Aufgaben und Prozessinstanzen. Die Reihenfolge der einzelnen Phasen sind in Abbildung 2.26 b) dargestellt und bestehen aus: Verständnis des Kontexts und der Daten, Datenverarbeitung, Optimierung der Parameter und Erstellung der Modelle, Evaluation der Datenmodelle anhand Aufgabenstellung und Entscheidungsprozess auf Basis der gewonnenen Erkenntnisse (Wirth & Hipp, 2000).

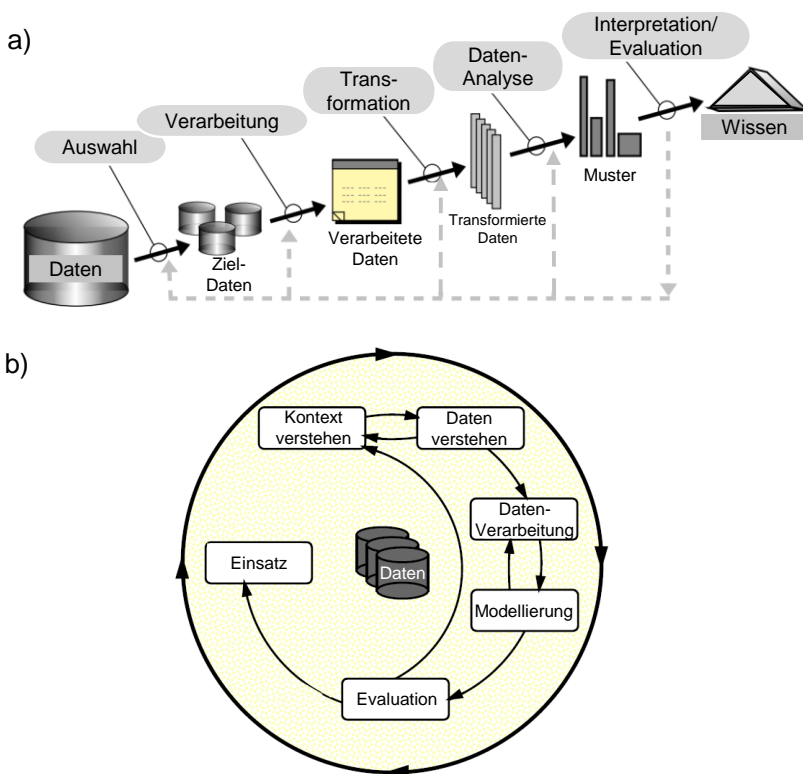


Abbildung 2.26: Veranschaulichung des a) KDD-Prozess (Fayyad et al., 1996, S. 41) und b) CRISP-DM für Data Mining (Wirth & Hipp, 2000, S. 5).

Beide Prozesse sind stark iterativ. Für den CRISP-Prozess werden mindestens drei Iterationen vorgeschlagen. Außerdem schafft der CRISP-Prozess ein transparentes Vorgehen zur Kommunikation zwischen den Stakeholdern und unterstützt die Planung und Dokumentation. Der KDD-Prozess nennt mehrfach die Wichtigkeit von Domänenwissen für die Wissensgewinnung. KDD und CRISP-DM bilden den Ausgangspunkt für die Methodenentwicklung in Kapitel 6.

2.3.3 Verstärkendes Lernen

In diesem Kapitel wird zunächst ein Überblick über das verstärkende Lernen gegeben. Anschließend wird der Soft-Actor-Critic-(SAC)-Algorithmus vorgestellt, der in dieser Arbeit Anwendung findet.

Beim verstärkenden Lernen lernt ein Agent eine Aufgabe in einer unbekannten Umgebung mit einem Trial-and-Error-Ansatz auszuführen. Er führt Aktionen a_t aus, die den Zustand s_t der Umgebung verändern. Als Folge seiner Aktion erhält der Agent einen neuen Zustand der Umgebung s_{t+1} und eine Belohnung r_{t+1} zurück (vgl. Abbildung 2.27). Ziel des Agenten ist es, diese Belohnung im Laufe des Trainings zu maximieren, damit ein ideales Verhalten anzunehmen und seine Aufgabe zu erfüllen (Sutton & Barto, 2018, S. 1–8). Diese Strategie des Agenten, die jedem Zustand eine Aktion zuordnet, wird als Strategie π bezeichnet.

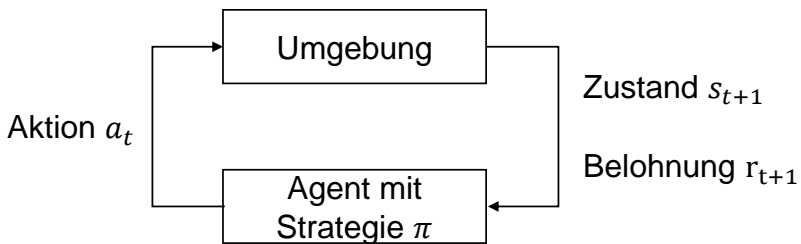


Abbildung 2.27: Schematische Darstellung des verstärkenden Lernens (Eigene Darstellung nach Sutton & Barto, 2018).

Verstärkendes Lernen zeigt herausragende Ergebnisse bspw. für das Lernen von Atari-Spielen (Mnih et al., 2013) oder das Rennen-Lernen des MIT Roboters Mini Cheetah (Margolis, Yang, Paigwar, Chen & Agrawal, 2022). Der Algorithmus AlphaZero ist sogar in der Lage durch Spielen gegen sich selbst, die drei Strategiespiel Schach, Go und Shogi zu lernen und die weltbesten Programme und Spieler zu schlagen (Silver et al., 2018).

Es existieren drei Arten von Agenten:

- Ein nutzenbasierter Agent lernt eine Nutzenfunktion für Zustände und damit die Aktion, die den erwarteten Ergebnisnutzen maximiert.
- Ein Q-lernender Agent lernt eine Aktion/Nutzen-Funktion anhand des Nutzens, der durch eine bestimmte Aktion in einem bestimmten Zustand erhalten wird.
- Ein Reflex-Agent lernt eine direkte Abbildung von Zuständen auf Aktionen.

Darüber hinaus gibt es passives und aktives Lernen. Beim passiven Lernen steht die Strategie des Agenten fest, er versucht deshalb den Nutzen von Zuständen für die Aufgabe zu lernen. Dagegen lernt der Agent beim aktiven Lernen erst diese Strategie im Trainingsprozess (Russell & Norvig, 2012, S. 960–961).

Für einfache Probleme kann die Nutzenfunktion bzw. die Aktion/Nutzenfunktion in einer Tabelle abgelegt werden. Für komplexe Probleme mit sehr großem Zustandsraum und kontinuierlichen Aktionen würden riesige Tabellen entstehen, deren Abfrage und Bedatung sehr viel Zeit braucht. Es wird also eine Generalisierung benötigt, um diesen Aufwand zu reduzieren. Hierfür werden Funktions-Approximationen verwendet. Als sehr geeignet zeigen sich hierbei KNN (vgl. 2.3.1) (Sutton & Barto, 2018, S. 195). Werden KNN mit mehreren tiefen Schichten verwendet, so spricht man von tiefem verstärkendem Lernen.

Werden die Aktionen und Zustände zur Optimierung der Strategie von der aktuellen Strategie generiert, so ist dies eine On-Strategie. Werden diese von einer Vielzahl vergangener Strategien generiert, so spricht man von Off-Strategie (Sutton & Barto, 2018, S. 100). Durch das Lernen aus vergangenen Strategien kann von einer breiten Erfahrungsbasis gelernt werden. Jedoch ist dieser Ansatz in einer sich schnell ändernden Umgebung weniger effizient als eine On-Strategie.

Im Bereich des verstärkenden Lernens existiert eine Vielzahl an Algorithmen. Die aktuell performantesten für eine Vielzahl an Problemstellungen sind Proximal-Policy-Optimization (Schulman, Wolski, Dhariwal, Radford & Klimov, 2017), Deep-Deterministic-Policy-Gradient (Lillicrap et al., 2015), Soft-Q-Learning (Haarnoja, Tang, Abbeel & Levine) und Soft-Actor-Critic (Haarnoja, Zhou, Abbeel & Levine).

In der späteren Entwicklungsaufgabe (vgl. Kapitel 7.2) wird ein Soft-Actor-Critic-Algorithmus verwendet, weshalb nachfolgend zunächst die Actor-Critic-Architektur und anschließend der Algorithmus selbst vorgestellt wird.

Der Soft-Actor-Critic-Algorithmus (SAC) ist derzeit einer der leistungsstärksten und robustesten Algorithmen des verstärkenden Lernens (Haarnoja et al.).

Ein Actor-Critic Algorithmus hat eine getrennte Speicherstruktur für die Strategie π und die Aktionswert-Funktion $Q(s, a)$. Die Strategie ordnet jedem Zustand $a = \pi(s)$ eine Aktion zu und wird „Actor“ genannt. Die Value-Funktion bestimmt die erwartete Belohnung für einen Zustand oder Zustands-Aktions-Paar und wird „Critic“ genannt, da es die Aktionen des Aktors kritisiert. Der TD-error ist das Maß für die Kritik und beschreibt den Fehler zwischen der Schätzung des Agenten für die Belohnung bei einem bestimmten Aktions-Zustandspaar und der tatsächlichen Belohnung. Aktionen mit einem positiven TD-error werden in Zukunft gestärkt, mit einem negativen TD-error abgeschwächt. (vgl. Abbildung 2.28)

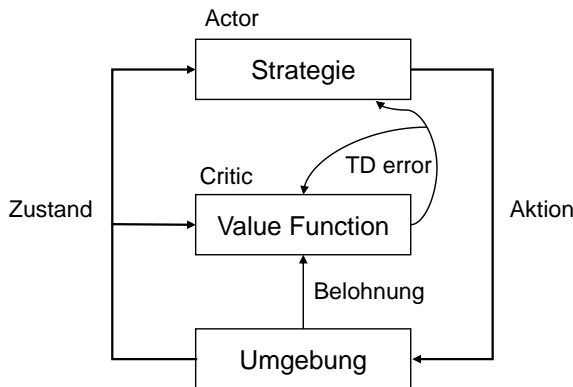


Abbildung 2.28: Actor-Critic Architektur (Eigene Darstellung nach Sutton & Barto, 1998). Der Critic bewertet die Aktionen des Actors (Strategie) mit einem Fehler (TD-Error), der die Differenz zwischen erwarteter Belohnung (Value) und tatsächlicher Belohnung darstellt.

Actor-Critic-Ansätze bieten den Vorteil, dass durch die separate Speicherung der Strategie eine Aktion mit geringem Rechenaufwand ausgewählt werden kann. Eine Methode, die ausschließlich Aktions-Werte lernt, müsste einen kontinuierlichen Aktionsraum aus einer unendlichen Anzahl an Aktionen wählen, was zu einem sehr hohen Rechenaufwand führt (Sutton & Barto, 1998, S. 257–259).

Soft-Actor-Critic. Der Soft-Actor-Critic-Algorithmus ist ein Off-Strategie Actor-Critic-Algorithmus, der neben der Belohnung auch die Entropie maximiert, die ein Maß für die Exploration darstellt. Bei der Exploration werden neue Aktionen ausgeführt, wodurch unbekannte Bereiche des Lösungsraums untersucht werden.

Der Begriff „soft“ bezieht sich dabei auf die Policy. Eine soft Policy wählt eine Aktion mit einer Wahrscheinlichkeit aus allen möglichen Aktionen aus (Sutton & Barto, 2018, S. 100).

Der Verwendung des SAC liegen folgende Randbedingungen zu Grunde. Ein kontinuierlicher hochdimensionaler Zustandsraum $S \in \mathbb{R}^n$, ein kontinuierlicher hochdimensionaler Aktionsraum $A \in \mathbb{R}^n$, sowie eine probabilistische Umgebung ergeben einen Markov-Decision-Process (MDP), der durch das Tupel (S, A, p, r) definiert wird. Die probabilistische Übertragungsfunktion $T(st, at, s_{t+1}) = \mathbb{P}(s_{t+1}|st, at)$ gibt dabei die Wahrscheinlichkeit an, dass eine Aktion $a_t \in A$ in einem Zustand $s_t \in S$ zu einem Zustand $s_{t+1} \in S$ führt. Zu einem Zeitpunkt t führt ein Agent aus einer Vielzahl möglicher Aktionen $A \in \mathbb{R}^n$ eine Aktion a_t aus. Dadurch ändert sich der Zustand $s_t \in S$ der Umgebung zum Zustand des nächsten Zeitschritts $s_{t+1} \in S$ und der Agent erhält eine Belohnung $r : S \times A \times S \rightarrow \mathbb{R}$. Eine Strategie $\pi(a|s) : S \rightarrow \mathbb{P}(A)$ wählt die nächste Aktion in Abhängigkeit des Zustands. Für einen begrenzten Zeithorizont H beschreibt $J = \mathbb{E}[\sum_{t=0}^H r(s_t, a_t, s_{t+1})]$ die erwartete zurückgegebene Belohnung einer Strategie. Das Ziel eines Agenten ist es, sein Verhalten so zu optimieren, dass die Belohnung maximiert wird, was mathematisch als $\pi^* = \operatorname{argmax} J(\pi)$ beschrieben werden kann.

Standardverfahren des verstärkenden Lernens maximieren die erwartete Summe der Belohnungen nach $\sum_{t=0}^T \mathbb{E}_{(s_t, a_t)} [r(s_t, a_t)]$. Der SAC hingegen maximiert neben dieser Belohnung gleichzeitig auch die Entropie, wodurch sich der Term $J(\pi) = \sum_{t=0}^T \mathbb{E}_{(s_t, a_t)} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t))]$ ergibt. Der Entropie-Term \mathcal{H} ist ein Maß für das stochastische Verhalten der Strategie und wird mit dem Temperaturparameter α gegen die Belohnung gewichtet. Der Hauptvorteil der Entropie-Maximierung ist eine größere Exploration der Strategie bei gleichzeitigem Verzicht von eindeutig aussichtslosem Verhalten, was zu einer insgesamt verbesserten Exploration führt.

SAC nutzt Funktions-Approximationen für die Aktion/Nutzen-Funktionen und die Strategie und alterniert zwischen der Optimierung aller Netzwerke mit stochastischem Gradientenabstiegsverfahren. Es wird eine parametrisierte Zustandswert-Funktion $V(s_t)$, mehrere weiche Aktionswert-Funktionen $Q(s_t, a_t)$ und eine Strategie $\pi(a_t|s_t)$ verwendet. Zum Beispiel kann die Zustandswert-Funktion als KNN und die Strategie als eine Gaußverteilung mit Mittelwert und Kovarianz abgeleitet aus einem KNN dargestellt werden. Für den genauen Ablauf des Algorithmus wird auf die Veröffentlichung von Haarnoja et. al verwiesen (Haarnoja et al.).

Die Vorteile des SAC sind seine hohe Stabilität gegenüber verschiedenen Hyperparametern und seine gute Performance für verschiedene Steuer- und Regelaufgaben, was anhand verschiedener Steuerbeispiele gezeigt wurde (Haarnoja et al.). Der SAC-Algorithmus wird in dieser Arbeit anhand der Stable-Baselines-Bibliothek implementiert (Raffin et al., 2021).

2.3.4 Zwischenfazit

In diesem Kapitel wurden eine Übersicht über die Methoden des maschinellen Lernens - überwachtes Lernen, nicht überwachtes Lernen und verstärkendes Lernen - gegeben und spezifische Algorithmen, die später in der Arbeit verwendet werden, im Detail vorgestellt. Im Bereich überwachtes Lernen wurden künstliche neuronale Netze und speziell rekurrente Netze beschrieben. Für nicht überwachtes Lernen wurden verschiedene Cluster-Verfahren vorgestellt und der DBSCAN-Algorithmus als dichtebasiertes Verfahren im Detail erläutert. Abschließend wurde das Konzept des verstärkenden Lernens dargestellt. Actor-Critic-Agenten und der Soft-Actor-Critic-Algorithmus wurden im Detail beschrieben.

2.4 Einsatz von maschinellem Lernen in Fahrzeuggetrieben

Forschungsgegenstand der Literaturrecherche ist der Einsatz von ML-Methoden in Fahrzeuggetrieben. Betrachtet wird dabei der Einsatz zur Unterstützung des Entwicklungsprozesses und im Betrieb des Getriebes. In der Recherche wird auch der Begriff KI genutzt, da ML-Methoden häufig mit diesem Überbegriff zusammengefasst werden. In der Auswertung des Stands der Forschung wird dann auf ML-Methoden fokussiert.

Zunächst wird der Einsatz von ML-Methoden in der Getriebesteuerung konventioneller Antriebe für relevante Steuerungsaufgaben, Betriebsstrategien und Entwicklungsmethoden qualitativ analysiert. Diese werden nach der Methodenverwendung überwachtes Lernen, verstärkendes Lernen und nicht überwachtes Lernen eingeordnet.

Anschließend wird für die Getriebesteuerung von Hybridfahrzeugen eine quantitative Literaturanalyse von Veröffentlichungen durchgeführt, um eine Verteilung auf verschiedene Anwendungsfälle zu bestimmen. Die Analyse nutzt die Suchbegriffe: artificial intelligence, machine learning, neural network, genetic algorithm, data mining, prediction, recognition, sowie powertrain, hybrid transmission control und vehicle. Als Datenbanken werden Science Direct und IEEE Xplore verwendet. Anschließend werden die Ergebnisse inhaltlich geprüft und doppelte Findungen aussortiert. Insgesamt werden 72 relevante Ergebnisse

identifiziert¹. Die daraus verwendete Literatur in Kapitel 2.4.2 findet sich im Literaturverzeichnis.

2.4.1 Konventionelle Fahrzeuggetriebe

Überwachtes Lernen. Veith, Krüger, Albers und Behrendt (2018) entwickeln einen virtuellen Drehmomentensensor, der anhand von BUS-Signalen mit Hilfe von KNN das Drehmoment im Antriebsstrang vorhersagt. Durch eine Datenvorverarbeitung kann mit der gleichen Menge an Trainingsdaten eine höhere Abbildungsgüte erzielt werden. Außerdem wird der Einsatz für eine Drehmomenten-Adaption vorgeschlagen. Die Optimierung der Hyperparameter wird nicht näher erläutert.

Einige Veröffentlichungen nutzen KNN, um Simulationsmodelle für die Antriebsstrangentwicklung zu verbessern. Zur Drehmomentenbestimmung einer Fahrzeugbremse nutzen Ćirović, Smiljanić und Aleksendrić (2014) ein rekurrentes neuronales Netz. Als Eingangsgrößen dienen Bremsendruck, Temperatur, und Fahrzeuggeschwindigkeit, die das Bremsenmoment berechnen. In diesem Ansatz werden 45 Netzwerkarchitekturen und drei Trainings-Algorithmen zur Optimierung genutzt. Auch Cao, Wang, Fujii und Tobler (2004) modellieren das Moment einer Bremse innerhalb eines Planetenradgetriebes mit einem hybriden neuronalen Netz, bei dem die Öltemperatur über eine vorgelagerte Berechnung bestimmt wird.

Cabrera Cano (2017) stellt einen Ansatz vor, Simulationsmodelle von Fahrzeuggetrieben mit KNN und vorgelagerten Laguerre-Filtern abzubilden. Dieses Modell wird mit Simulationsdaten eines bereits vorhandenen Simulationsmodells trainiert, mit dem Ziel ein Modell mit gleichbleibender Modellgüte, aber deutlich verringerter Simulationszeit zu erhalten. Mit diesem Ansatz konnte die Simulationszeit um den Faktor 10 reduziert werden, was eine Echtzeitsimulation ermöglicht.

Zur Unterstützung von Entwickelnden existieren Ansätze, das menschliche Komfortempfinden und die Beurteilung der NVH (Noise Vibration Harshness) -Eigenschaften mit Hilfe von KNN zu modellieren, um somit spätere Produkteigenschaften zuverlässig zu modellieren (Lerspalungsanti, 2010). Dies wird am Beispiel des Anfahrkomforts für verschiedene Fahrzeuge gezeigt und im Sinne der ganzheitlichen Antriebsstrangentwicklung auf NVH-Eigenschaften des

¹ Eine Liste der 72 Ergebnisse ist im Anhang auf S.XIX enthalten.

Getrieberassessments übertragen. Hierfür bewerten Experten zunächst ihr subjektives Komfortempfinden. Mit diesen Daten werden dann KNN trainiert, die die Zuordnung zwischen Kennwerten der Messungen und den Expertenbewertungen abbilden. Auch für die Bewertung des Schaltkomforts wurden solche Verfahren entwickelt. Wolter, Ruoff, Lutz, Bause und Ott (2018) verwenden diesen Ansatz für Handschaltgetriebe und Kawakami, Ide, Moriyama, Hoki und Muramatsu (2020) für ein Automatikgetriebe.

Verstärkendes Lernen. In der Automatikgetriebeentwicklung werden erste Regelvorgänge mit Methoden des maschinellen Lernens optimiert. Dazu gehören beispielsweise der Gangwechsel, der Anfahrvorgang oder die Unterdrückung des Kupplungsrupfens (Gaiselmann, Altenburg, Studer & Peters, 2022; Lampe, Serway Roland, Siestrup & Gühmann, 2019; Sommer Obando, 2016). Alle diese Ansätze verwenden verstärkendes Lernen, bei dem ein intelligenter Agent in einer Antriebsstrangsimulationsumgebung ein Verhalten lernt, welches eine Belohnungsfunktion maximiert. Lampe et al. (2019) stellen auch ein Sicherheitskonzept vor, das beschreibt, wie ein Agent auf dem Steuergerät integriert werden kann. Der Fokus dieser Arbeit liegt auf der Reduzierung des Applikationsaufwands und der Verbesserung der Regelgüte. Die Konzepte werden in der Simulation und teilweise am Prüfstand validiert, aber noch nicht auf einem Seriensteuergerät eingesetzt. Alle Ansätze beschreiben, wie im konkreten Anwendungsfall Aktions-, Zustandsraum, Trainingsalgorithmus und Belohnungsfunktion gewählt werden.

Nicht überwachtes Lernen. Erste Arbeiten zeigen den Einsatz von maschinellem Lernen zur Datenauswertung in der Antriebsstrangentwicklung. Neben KNN zur Objektivierung der Fahrbarkeit verwenden X. Zhang, Ebner, Arntz, Ramsauer und Küçükay (2021) eine Ausreißererkennung auf Basis des Cluster-Algorithmus kMeans, um Parameter in der Software zu identifizieren, die für eine schlechte Fahrbarkeit verantwortlich sind. Zur Vorauswahl der Parameter wird eine Korrelationsanalyse durchgeführt. Zur Generierung von realistischen Fahrzyklen aus Fahrzeugdaten wird ein kMeans-Cluster-Algorithmus genutzt (Förster, Inderka & Gauterin, 2020). Auch Tewiele, Driesch, Weber und Schramm (2018) nutzen einen Cluster-Algorithmus zur Generierung von Fahrzyklen aus realen Fahrzeugdaten speziell für batterieelektrische Fahrzeuge. Um diese Datenbasis zu erweitern, werden zusätzlich Simulationsdaten benutzt. Der konkrete Einsatz von Cluster-Algorithmen wird nicht näher beschrieben. Kistner, Sanzenbacher, Munier und Fietkau (2019) verwenden Korrelationen, Regressionen und Entscheidungsbäume, um wichtige Einflussfaktoren für die Geräuschanalyse des Anfahrens mit nasslaufender Kupplung zu identifizieren. Die konkrete Auswahl und Anwendung der Algorithmen wird nicht näher beschrieben. Krätschmer, Lutchen

und Reuss (2021) stellen ein Konzept vor, wie Messdaten am Prüfstand offline mit ML ausgewertet werden können, um die Fehlererkennung zu unterstützen. Welche ML-Algorithmen zur Datenauswertung genutzt werden, wird nicht näher erläutert.

2.4.2 Hybride Fahrzeuggetriebe

Anhand der quantitativen Recherche werden 71 relevante Veröffentlichungen identifiziert, die sich mit dem Einsatz von maschinellem Lernen in hybriden Getrieben beschäftigen. 77 % der Veröffentlichungen optimieren die Betriebsstrategie, wohingegen 10 % für Diagnosefunktionen und 13 % für die Steuerung und Regelung eingesetzt werden.

Harold, Prakash & Hofman stellen ein Verfahren vor, das die Betriebsstrategie anhand bereits durchgeführter Fahrmanöver erlernt. Ziel ist es, die Betriebsstrategie an das spezifische Fahrverhalten des Fahrers anzupassen. Es wird eine Kombination aus dynamischer Programmierung und überwachtem Lernen verwendet. T. Liu, Tang, Wang, Yu & Hu (2019) nutzen einen genetischen Algorithmus, der die optimale Betriebsstrategie für verschiedene Fahrzyklen ermittelt. Diese Strategie wird mittels überwachtem Lernen auf ein KNN trainiert, um eine Echtzeitfähigkeit der Strategie zu ermöglichen. Lin, Wang, Bogdan, Chang & Pedram (2014) entwickeln einen Ansatz des verstärkenden Lernens zum Erlernen einer optimalen Betriebsstrategie. Yang, Han, Xiang, Liu & Hou (2021) und Biswas, Anselma & Emadi (2019) beschreiben Möglichkeiten, diesen Ansatz zu erweitern und zu verbessern.

Im Bereich hybrider Antriebsstrang-Regelalgorithmen gibt es erste Ansätze zum Einsatz von ML-Methoden. Beispielsweise wird der Wiederstart des Motors über das Trägheitsmoment des Antriebsstrangs optimiert (Wang, Hou, Du, Xu & Zhou, 2020). Es wird eine adaptive modellprädiktive Regelung (MPR) (vgl. Kapitel 2.1.4) entwickelt, die den Wiederstart des Motors regelt, um das Fahrverhalten zu verbessern und gleichzeitig den Energieverbrauch zu senken. Ein genetischer Algorithmus wird genutzt, um den Momentenverlauf beim Beschleunigungsvorgang zu optimieren, mit dem Ziel den Fahrkomfort zu steigern (Bovee & Rizzoni, 2016).

Vadamalu & Beidl (2016) verwenden einen MPR-Ansatz, um die Schwingungen im Antriebsstrang mit Hilfe der EM aktiv zu reduzieren und damit die Fahrbarkeit eines Hybridantriebs zu verbessern.

Reuss, Liao, Gitt & Schröder (2019) stellen einen Ansatz vor, die Applikation im Fahrzeug zu automatisieren. Sie verwenden ein Kriging-Modell², um Parametersätzen der Applikation objektivierte Bewertungsnoten zuzuweisen. Anschließend werden weitere Betriebspunkte in einer adaptiven Versuchsplanung am Prüfstand gemessen, um dieses Modell zu verbessern und den optimalen Parametersatz zu ermitteln, der die Bewertungsnote maximiert. Schließlich wird über KNN der Zusammenhang zwischen Zustandsgrößen (z. B. Fahrpedalwinkel) und den optimalen Applikationsparametern gelernt. Dadurch ergibt sich eine gleichbleibend gute Applikation für Anfahrvorgänge mit unterschiedlichen Pedalwinkeln im Vergleich zu einem konstanten Parametersatz.

Analog zur Objektivierung der Komfortbewertung des Schaltvorgangs entwickelt Birkhold (2013) eine Methode, die die Entwicklung des Wiederanlaufsystems von Hybridantrieben durch Komfortobjektivierung unterstützt. Sie nutzen KNN, um aus Messdaten einen Zusammenhang zwischen subjektiven Expertennoten und Parametern herzustellen. Auch Reuss et al. (2019) nutzen diese Methode, um die Bewertung des Motorwiederstarts zu objektivieren.

Am IPEK - Institut für Produktentwicklung gibt es erste Forschungsarbeiten, die sich damit beschäftigen, ML-Methoden in der Antriebsstrangentwicklung einzusetzen. Den Einsatz von verstärkendem Lernen untersuchen Sommer Obando (2016) zur Unterdrückung von Kupplungsrupfen und Altenburg, Bause & Albers (2022) mit dem Erlernen des Einschaltvorgangs bzw. des Gangwechsels durch einen trainierten Agenten (Gaiselmann et al., 2022).

Zur Unterstützung der Entwickelnden existieren Ansätze, das menschliche Komfortempfinden und die Beurteilung der NVH-Eigenschaften mit Hilfe von KNN zu modellieren, um somit spätere Produkteigenschaften zuverlässig zu modellieren. Bereits vor über 10 Jahren entwickelte Lerspalungsanti (2010) einen ganzheitlichen Ansatz für diese Aufgabe und zeigte die Übertragbarkeit der vorgestellten Methoden und Werkzeuge auf weitere Anwendungen in der Antriebsstrangentwicklung. Dabei wurden der Anfahrvorgang sowie das Getrieberasseln untersucht. Wolter et al. (2018) sowie Birkhold (2013), beschäftigten sich mit der Objektivierung des Schaltvorgangs eines Handschalters bzw. dem Motorwiederstart beim Hybrid mit Hilfe von KNN. Veith et al. (2018) nutzen ebenfalls KNN, um anhand von BUS-Signalen das Drehmoment im Antriebsstrang vorherzusagen und damit einen virtuellen Drehmomentensensor darzustellen. Außerhalb des

² Das Kriging-Verfahren ist auch als Gauß-Prozess-Regression bekannt.

Antriebsstrangbereichs entwickeln Dörr, Ries, Gwosch und Matthiesen (2019) einen Daten-Logger für Power-Tools und eine ML-basierte Zeitreihenanalyse, um anhand der Messdaten die Nutzung eines Power-Tools zu erkennen. Außerdem präsentieren Heyden et al. (2022) einen Tandem-Ansatz, um in dezentralen Datensätzen lokale und globale Ausreißer zu identifizieren, was am Beispiel von Power-Tool-Daten gezeigt wird.

Die Ergebnisse dieser Literaturrecherche wurden bereits im Dezember 2021 veröffentlicht und wurden durch weitere Veröffentlichungen aus den Jahren 2021 und 2022 ergänzt (Schuchter, Bause & Albers, 2021).

2.4.3 Zwischenfazit

Wie der Stand der Forschung zeigt, werden in Fahrzeuggetrieben erste Anwendungen von ML-Methoden erforscht. Methoden des überwachten Lernens werden überwiegend zur Modellbildung von Simulationsmodellen genutzt, die in der Entwicklung von Fahrzeuggetrieben eingesetzt werden. Methoden des nicht überwachten Lernens werden zum Clustering von Fahrzyklen verwendet, die Anwendung im Antriebsstrang wird teilweise nicht detailliert beschrieben.

In hybriden Antriebssträngen werden ML-Methoden hauptsächlich zur Optimierung der Betriebsstrategie eingesetzt. Dieser Bereich ist bereits stark erforscht. Gleichzeitig gibt es einige Ansätze, die Komfortbewertung mit KNN zu objektivieren. Bisher gibt es nur begrenzte Forschung zur Optimierung und Ansteuerung von Hybridantriebssträngen. Hier werden bisher MPR-Ansätze weitgehend untersucht. Der Begriff Ansteuerung von Hybridantriebssträngen wird später anhand der in Kapitel 5 identifizierten Herausforderungen konkretisiert.

Am IPEK - Institut für Produktentwicklung gibt es einen ersten Einsatz von ML im Kontext der Antriebsstrangentwicklung. Bisher beschäftigen sich nur Altenburg und Sommer Obando mit dem konkreten Einsatz von ML in der Steuerung und Regelung von Fahrzeuggetrieben und dies nur mit verstärkendem Lernen, sowie Veith mit einem überwachtem Lernansatz. Nicht überwachtes Lernen z.B. zur Unterstützung der Validierung durch Datenanalysen wurde im Antriebsstrang bisher nicht untersucht.

3 Forschungsbedarf und Zielsetzung

Nachfolgend werden basierend auf dem Stand der Forschung der Forschungsbedarf aufgezeigt und die Zielsetzung abgeleitet. Darauf aufbauend werden zugehörige Forschungsfragen formuliert.

3.1 Forschungsbedarf

Die Entwicklung von Getrieben für Hybridantriebe steht zahlreichen Herausforderungen gegenüber. Durch die Kombination mechanischer, tribologischer, elektrischer und hydraulischer Systeme weisen diese mechatronischen Systeme eine hohe Komplexität auf. Gleichzeitig existieren hohe Anforderungen an Komfort, Dynamik und Verbrauch über die gesamte Lebensdauer. Zusätzlich ergeben sich durch die Elektrifizierung weitere Fahrfunktionen wie der Motorwiederstart, elektrisches und hybrides Anfahren, die die Komplexität weiter erhöhen (vgl. Kapitel 2.1).

Zugleich zeigen ML-Methoden herausragende Ergebnisse in vielen Bereichen wie der Bilderkennung, Sprachverarbeitung, komplexen Strategiespielen oder der Diagnose von Krankheiten (vgl. Kapitel 1).

Wie im Stand der Forschung bereits aufgezeigt, werden ML-Methoden in Fahrzeuggetrieben, speziell in Hybridantrieben hauptsächlich für die Optimierung des Energie-Managements eingesetzt. Für die Ansteuerung existieren erste Ansätze. Eine konkrete technische Umsetzung in Serienfahrzeugen ist nicht bekannt (vgl. Kapitel 2.4). Der bisherige Einsatz von ML in Fahrzeuggetrieben beschränkt sich auf die Verbesserung von Simulationsmodellen für die Entwicklung und die Methodenanwendung ist teilweise nicht detailliert beschrieben.

Zusätzlich müssen ML-Methoden, die größtenteils in digitalen Anwendungen genutzt werden, adaptiert und für den Kontext der Produktentwicklung nutzbar gemacht werden. Dabei gilt es auch die Anwendbarkeit dieser Methoden durch die Getriebeentwickelnden zu untersuchen.

Deshalb soll diese Arbeit eine Brücke schlagen ML-Methoden in Antriebssystemen, speziell für die Ansteuerung von parallelen hybriden Fahrzeuggetrieben

mehrwertbringend einzusetzen. Parallele Hybridantriebe sind dabei am weitesten verbreitet, weshalb sich diese Forschungsarbeit auf diese fokussiert.

Die Forschungsumgebung bei der Mercedes-Benz Group AG bietet hierbei die einmalige Chance die entwickelten Methoden auf praxisrelevante Entwicklungsaufgaben anzuwenden und zu evaluieren.

3.2 Zielsetzung

Ziel der Forschungsarbeit ist es, Entwickelnde mit dem Einsatz von ML-Methoden in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen.

Zusätzlich soll diese Arbeit einen Beitrag zur KaSPro – Karlsruher Schule für Produktentwicklung leisten und diese durch eine ML-Methodenbeschreibung im Kontext der Getriebefunktionsentwicklung erweitern und die Mehrwerte und Anwendbarkeit der Methoden anhand aktueller Entwicklungsaufgaben aufzeigen.

Ausgehend von dieser Zielsetzung werden die folgenden Forschungsfragen formuliert. Diese ermöglichen eine strukturierte Erarbeitung der Inhalte, operationalisieren die Zielsetzung und müssen zur Zielerreichung der Forschungsarbeit beantwortet werden.

Forschungsfrage I (FF-I)

Welche Herausforderungen existieren in der Entwicklung von Steuerungs- und Regelungsalgorithmen von parallelen hybriden Fahrzeuggetrieben?

Forschungsfrage 2, 3 und 4 bauen auf Analyseergebnissen der ersten Forschungsfrage auf.

Forschungsfrage II (FF-II):

Wie müssen die Methoden des maschinellen Lernens gestaltet werden, um Entwickelnde bei der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen?

Forschungsfrage III (FF-III):

Welche messbaren Mehrwerte und Potentiale ergeben sich durch die Methoden des maschinellen Lernens?

Forschungsfrage IV (FF-IV):

Wie wird die Anwendbarkeit der entwickelten Methoden des maschinellen Lernens durch Entwickelnde bewertet?

4 Forschungsvorgehen

In diesem Kapitel wird das methodische Vorgehen zur Beantwortung der Forschungsfragen erläutert. Zunächst wird die Forschungsmethode Design Research Methodology (DRM) nach Blessing und Chakrabarti (2009) beschrieben. Es folgt eine Beschreibung der Forschungsumgebung, in welcher das Forschungsvorhaben durchgeführt wird. Abschließend werden das konkrete Vorgehen der Forschungsarbeit sowie das Forschungsdesign vorgestellt.

4.1 Forschungsmethode

Ein forschungsmethodisches Vorgehen unterstützt durch dessen Systematik und Methoden die Wissenschaftlichkeit und Validität des zu bearbeitendem Forschungsvorhabens. Die Design Research Methodology nach Blessing und Chakrabarti (2009) dient als Grundlage für die vorliegende Arbeit.

Die DRM gliedert sich in die folgenden vier Phasen:

- Für die *Klärung des Forschungsgegenstandes* wird der Stand der Forschung untersucht und damit das Forschungsthema eingeordnet und abgegrenzt. Darauf aufbauend wird eine Zielsetzung abgeleitet.
- In der *Deskriptiven Studie I* wird ein tieferes Verständnis der Ausgangssituation durch Analysen ermittelt. Es können bestehende Untersuchungen herangezogen oder diese durch eigene Untersuchungen erweitert werden.
- In der *Präskriptiven Studie* werden diese Erkenntnisse genutzt und eine konkrete Unterstützung für die Produktentwicklung erarbeitet.
- Die *Deskriptive Studie II* evaluiert abschließend die entwickelte Unterstützung.

Je nach Forschungsprojekt kann der Umfang der einzelnen Phasen variieren. Dabei gibt es innerhalb der einzelnen Phasen drei verschiedene Arten von Studien: literatur-basierte Studien, umfassende Studien und initiale Studien. Umfassende Studien ergänzen literatur-basierte Studien durch weitere empirische Analysen. Initiale Studien zeigen die Konsequenzen der Forschungsergebnisse und bereiten die Ergebnisse für weitere Nutzende auf (Blessing & Chakrabarti, 2009, S. 18).

Auf Basis der vorgestellten Phasen und Arten der Studien lassen sich sieben Forschungsprojekttypen ableiten (vgl. Abbildung 4.1).

| Projekt- typ | Klärung des Forschungs- gegenstandes | Deskriptive Studie I | Präskriptive Studie I | Deskriptive Studie II |
|-----------------|--|-------------------------|---------------------------------|--------------------------|
| 1 | Literatur-basiert → | Umfassend | | |
| 2 | Literatur-basiert → | Umfassend → | Initial | |
| 3 | Literatur-basiert → | Literatur-basiert → | Umfassend → | Initial |
| 4 | Literatur-basiert → | Literatur-basiert → | Literatur-basiert/ Umfassend | Umfassend |
| 5 | Literatur-basiert → | Umfassend → | Umfassend → | Initial |
| 6 | Literatur-basiert → | Literatur-basiert → | Umfassend → | Umfassend |
| 7 | Literatur-basiert → | Umfassend → | Umfassend → | Umfassend |

Abbildung 4.1: Projekttypen im DRM-Modell (Eigene Darstellung nach Blessing & Chakrabarti, 2009).

Diese Forschungsarbeit lässt sich Typ 5 zuordnen. Die Klärung des Forschungsgegenstandes erfolgt auf Basis der Literatur. Um ein Verständnis der Herausforderungen in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu erhalten, wird eine umfassende Studie anhand von Experteninterviews in der Forschungsumgebung und einer Systemanalyse anhand des C&C²-Ansatzes durchgeführt. In der Präskriptiven Studie werden unterstützende Methoden auf Basis der Vorgehensmodelle der Literatur an den Entwicklungskontext adaptiert. Abschließend werden in der Deskriptiven Studie II die Methoden angewendet und evaluiert.

4.2 Empirische Methoden

In dieser Arbeit werden verschiedene empirische Methoden genutzt, um die Beantwortung der Forschungsfragen zu ermöglichen. Deren Anwendungsfeld, sowie Vor- und Nachteile werden anhand der nachfolgenden Methodensteckbriefe nach Marxen (2014) dargestellt. Aspekte, die den Methodeneinsatz für die vorliegende Arbeit begründen, sind in Fettdruck hervorgehoben.

Als eine Methode zur Beantwortung der ersten Forschungsfrage werden leitfaden-geführte Experteninterviews durchgeführt (vgl. Abbildung 4.2). Damit werden Daten

hinsichtlich der Herausforderungen der Getriebefunktionsentwicklung direkt mit involvierten Entwicklungs- und Projektmitarbeitenden erhoben. Die Interviews werden anhand einer qualitativen Inhaltsanalyse ausgewertet und ein induktives Vorgehen zur Kategorienbildung genutzt. Über die Häufigkeit der Nennung kann eine quantitative Bewertung der Kategorien stattfinden.

| Interview | |
|----------------|---|
| Anwendungsfeld | <ul style="list-style-type: none"> • Empirische Forschung, Analyse von realen Entwicklungsprozessen • Experimentelle Studien, Evaluation in kontrollierten Umgebungen • Anwendungsstudien, Einsatz von Entwicklungsunterstützung in realen Prozessen • Felder der Entwicklungsmethodik, in der die Wahrnehmung der Menschen bzgl. einer Situation oder einem Prozess von Interesse ist |
| Vorteile | <ul style="list-style-type: none"> • Nachfragen zur Präzisierung der Antworten sind möglich, Missinterpretationen können vermieden werden • Forscher kann eingreifen • Es stehe eine große Anzahl an Interview-Tools zur Verfügung |
| Nachteile | <ul style="list-style-type: none"> • Aufwändige Analyse bzw. Transkription der Interviews • Erschwerter Zugang zu großen Auswahlgruppen, da Interviews mit einem hohen zeitlichen Aufwand verbunden sind |

Abbildung 4.2: Methodensteckbrief – Interview (Eigene Darstellung nach Marxen, 2014).

In einer Anwenderstudie wird die Nutzbarkeit und Anwendbarkeit der Methoden in der Forschungsumgebung untersucht. Um die Meinung der Nutzenden hinsichtlich der Qualität, den Unterstützungsgrad und die Anwendbarkeit dieser Methoden zu erheben, wird ein Fragebogen genutzt (vgl. Abbildung 4.3).

| Fragebogen | |
|----------------|---|
| Anwendungsfeld | <ul style="list-style-type: none"> • Empirische Forschung, Analyse von realen Entwicklungsprozessen • Experimentelle Studien, Evaluation in kontrollierter Umgebung • Anwendungsstudien, Einsatz von Entwicklungsunterstützung in realen Prozessen • Felder der Entwicklungsmethodik, in denen die Wahrnehmung von Menschen bezüglich einer Situation oder einem Prozess von Interesse ist. |
| Vorteile | <ul style="list-style-type: none"> • Direkte Datenerfassung • Einfacher Zugang zu großen Auswahlgruppen, vor allem mit Online-Umfragen • Es steht eine große Anzahl von Online-Umfrage-Tools zur Verfügung |
| Nachteile | <ul style="list-style-type: none"> • Forscher kann nicht eingreifen • Präzise Fragen müssen formuliert werden. Missverständnisse durch die Teilnehmern können die komplette Studie ruinieren. |

Abbildung 4.3: Methodensteckbrief – Fragebogen (Eigene Darstellung nach Marxen, 2014).

4.3 Forschungsumgebung: Mercedes-Benz Group AG

Die Forschungsarbeit ist in Kooperation mit dem Entwicklungsbereich der Mercedes-Benz Group AG entstanden. Diese Untersuchungsumgebung dient dabei zur Erhebung empirischer Daten zur Systemanalyse in der Deskriptiven Studie I sowie zur Evaluierung der Methoden in der Deskriptiven Studie II anhand praxisrelevanter Entwicklungsaufgaben.

Um die Ergebnisse dieser Arbeit einordnen und deren Übertragbarkeit bewerten zu können, wird nachfolgend eine Kurzvorstellung über das Unternehmen sowie der Entwicklungsabteilung gegeben.

Kurzvorstellung Mercedes Benz Group AG. Die Anfänge der Mercedes-Benz Group AG reichen bis in das Jahr 1883 zurück, in dem die Benz & Co Rheinische Gasmotorenfabrik gegründet wurde und 1926 mit der Daimler-Motoren-Gesellschaft zur Daimler-Benz AG fusionierte. Bei der Mercedes-Benz Group AG arbeiteten 2021 verteilt auf Standorte in 44 Länder 172.000 Mitarbeitende. Der Hauptsitz ist Stuttgart-Untertürkheim. Das Produktportfolio reicht von Kompaktfahrzeugen über Sport Utility Vehicles (SUV) und Sportlimousinen hin zu Fahrzeugen des Luxussegments. Im Jahre 2021 verkaufte die Mercedes-Benz Group AG insgesamt 2,75 Millionen Fahrzeuge und erwirtschaftete einen Umsatz von 168 Mrd. € und einen Gewinn von 23,4 Mrd. €.¹

Kurzvorstellung Abteilung Funktionsentwicklung und Applikation Front-Antriebe. Die Untersuchungen der Forschungsarbeit finden in der Abteilung Funktionsentwicklung und Applikation Front-Antriebe bei der Mercedes-Benz Group AG statt, in der die Software für konventionelle und hybridisierte Doppelkupplungsgetriebe entwickelt wird. Die Getriebesoftware umfasst dabei unter anderem das Schaltprogramm, Schaltablauf, Diagnose-Funktionen, Ansteuerung der Hydraulik, Sensorsignalverarbeitung und Funktionale Sicherheit. Diese Funktionen werden entwickelt sowie Applikations-Parameter des Getriebes zur Verbesserung der Fahrbarkeit angepasst.

¹ Geschäftsbericht Mercedes-Benz-Group AG 2021 (www.group.mercedes-benz.com)

4.4 Vorgehensweise und Forschungsdesign

Abbildung 4.4 gibt eine Übersicht über das Vorgehen der Forschungsarbeit. Die einzelnen Kapitel werden dabei den Phasen der DRM zugeordnet und deren Ziele dargestellt. Darüber hinaus werden die dabei verwendeten Methoden aufgezeigt.



Abbildung 4.4: Forschungsvorgehen der Arbeit in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).

Zur Klärung des Forschungsgegenstands und Ableitung einer Forschungslücke wird in Kapitel 2 der Stand der Forschung unter anderem zu hybriden Fahrzeuggetrieben, ML und der bisherige Einsatz von ML-Methoden in Fahrzeuggetrieben vorgestellt. Darauf aufbauend wird in Kapitel 3 das Forschungsziel der Arbeit abgeleitet, was anhand der Forschungsfragen operationalisiert wird.

In der Deskriptiven Studie I (Kapitel 5) wird der Forschungsbedarf anhand der Herausforderungen in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben konkretisiert. Diese werden mit Hilfe von Experteninterviews in der Forschungsumgebung abgeleitet. Experteninterviews bieten den Vorteil einer retropektiven Betrachtung durch Experten mit mehrjähriger Berufserfahrung direkt aus der Praxis.

Eine anschließende Systemanalyse eines hybriden Doppelkupplungsgetriebes der Mercedes-Benz Group AG ergänzt weitere Herausforderungen und präzisiert die Einflussfaktoren auf die Ansteuerung anhand des C&C²-Ansatzes. Dabei liegt der Fokus auf den Steuer- und Regelaufgaben des Getriebes.

In der Präskriptiven Studie (Kapitel 6) erfolgt die Synthese von unterstützenden methodischen Lösungsansätzen durch die Adaption der bestehenden ML-Methoden an den Entwicklungskontext. Die drei Hauptmethoden des ML überwachtes Lernen, verstärkendes Lernen und nicht überwachtes Lernen werden hierfür eingesetzt.

Die abgeleiteten Methoden werden in der Deskriptiven Studie II auf aktuelle Entwicklungsaufgaben der Mercedes-Benz Group AG angewendet und anhand messbarer Erfolgskriterien evaluiert (Kapitel 7). Zu diesen gehören Kupplungstemperaturmodelle, die Optimierung einer Kupplungsfüllung sowie eine Datenanalyse zum hybriden Motorwiederstart.

In Kapitel 8 wird die Nutzbarkeit und Anwendbarkeit der Methoden in Studien untersucht. Für zwei der entwickelten Methoden wird ein Workshop mit anschließender Evaluation anhand eines Fragebogens durchgeführt. Für die dritte lösen Anwendende eine Aufgabenstellung mit Hilfe einer Anleitung und eines entwickelten Software-Tools. Die Ergebnisse werden ausgewertet und die Anwendenden evaluieren die Methode anhand eines Fragebogens. Diese Methoden werden genutzt, um die Wahrnehmung des potentiellen Anwendenden bezüglich der Methoden zu untersuchen.

Abschließend gibt Kapitel 9 eine Zusammenfassung der Forschungsarbeit und zeigt Anknüpfungspunkte für weitere Arbeiten auf.

5 Herausforderungen in der Getriebefunktionsentwicklung von parallelen hybriden Fahrzeuggetrieben

Dieses Kapitel leitet die Herausforderungen in der Getriebefunktionsentwicklung von parallelen hybriden Fahrzeuggetrieben ab, um die erste Forschungsfrage zu beantworten. Zusätzlich wird damit der bereits in Kapitel 3.1 abgeleitete Forschungsbedarf konkretisiert. Aufbauend darauf können dann die ML-Methoden Anwendung finden.

Forschungsfrage I (FF-I)

Welche Herausforderungen existieren in der Entwicklung von Steuerungs- und Regelungsalgorithmen von parallelen hybriden Fahrzeuggetrieben?

Abbildung 5.1 gibt eine Übersicht über die Methoden und Inhalte von Kapitel 5. Zunächst werden in Kapitel 5.1 die Herausforderungen anhand von leitfadengeführten Experteninterviews abgeleitet. Diese werden mit einer qualitativen Inhaltsanalyse ausgewertet und anhand von Kategorien zusammengefasst. Über die Häufigkeit der Nennung kann eine quantitative Bewertung der Kategorien stattfinden. Die Experteninterviews werden mit Experten der Mercedes-Benz Group AG durchgeführt. Eine Übertragbarkeit der Ergebnisse auf andere Unternehmen ist denkbar, muss aber aufgrund von unternehmensspezifischen Faktoren und Marktbedingungen kritisch geprüft werden.

Anschließend wird in Kapitel 5.2 eine Systemanalyse eines parallelen Hybridgetriebes durchgeführt. Dabei werden zunächst die Fahrfunktionen des allgemeinen Systems Parallel-Hybrid analysiert. Danach werden am Beispiel eines hybriden DKG der Mercedes-Benz Group AG das Getriebesystem und die Kupplungsansteuerung mit dem C&C²-Ansatz analysiert. Damit werden fehlende Herausforderungen ergänzt und die bereits identifizierten Herausforderungen konkretisiert. Mit Hilfe des C&C²-Ansatzes werden ein Verständnis der Wirkzusammenhänge geschaffen und Einflussfaktoren auf die Ansteuerung abgeleitet. Kapitel 5.3 gibt abschließend ein Zwischenfazit.

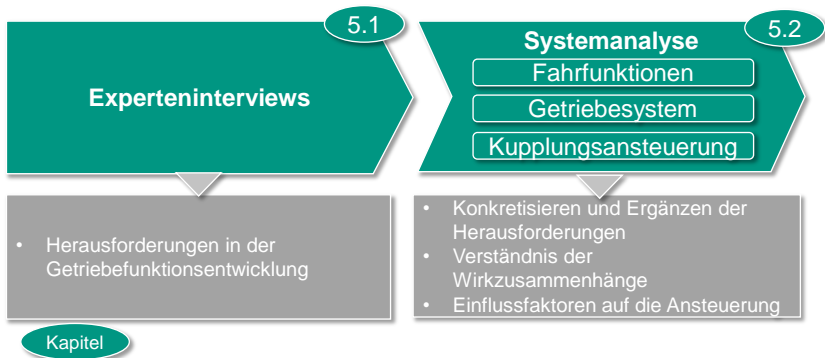


Abbildung 5.1: Vorgehen der ersten Deskriptiven Studie

5.1 Experteninterviews

Ziel der Experteninterviews¹ ist es, die Herausforderungen in der Entwicklung von Steuerungs- und Regelungsalgorithmen direkt am Nutzenden zu ermitteln und von deren Erfahrungswerten zu profitieren. Mit dieser Methode können umfangreiche Informationen von verschiedenen Praxiserfahrungen gesammelt und ausgewertet werden. Das verfügbare Sachwissen der Befragten kann direkt abgefragt werden und in kurzer Zeit viele Erkenntnisse liefern. Die spätere Methodenentwicklung kann dann gezielt an den identifizierten Herausforderungen anknüpfen.

Auf der Basis eines Interviewleitfadens² werden die Experten nach den Herausforderungen in der Entwicklung von Steuerungs- und Regelungsalgorithmen in Doppelkupplungsgetrieben befragt. Es wird nicht gezielt nach Herausforderungen gefragt, die nur bei hybriden Getrieben auftreten. Dies hat folgende Gründe:

¹ Die Anwendungsfelder, sowie Vor- und Nachteile des Interviews als empirische Methode werden auf S.62 erläutert.

² Der Interviewleitfaden ist dem Anhang auf S.XXIV beigelegt.

- Hybridisierte Getriebe besitzen sehr große Übernahmevariationen von konventionellen Getrieben. Aufgrund der Wechselwirkungen im Gesamtsystem entstehen Herausforderungen, die die Befragten bei einer gezielten Befragung nicht unbedingt der Hybridisierung zuordnen würden.
- Herausforderungen können im Übernahmeanteil des konventionellen Getriebes aufgrund der Wechselwirkungen durch die Neuentwicklungsanteile der Hybridisierung entstehen.

Somit kann mit dieser Befragung ein möglichst objektives Bild abgeleitet werden. In den Interviews wird gezielt nach Ursachen und Folgen der genannten Herausforderungen gefragt, um das Verständnis zu vertiefen und gleichzeitig die Zusammenhänge zu erkennen.

Zunächst wurden Personen anhand der folgenden Kriterien für das Studienpanel bestimmt:

- Entwickelnder in der Funktionsentwicklung von Automatikgetrieben
- Erfahrung mit konventionellen und hybriden Getrieben
- mind. 5 Jahre Berufserfahrung

Unter Beachtung dieser Kriterien wurden 14 Experten ausgewählt. Für die Interviews war es nicht möglich eine Transkription durchzuführen, weshalb während des Interviews Notizen gemacht wurden, auf deren Basis im Anschluss an das Interview Gedächtnisprotokolle erstellt wurden. Die Interviews dauerten zwischen 30 und 50 Minuten. Da Gedächtnisprotokolle keine wortwörtliche Wiedergabe darstellen, könnten möglicherweise vom Forscher interessante oder wichtige Punkte nicht aufgenommen werden. Aufgrund der breiten Informationsbasis durch die Anzahl der geführten Interviews, ist anzunehmen, dass dies ausgeglichen werden kann.

Die Ergebnisse werden anhand einer qualitativen Inhaltsanalyse ausgewertet. Dabei wird ein induktives Vorgehen zur Kategorienbildung nach Mayring genutzt (Mayring, 2016, S. 114). Anschließend werden die Kategorien anhand der Häufigkeit der Nennung quantitativ bewertet (vgl. Abbildung 5.2). Hierbei liegt der Fokus nicht auf konkreten Prozentsätzen, sondern auf wesentlichen Unterschieden der Häufigkeiten.

Die aufgeführten Kategorien von Herausforderungen werden nachfolgend anhand der Interviewergebnisse erläutert und interpretiert.

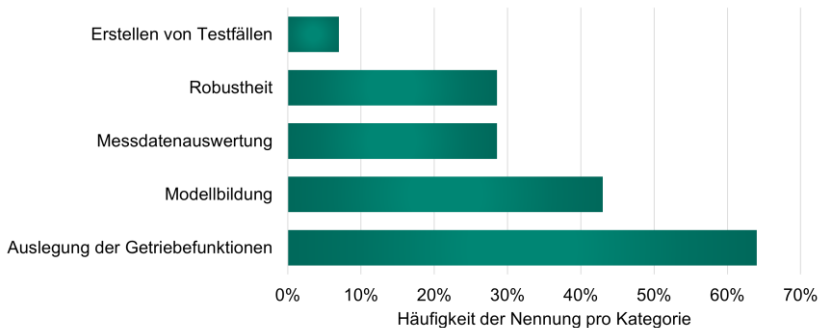


Abbildung 5.2: Ergebnisse der Experteninterviews, Häufigkeit der Nennung in Prozent pro Kategorie

Auslegung der Getriebefunktionen. Neun der 14 Befragten nennen die Auslegung der Getriebefunktionen (vgl. Kap 2.1.3) als Herausforderung. Nachfolgend werden einige technische Herausforderungen erläutert:

Aufgrund der Elektrifizierung werden elektrische Hydraulikpumpen eingesetzt, die als Element des Referenzsystems im Vergleich zur vorherigen Systemgeneration ein für den Entwickelnden bisher unbekanntes Verhalten aufweisen. Darüber hinaus werden Teilsysteme wie Ventile oder Drucksensoren ersetzt oder aus Kostengründen reduziert. Die hydraulische Strecke, Geometriedaten der Kupplungen, Strom-Druck-Kennlinien und Reibungszahlen sind neu zu ermitteln. Das gesamte Hydraulikverhalten besitzt dadurch ebenfalls einen hohen Neuheitsgrad für den Entwickelnden. Auch die Applikation der Kupplungsfüllung ist aufgrund des komplexen Hydraulikverhaltens aufwendig umzusetzen. In der Hydraulik kommt es zu Totzeiten, wobei eine parallele Durchführung von Aufgaben die Gesamtdynamik des Systems verbessern kann. Dies zu identifizieren, ist zeitlich aufwendig. Diese Änderungen lassen sich im Sinne der SGE als große Ausprägungsvariation beschreiben.

Im Schaltablauf gibt es zahlreiche Zustände. Es muss während der Auslegung sichergestellt werden, dass alle Zustände von den Algorithmen abgedeckt sind, damit es zu keinem Deadlock kommt. Fehlt z.B. eine Rückmeldung des Ölkoordinators, ist keine Schaltung möglich.

Speziell beim Hybrid ist die Applikation des Anfahrvorgangs aufgrund unterschiedlicher Leerlaufdrehzahlen aufwendig. Es kann zu einem Überfüllen der Kupplung kommen, was für den Fahrer als Ruck spürbar ist.

Zur Schwingungsreduzierung im Antriebsstrang werden die Kupplungen im Lastzustand in geringem Schlupf betrieben, auch Mikroschlupf genannt. Regler für diesen Mikroschlupfbetrieb besitzen sehr viele Abhängigkeiten, was die Reglerauslegung erschwert.

Aufgrund des Zusammenspiels mehrerer Teilsysteme des Antriebsstrangs haben Änderungen am Momenten-, Fahrpedal- oder Regelungsverhalten direkte Auswirkungen auf das Gesamtantriebsstrangverhalten. Diese Einflüsse auf das Gesamtsystemverhalten müssen während der Auslegung beachtet werden. Wie in der allgemeinen Antriebsstrangentwicklung existieren Zielkonflikte in der Auslegung z.B. zwischen Komfort und Dynamik.

Dadurch, dass Hardwareentscheidungen oft erst spät im Projekt getroffen werden, muss die Software möglichst alle Hardware-Varianten bereits im Vorfeld abdecken können. Die Software muss deshalb von Anfang an möglichst robust und parametrisiert ausgelegt werden. Kann eine spätere Hardware nicht von der Software korrekt angesteuert werden, führt dies zu aufwendiger Nacharbeit.

Die Komplexität der Software erschwert die Auslegung zusätzlich. Ursachen dafür sind die zahlreichen Betriebszustände, die abgedeckt werden müssen. Aufgrund der Komplexität werden zusätzliche Funktionen in der Software implementiert, um Einzelfälle von Betriebszuständen abzufangen. Dies führt wiederum zu einer komplexen Softwarestruktur. Die weiteren Folgen daraus sind neben der erschwerten Auslegung Fehler bei Schnittstellenänderungen von Modulen der Software und eine aufwendige Analyse von Fehlern. Zusätzlich existiert eine hohe Varianz in der Software selbst aufgrund von verschiedenen Fahrzeugmodellen und Softwareständen.

Modellbildung. Neben der Auslegung der Ansteuerung müssen die in der Software verwendeten Abbildungen der physikalischen Systemeigenschaften eine hohe Genauigkeit besitzen. Zu diesen Modellen gehören die Abbildungen des Reibverhaltens, der hydraulischen Strecke, der Kupplungshysterese, der Kupplungstemperatur und des Touch-Points.

Um die geforderten Drehmomentengenauigkeiten der Kupplungen sicherzustellen, muss ein genaues Modell des Reibungssystems vorliegen. Aufgrund der hohen Abhängigkeit von vielen Größen, ist z.B. die in den Berechnungsmodellen verwendete Reibungszahl sehr schwer zu bestimmen. Ebenso muss die Hysterese der Kupplung genau abgebildet werden. Die aktuellen Modelle sind hierbei teilweise ungenau. Für eine komfortable Schaltung muss der Touch-Point der Kupplung sehr genau bestimmt werden. Der Touch-Point ist dabei der Kupplungsdruck, bei dem die Lamellen anliegen, aber gerade kein Moment übertragen. Allein den

Drehzahleinfluss genau abzubilden ist sehr komplex. Auch die Kupplungstemperaturmodelle weisen im Vergleich zu Validierungsmessungen der Temperatur im Fahrzeug große Abweichung auf. Aufgrund der Hybridisierung entstehen weitere Lastfälle, wie rein elektrisches Anfahren, die von den Modellen ebenfalls abgedeckt werden müssen. Neben den Kupplungen werden für die Abbildung der hydraulischen Strecke wie der Hysterese der Ventile und der Temperaturabhängigkeit ebenfalls Modelle benötigt.

Ursachen für die Herausforderungen in der Modellbildung sind die hohe Komplexität aufgrund der hohen Anzahl an Betriebszuständen sowie des nichtlinearen Verhaltens der Hydraulik und der tribologischen Systemeigenschaften der Kupplungen. Außerdem unterscheiden sich Teilsysteme der Prüfstände von den späteren Serienfahrzeugen, wodurch Modelle mit einer hohen Abbildungsgüte am Prüfstand im Serienfahrzeug nicht mehr vollständig gültig sind. Außerdem können auf Prüfständen nicht alle Betriebszustände abgefahren werden, wodurch diese erst im Fahrzeug validiert werden können. Die Folgen daraus sind ein hoher Aufwand in der Modellbildung, der sich durch viele Messungen an Prüfständen oder Nacharbeit am Serienfahrzeug zeigt. Außerdem reduzieren Modelle mit geringer Abbildungsgüte die Produktqualität wie z.B. den Schaltkomfort.

Messdatenauswertung. Vier der 14 Befragten nennen die Messdatenauswertung als Herausforderung. Bei der Detailauswertung von Messdaten in der Funktionsentwicklung müssen als Validierungsaktivität Signalverläufe kontrolliert werden, um die korrekte Funktion sicherzustellen. Auswerteskripte werden teilweise von den PE-Teams selbst geschrieben, können aber nicht immer alle Fälle abfangen. Hierbei fehlt es an Automatisierung. Die Folge ist eine aufwendige manuelle Auswertung. Fehler im Feld werden anhand der Fehlerbeschreibung durch den Kunden und des Fehlerberichts aus dem Getriebe ausgewertet. Dabei ist Erfahrung mit bereits bekannten Fehlern essenziell. Ist diese nicht vorhanden, so müssen eine Vielzahl von Signalen analysiert werden, was zu einer aufwendigen Analyse führt.

Robustheit. Die Ansteuerung muss auch eine Robustheit gegenüber Fertigungstoleranzen, Alterung, Verschleiß und Umwelteinflüssen wie Umgebungstemperatur besitzen. Die genannten Modelle müssen dies ebenfalls abbilden können. Das Temperaturspektrum liegt dabei zwischen -30°C und 50°C Außentemperatur, bei der alle Getriebefunktionen bei geforderter Qualität erfüllt werden müssen. Dies führt insgesamt zu einer aufwendigen Validierung der Funktionen. Umgebungstemperaturen von bis zu -30°C können in Kältekammern erzeugt werden, bestehen aber nur für wenige Minuten, was zu einem langwierigen Testprozess führt. Verschleißgrenzen fallen teilweise erst im Feld auf und werden

als Anforderung in den Entwicklungsprozess zurückgeführt. Hoher Verschleiß ist dabei schwierig auf Prüfständen oder in Entwicklungsfahrzeugen zu reproduzieren. Die bereits beschriebene Beeinflussung anderer Teilsysteme aufgrund von Funktionsänderungen führt zu einem systematischen Abtesten aller Funktionen bei kleiner Änderung einer Funktion.

Folgen einer geringen Robustheit sind eine geringe Fahrbarkeitsqualität von Serienfahrzeugen und können im schlimmsten Fall zu einem Ausfall im Feld führen.

Erstellen von Testfällen. Abschließend wird auch das Erstellen von Testfällen als Teil der Validierungsaktivitäten für die Funktionsabsicherung als Herausforderung genannt. Diese ist sehr aufwendig, da Testfälle die zahlreichen Betriebszustände des Getriebes abdecken müssen und gleichzeitig manuell erstellt werden.

Tabelle 5.1 fasst die Ursachen, Folgen und abgeleiteten Erkenntnisse aus den genannten Herausforderungen zusammen.

Tabelle 5.1 Ursachen, Folgen und abgeleitete Erkenntnisse der Experteninterviews

| Kategorie | Ursache | Folgen | Abgeleitete Erkenntnisse |
|--|---|---|--|
| Auslegung der Getriebe-Funktionen | Hoher Neuheitsgrad des Hydrauliksystems durch hohen Anteil an Gestalt- und Prinzipvariationen | Mehraufwand in der Auslegung | Umgang mit hohem Neuheitsgrad erleichtern |
| | Wechselwirkungen Gesamtantriebsstrang | Mehraufwand in der Auslegung | Analyse der Wechselwirkungen unterstützen |
| | Software für verschiedene PGs funktionsfähig | Mehraufwand in der Auslegung | Varianz der Software reduzieren |
| | Hohe Komplexität und Varianz der Software | Mehraufwand in der Auslegung, Fehler bei Schnittstellen änderungen von Softwaremodulen, aufwendige Analyse von Fehlern | Komplexität und Varianz der Software reduzieren |
| Modellbildung | hohe Komplexität aufgrund der hohen Anzahl an Betriebszuständen , sowie des nichtlinearen Verhaltens der Hydraulik und der Tribologie der Reibelemente. | Komplexe mathematische Modelle , viele Messungen an Prüfständen oder Nacharbeit am Serienfahrzeug | Modellbildungsaufwand reduzieren und Modellgüte erhöhen |
| Robustheit | Robustheit gegenüber weitem Temperaturbereich | Aufwändige Validierung | Aufwand Validierung reduzieren |
| | Robustheit gegenüber Fertigungstoleranzen | Aufwändige Validierung | Aufwand Validierung der reduzieren |
| Messdaten-auswertung | Manuelle Auswertung von Messdaten | Hoher Aufwand , fehlende Erkenntnisse | Aufwand Messdatenauswertung reduzieren |
| Testfallerstellung | Manuelle Testerstellung für zahlreiche Betriebszustände | Hoher Aufwand bei der Testerstellung | Aufwand Testfallerstellung reduzieren |

Wie bereits in der Einleitung des Kapitels erwähnt, muss bei den Interviewergebnissen beachtet werden, dass diese nur mit Experten der Mercedes-Benz Group AG durchgeführt wurden. Außerdem entwickeln die Befragten Doppelkupplungsgetriebe, weshalb sich ihre Aussagen auf dieses System beziehen. Eine Übertragbarkeit der Ergebnisse auf andere Unternehmen ist denkbar, muss aber aufgrund von unternehmensspezifischen Faktoren und Marktbedingungen kritisch geprüft werden. Außerdem könnten die Befragten Herausforderungen, die in Entwicklungsprojekten zum Zeitpunkt der Interviews aufgetreten sind, unterbewusst übergewichten.

5.2 Systemanalyse

Anhand der Systemanalyse werden die aus den Experteninterviews ermittelten Herausforderungen zunächst ergänzt. Dabei werden die Fahrfunktionen des Antriebsstrangs und die gesamte Ansteuerung des Getriebes betrachtet.

Anschließend wird das Getriebesystem vorgestellt, und die Kupplungsaktuierung als zentrale Funktion der Getriebeansteuerung mit Hilfe des C&C²-Ansatzes analysiert (vgl. Kapitel 2.2.4), um Wirkzusammenhänge zu ermitteln und Einflussfaktoren auf die Ansteuerung abzuleiten.

Die Analyse wird am Beispiel eines hybriden DKG der Mercedes-Benz Group AG durchgeführt.

5.2.1 Fahrfunktionen

Durch die Elektrifizierung des Antriebsstrangs entstehen eine Reihe neuer Fahrfunktionen. Bei Hybridantrieben wird rein elektrisches Fahren und hybrides Fahren möglich. Darüber hinaus kommt die Transition zwischen diesen Fahrmodi hinzu. Insgesamt ergeben sich für die rein elektrische Fahrt die Fahrfunktionen: Anfahren, Beschleunigen, Konstant-Fahrt und Verzögern. Diese Funktionen ergeben sich analog für das hybride Fahren. Als Übergang zwischen dem elektrischen und hybriden Fahrmodus dient der Motorwiederstart des VM. Bei der hybriden Fahrt ergänzt sich der Stopp des VM als Übergangsfunktion zur elektrischen Fahrt.

Neben diesen Zuständen und Transitionen kann während der Fahrt eine Lastpunktverschiebung des VM stattfinden. Dabei wird die EM als Generator betrieben, was die Last des VM erhöht. Mit der erzeugten elektrischen Energie wird

die Batterie geladen. Dadurch verschiebt sich der Betriebspunkt des VM in einen höheren Lastbereich und damit in einen Betriebspunkt höherer Effizienz, wodurch der Gesamtwirkungsgrad des Antriebs steigt. Wird zusätzliche Leistung benötigt, so kann durch „Boosten“ ein höheres Moment der EM angefordert werden, um die maximale Systemleistung beider Kraftmaschinen zu nutzen.

Um die hybriden Fahrfunktionen hinsichtlich einer Herausforderung zu bewerten, werden die Kriterien Fahrsituation, Interaktion des Fahrers, Komfortrelevanz und Anzahl und Art der Steueraufgaben genutzt (vgl. Abbildung 5.3). Fahrfunktionen in dynamischen Fahrsituationen, eine aktive Interaktion des Fahrers, hohe Komfortrelevanz und eine hohe Anzahl an Komponenten zur Ansteuerung stellen für PE-Teams eine Herausforderung dar.

| Kriterium | Ausprägung | |
|-----------------------------------|------------|-----------|
| Fahrsituation | stationär | dynamisch |
| Interaktion des Fahrers | passiv | aktiv |
| Komfortrelevanz | gering | hoch |
| Anzahl und Art der Steueraufgaben | gering | hoch |

Abbildung 5.3: Bewertungskriterien der Fahrfunktionen

Die zusammenfassende Bewertung der Fahrfunktionen ist in Tabelle 5.2 dargestellt.

Tabelle 5.2 Bewertung der Fahrfunktionen

| Fahrmodus | Fahrfunktionen | Fahrsituation | Komfortrelevanz | Interaktion Fahrer | Steueraufgaben |
|--------------|-------------------|---------------|-----------------|--------------------|-------------------|
| E-Fahrt | Anfahren | Dynamisch | ja | Aktiv | EM |
| | Konstant-Fahrt | Stationär | ja | passiv | EM |
| | Verzögern | Dynamisch | ja | Aktiv | EM, Bremse, K1/K2 |
| Hybrid-Fahrt | Anfahren | Dynamisch | ja | Aktiv | VM, EM, K1/K2 |
| | Konstant-Fahrt | Stationär | ja | passiv | VM |
| | Verzögern | Dynamisch | ja | Aktiv | EM, Bremse, K1/K2 |
| | Boosten | Dynamisch | ja | passiv | EM |
| | Lastpunkterhöhung | Stationär | ja | Passiv | EM, VM |
| Transition | Wiederstart VM | Dynamisch | ja | Aktiv | EM, VM, K0, K1/K2 |
| | Stopp VM | Stationär | ja | Passiv | VKM, K0 |

Die Fahrfunktion Konstant-Fahrt findet während der stationären Fahrt statt und wird nicht durch den Fahrer beeinflusst. Lediglich die jeweilige Kraftmaschine muss dabei betrieben werden. Als Basis-Funktion des Antriebs stellt die Konstant-Fahrt deshalb keine Herausforderung dar.

Beim elektrischen Anfahren beschleunigt das Fahrzeug aus dem Stillstand durch das Moment der EM. Dabei ist das Anfahren mit und ohne Verbrenner identisch. Die EM stellt eine Leerlaufdrehzahl analog zum Verbrennungsmotor dar, die jeweilige Fahrkupplung wird aktuiert und im Schlupf betrieben, bis dieser durch die steigende Getriebeeingangsdrehzahl auf null reduziert wird.

Beim elektrischen Verzögern wird durch die EM ein Bremsmoment erzeugt, das die Batterie lädt. Dieser Vorgang wird auch als regeneratives Bremsen bezeichnet. Wird durch den Fahrer ein Bremsmoment angefordert, das größer ist als das maximale Rekuperationsmoment, so muss diese Momentendifferenz zusätzlich durch die mechanische Fahrzeugbremse gestellt werden. Der Übergang zwischen diesen Vorgängen darf für den Fahrer nicht spürbar sein. Es ist deshalb eine sehr hohe Genauigkeit an die Regelung der mechanischen Bremse gestellt. Da diese Fahrfunktion während einer dynamischen Fahrsituation auftritt, durch den Fahrer aktiv ausgelöst wird, die Komfortrelevanz hoch ist und EM und mechanische Bremse genau abgestimmt werden müssen, kann das regenerative Bremsen als Herausforderungen eingeordnet werden. Dieser Vorgang tritt auch beim hybriden Fahren auf. Beim Verzögern kann auch immer ein Gangwechsel stattfinden.

Fordert der Fahrer während der elektrischen Fahrt ein Moment, das größer ist als das maximale EM-Moment, so wird der VM gestartet. Das Zustarten kann außerdem auftreten, wenn die Batterie einen niedrigen SOC erreicht.

Beim Motorwiederstart müssen, speziell in dem betrachteten System eines hybriden Doppelkupplungsgetriebes in P2-Anordnung, eine Vielzahl von Steuer- und Regelaufgaben ausgeführt werden.

Das Zustartmoment des VM wird von der EM über die Trennkupplung K0 gestellt. Da die EM gleichzeitig ein Abtriebsmoment über die Fahrkupplungen erzeugt, ist das Moment für den Motorwiederstart begrenzt. Deshalb wird eine Drehzahlüberhöhung der EM genutzt. Beim Zustart wird die belastete Fahrkupplung in den Schlupf gebracht. Dadurch kann die EM eine höhere Drehzahl erreichen. Anschließend wird die Trennkupplung K0 geschlossen. Die EM wird dadurch abgebremst, wodurch dem Motorstart ein zusätzliches dynamisches Moment zur Verfügung steht. Über den Anpressdruck der K0 kann das Drehmoment eingestellt werden, sodass der VM auf die Zieldrehzahl beschleunigt. Erreicht der VM die Zieldrehzahl und das Zielmoment, so muss dieses Moment über die Fahrkupplung an den Abtrieb gebracht werden. Die Komfortanforderungen sind hoch. Der Zustart soll vom Fahrer nicht wahrgenommen werden und möglichst schnell erfolgen, um ein dynamisches Fahrverhalten zu ermöglichen. Durch die hohe Anzahl an Steueraufgaben, die hohen Komfortanforderungen und die aktive Auslösung des

Vorgangs durch den Fahrer während eines dynamischen Fahrvorgangs handelt es sich beim Motorwiederstart ebenfalls um eine Herausforderung.

Beim hybriden Anfahren beschleunigt das Fahrzeug aus dem Stillstand durch Momente des VM und der EM. Diese müssen dabei genau abgestimmt werden. Der eigentliche Anfahrprozess wird dabei über die Fahrkupplung geregelt. Diese bestimmt durch den Kupplungsdruck das übertragbare Moment. Darüber hinaus wird das Anfahren aktiv vom Fahrer eingeleitet und besitzt hohe Komfortrelevanz, weshalb auch diese Fahrfunktion eine Herausforderung darstellt.

Beim Boosten und der Lastpunkterhöhung wird durch die EM ein Moment bzw. Bremsmoment gestellt. Analog zur Lastpunkterhöhung findet das Boosten passiv statt. Die Betriebsstrategie entscheidet abhängig vom Fahrerwunsch, ob das Boosten ausreicht oder der VM zusätzlich gestartet werden muss. Da beide Fahrfunktionen lediglich durch eine Ansteuerung der EM durchgeführt werden, stellen sie aus Getriebebesicht keine Herausforderung dar.

Beim Übergang vom hybriden in den elektrischen Fahrmodus wird der VM gestoppt. Da der VM durch die Trennkupplung K0 vom Antrieb abgetrennt werden kann, hat das Stoppen des VM nur sehr geringen Einfluss auf den Komfort. Deshalb stellt auch diese Fahrfunktion aus Getriebebesicht keine Herausforderung dar.

5.2.2 Getriebesystem

Bereits im Stand der Forschung wurde das technische System des hybriden DKG sowie die sich daraus ableitende Software-Architektur beschrieben (vgl. Kapitel 2.1). Die nachfolgende Systemanalyse baut darauf auf und fokussiert sich auf die Funktionen, die die Getriebeansteuerung ermöglicht. Diese werden anschließend genauer betrachtet, um vorhandene Herausforderungen zu identifizieren. Abbildung 5.4 zeigt die Prinzipskizze eines hybridisierten DKG und dessen Antriebsstrangkomponenten. Der Gesamtantriebsstrang ist als Frontantrieb aufgebaut.

Das System-under-Investigation (Sul) der Analyse ist der gesamte Antriebsstrang, wobei der Fokus auf den Schaltelementen (K0, K1, K2), dem Gangsteller, den Hydraulikkreisläufen für die Kupplungsbetätigung, den Kühlölkreis, der Gangsteller-Hydraulik und den Getriebegängen liegt. Die Ansteuerung des Getriebes in der Ausprägung der Getriebesoftware ist das System-in-Development (SiD).

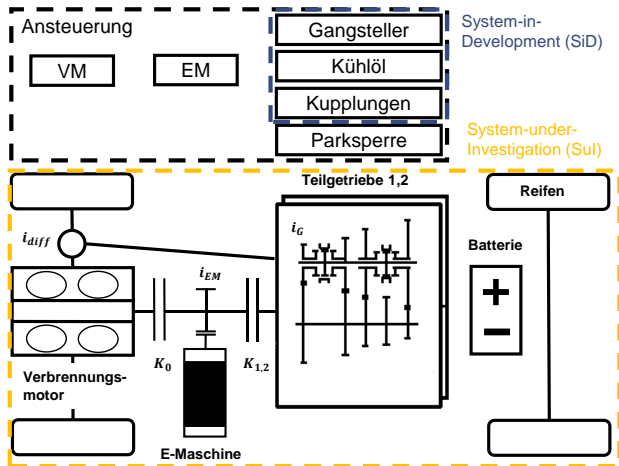


Abbildung 5.4: Prinzipskizze eines hybriden DKG in P2-Anordnung mit Ansteuerelementen

Beim elektrifizierten Doppelkupplungsgetriebe in P2-Anordnung sitzt die EM zwischen VM und Getriebe. In den Entwicklungsaufgaben in Kapitel 7 werden zwei Systemgenerationen betrachtet. Für die $G_n^{\{hybrid\}}$ ist die Übersetzung i_{EM} gleich eins. Die EM sitzt koaxial zur Getriebeeingangswelle ohne eine Zahnradgetriebestufe (Kapitel 7.1 und Kapitel 7.3). Die Doppelkupplung ist im Rotor der EM integriert (vgl. Abbildung 5.5).

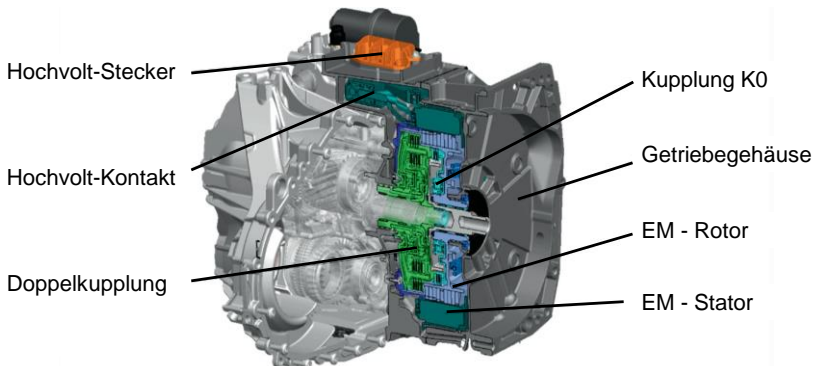


Abbildung 5.5: Hauptkomponenten des Getriebes der Systemgeneration $G_n^{\{hybrid\}}$ (Maisch et al., 2019).

Für die $G_{n+1}^{\{hybrid\}}$ ist die Übersetzung i_{EM} größer eins und die EM sitzt achsparallel zum Getriebe und überträgt ihr Moment über eine Zahnradstufe (Kapitel 7.2).

Der VM kann über die Trennkupplung K0 vom Antrieb abgetrennt werden, was den rein elektrischen Fahrbetrieb ermöglicht. Nach der EM folgen im Kraftfluss die beiden Fahrkupplungen K1 und K2, die als Schalt- und Anfaherelemente dienen, sowie die entsprechenden Teilgetriebe 1 und 2.

Beim Übergang vom rein elektrischen in den hybriden Fahrmodus wird der VM für beide Systemgenerationen über die EM gestartet. Dabei wird die Trennkupplung K0 geschlossen und ein Drehmoment an den VM übertragen, sodass dieser startet. Das Erreichen der Zieldrehzahl findet dann über das eigene VM-Moment statt. Kurz vor dem Schließen der Trennkupplung K0 wird die belastete Fahrkupplung im Schlupf betrieben, um die EM-Drehzahl anzuheben. Dieses Drehzahlniveau wird beim Zustart des VM wieder abgebaut, um ein zusätzliches Start-Moment bereitzustellen. Bei eingesetzten Kupplungen K0, K1 und K2 handelt es sich um nasslaufende Mehrscheibenlamellenkupplungen (vgl. Kapitel 2.1.3).

Die Ansteuerung des Doppelkupplungsgetriebes findet über mehrere Aktuatoren statt. Für das Anfahren, Schaltvorgänge und den Motorwiederstart müssen die entsprechenden Fahrkupplungen K1 und K2, sowie die Trennkupplung K0 angesteuert werden. Der Gangsteller zur Aktuierung der Gänge, der Kühl- und Schmierölkreislauf sowie die Parksperre sind über einen eigenen Hydraulikkreislauf verbunden. Die Kupplungen K1, K2 und K0 besitzen einen davon unabhängigen Hydraulikkreislauf. Jedes dieser Elemente wird bedarfsgerecht über Hydraulikventile angesteuert werden. Abschließend werden auch die Ansteuerung des VM und der EM wie Drehmoment und Drehzahl von den Getriebesignalen bestimmt.

Diese Getriebefunktionen werden angelehnt an die Software-Architektur eines konventionellen DKG (vgl. Kap.2.1.3) geordnet und um hybride Fahrfunktionen ergänzt. Zur Übersichtlichkeit wird die Parksperrenansteuerung nicht abgebildet (vgl. Abbildung 5.6).

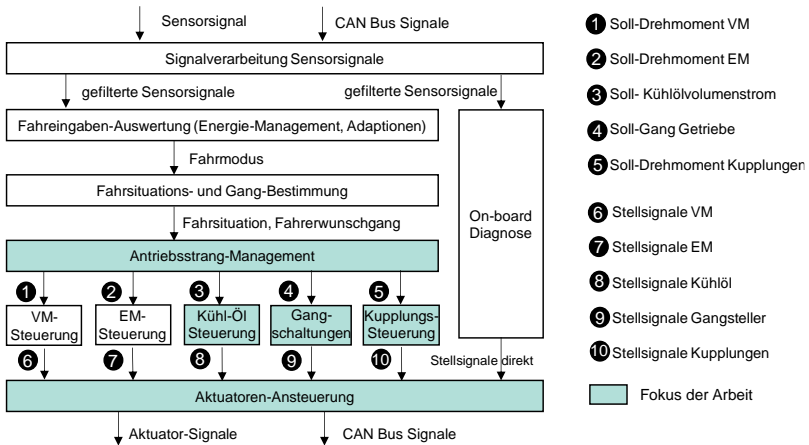


Abbildung 5.6: Softwarearchitektur eines hybriden Doppelkupplungsgetriebes

Neben diesen direkten Ansteuerungsfunktionen gibt es weitere übergeordnete Getriebefunktionen. Dazu gehören die Signalverarbeitung der Sensorsignale, Fahreingaben-Auswertung, Fahrsituations- und Gangbestimmung, das Antriebsstrang-Management und die On-Board Diagnose. Diese übergeordneten Funktionen werden im Rahmen der Forschungsarbeit nicht näher untersucht. Auch die Steuerung- und Regelung des VM und der EM werden nicht näher betrachtet, sondern dienen als Schnittstellen zum Getriebesystem.

Die Kupplungsansteuerung ist die zentrale Funktion der Getriebeansteuerung, da die Kupplungen für die Fahrfunktionen Anfahren und Motorwiederstart im hybriden Antriebsstrang sowie für den Gangwechsel verantwortlich sind. Die Ansteuerung wirkt sich wiederum direkt auf die Fahrzeugdynamik und den Fahrkomfort aus. Deshalb wird die Kupplungsansteuerung nachfolgend näher analysiert, um die Wirkzusammenhänge zu verstehen und Einflussfaktoren auf die Ansteuerung abzuleiten.

5.2.3 Kupplungsansteuerung

Abbildung 5.7 zeigt den Steuer- und Regelkreis der Kupplungsansteuerung. Das Steuergerät stellt einen Soll-Druck der Kupplung über die Kupplungsaktuatorik ein. Dies wird über ein Magnet-Regel-Ventil mit einer Strom-Druck-Kennlinie umgesetzt. Über einen Drucksensor wird der Druck im Kupplungskolben gemessen und

zurückgeführt. Ein PID-Regler in der Getriebe-Software nutzt dieses Signal, um den Soll-Druck einzuregeln. Analog wird zum Beispiel für eine Kupplungsfüllung eine Steuerung und somit eine direkte Vorgabe des Solldrucks vorgegeben. Dieser wird über die Ventilkennlinie wiederum in Sollströme umgerechnet. Adaptionen für die Reibungszahl und den Touch-Point gleichen Einflüsse wie Alterung oder Verschleiß über Lebensdauer aus.

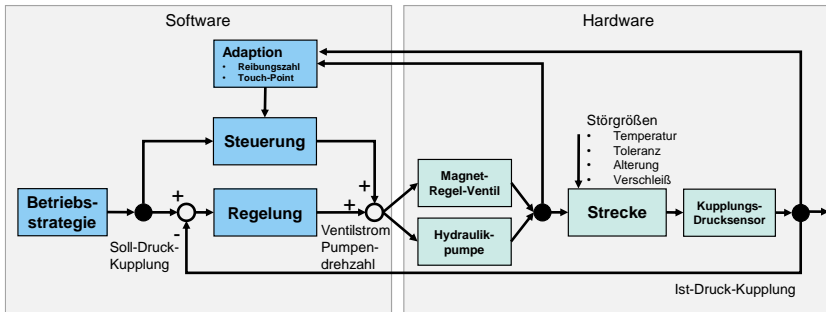


Abbildung 5.7: Steuer- und Regelkreis Kupplungsansteuerung

Die physikalische Strecke des Systems besteht aus Aktuatorik (Magnet-Regel-Ventil, Hydraulikpumpe), dem hydraulischen System aus Leitungen und Blenden, sowie der Kupplungsmechanik bestehend aus Lamellen der jeweiligen Kupplung (vgl. Abbildung 5.8).

Die Kupplungsansteuerung setzt folgende Funktionen um:

- Schließen der Kupplung (Füllen)
- Drehmoment der Kupplung einstellen
- Öffnen der Kupplung (Leeren)

Da die Funktion „Drehmoment der Kupplung einstellen“ die Hauptfunktion der Kupplung darstellt und diese beim Anfahren sowie Motorwiederstart wichtig ist, wird diese nachfolgend analysiert. Für die Ansteuerung wird dabei die untergeordnete Funktion „Kupplungsdruck einstellen“ analysiert, worin auch der Modellzweck besteht. Zur Vereinfachung wird nur ein Zustand betrachtet, kein zeitlicher Verlauf der Funktion. Nachfolgend werden anhand des C&C²-Ansatzes die Wirkzusammenhänge der Funktion und die Einflussfaktoren auf die Ansteuerung abgeleitet.

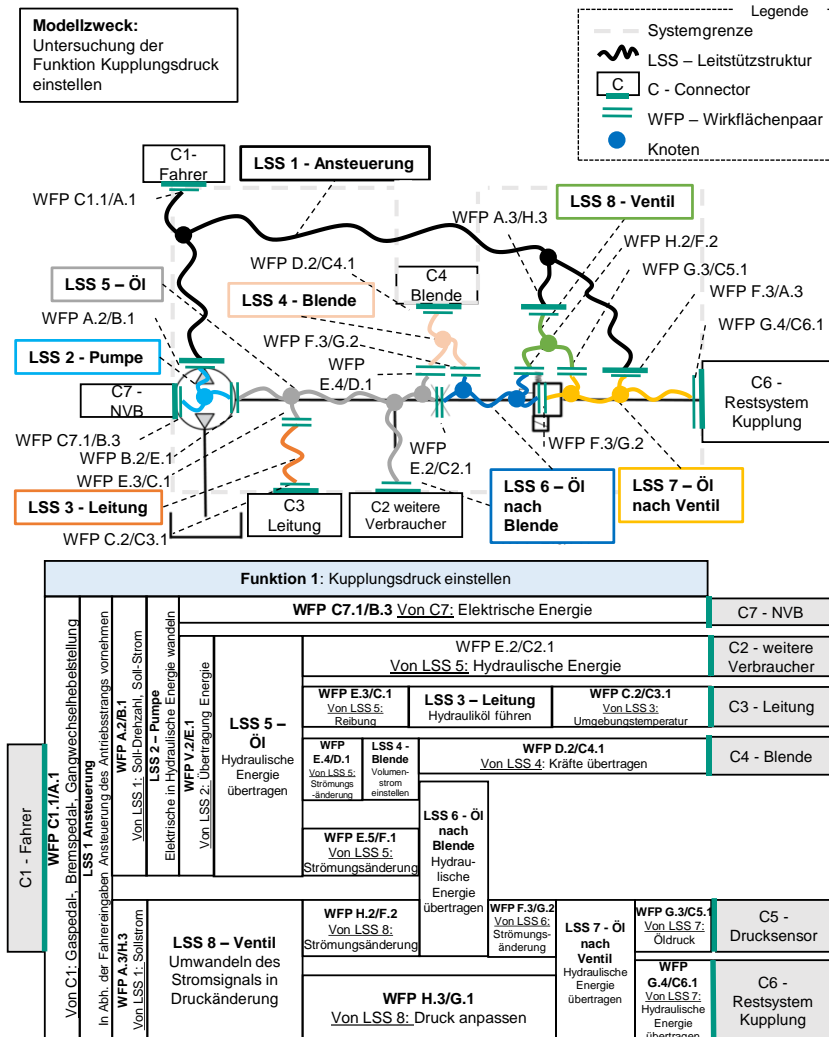


Abbildung 5.8: Identifikation von Einflussfaktoren auf die Kupplungsansteuerung anhand des C&C²-Modells der Kupplungsansteuerung und dem zugehörigen Wirknetz

Abbildung 5.8 zeigt das C&C²-Modell der hydraulischen Kupplungsaktuierung und das entsprechende Wirknetz. Die Systemgrenze bildet dabei die Eingaben des Fahrers sowie das hydraulische Kupplungsaktuierungssystem von der Hydraulikpumpe bis zum Kupplungskolben. Die Aktuierung wird nur für eine Kupplung dargestellt.

Zur Vereinfachung des Modells wird das Element des Knoten genutzt. Wenn mehrere LLS einen Ringschluss bilden, können diese zu einem Knoten konzentriert werden (Tröster, Prager, Schmidt & Albers, 2023). Zur Übersichtlichkeit werden die durch den Knoten konzentrierten LSS im Wirknetz als eine LLS dargestellt.

Über die LSS 1 der Ansteuerung werden die Fahrereingaben, die als Connector abgebildet sind (C1) anhand der Steuerungs- und Regelungsalgorithmen in Signale an die Hydraulikpumpe sowie an das Magnet-Regel-Ventil übermittelt.

Die LSS 2 - Hydraulikpumpe wandelt elektrische Energie, die über den Connector 7 der Nieder-Volt-Batterie (NVB) eingebracht wird, anhand der Vorgabe der Ansteuerung in mechanische Energie um. Über die Ansteuerung werden die Pumpendrehzahl, die den Volumenstrom stellt und ein Strom, der den Hydraulikdruck stellt, eingestellt. Diese Energie wird über ein Wirkflächenpaar an die LSS 5 Hydrauliköl übertragen. Schwingungen der Hydraulikpumpe werden dabei ebenfalls übertragen. Elektrische Hydraulikpumpen werden benötigt, um die Hydraulikleistung zur Verfügung zu stellen, auch wenn der VM abgeschaltet ist. Außerdem können diese unabhängig von der VM-Drehzahl betrieben werden und benötigen deutlich weniger Leistung für den Betrieb (Fischer et al., 2016, S. 211). Zur Steigerung der Energieeffizienz der Pumpe wird ein möglichst kleiner Ölhaushalt genutzt, wodurch der Systemdruck des Hydrauliksystems deutlich sensibler auf Volumenänderungen reagiert.

Das Hydrauliköl ist aufgrund der unterschiedlichen Eigenschaften hinsichtlich Strömung und Druck in der Gesamthydraulik in drei einzelne LSS unterteilt. Die LSS 5 nimmt die mechanische Energie der Pumpe auf. An der Blende wird die Strömung homogenisiert, wodurch das Öl nach der Blende als LSS 6 betrachtet wird. Das Magnet-Regel-Ventil stellt anschließend den Kupplungsdruck ein. Das Öl nach dem Ventil wird als LSS 7 betrachtet. Über WFP stehen die einzelnen Abschnitte miteinander in Kontakt.

Die LSS 5, 6 und 7 haben die Hauptfunktion die hydraulische Energie an das Restsystem Kupplung (C6) zu übertragen. Diese ermöglichen somit die übergeordnete Funktion, das Kupplungsmoment zu einzustellen.

Die LSS 5, 6 und 7 stehen alle im Kontakt zu den Leitungen. Zur Vereinfachung ist nur das WFP zu LSS 5 im C&C²-Modell dargestellt. Der Kontakt zwischen Öl und Leitung erzeugt Reibung, die zu Energieverlusten führt, welche in Form von Wärme an das Öl und an die Leitungen abgegeben werden. Die LSS 3 Leitung besitzt mechanische Eigenschaften wie eine Steifigkeit und Wärmeleitungseigenschaften. Über den Connector C3 Leitung wird die Umgebungstemperatur aufgeprägt. Die Eigenschaften der LSS 5, 6 und 7 sind abhängig von der Temperatur. Bei niedrigen Temperaturen ($<0^{\circ}\text{C}$) ist die Viskosität des Hydrauliköls deutlich höher als bei hohen Temperaturen ($>100^{\circ}\text{C}$). Die Getriebeöltemperatur ist abhängig von der Außentemperatur und vom Fahrbetrieb, der das Öl durch auftretende Verlustleistungen erwärmt. Grundsätzlich muss der Antriebsstrang über einen großen Getriebeöltemperaturbereich von -30°C bis 140°C voll funktionsfähig sein (Fischer et al., 2016, S. 224). Über die Lebensdauer kommt es zu Alterungserscheinungen des Öls, die die hydraulischen Eigenschaften verändern.

Die LSS Blende besitzt einen konstanten Querschnitt. Sie dient dazu bei schwankenden Drücken einen konstanten Volumenstrom hinter der Öffnung zu erzeugen. Diese Funktion wird über das WFP zwischen LSS 5 und LSS Blende, der LSS Blende, sowie über das WFP zwischen LSS Blende und LSS 6 umgesetzt.

Das Regel-Magnet-Ventile, das neben der Hydraulikpumpe die hydraulische Aktuatorik darstellt, stellt den Kupplungsdruck ein. Über eine Strom-Druck-Kennlinie wird der anliegende elektrische Strom in einen Soll-Druck umgesetzt. Diese wird für verschiedene Temperaturen im Verbund mit der elektrohydraulischen Steuerereinheit messtechnisch ermittelt. Dann werden am End-of-Line-Prüfstand in der Produktion alle Ventile nochmal bei einer Temperatur von 60°C vermessen und abhängig vom Ergebnis wird ein Offset auf die Kennlinie addiert, um Fertigungstoleranzen auszugleichen. Diese Funktionen werden über die WFP zwischen LSS 6 und LSS Ventil und des WFP zwischen LSS Ventil und LSS 7 umgesetzt.

Der Drucksensor wird über ein WFP zur LSS 7 abgebildet und misst den Kupplungsdruck. Dabei können Abweichungen zwischen der Sensormessung und dem tatsächlichen Ist-Druck vorliegen. Betrachtet man die Wirkkette von der Pumpe bis zu Connector 6, so entstehen aufgrund der Systemeigenschaften wie der Trägheit des Öls, Totzeiten zwischen der Aktuierung der Pumpe und der Ventile und einer Volumenstrom- und Druckänderung am Kupplungskolben. Das Strömungsverhalten der Hydraulik ist grundsätzlich nichtlinear.

Die Eigenschaften des Restsystems der Kupplung haben starken Einfluss auf die Ansteuerung und werden über den Connector 6 abgebildet. Dieser führt die

Eigenschaften des tribologischen Systems der Lamellenkupplung sowie des Antriebs und Abtriebs auf die Ansteuerung zurück.

Da der Fokus auf der Ansteuerung liegt, werden nachfolgend die relevanten Einflussgrößen der Lamellenkupplung, wie Reibungszahl und Hysterese, sowie des Antriebsstrangs wie An- und Abtrieb der Kupplung, auf die Ansteuerung beschrieben. Für eine Detail-Analyse des Systems Lamellenkupplung mit dem C&C²-Ansatz wird auf den Forschungsbericht von Basiewicz (2020) verwiesen.

Das Eingangsdrehmoment und die Eingangsdrehzahl der Kupplung werden auf die Stahllamellen aufgebracht. Über das WFP Reibkontakt wird das Drehmoment von den Stahllamellen auf die Belagslamelle übertragen. Das Ausgangsdrehmoment der Kupplung wird anschließend auf den Restantriebsstrang übertragen. Fahrwiderstände, Schwingungen und Elastizität des Restantriebsstrangs haben hier Einfluss. Für den Fahrkomfort ist es entscheidend ein genaues Kupplungsmoment einstellen zu können, da Änderungen des Kupplungsmoments zu Änderungen des Abtriebsmoments und somit zu einem Fahrzeugruck führen. Speziell in dieser Hybridtopologie wird für den Motorwiederstart über die Trennkupplung K0 eine hohe Momentengenauigkeit benötigt. Die Momentenübertragung der Kupplung im Reibkontakt wird nach Gleichung 2.6 aus Kapitel 2.1.3 berechnet.

Die Anzahl der Lamellen z , die Reibfläche A und der mittlere Reibradius r_m sind geometrische Konstanten der Kupplung. Der Kupplungsdruck p kann mit hoher Genauigkeit mit einem Drucksensor bestimmt werden. Die Reibungszahl μ ist ein Funktional von Temperatur, Gleitgeschwindigkeit und Flächenpressung und ändert sich deshalb im Betrieb ständig (vgl. Kapitel 2.1.2). Die Reibungszahlen werden in Messungen aufwendig bestimmt und sind in Kennfeldern in der Software hinterlegt. Somit stellt die genaue Reibungszahlbestimmung über Lebenszeit eine Herausforderung dar. Der Einfluss von Fertigungstoleranzen und Verschleiß muss hierbei ebenfalls berücksichtigt werden. Dies wird mit Adaptionen über Lebensdauer umgesetzt. Adaptionsgrenzen und –geschwindigkeiten werden anhand aufwendiger Messungen an Prüfständen und in Fahrzeugen ermittelt (vgl. Kapitel 2.1.3).

Das Kupplungsübertragungsverhalten wird außerdem stark durch die Hysterese-Eigenschaften der Kupplung beeinflusst. Hierbei kann man zwischen einem hydraulischen und einem mechanischen Anteil der Hysterese unterscheiden. Insbesondere der mechanische Anteil kann kupplungsindividuell stark unterschiedlich ausfallen. Dieser ist unter anderem abhängig von der Steifigkeit des Lamellenpakets, die wiederum von der Geometrie der Lamellen, Eigenschaften der Rückstellfeder, Reibeigenschaften in den Kontaktflächen und den

Fertigungstoleranzen abhängt. Die Identifizierung dieser Modelle ist aufgrund des Reibverhaltens sehr komplex und aufwendig (vgl. Kapitel 2.1.2).

Über das Hydrauliksystem werden neben der im Detail dargestellten Kupplung zwei weitere Kupplungen aktuiert. Die weiteren Restsysteme der Verbraucher werden über ein WFP und den Connector 2 abgebildet. Diese benötigen einen Volumenstrom, um ihre Funktion zu erfüllen. Fordert eine Kupplung kurzfristig einen hohen Volumenstrom, so kann es im Hydrauliksystem zu einem Druckeinbruch kommen. Dieser führt wiederum zu einem Drehmomenteneinbruch einer belasteten Kupplung, was in einem Abfall der Fahrzeugbeschleunigung und somit einem Fahrzeugsruck resultiert. Diese Wechselwirkungen müssen zwingend in der Auslegung der Ansteuerung berücksichtigt werden.

Aufgrund des fraktalen Charakters des C&C²-Ansatz kann die Analyse noch auf tieferen Ebenen mit Fokus auf einzelne Wirkmechanismen oder über den zeitlichen Verlauf einer Funktion durchgeführt werden. Ergebnisse können dabei in die Modellierung eines Hydrauliksimulationsmodells einfließen.

Aus der Systemanalyse der Kupplungsansteuerung gehen folgende Herausforderungen hervor: stark temperaturabhängiges und nichtlineares Hydraulikverhalten, eine aufwendig und oft nur ungenau bestimmte Reibungszahl, der Einfluss von Temperaturen, Fertigungstoleranzen, Alterung und Verschleiß und aufwendige Adaptionen, die dem entgegenwirken sollen. Speziell in diesem hybriden Antriebsstrang erschweren elektrische Hydraulikpumpen aufgrund ihrer hohen Drucksensibilität sowie die hohe Momentengenauigkeit der K0 die Auslegung.

5.3 Zwischenfazit

In Kapitel 5 wurden die Herausforderungen in der Getriebefunktionsentwicklung von parallelen hybriden Fahrzeuggetrieben anhand leitfadengeführter Experteninterviews und Systemanalyse abgeleitet. Damit kann Forschungsfrage I (FF-I) beantwortet werden.

Forschungsfrage I (FF-I)

Welche Herausforderungen existieren in der Entwicklung von Steuerungs- und Regelungsalgorithmen von parallelen hybriden Fahrzeuggetrieben?

In den Experteninterviews mit insgesamt 14 Getriebeentwickelnden mit mehrjähriger Berufserfahrung wird die Auslegung der Getriebefunktionen am häufigsten als Herausforderung genannt. Die Ursachen hierfür sind ein hoher Neuheitsgrad des Hydrauliksystems durch einen hohen Anteil an Attribut- und Prinzipvariationen, Wechselwirkungen im Gesamtantriebsstrang und eine hohe Komplexität und Varianz der Software. Dies führt zu einem hohen zeitlichen Aufwand in der Auslegung. Am zweithäufigsten wird die Bildung von Modellen für die Steuergeräte-Software als Herausforderung genannt. Ursächlich dafür ist die hohe Komplexität aufgrund der hohen Anzahl an Betriebszuständen sowie des nichtlinearen Verhaltens der Hydraulik und der Tribologie der Reibelemente. Daraus folgen komplexe mathematische Modelle, für die viele Messungen an Prüfständen sowie teilweise Nacharbeit am Serienfahrzeug notwendig sind.

Bei der Analyse der Fahrfunktionen stellen der Motorwiederstart und das hybride Anfahren eine Herausforderung dar. Wie die Analyse der Kupplungsansteuerung anhand des C&C²-Ansatzes zeigt, fokussieren sich die Herausforderungen in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben auf die Kupplungsansteuerung aufgrund des stark temperaturabhängigen und nichtlinearen hydraulischen und tribologischen Systemverhaltens. Gleichzeitig bedarf es auch einer hohen Robustheit gegenüber Umwelteinflüssen wie Temperatur, Toleranzen, Alterung und Verschleiß. Diese wird durch aufwendige Adaptionen, die Parameter über Lebensdauer einstellen, umgesetzt. Speziell im hybriden Antriebsstrang erschweren elektrische Hydraulikpumpen aufgrund ihrer hohen Drucksensibilität sowie die hohe Momentengenauigkeit der K0 die Auslegung.

Nachfolgend sind alle identifizierten Herausforderungen zusammengefasst:

- Auslegung der Steuerung und Regelung (spez. Kupplungsansteuerung)
- Erstellung von Modellen der Getriebesoftware
- Messdatenauswertung
- Fahrbarkeit - Motorwiederstart
- Hybrides Anfahren

Im Abgleich mit der Forschungslücke wird auf die Auslegung der Steuerung und Regelung, sowie auf die in der Software verwendeten Modelle, speziell virtuelle Sensoren und Streckenmodelle, eingegangen.

6 Methoden des maschinellen Lernens im Kontext der Getriebefunktionsentwicklung

In diesem Kapitel werden ML-Methoden speziell für den Kontext der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben adaptiert, um den Anwendenden zielgerichtet und ganzheitlich zu unterstützen und die in Kapitel 5 identifizierten Herausforderungen anzugehen. Abbildung 6.1 gibt eine Übersicht über die Methoden und Inhalte dieses Kapitels. Dabei wird Forschungsfrage II beantwortet:

Forschungsfrage II (FF-II)

Wie müssen die Methoden des maschinellen Lernens gestaltet werden, um Entwickelnde bei der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen?

Wie im Stand der Forschung beschrieben, existieren bereits erste Ansätze ML-Methoden in Fahrzeuggetrieben einzusetzen. In Kapitel 6.1 werden diese Methodennutzung untersucht und Schwächen abgeleitet, an der die in diesem Kapitel beschriebenen Methoden ansetzen können. Kapitel 6.2 beschreibt dann die übergreifenden Zielsysteme für die zu entwickelnden Methoden.

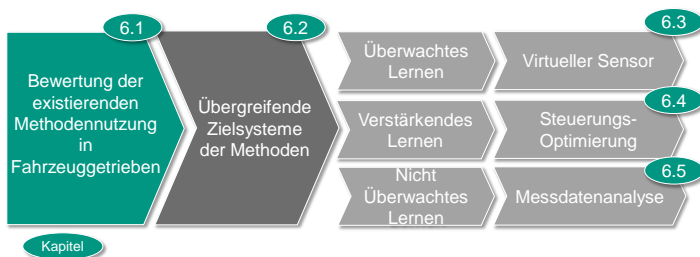


Abbildung 6.1: Vorgehen der Präskriptiven Studie

In den darauffolgenden Kapiteln werden die bestehenden Methoden und ML-Vorgehensmodelle an den Kontext der Produktentwicklung adaptiert und somit nutzbar gemacht. Kapitel 6.3 stellt die Methode Modellierung von virtuellen Sensoren mit Künstlichen Neuronalen Netzen vor, Kapitel 6.4 die Methode Steuerungs-Optimierung mit verstärkendem Lernen und Kapitel 6.5 die Methode Messdatenanalyse mit Clustering-Algorithmen. Jedes Unterkapitel schließt mit der Anwendung und Übertragbarkeit der jeweiligen Methode und beschreibt die unterliegenden Randbedingungen. Abschließend gibt Kapitel 6.6 ein Zwischenfazit.

6.1 Bewertung der existierenden Methodennutzung in Fahrzeuggetrieben

In Kapitel 2.4 wurde unter Vorbehalt der Vollständigkeit die Nutzung von ML-Methoden in Fahrzeuggetrieben erläutert. In diesem Abschnitt werden diese Ansätze für den Einsatz für die in Kapitel 5 identifizierten Herausforderungen untersucht und deren Schwächen aufgezeigt.

Überwachtes Lernen. Methoden des Überwachten Lernens z.B. KNN werden hauptsächlich zur Modellbildung von Simulationsmodellen genutzt, die in der Entwicklung von Fahrzeuggetrieben eingesetzt werden (Cabrera Cano, 2017; Cao et al., 2004; Ćirović et al., 2014), jedoch nicht für die Erstellung von virtuellen Sensoren. Nur Veith, Krüger, Albers und Behrendt (2018) gehen hier auf die Erstellung eines virtuellen Sensors für das Drehmoment ein. Allerdings fehlt hierbei eine konkrete Beschreibung der Hyperparameteroptimierung. Die Objektivierung des Fahrkomforts bei manuellen Getrieben, Automatikgetrieben und des hybriden Motorwiederstarts können nicht zur Modellierung von virtuellen Sensoren genutzt werden (Birkhold, 2013; Kawakami et al., 2020; Wolter et al., 2018).

Bei allen Methoden werden die Datenaufbereitung und die konkrete Auswahl der Eingangsgrößen, sowie das Erstellen von Trainings-, Validierungs- und Testdaten nicht näher beschrieben. Außerdem zeigen keine der Autoren eine systematische Beschreibung der Datengenerierung z.B. anhand von Fahrmanövern oder verschiedenen Validierungsumgebungen. Auch die Hyperparameteroptimierung wird nur unzureichend erläutert. Erklärbarkeitsansätze, die für Modelle in der Fahrzeugsoftware Transparenz schaffen können, werden nicht vorgestellt.

Verstärkendes Lernen. Der Einsatz von verstärkendem Lernen in Fahrzeuggetrieben für den Gangwechsel, den Anfahrvorgang oder der Unterdrückung von Kupplungsrupfen wird ausreichend beschrieben (Gaiselmann et al., 2022; Lampe et al., 2019; Sommer Obando, 2016). Alle Ansätze beschreiben,

wie im konkreten Anwendungsfall Aktions- und Zustandsraum, Trainingsalgorithmus und Belohnungsfunktion gewählt werden. Für alle Anwendungsfälle ist es das Ziel, den trainierten Agenten auf dem Getriebesteuergerät einzusetzen. Der Einsatz von verstärkendem Lernen als parameterfreies Optimierungsverfahren zur Unterstützung der Auslegung wurde nicht untersucht.

Nicht überwachtetes Lernen. Der Einsatz von Datenanalysemethoden mit Cluster-Algorithmen wird im Antriebsstrangbereich für das Ableiten von Fahrzyklen genutzt (Förster et al., 2020; Tewiele et al., 2018). Für Anwendungsfälle im Antriebsstrang für Fehlererkennung auf dem Prüfstand (Krätschmer et al., 2021) und Analyse von Geräuschphänomenen (Kistner et al., 2019) wird der Einsatz von Cluster-Methoden sehr allgemein beschrieben. Methodenauswahl und konkrete Umsetzung werden nicht aufgegriffen. Nur X. Zhang, Ebner, Arntz, Ramsauer und Küçükay (2021) beschreiben den konkreten Einsatz in der Getriebeentwicklung. Es werden Fahrzeugmessdaten analysiert, um Parameter zu identifizieren, die einen negativen Einfluss auf die Fahrbarkeit haben.

Insgesamt fehlt es bei allen drei Methoden an der Beschreibung eines konkreten Vorgehensmodells. Speziell beim überwachten Lernen können eine systematische Datengenerierung und Erklärbarkeitsansätze die Entwicklung unterstützen. Beim verstärkenden Lernen wurde die Nutzung für eine parameterfreie Optimierung noch nicht untersucht. Bei der Anwendung von Cluster-Algorithmen in der Fahrzeugentwicklung fehlt es insgesamt an Forschung an weiteren Anwendungsfällen. Zusätzlich existiert im Bereich der ML-Methoden eine Vielzahl an komplexen Algorithmen und Optimierungsverfahren. Um Entwickelnden einen einfachen Umgang mit diesen Methoden zu ermöglichen, sollten zusätzlich Erfahrungen im Umgang mit ML-Methoden und den teils komplexen Teilschritten berücksichtigt werden. Des Weiteren ist der Umfang an bisher untersuchten Anwendungsfällen speziell in hybriden Fahrzeuggetrieben gering, sodass die Mehrwerte der Methoden messbar an weiteren aktuellen Anwendungsfällen aufgezeigt werden sollten. Abschließend fehlt eine Bewertung der Nutzbarkeit und Anwendbarkeit der Methoden für die Entwickelnden.

6.2 Übergreifende Zielsysteme der Methoden

Aufbauend auf den Ergebnissen von Kapitel 5 werden nachfolgend der Abgleich zwischen Ist- und Soll-Zustand aufgezeigt, das übergreifende Zielsystem der einzelnen Methoden formuliert und messbare Erfolgskriterien für die Bewertung der Methoden abgeleitet (vgl. Abbildung 6.2).

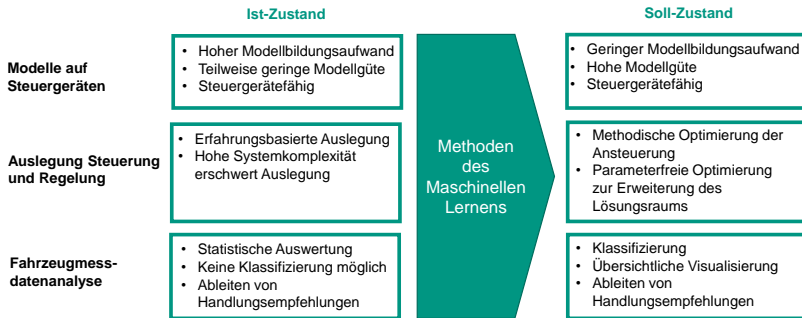


Abbildung 6.2: Ist- und Soll-Zustand für Modelle auf Steuergeräten, Auslegung Steuerung und Regelung und Messdatenanalyse

Modelle auf Steuergeräten. Zu diesen Modellen gehören virtuelle Sensoren und Streckenmodelle. Streckenmodelle rechnen im Rahmen einer Steuerung eine Führungsgröße in eine Stellgröße um. Streckenmodelle haben hohe Anforderungen an ihre Genauigkeit, um eine optimale Ansteuerung zu ermöglichen. Ungenauigkeiten können dabei zu Komforteinbußen führen.

Um eine Rückmeldung aus dem realen System zu erhalten, werden Sensoren verwendet. Diese messen Drehzahlen, Drücke, Temperaturen oder Positionen. Aus Kosten- und Bauraumgründen können nicht alle relevanten Größen über Sensoren gemessen werden. Deshalb werden virtuelle Sensoren benötigt. Diese berechnen aus verschiedenen Größen in der Software eine Sensorsersatzgröße (vgl. Kapitel 2.1.5). Sind z.B. Drücke oder Temperaturen genau bekannt, so kann von der Ansteuerung die jeweils optimale Steuergröße ausgegeben werden. Zum Beispiel wird beim Motorwiederstart über die Trennkupplung K0 ein Modell zur Momentenberechnung eingesetzt, um das an die K0 gestellte Moment zu bestimmen. Diese Momentenberechnung könnte über einen virtuellen Sensor umgesetzt werden.

In der Getriebeentwicklung wird der Ansatz der theoretischen Modellbildung in Form von Brightgrey-Box Modellen verfolgt. Die Modelle werden anhand mathematischer Gleichungen aufgebaut und die Modellparameter anschließend aufwendig am Prüfstand mit Experimenten ermittelt (vgl. Kapitel 2.1.5). Der hohe Aufwand der Parameterbestimmung hängt auch damit zusammen, dass hydraulische oder thermische Systeme ein stark nichtlineares Verhalten aufweisen. Messfehler und Störungen können die Modellgenauigkeit verschlechtern. Dabei steigt mit dem Detaillierungsgrad der Modellierungsaufwand. Da diese Modelle auch die Anforderungen der Steuergeräte hinsichtlich Echtzeitfähigkeit und geringem

Speicherbedarf erfüllen müssen, können nur einfache Modelle auf dem Steuergerät hinterlegt werden, was wiederum die Modellgenauigkeit einschränkt.

Deshalb wird für die Modellbildung von virtuellen Sensoren eine Methode benötigt, die den Modellbildungsaufwand reduziert, die Modellgüte erhöht und auf einem Getriebesteuergerät integrierbar ist. Da der Modellbildungsaufwand im Rahmen der Forschungsarbeit nicht vollständig bestimmt werden kann, wird die Steigerung der Modellgüte im Vergleich zu bisher genutzten Modellen sowie die Steuergerätefähigkeit als Erfolgskriterium herangezogen. Die Modellgüte beschreibt dabei die Abweichung zwischen Modell und einer realen Messung. Diese kann über verschiedene Metriken erfasst werden, die später bei der Anwendung der Methode auf eine Entwicklungsaufgabe ausgewählt werden. Die Steuergerätefähigkeit wird über den Speicherbedarf und somit der Parameteranzahl des Modells bewertet.

Auslegung der Steuerung und Regelung. Allgemein werden für die verschiedenen Regler Standard-Regler mit Proportional (P), Integral (I) und Differential (D) Anteil verwendet. Für die Bestimmung der Parameter existieren diverse Einstellregeln (vgl. Kapitel 2.1).

Kompensationsregler und modellprädiktive Regler kommen in Getrieben nicht zum Einsatz, da einerseits ein sehr genaues Streckenmodell für den Kompensationsregler bekannt sein muss und modellprädiktive Regelungen nicht echtzeitfähig auf dem Getriebesteuergerät ausgeführt werden können.

In der Getriebeentwicklung werden die Regler-Parameter häufig anhand von Erfahrungswerten eingestellt oder durch Berechnungen in Abhängigkeit von anderen Größen in der Software bestimmt (vgl. Kapitel 2.1.4). Die Auslegung dieser Parameter wird durch das bereits beschriebene komplexe Reib- und Hydraulikverhalten der Regelstrecke erschwert. Außerdem kommt es bei der Auslegung häufig zu Zielkonflikten, wie z.B. der klassische Zielkonflikt zwischen Komfort und Dynamik von Getriebebeschaltungen. Diese Herausforderungen gelten ebenfalls für den hybriden Antriebsstrang.

Ein Ansatz die Auslegung zu unterstützen, ist die Analyse von Messdaten. In der Entwicklung und Applikation der Fahrzeuggetriebe existiert eine große Menge an Simulations-, Prüfstands- und Fahrzeugmessdaten. Hauptsächlich Dauerlaufdaten werden statistisch ausgewertet und zur Validierung genutzt. Messungen werden größtenteils händisch oder mit einfachen statistischen Methoden ausgewertet.

Deshalb wird im Bereich der Auslegung von Ansteuerungsfunktionen eine Methode benötigt, dem Entwickelnden ein stringentes Vorgehen an die Hand gibt, den Umgang mit der Komplexität unterstützt und die Genauigkeit der Ansteuerung

erhöht. Als messbares Erfolgskriterium eignet sich hier die Genauigkeit der Ansteuerung. Auch hier wird der Begriff Genauigkeit später bei der Anwendung der Methode auf eine konkrete Entwicklungsaufgabe formuliert.

Messdatenanalyse. Die Auslegung kann ebenfalls durch eine Methode unterstützt werden, die Fahrzeugmessdaten analysiert. Bisher werden hierfür statistische Methoden verwendet. Diese haben jedoch den Nachteil, dass sie keine mehrdimensionalen Cluster in den Daten bestimmen können. Es ist hierbei ein großes Potential zum Erkenntnisgewinn durch die Identifizierung von Clustern vorhanden. Deshalb wird eine Methode benötigt, die durch Klassifizierung und eine kompakte Visualisierung der Auswertungsergebnisse dem Entwickelnden Informationen zugänglich macht, die bei der Synthese des Zielsystems und Objektsystems (vgl. Kapitel 2.2.1) unterstützen. Dadurch kann ein Beitrag zu den Validierungsmethoden geleistet werden. Als Erfolgskriterium soll hierbei das Potential für den Erkenntnisgewinn aufgezeigt werden.

6.3 Modellierung von virtuellen Sensoren mit Künstlichen Neuronalen Netzen

Aus Kosten- und Bauraumgründen werden reale Sensoren in Fahrzeuggetrieben durch virtuelle Sensoren ersetzt. Diese bilden dabei die Schnittstelle zwischen realen physikalischen Größen der Hardware und virtuellen Größen der Getriebesoftware.

Gleichzeitig können virtuelle Sensoren als Ausprägungen von virtuellen Koppelsystemen nach Pinner (2017) verstanden werden. Diese verbinden Teilsysteme einer Validierungsumgebung miteinander, ohne relevantes Systemverhalten zu ergänzen (vgl. Kapitel 2.2.3). Eine Ausprägung eines virtuellen Koppelsystems wandelt physikalische Eingangsgrößen in virtuelle Ausgangsgrößen um.

Das Vorgehen zur Modellierung von virtuellen Sensoren mit Künstlichen Neuronalen Netzen (KNN) im Kontext der Ansteuerung eines parallelen hybriden Fahrzeuggetriebes ist in Abbildung 6.3 dargestellt. Es baut dabei auf dem CRISP-ML (Q) Modell aus Kapitel 2.3.1 auf. Der Fokus in der Methodenbeschreibung liegt auf der Datengenerierung und Definition der Modelleingangsgrößen. Bei den automatisierten Prozessschritten wird vor allem darauf eingegangen, welche Parameter relevant sind und wie diese bestimmt werden. Der Schritt der Erklärbarkeitsansätze unterstützt die Validierung des Modells im Umgang mit KNN als Black-Box-Modelle. Ein großer Vorteil von ML-Methoden liegt in der Automatisierung. Diese sollte für die in Abbildung 6.3 markierten Schritte genutzt werden, sodass sich die Entwicklungsarbeit auf die manuellen Schritte fokussieren kann. Außerdem sind die automatisierbaren Schritte sehr gut skalierbar. Trainingsdatensätze können beliebig vergrößert werden. Die Programmiersprache Python bietet den großen Vorteil, dass für sehr viele ML-Methoden Bibliotheken existieren, wodurch eine große Funktionalität vorhanden ist und gleichzeitig der Implementierungsaufwand geringgehalten wird.

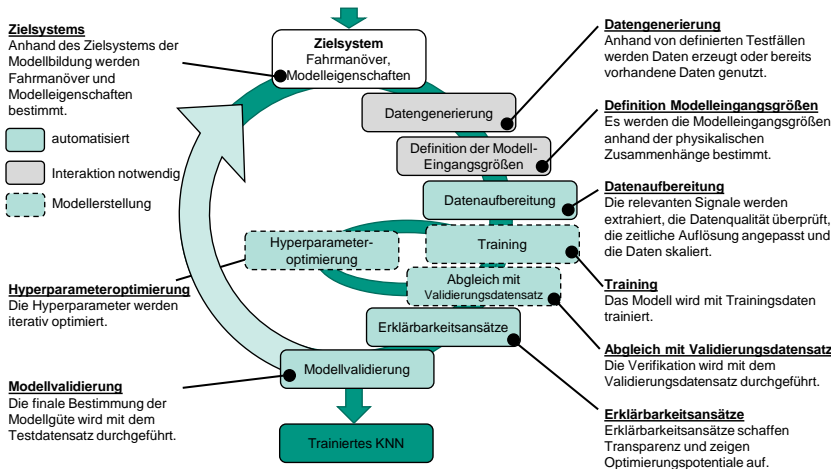


Abbildung 6.3: Vorgehen Modellierung von virtuellen Sensoren mit KNN

6.3.1 Zielsystem

Im ersten Schritt der Methode wird das Zielsystem der Modellbildung definiert. Dazu gehören die Definition von abzubildenden Fahrmanövern und Modelleigenschaften. Die Definition der Fahrmanöver beinhaltet auch:

- Umgebungseinflüsse: z.B. Außentemperatur, Höhe über NN, Luftfeuchte
- Alterung, Verschleiß, Fertigungstoleranzen

Zu den Modelleigenschaften zählen:

- Parameteranzahl (Netzgröße) aufgrund gegebener Speichieranforderungen,
- Kriterien zur Bewertung der Modellgüte,
- und zeitliche Auflösung der Vorhersage.

Die Datengrundlage zur Modellerstellung bildet die Basis für das Modellverhalten. Für Fahrmanöver, die in den Trainingsdaten enthalten sind, kann das Modell genaue Aussagen liefern (Interpolation). Für Fahrmanöver mit hoher Ähnlichkeit ist eine Modellübertragbarkeit möglich (Extrapolation). Die Erzeugung der Daten für das Modelltraining wird systematisch anhand von Fahrmanövern in definierten

Testumgebungen durchgeführt. Testumgebungen können dabei Simulationen (rein virtuell), Getriebe- oder Antriebsstrangprüfstände (gemischt physisch-virtuell) oder Fahrversuche (rein physisch) sein. Fahrmanöver können aus definierten Fahrsituationen, Getriebefunktionen (z.B. Gangwechsel) oder aus Fahrzyklen abgeleitet werden (vgl. Abbildung 6.4).

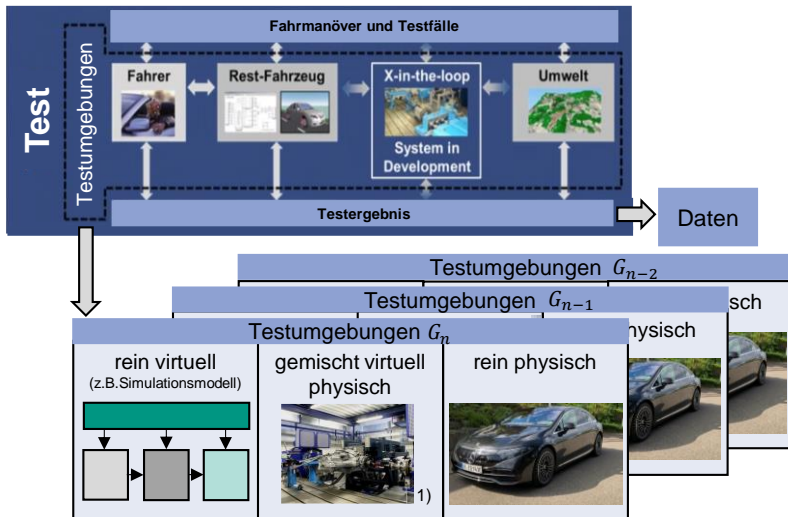


Abbildung 6.4: Datengenerierung anhand von Fahrmanövern und Testumgebungen, 1) nach (Ott)

Um möglichst früh im Produktentstehungsprozess Daten zu generieren, sind rein virtuelle oder gemischt physisch-virtuelle Validierungsumgebungen zu bevorzugen. Daten aus dem Fahrversuch bieten den Vorteil, dass alle Teilsysteme einen sehr hohen Reifegrad besitzen und somit die Daten dem späteren Serienbetrieb am nächsten kommen. Sind bereits Daten vorhanden, muss geprüft werden, ob diese anhand der relevanten Fahrmanöver und korrekt aufgebauten Testumgebungen erzeugt worden sind.

Im IPEK-XiL-Ansatz werden über das Restsystem „Umgebung“ äußere Einflüsse wie die Umgebungstemperatur berücksichtigt. In der Getriebeentwicklung ist diese von großer Bedeutung, da sie das hydraulische Verhalten des Systems stark beeinflusst.

Sind für die Funktionalität des virtuellen Sensors Alterungs- und Verschleißeffekte sowie Fertigungstoleranzen entscheidend, ist dies ebenfalls zu berücksichtigen. Dies kann über die Verwendung von Messdaten aus Validierungsumgebungen basierend auf Teilsystemen der gleichen Systemgeneration umgesetzt werden. Existieren vorangehende Systemgenerationen mit geringem Prinzip- und Ausprägungsvariationsanteil, so können deren Entwicklungsdaten ebenfalls genutzt werden. Deren Verwendung muss davor kritisch geprüft werden.

Abschließend werden konkrete Modelleigenschaften definiert, wie die geforderte zeitliche Auflösung der Modellvorhersage sowie die Parameteranzahl des Netzwerks aufgrund der Speicheranforderungen auf dem Getriebesteuergerät. Dazu gehören auch Kriterien zur Bewertung der Modellgüte, um das trainierte Modell hinsichtlich seiner Leistungsfähigkeit zu beurteilen. Die Modellgüte wird über die Abweichung zwischen Modellvorhersage und realer Messung über einen Testdatensatz bestimmt. Es existiert eine Vielzahl an statistischen Metriken, um die Modellgüte zu bewerten. Dazu gehören unter anderem der maximale absolute Betragsfehler, der mittlere quadratische Fehler oder das Bestimmtheitsmaß (vgl. Tabelle 2.1). Die Wahl der Metrik ist abhängig von der jeweiligen Zielsetzung. Mean-Squared-Error (MSE) und Root-Mean-Squared-Error (RMSE) bestimmen die Abweichung quadratisch, wodurch große Abweichungen stärker gewichtet werden.

Das beschriebene Zielsystem wird iterativ anhand der Validierungsergebnisse des Modells angepasst und erweitert.

6.3.2 Datengenerierung

Die Daten für die Modellerstellung werden anhand der in 6.3.1 beschriebenen Fahrmanöver und Validierungsumgebungen generiert.

Die Datengenerierung kann ein sehr ressourcen-intensiver Vorgang sein. Vor allem rein-physische oder gemischt physisch-virtuelle Umgebungen sind kostenintensiv. Umgebungstemperaturen von bis zu -30 °C müssen aufwendig in Kältekammern erzeugt werden. In der Praxis kann der Ressourcen-Aufwand reduziert werden, indem Daten von bereits durchlaufenen Testfällen wie z.B. Dauerlaufdaten oder Kältetests genutzt werden. Deren Verwendung muss anhand der definierten Fahrmanöver und Umgebungsbedingungen vorab geprüft werden. Auch die gleichzeitige Verwendung von Daten aus unterschiedlichen Validierungsumgebungen kann den Ressourcen-Aufwand reduzieren. In Prüfstands- oder Fahrversuchen unterrepräsentierte Fahrmanöver können durch kostengünstige Simulationen ergänzt werden.

Bei der Erstellung von Trainingsdaten ist zu beachten, dass die Fahrmanöver homogen gewichtet werden, um eine Über- oder Unteranpassung zu vermeiden.

Die Robustheit des Modells kann gesteigert werden, in dem Fahrmanöver mehrfach durchfahren und hinterlegt und Messungen aus verschiedenen Fahrzeugen genutzt werden.

Bei der Datengenerierung sind die Eigenschaften der Messsensorik zur Messung der Modellzielgröße wie Messgenauigkeit, Messbereich und Messfrequenz zu berücksichtigen. Einschränkungen müssen bekannt sein und in der Nutzung des Modells einbezogen werden.

Der Gesamtdatensatz wird in einen Trainings-, Validierungs- und Testdatensatz aufgeteilt. Der Testdatensatz dient dabei als abschließende Validierung des Modells, um dessen Modellgüte zu bewerten. Dieser Datensatz geht nicht in das Training oder die Hyperparameteroptimierung ein und ist somit für das Modell unbekannt. Der Testdatensatz kann identische Fahrmanöver zum Trainingsdatensatz enthalten, es müssen dabei aber jeweils unterschiedliche Messungen vorliegen. Fahrzyklen können ebenfalls als Testdaten genutzt werden. Nach Literaturwerten wird das Verhältnis von Trainings-, Validierungs- und Testdatensatz im Bereich 70 % Training, 15 % Validierung und 15 % Test gewählt.

6.3.3 Definition der Modelleingangsgrößen

Nach der Definition des Zielsystems und der Datengenerierung folgt die Definition der Modelleingangsgrößen.

Als Modelleingangsgrößen sind diejenigen CAN-Bus-Signale zu wählen, die den größten Einfluss auf die Modellzielgröße haben. Gleichzeitig wird dadurch auch deren Anzahl geringgehalten, was die spätere Modellgröße minimal hält. Am einfachsten ist dies durch Verwendung der physikalischen Einflussgrößen umzusetzen. Dazu werden analog zur theoretischen Modellbildung physikalische Gleichungen aufgestellt, um aus vorhandenen Eingangsgrößen die Zielausgangsgrößen des Modells zu berechnen. Sind bereits physikalisch modellierte virtuelle Sensoren vorhanden, so können deren Eingangsgrößen als erster Ausgangspunkt für das datenbasierte Modell herangezogen werden.

Alternativ können Filter- oder Wrapper-Verfahren für die Wahl der Eingangsgrößen genutzt werden. Filter-Verfahren berechnen anhand der Korrelation zur Modellzielgröße die Relevanz der Eingangsgröße. Wrapper-Verfahren testen eine

große Anzahl an Kombinationen von Eingangsgrößen, um die zur Ermittlung der Modellzielgröße optimalen Eingangsgrößen zu identifizieren.

Die Eingangsgrößen sind idealerweise so zu wählen, dass sie als CAN-Bus-Signale im späteren Serienfahrzeug vorhanden sind. Dabei können die Eingangsgrößen Sensormessgrößen innerhalb des Antriebsstrangs sein, oder aus diesen in anderen Teilen der Getriebesoftware berechnet werden. In Automatikgetrieben gehören dazu Drehzahlen, Hydraulikdrücke, Positionen und Temperaturen. Eine Korrelationsanalyse kann unterstützen, linear abhängige und somit überflüssige Eingangsgrößen zu erkennen. Systemwissen und Systemverständnis des Anwenders sind zentrale Faktoren, um die relevanten Eingangsgrößen zu bestimmen.

In dieser Arbeit wird der Ansatz der physikalischen Herleitung genutzt, da dieser den Vorteil besitzt, dass alle wichtigen Eingangsgrößen mit sehr großer Wahrscheinlichkeit berücksichtigt werden und die Rechenkosten zur Bestimmung dieser Größen gering sind. Eine automatisierte Bestimmung der Eingangsgrößen, aus denen anschließend manuell gewählt werden kann, ist grundsätzlich denkbar. In der Anwenderstudie in Kapitel 8.4 wird untersucht, ob Anwendende in diesem Schritt durch eine Automatisierung unterstützt werden wollen.

6.3.4 Datenaufbereitung

Die Datenaufbereitung als vierter Schritt der Methode (vgl. Abbildung 6.3) unterteilt sich in fünf Abschnitte:

- Extrahieren der relevanten Signale aus den Messdaten
- Zusammenfügen zu einem Gesamtdatensatz
- Prüfen des Datensatzes auf fehlende Werte und Ausreißer
- Anpassen der zeitlichen Auflösung
- Skalieren der Daten

In einem ersten Schritt werden die identifizierten Eingangsgrößen als Signale aus den Fahrzeugmessdaten extrahiert. Ein gängiges Dateiformat für CAN-Bus-Daten in der Getriebeentwicklung ist das MF4-Dateiformat. Zur Extraktion der Daten wird die Verwendung der Python-Bibliothek `asammdf` empfohlen, die das MF4-Format nutzerfreundlich in pythontypische Strukturen wie Dataframes überführen kann (Hrisca, 2024).

Anschließend werden die einzelnen Messungen zu einem Gesamtdatensatz zusammengefügt. Um eine hohe Datenqualität sicherzustellen, werden die Daten auf fehlende Daten (Not a Number-Einträge) und Ausreißer durchsucht, die

außerhalb des plausiblen Wertebereichs liegen. Methoden, Ausreißer zu identifizieren, sind das Setzen von Maximal- oder Minimalwerten oder die Relation der Datenpunkte zum Mittelwert. Werden im Datensatz fehlende Einträge oder Ausreißer identifiziert, können diese aus dem Datensatz entfernt werden.

Je nach Anwendungsfall wird die zeitliche Auflösung der Daten angepasst und/oder Rauschen aus den Messsignalen gefiltert. Die zeitliche Auflösung kann angepasst werden, um Trainingszeiten zu reduzieren. Eine gleichbleibende Modellgüte steht im Konflikt dazu. Zur Filterung sind aus der Literatur viele Methoden bekannt, wie zeitdiskrete oder digitale Filter (Mildenberger, 1992).

Da die Modelleingangsgrößen unterschiedliche Größenordnungen besitzen, werden die Daten skaliert. Die zwei Hauptverfahren sind die Standardisierung und die Normalisierung. Bei der Standardisierung wird der Mittelwert einer Größe auf 0 gesetzt, die Standardabweichung auf 1. Bei der Normalisierung werden die Werte zwischen 0 und 1 skaliert. Als Skalierungs-Verfahren wird das Normalisierungs-Verfahren der Quantil-Normalisierung genutzt (vgl. Kapitel 2.3.1). Der große Vorteil dieses Verfahrens liegt darin, dass Ausreißer die Normalisierung nicht beeinflussen und somit die Messdaten im skalierten Raum homogener verteilt sind. Ein Beispiel für einen solchen Normalisierer ist der RobustScaler der Python-Bibliothek Scikit-learn (Pedregosa et al., 2011).

Der Vorteil der Verwendung von CAN-Bus-Signalen aus dem Getriebesteuergerät besteht darin, dass diese Signale bereits ein identisches Datenformat sowie eine hohe Datenqualität aufweisen. Das Prüfen der Daten auf fehlerhafte Einträge oder Ausreißer findet unabhängig davon statt.

Zum Vergleich: Bei der Nutzung von ML im Bereich Business Intelligence liegen Daten aus unterschiedlichen Geschäftsbereichen und Programmen vor, was zu vielen verschiedenen Datenformaten führt. Durch das Zusammenführen aus verschiedenen Systemen ist die Fehleranfälligkeit hoch. Die hohe Datenqualität von CAN-Bus-Signalen vereinfacht hier das Vorgehen.

6.3.5 Modellerstellung

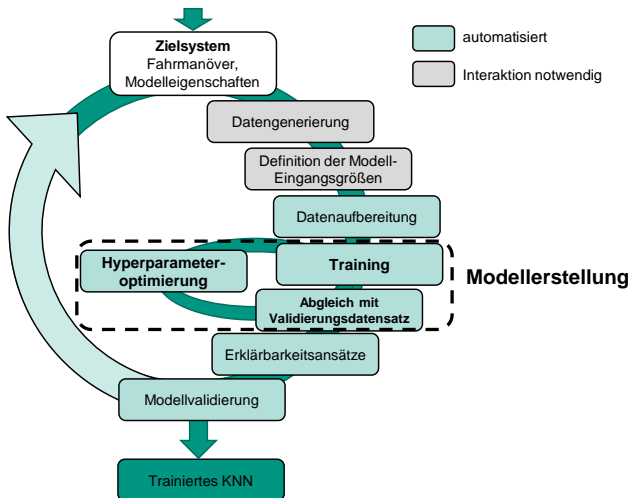


Abbildung 6.5: Vorgehen Modellierung von virtuellen Sensoren mit KNN, vereinfachte Darstellung, Modellerstellung hervorgehoben

Die Modellerstellung setzt sich aus den drei Schritten Training, Abgleich mit Validierungsdatensatz und Hyperparameteroptimierung zusammen (vgl. Abbildung 6.5).

Training. Der Trainingsvorgang eines KNN wurde bereits im Stand der Forschung beschrieben (vgl. Kapitel 2.3.1). Ein klassisches Verfahren hierfür ist ein Back-Propagation Ansatz, bei dem über ein Gradienten-Abstiegsverfahren die Gewichte und Aktivierungswerte angepasst werden, um den Modellfehler zu minimieren. Der regelmäßige Abgleich des Modells während des Trainings wird mit dem Validierungsdatensatz durchgeführt.

Um das Training zu beenden, wird ein Abbruchkriterium definiert. Möglich ist dabei der Abbruch nach einer definierten Anzahl an Trainingsepochen (3), der Abbruch bei Unterschreiten eines definierten Modellfehlers (2) oder der Abbruch nach einer definierten Anzahl von nachfolgenden Trainingsepochen ohne Verbesserung des Validierungsfehlers (1). Der Nachteil aller Verfahren liegt darin, dass sie nicht sicherstellen, dass sie das Training an der Epoche des globalen Minimums des Validierungsfehlers beenden. (vgl. Abbildung 6.6). Um ein Abbruchkriterium zu bestimmen, wird ein Training mit hoher Epochenzahl gestartet, um das Konvergenzverhalten zu untersuchen.

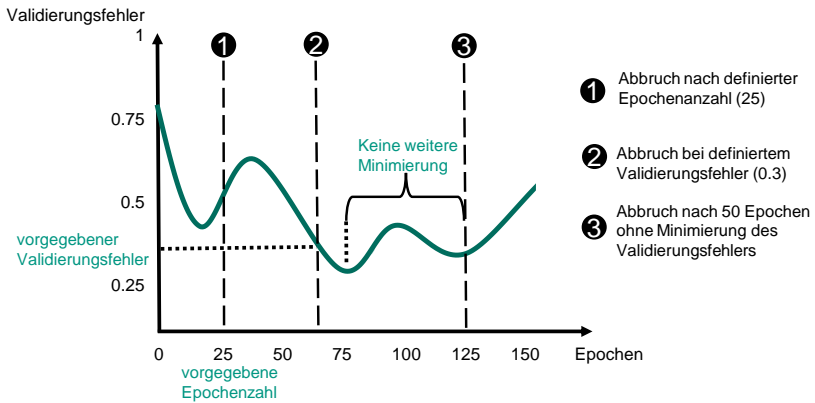


Abbildung 6.6: Darstellung drei verschiedener Abbruchkriterien des Modelltrainings anhand des Validierungsfehlers

Abgleich mit Validierungsdatensatz. In diesem Schritt macht das KNN eine Vorhersage auf den Validierungsdatensatz. Damit wird die Generalisierungsfähigkeit des Netzes untersucht und eine Überanpassung an den Trainingsdatensatz verhindert. Dieser Schritt wird iterativ durchgeführt, um die Hyperparameter des KNN zu optimieren. Während des Trainings und der Hyperparameteroptimierung erfolgt dieser Abgleich nach jeder Trainingsepisode.

Hyperparameteroptimierung. Die Leistungsfähigkeit des KNN wird von den Netzwerk- und Trainingsparametern beeinflusst. Die Netzwerkarchitektur wird durch die Anzahl an Schichten und Neuronen pro Schicht festgelegt. Eine größere Anzahl an Schichten und Neuronen erhöht die Anzahl an Parametern des KNN, wodurch sich theoretisch jede Funktionalität abbilden lässt (Russell & Norvig, 2012, S. 850). Jedoch muss ein Optimum der Netzgröße bestimmt werden, da sehr große Netze zur Überanpassung neigen und große Datenmengen benötigen, um alle Parameter zu bestimmen.

Rekurrente Netzwerke nutzen sequenzielle Daten einer zeitlichen oder räumlichen Abfolge und eignen sich deshalb für die Abbildung von virtuellen Sensoren. Neben rekurrenten neuronalen Netzen haben sich zur Abbildung von Zeitreihen in den letzten Jahren auch zeitlich faltende neuronale Netze als geeignet gezeigt (vgl. Kapitel 2.3.1). Im Rahmen dieser Arbeit werden aufgrund ihrer breiten Anwendung rekurrente Netzwerke betrachtet.

Da virtuelle Sensoren einen möglichst geringen Speicherbedarf auf Getriebesteuergeräte einnehmen sollen, sind kleine Netze mit ein bis drei Schichten

und jeweils zwei bis 64 Knoten zu untersuchen. Die Anzahl an Knoten pro Schicht wird über Zweierpotenzen parametrisiert.

Bei rekurrenten neuronalen Netzen ist ein weiterer Optimierungsparameter die Sequenzlänge. Dieser beschreibt die Anzahl der zeitlich zurückliegenden Zeitschritte der Eingangsgrößen, die für das Training verwendet werden. Dies sollte in Abhängigkeit der zeitlichen Auflösung so gewählt werden, dass ein dynamischer Vorgang des Systems abgebildet kann. Liegt zum Beispiel die Dauer eines Druckanstiegs bei einem Drucksensor einer Kupplung bei 100ms, sollte das Modell diesen Zeitbereich abdecken.

Ein weiterer Parameter ist die Stapelgröße, die die Länge des Datensatzes beschreibt, mit der ein Lernvorgang durchgeführt wird. Das Zerlegen des Gesamtdatensatzes in mehrere Teile minimiert den benötigten Speicher beim Trainieren und beschleunigt das Training. Eine zu kleine Stapelgröße kann jedoch zu einer verrauschten Schätzung des Fehlergradienten des Lernvorgangs führen, was wiederum in einem ungenauen Modell resultiert. Die Stapelgröße wird ebenfalls über Zweierpotenzen parametrisiert (z.B. 4, 8, 16, 32, 64).

Zur Hyperparameteroptimierung existieren verschiedene Verfahren: Grid-Search, Random-Search, Bayes'sche Optimierung und populationsbasierte Ansätze (vgl. 2.3.1).

Bei kleinen Modellen mit einer geringen Anzahl an Hyperparametern wird eine GridSearch empfohlen. Dadurch wird die Identifizierung des globalen Optimums im Parameterraum sichergestellt. Oft liegt aber eine große Anzahl an Parametern und damit auch Parameterkombinationen vor. Die Verwendung einer RandomSearch ist dabei gebräuchlich. Da dieser den Parameterraum allerdings nur zufällig absucht, wird bei großen Parameterräumen (>5000 Parameterkombinationen) die Verwendung eines Bayesischen Optimierungsverfahrens empfohlen. Für große Modelle mit einem großen Hyperparameterraum eignen sich populationsbasierte Ansätze wie z.B. genetische Algorithmen. Allgemein können Stopp-Kriterien genutzt werden, um Parameterkombinationen mit einem großen Fehler schon frühzeitig abubrechen und damit die Optimierungsdauer zu reduzieren. Ab welcher Parameterkombinationsanzahl ein großer Parameterraum vorliegt, muss für den konkreten Anwendungsfall in Kombination mit der vorhandenen Rechenleistung bestimmt werden.

Die Python Bibliothek ray-tune (Liaw et al., 2018) ermöglicht eine einfache Implementierung der Optimierung, Tensorboard (Abadi et al., 2016) eine automatisierte Visualisierung der Ergebnisse.

Einige Parameter können aufgrund der guten Ergebnisse aus der Literatur für viele Anwendungsfälle übernommen werden. Dazu gehören der Neuronentyp LSTM und der Optimierungsalgorithmus der Netzwerkparameter Adam (vgl. Kapitel 2.3.1).

Die Durchführung der Schritte Training, Validierung und Hyperparameteroptimierung erfolgen wie in Abbildung 6.3 dargestellt iterativ.

6.3.6 Erklärbarkeitsansätze

Als vorletzter Schritte der Methode werden Erklärbarkeitsansätze genutzt, um die Modelloptimierung zu unterstützen. Da KNN Black-Box-Modelle darstellen, ist deren Verhalten nicht direkt erklärbar oder für unbekannte Modellbereiche bestimmbar. Neben den Modellparametern können deshalb auch Erklärbarkeitsansätze zur Modelloptimierung genutzt werden. Damit kann bspw. die Relevanz der Eingangsgrößen bewertet werden. Das Gebiet der Explainable AI umfasst dabei eine Vielzahl von Methoden. In diesem Abschnitt werden zwei Methoden beschrieben. Im Vorgehensmodell aus dem Stand der Forschung (vgl. Abbildung 2.20) ist dieser Schritt nicht enthalten. Er wird in dieser Arbeit hinzugefügt, um die Validierung zu unterstützen.

Sensitivitätsanalyse. Bei der Sensitivitätsanalyse wird eine Eingangsgröße des Testdatensatzes verändert, während alle weiteren Eingangsgrößen konstant gehalten werden. Eine Eingangsgröße kann z.B. um einen konstanten Prozentsatz vergrößert oder verkleinert werden. Dadurch wird der Einfluss dieser Eingangsgröße auf die Modellzielgröße direkt ersichtlich. Die Veränderung der Modellzielgröße kann dann mit dem erwarteten Modellverhalten oder dem Verhalten eines physikalischen Modells abgeglichen werden. So gewinnt man eine Aussage über die Generalisierbarkeit des Modells. Außerdem kann so der Einfluss der Eingangsgrößen ermittelt werden. Hat die Variation einer Eingangsgröße nur sehr geringen Einfluss auf die Modellgröße, so kann diese Eingangsgröße entfallen.

Local Interpretable Model-Agnostic Explanations (LIME). Der bereits in Kapitel 2.3.1 beschriebene LIME-Ansatz schafft ein lokales Ersatzmodell des KNN zu einem festgelegten Zeitpunkt, um den Einfluss der Modelleingangsgrößen auf die Vorhersage erklärbar zu machen. Dieses Wissen kann genutzt werden, um die Eingangsgrößen anzupassen, oder um eine nachvollziehbare Erklärbarkeit des Modells für Entwickelnde zu ermöglichen.

6.3.7 Modellvalidierung

Nach der Hyperparameter- und Modelloptimierung mit Erklärbarkeitsansätzen werden mehrere Modelle trainiert und diese auf den Testdatensatz angewendet. Es werden zehn Modelle trainiert, um eine statistische fundierte Aussage über die Modellperformanz treffen zu können. Zur Bewertung wird der Mittelwert der Bewertungsgröße über alle Modelle gebildet.

Erfüllt das Modell die im Zielsystem festgelegten Anforderungen nicht, so beginnt eine Iteration des Vorgehens (vgl. Abbildung 6.3). Auf Grundlage der Validierungsergebnisse werden das Zielsystem, die Modelleingangsgrößen und die Datensätze iterativ angepasst und erweitert und die nächsten Methodenschritte wiederum darauf aufbauend durchgeführt. Werden die Anforderungen des Zielsystems erfüllt, so ist das Vorgehen abgeschlossen.

6.3.8 Anwendung und Übertragbarkeit

Die Methode Modellierung von virtuellen Sensoren mit KNN lässt sich in den Bereich der experimentellen Modellbildung einordnen (Isermann, 2007, 47–51). Die experimentelle Modellbildung zeigt in zahlreichen Gebieten die Eignung zur Beschreibung von nichtlinearen Zusammenhängen, weshalb mit dieser Methode komplexe und mathematisch aufwendig zu beschreibende Systeme abgebildet werden können (Russell & Norvig, 2012, S. 850).

Anhand des IPEK-XiL-Ansatzes werden Fahrmanöver und Testumgebungen systematisch definiert, um Daten zur Modellerstellung zu generieren. Die Kombination von Daten aus virtuellen, gemischt physisch-virtuellen und rein-physischen Testumgebungen ist möglich. Dadurch können über verschiedene durchgängige Validierungsumgebungen Daten genutzt werden.

Der Einsatz von rechenstarken Maschinen oder Cloud-Computing kann die Trainingsdauer reduzieren.

Grundsätzlich ist die Anwendung der Methode auf eine Vielzahl an Entwicklungsbeispielen in der Getriebesoftware denkbar. Mögliche Anwendungsfälle sind die Reibungszahlbestimmung, Druckberechnung oder Temperaturmodelle. Im Kontext der SGE - Systemgenerationsentwicklung nach Albers können Teile des Objektsystems z.B. Netzwerkarchitekturen oder Modelleingangsgrößen für nachfolgende Entwicklungsgenerationen des virtuellen Sensors übernommen werden.

6.4 Steuerungs-Optimierung mit verstärkendem Lernen

Verstärkendes Lernen lässt sich in drei Formen in der Getriebefunktionsentwicklung einsetzen:

- Optimierung von Steuervorgängen.
- Ein trainierter Agent übernimmt eine Steuer- oder Regelaufgabe auf dem Getriebesteuergerät.
- Ein trainierter Agent übernimmt eine Steuer- oder Regelaufgabe auf dem Getriebesteuergerät, sammelt im Fahrbetrieb weitere Daten und lernt für ein spezifisches Fahrzeug hinzu.

Die Methodenanwendung widmet sich dem Einsatz von verstärkendem Lernen als ein parameterfreies Optimierungsverfahren zur Unterstützung der Auslegung von Ansteuerungen. Ziel ist es, eine Ansteuerung zu erhalten, die Entwickelnde als eine erste Parametrierung nutzen können. Darauf aufbauend wird die abgestimmte Anpassung auf den Antriebsstrang durchgeführt. Dadurch entfällt ein Großteil der manuellen Optimierungen. Nach Schröter (2013) lassen sich Optimierungsverfahren über das erweiterte IPEK-XiL-Framework durchgängig in den Produktentwicklungsprozess integrieren. Das System „Optimierer“ weist dabei Schnittstellen zu den Systemen „Fahrer“, „Fahrzeug“ und „Umwelt“ auf (vgl. Abbildung 6.7).

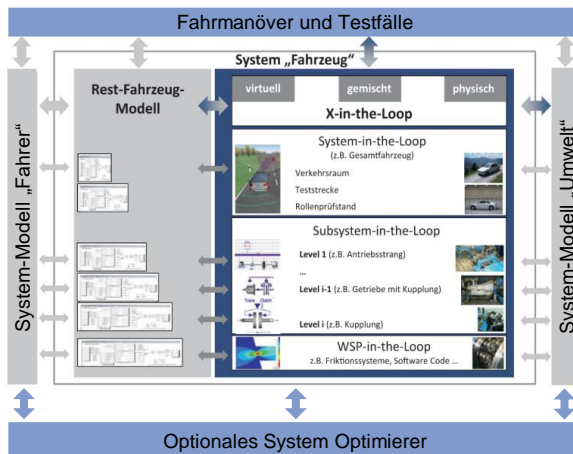


Abbildung 6.7: Erweitertes IPEK-XiL-Framework mit dem optionalen System Optimierer (Eigene Darstellung nach Schröter, 2013).

Dieses Kapitel thematisiert die Anwendung der Methode. Bezüglich der theoretischen Details des verstärkenden Lernens wird auf den Stand der Forschung verwiesen (vgl. Kapitel 2.3.3). Das Vorgehen zur Anwendung der Methode Steuerungs-Optimierung mit verstärkendem Lernen im Kontext der Ansteuerung eines parallelen hybriden Fahrzeuggetriebes ist in Abbildung 6.8 skizziert. Dabei können die Schritte Training und Validierung in einer rein virtuellen Validierungsumgebung automatisiert durchgeführt werden.

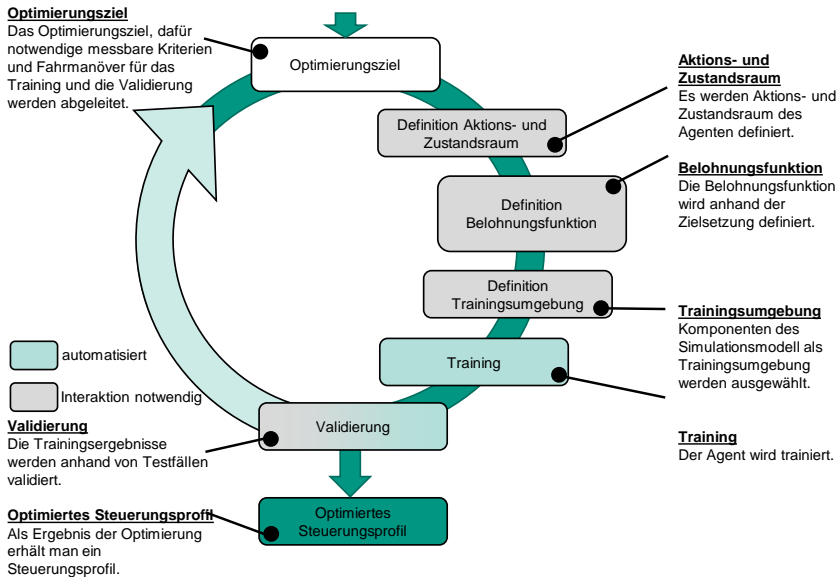


Abbildung 6.8: Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen

6.4.1 Optimierungsziel

Im ersten Schritt der Methode wird das Optimierungsziel bestimmt. Aus dem Optimierungsziel werden dann messbare Kriterien zum Nachweis der Zielerreichung abgeleitet. Außerdem werden Fahrmanöver für das Training und die Validierung definiert. Zum Beispiel kann das Optimierungsziel für einen Gangwechsel ein hoher Schaltkomfort bei gegebener Schaltzeit sein. Als messbare Kriterien könnten hierfür der Fahrzeugruck zur Bewertung des Schaltkomforts sowie die Dauer der Schaltung und als Fahrmanöver eine 2-3 Schaltung herangezogen

werden. Für die Fahrmanöver werden entsprechende Randbedingungen definiert, wie Start- und Endzeitpunkt oder Fahrzeugzustand (z.B. Geschwindigkeit, Lenkwinkel, Steigung).

6.4.2 Definition Aktions- und Zustandsraum

Bei der Anwendung auf ein Steuerungsproblem beschreibt der Aktionsraum die Steuergrößen. Im Kontext eines parallelen hybriden Fahrzeuggetriebes können das Ventilströme der Kupplungen, Ansteuerung der Hydraulikpumpen der Kupplungen (Strom, Drehzahl), des Gangstellers oder des Kühllöls sein. Je nach Anwendungsfall ist der Aktionsraum eindimensional (eine Steuergröße) oder mehrdimensional (mehr als eine Steuergröße). Um ein sprunghaftes Verhalten der Aktion zu verhindern, wird die Nutzung des Gradienten der relevanten Steuergröße als Aktion empfohlen. Dadurch ist ein stetiger Signalverlauf sichergestellt.

Der Zustandsraum beschreibt den Zustand der Umgebung aufgrund dessen der Agent eine Aktion ausführt. Hierbei werden Größen gewählt, die das physikalische Verhalten des Systems möglichst genau beschreiben. Im Kontext eines parallelen hybriden Fahrzeuggetriebes sind dies Sensorgrößen wie Hydraulikdrücke, Drehzahlen oder Temperaturen. Auch in der Software berechnete Größen wie Drehmomente oder Kupplungspositionen können hierbei verwendet werden.

6.4.3 Definition Belohnungsfunktion

Die Belohnungsfunktion bildet das zentrale Element des verstärkenden Lernens und gibt an, ob eine Aktion einen positiven oder negativen Einfluss auf die Zielgrößen der Steueraufgabe hat. Der Agent optimiert sein Verhalten so, dass die Belohnungsfunktion maximiert wird. Sie ist maßgeblich für die Leistungsfähigkeit des Ansatzes.

In der Literatur existieren wenige konkrete Beschreibungen von Belohnungsfunktionen. Oft wird die Belohnungsfunktion mit einem negativen Vorzeichen versehen, damit eine Minimierung der Funktionsterme zu einer Maximierung der Belohnung führt. In die Belohnungsfunktion gehen die Zielgrößen Z aus dem Zielsystem, wenn möglich, direkt ein. Ein standardmäßiger Ansatz ist die Verwendung von Fehlerquadraten (Lampe et al., 2019; Sommer Obando, 2016). Diese beschreiben das Quadrat der Abweichung zwischen einer Modellgröße und einer Zielgröße. Größere Abweichung von der Zielgröße werden dadurch stärker gewichtet.

Existieren mehrere Zielgrößen, die in die Belohnungsfunktion eingehen, so können diese normiert und gewichtet werden. Die Gewichtung muss dabei anwendungsfallbezogen bestimmt werden. Zielkonflikte können dadurch nicht aufgelöst werden.

Zusätzlich können Bestrafungs- und Belohnungsterme $P(Z_1, Z_2, \dots)$ addiert werden, um ein schlechtes Verhalten zu bestrafen, oder ein positives Verhalten, das zur Aufgabenerfüllung beiträgt, zu belohnen. Zum Beispiel kann eine Bestrafung für das Nicht-Erreichen des Steuerziels gegeben werden. Somit kann eine allgemeine Beschreibung einer Belohnungsfunktion r abgeleitet werden (vgl. Gleichung 6.1).

$$r = - \left(c_1 \frac{(Z_1 - Z_1(t))^2}{Z_{1norm}} + c_2 \frac{(Z_2 - Z_2(t))^2}{Z_{2norm}} + \dots \right) - P(Z_1, Z_2, \dots) \quad 6.1$$

Vorrangig für die Definition der Belohnungsfunktion ist es, die Bewertung komplett zu objektivieren und in einer einzelnen Metrik abzubilden.

Wie in diesem Abschnitt gezeigt, wird ein sehr gutes Verständnis des zu optimierenden Vorgangs benötigt, um alle Terme der Belohnungsfunktion zu bestimmen. Zielgrößen der Ansteuerung, Grenzwerte sowie Grenzfälle müssen den Entwickelnden bekannt sein. Außerdem ist hier festzuhalten, dass die Belohnungsfunktion nicht einmalig bestimmt, sondern im Laufe der Anwendung kontinuierlich verbessert wird, um eine optimale Ansteuerung zu ermöglichen. Dabei wird der Ablauf in Abbildung 6.8 einmal komplett durchlaufen, bevor die Belohnungsfunktion angepasst wird.

6.4.4 Definition Trainingsumgebung

Nachdem das Optimierungsziel, Aktions- und Zustandsraum, sowie die Belohnungsfunktion festgelegt sind, wird die Trainingsumgebung definiert (vgl. Abbildung 6.11)

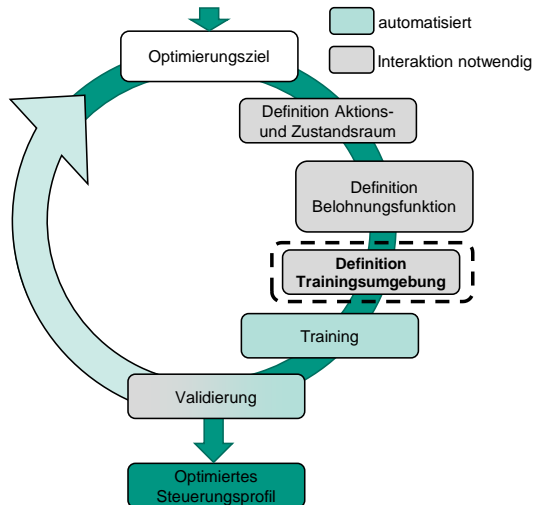


Abbildung 6.9: Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen, vereinfachte Darstellung, Definition Trainingsumgebung hervorgehoben

Das Training des Agenten ist in verschiedenen Validierungsumgebungen nach dem IPEK-XiL-Ansatz (vgl. Kapitel 2.2.3) durchführbar: in einer rein virtuellen Umgebung, verschiedene Ausprägungen gemischt physisch-virtueller Umgebungen oder in einer rein physischen Umgebung.

Das Training in einer virtuellen Umgebung bietet die Vorteile, dass keine Belegung von Prüfständen oder Versuchsfahrzeugen besteht, keine Schutzmechanismen für den Fall einer kritischen Aktion des Agenten notwendig sind und Simulationsdauer kleiner Echtzeit möglich ist. Daher ist das Training in der virtuellen Umgebung zu empfehlen.

Damit anschließend die Übertragbarkeit der Trainingsergebnisse von der Simulation in das Fahrzeug gegeben ist, benötigt das verwendete Modell eine hohe Abbildungsgüte. Gleichzeitig müssen aber auch die Rechenzeiten für eine geringe Trainingszeit so niedrig wie möglich gehalten werden.

Es muss auch definiert werden, welche Modelle für den jeweiligen Steuervorgang benötigt werden und für welche Teilsysteme welche Modellierungsgüte benötigt wird. Der IPEK-XiL-Ansatz unterstützt diesen Prozess. Beispielhaft ist für einen Gangwechselvorgang ein erweitertes IPEK-XiL-Framework in Abbildung 6.10 dargestellt.

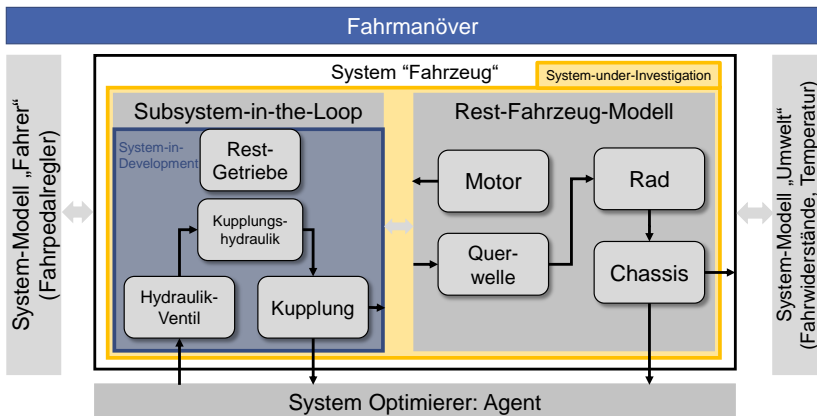


Abbildung 6.10: Verknüpfung der rein virtuellen Trainingsumgebung und des Agenten anhand des IPEK-XiL-Frameworks am Beispiel eines Gangwechsels

Simulationsmodelle können durch den Export im Format einer Functional Mock-up Unit Format (FMU) einfach mit der Python-Umgebung verknüpft werden. Es wird somit keine zusätzliche Simulationssoftware benötigt.

6.4.5 Training

Für verstärkendes Lernen steht eine Vielzahl an Algorithmen zur Auswahl. Die Hauptalgorithmen sind Q-Learning, Policy-Gradient und Actor-Critic-Ansätze (Arulkumaran, Deisenroth, Brundage & Bharath, 2017). Der Soft-Actor-Critic-Algorithmus (SAC) (Haarnoja et al.) zeigt aktuell die beste Performance über verschiedene Anwendungsbeispiele hinweg und besitzt eine hohe Stabilität hinsichtlich der Reproduzierbarkeit der Ergebnisse. Außerdem ist er wenig sensibel gegenüber Änderungen der Hyperparameter. Dadurch kann die Default-Parametrierung des Algorithmus genutzt werden. Aufgrund dieser Argumente wird die Verwendung des SAC-Algorithmus empfohlen (vgl. Kapitel 2.3.3). Die Python-Bibliothek Stable-Baselines kann für eine einfache Implementierung des Algorithmus genutzt werden (Raffin et al., 2021).

Während des Trainings wird der Steuervorgang wiederholt durchgeführt. Der Agent hat dabei die Aufgabe die Belohnung zu maximieren. Im Trainingsverlauf wird der trainierte Agent auf einen oder mehrere Testfälle angewendet, um dessen Verhalten anhand der Belohnungsfunktion zu bewerten. Testfälle können dabei definierte

Fahrmanöver wie im Beispiel des Gangwechsels eine 2-3 Schaltung sein. Liegt die Änderung der Belohnung auf einen oder mehrere Testfälle unter einem definierten Wert, so konvergiert der Agent auf eine Ansteuerung. Die Optimierung des Agenten, übernimmt dabei der ausgewählte Algorithmus.

6.4.6 Validierung

Ebenso wie die Trainingsumgebung kann die Validierung nach dem IPEK-XiL-Ansatz auf verschiedenen Ebenen und in unterschiedlichen Validierungsumgebungen erfolgen (Altenburg et al., 2022). Während des Trainings wird die Leistungsfähigkeit des Agenten bereits mit Testfällen in der Trainingsumgebung validiert. Gemessen wird die Leistungsfähigkeit dabei an der erzielten Belohnung.

Erfüllt das Ergebnis der Optimierung die im Zielsystem festgelegten Anforderungen nicht, so beginnt eine weitere Iteration des Vorgehens (vgl. Abbildung 6.8). Auf Grundlage der Validierungsergebnisse werden das Zielsystem, Aktions- und Zustandsraum, Belohnungsfunktion und Trainingsparameter iterativ angepasst und erweitert und die nächsten Methodenschritte wiederum darauf aufbauend durchgeführt. Werden die Anforderungen des Zielsystems erfüllt, so ist das Vorgehen abgeschlossen. Die Optimierung in der rein virtuellen Umgebung ist dadurch mit geringen Kosten verbunden.

Erst wenn in der rein virtuellen Umgebung ein gutes Trainingsergebnis vorliegt, wird dieses in einer gemischt physisch-virtuellen oder komplett physischen Umgebung validiert.

Die validierten Ergebnisse können dann als erste Parametrisierung für eine Ansteuerung verwendet werden. Je nach Reifegrad der genutzten Modelle können noch größere Abweichungen zwischen Simulationsergebnissen und Fahrzeugmessungen auftreten.

6.4.7 Anwendung und Übertragbarkeit

Die Methode des verstärkenden Lernens eignet sich zur parameterfreien Optimierung von Steuer- und Regelvorgängen. Das verstärkende Lernen besitzt dabei die Eigenschaft, eine Steuer- oder Regelaufgabe in einem unbekannten System zu erlernen. Dies wurde bereits erfolgreich in Strategiespielen, der Robotik und weiteren Problemstellungen gezeigt (vgl. Kapitel 1).

Bereits in der frühen Phase der Produktentwicklung kann die Methode genutzt werden, sobald Simulationsmodelle vorhanden sind. Die Ergebnisse müssen anschließend in einer geeigneten Validierungsumgebung validiert werden.

Zur Anwendung der Methode wird eine Trainings- und Validierungsumgebung benötigt. Diese können anhand des IPEK-XiL-Ansatzes aufgebaut werden. Eine rein virtuelle Trainingsumgebung senkt die Trainingskosten und gewährleistet Sicherheit. Durch die Digitalisierung der Antriebsstrangentwicklung sind im Umfeld der Getriebefunktionsentwicklung virtuelle Modelle des Antriebsstrangs vorhanden, die hierfür genutzt werden können. Es wird eine hohe Modellgüte gefordert, um das Trainingsverhalten später auf das Fahrzeug übertragen zu können. Die Trainingsdauer kann mehrere Stunden bis Tage andauern, je nach Komplexität der Problemstellung und Rechendauer der Modelle. Die Verwendung von virtuellen Maschinen oder Cloud-Computing wird empfohlen, um die Rechenzeit zu reduzieren.

Grundsätzlich ist die Anwendung der Methode unter den oben genannten Randbedingungen auf viele Vorgänge in hybriden Automatikgetrieben vorstellbar wie z.B. der Gangwechsel, Kupplungsfüllung oder Motorwiederstart.

6.5 Messdatenanalyse mit Cluster-Algorithmen

Dieses Kapitel stellt die dritte Methode Messdatenanalyse mit Cluster-Algorithmen im Kontext der Ansteuerung eines parallelen hybriden Fahrzeuggetriebes vor. Die Methode wird auf die Analyse von Fahrzeugmessdaten zur Unterstützung der Funktionsauslegung angewendet. Der Fokus in diesem Kapitel liegt auf der Anwendung der Methode. Für die theoretischen Details der Korrelationsanalyse und der Cluster-Algorithmen wird auf den Stand der Forschung verwiesen (vgl. Kapitel 2.3.2).

Die Datenanalyse unterstützt die Entwicklungsaktivität der Validierung, indem sie Testergebnisse eines Testfalls auswertet (vgl. Abbildung 6.11). Zunächst werden aus dem Produktprofil Validierungsbedarfe und daraus Validierungsziele abgeleitet. Aus den Validierungszielen werden wiederum Testfälle definiert. Aus der Durchführung der Testfälle resultieren Testergebnisse unter anderem in Form von Messdaten, die als Ausgangspunkt dieser Methode dienen (Mandel et al., 2020).

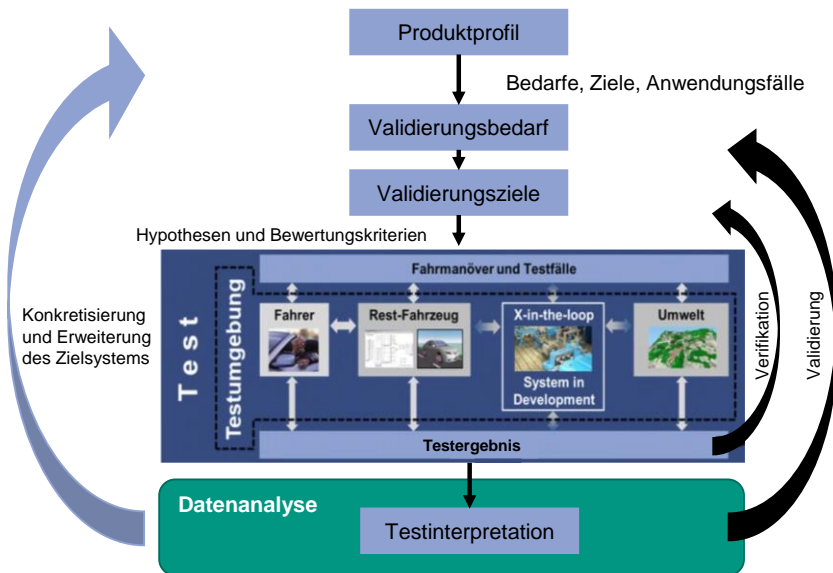


Abbildung 6.11: Korrelation zwischen Produkt-Kreation und Validierung (Eigene Darstellung nach Mandel et al., 2020).

Ein großer Vorteil der Datenanalyse mit ML-Methoden liegt in der Automatisierung. Diese sollte im Vorgehensmodell der Methode für die in Abbildung 6.12 markierten

Schritte genutzt werden, sodass sich die Entwicklungsarbeit auf die manuellen Schritte fokussieren kann. Die Automatisierung ermöglicht auch eine Skalierung der Methode auf große Datensätze. Das vorgestellte Vorgehensmodell baut dabei auf dem CRISP-DM Modell aus Kapitel 2.3.2 auf.

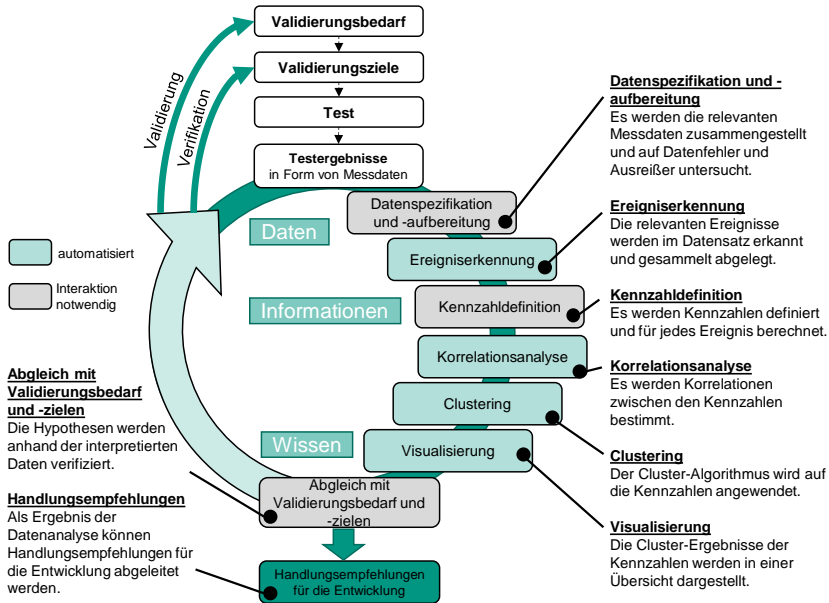


Abbildung 6.12: Vorgehen Messdatenanalyse mit Cluster-Algorithmen

6.5.1 Testergebnisse

Aus dem Produktprofil ergibt sich ein Validierungsbedarf, um den Kundennutzen zu validieren. Aus diesem werden Validierungsziele abgeleitet, die anhand von Hypothesen und Kriterien beschrieben werden. Auf Basis der Validierungsziele werden wiederum Testfälle definiert, die in definierten Validierungsumgebungen durchgeführt werden. Das Resultat der Testfälle sind Testergebnisse. Die Testergebnisse werden zum Abgleich der Kriterien der Validierungshypothesen genutzt (Verifikation), die interpretierten Testergebnisse zum Abgleich mit dem Validierungsbedarf (Validierung).

Hier setzt die Methode der Datenanalyse an und unterstützt die Testinterpretation. Abhängig von der Validierungsumgebung können die Testergebnisse in Form von Simulationsdaten, Prüfstands- oder Fahrversuchsdaten vorliegen. Im Fahrversuch liegen die Daten des Fahrzeugantriebs speziell des Getriebes als CAN-Bus-Signale vor. Ein gängiges Datenformat ist dabei MF4. Basierend auf der Definition des Wissensmanagements nach North (2016, S. 37) werden die Testergebnisse in Form von Daten zunächst in Informationen und abschließend in Wissen überführt. Dieses Wissen wird zum Abgleich mit dem Validierungsbedarf sowie zur Konkretisierung und Erweiterung des Zielsystems genutzt.

6.5.2 Datenspezifikation und -aufbereitung

Zunächst werden die Signale spezifiziert, die zur Analyse benötigt werden. Zu den Fahrzeugmessdaten gehören alle Getriebesignale aus dem Getriebesteuergerät. Die Verwendung von Signalen durch zusätzliche Messtechnik (z.B. weitere Drucksensoren im Hydrauliksystem) im Fahrzeug ist ebenfalls möglich. Zusätzlich kann spezifiziert werden, ob Messdaten aus mehreren Fahrzeugen benötigt werden, um den Einfluss von Fertigungstoleranzen oder Verschleiß und Alterung zu berücksichtigen. Treten Phänomene nur in einem bestimmten Fahrzeug auf, so werden nur Daten aus diesem Fahrzeug zur Analyse genutzt. Zum Vergleich können dann Fahrzeugdaten aus unauffälligen Fahrzeugen herangezogen werden.

Bei der Datenspezifikation muss auch die Häufigkeitsverteilung der Messdaten betrachtet werden, um diese für Cluster-Effekte in der Auswertung berücksichtigen zu können. Als Verteilungsgrößen können z.B. Last- (VM- und EM-Drehmoment) und Geschwindigkeitsbereiche herangezogen werden.

Um eine hohe Datenqualität sicherzustellen, werden die Daten auf fehlende Datenpunkte und Ausreißer untersucht, die außerhalb des plausiblen Wertebereichs liegen. Einfache Methoden Ausreißer zu identifizieren, sind das Setzen von Maximal- oder Minimalwerten oder die Relation der Datenpunkte zum Mittelwert. Werden im Datensatz fehlende Einträge oder Ausreißer identifiziert, werden diese aus dem Datensatz gelöscht. Der Vorteil der Verwendung von CAN-Bus-Signalen des Getriebesteuergeräts besteht darin, dass diese Signale das gleiche Datenformat aufweisen und die Datenqualität hoch ist. Fehlerhafte Daten oder Ausreißer sind nicht zu erwarten, auf diese wird dennoch geprüft.

Grundsätzlich können aufgrund der Automatisierbarkeit und daraus resultierenden Skalierbarkeit Datensätze beliebiger Größe genutzt werden.

6.5.3 Ereigniserkennung

Auf den Schritt der Datenspezifikation und -aufbereitung folgt die Ereigniserkennung. Die für die Untersuchung relevanten Ereignisse aus den Messdaten werden erkannt, der zeitlich relevante Bereich ausgeschnitten und gesammelt. Die Ereignisse können dabei direkt über definierte Signale, die bei Auftreten des Ereignisses ihren Zustand ändern oder über die Definition von Grenzwerten von ereignisrelevanten Signalen erkannt werden.

Ist eine direkte Erkennung oder die Definition von Grenzwerten nicht möglich bzw. deckt nicht alle Ereignisse robust ab, so kann auch ein maschinelles Lernverfahren zur Erkennung genutzt werden. Dabei wird durch überwachtes Lernen ein Modell mit Messdaten des Ereignisses trainiert und lernt dadurch die Zuordnung von Signalverläufen zu dem spezifischen Ereignis.

6.5.4 Kennzahldefinition

Um die zeitreihenbasierten Fahrzeugmessdaten auswerten und Cluster bilden zu können, werden Kennzahlen benötigt, die anhand der Zeitreihen berechnet werden (vgl. Abbildung 6.13). Hierbei eignen sich statistische Kenngrößen wie Mittelwert, Minima und Maxima oder Gradienten der CAN-Bus-Signale. Der Anwendende nutzt dabei sein Systemverständnis, um geeignete Kennzahlen zu ermitteln. Die Kennzahlen können wiederum rückwirkend einfach angepasst werden, falls sich aus der Validierung neue Erkenntnisse ergeben. Durch die Ereigniserkennung und Kennzahldefinition erhalten die Daten eine Bedeutung und werden somit zu Informationen. Zum Beispiel könnten Kennzahlen für eine 2-3 Schaltung Fahrzeuggeschwindigkeit, Getriebeeingangs- und Ausgangsdrehzahl zum Schaltzeitpunkt, Schaltzeit, Fahrzeugruck, berechnete Momente und -drücke der zu- und abschaltenden Kupplungen sein.

6.5.5 Korrelationsanalyse

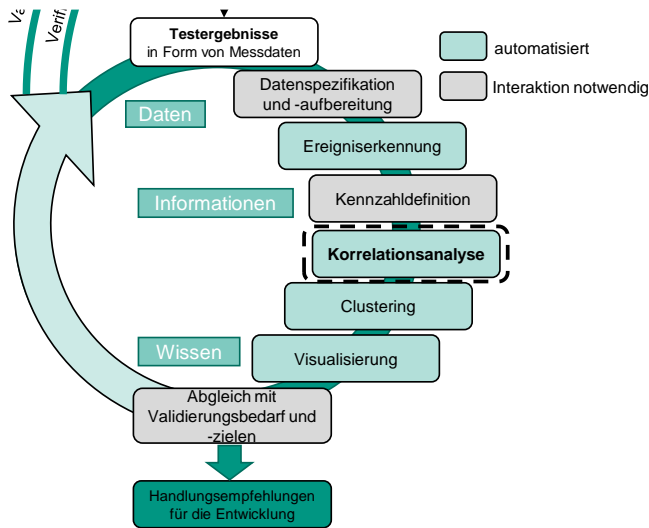


Abbildung 6.13: Vorgehen Messdatenanalyse mit Cluster-Algorithmen, Ausschnitt und vereinfachte Darstellung, Korrelationsanalyse hervorgehoben

Nach der Kennzahldefinition erfolgt die Korrelationsanalyse (vgl. Abbildung 6.13). In Datensätzen können lineare Abhängigkeiten zwischen Signalen oder Kennzahlen auftreten. Bei linear abhängigen Größen enthält eine Größe bereits die Informationen, weshalb die zweite Größe nicht mehr direkt zur Informationsgewinnung benötigt wird. Deshalb kann diese Größe aus dem Datensatz entfernt werden, was wiederum die Komplexität des Datensatzes reduziert. Außerdem können Korrelationen Wissen und Verständnis über das Gesamtsystem generieren.

In einer Korrelationsanalyse (vgl. Kapitel 2.3.2) werden die linearen Abhängigkeiten von Kennzahlen bestimmt und graphisch dargestellt. Für jeden Matrixeintrag von zwei Kennzahlen wird dabei der Korrelationskoeffizient nach Bravais und Pearson zwischen den beiden Größen für den Datensatz berechnet. Beispielsweise können als Kenngröße bei einem Schaltvorgang im Getriebe die Schaltzeit sowie der Fahrzeugruck herangezogen werden. Im Regelfall liegt deren Korrelation zwischen -0,5 und -1, was bedeutet, dass bei Schaltungen mit höherer Schaltzeit ein geringerer Fahrzeugruck vorliegt und umgekehrt. Dies ist aus der Praxiserfahrung heraus plausibel.

6.5.6 Clustering

Das Clustering der im vierten Schritt definierten Kennzahlen bildet den zentralen Methodenschritt. Clustering ist ein nicht überwachtes Lernverfahren, das Punkte einer Datenmenge einem Cluster zuordnet (vgl. Abbildung 6.14). Es existieren verschiedenste Cluster-Methoden. Das Python-Modul Scikit-Learn bietet eine gute Übersicht und eine einfache Implementierung der Algorithmen (Pedregosa et al., 2011). Zur Auswahl eines Algorithmus können folgende Kriterien herangezogen werden: Homogenität der Cluster-Größe und der Cluster-Formen, sowie die Cluster-Anzahl. Tabelle 6.1 gibt eine Übersicht der Algorithmen und für welche Cluster-Eigenschaften sie sich eignen (vgl. Kapitel 2.3.2).

Tabelle 6.1 Cluster-Algorithmen und Auswahlkriterien

| Algorithmus | Beispiel | Cluster-Größe | Cluster-Form | Cluster-Anzahl |
|-----------------|----------|---------------|--------------|----------------|
| Partitionierend | kMeans | homogen | homogen | vorgegeben |
| Dichtebasiert | DBSCAN | heterogen | heterogen | frei |
| Hierarchisch | Ward | heterogen | heterogen | vorgegeben |

Ist keine Cluster-Form a priori bekannt, können initial mehrere Verfahren für einen Vergleich genutzt werden. Die meisten Cluster-Verfahren besitzen wenige anpassbare Parameter. Diese beschreiben die Anzahl der Cluster oder die Cluster-Größe über Abstands- oder Dichteigenschaften. Je nach Verfahren gibt es hierfür Einstellregeln oder verschiedene Parameterkombinationen können miteinander verglichen werden. Vor dem Clustering wird der Datensatz skaliert, um Effekte über unterschiedliche Skalen zu vermeiden.

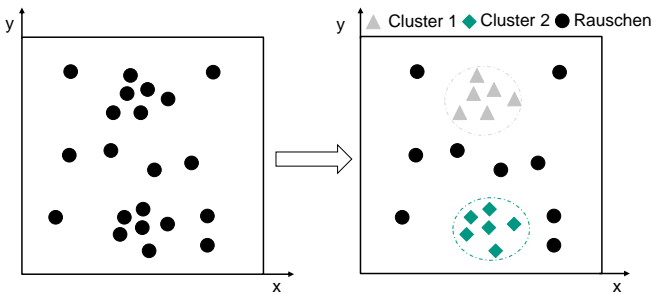


Abbildung 6.14: Clustering am Beispiel des DBSCAN Algorithmus

6.5.7 Visualisierung

Die Berechnung von Clusterzentren und -grenzen ermöglicht eine quantitative Beschreibung der Cluster. Dadurch kann z.B. der y-Anteil eines zweidimensionalen Clusters bei festgelegtem x-Anteil in einem Säulendiagramm dargestellt werden. Alle Säulendiagramme der Kennzahlen werden dann in einer Übersicht zusammengefasst. Dieser Schritt ermöglicht die automatisierte Auswertung und Visualisierung der Kennzahlen und generiert Informationen über das System (s. Abbildung 6.15).

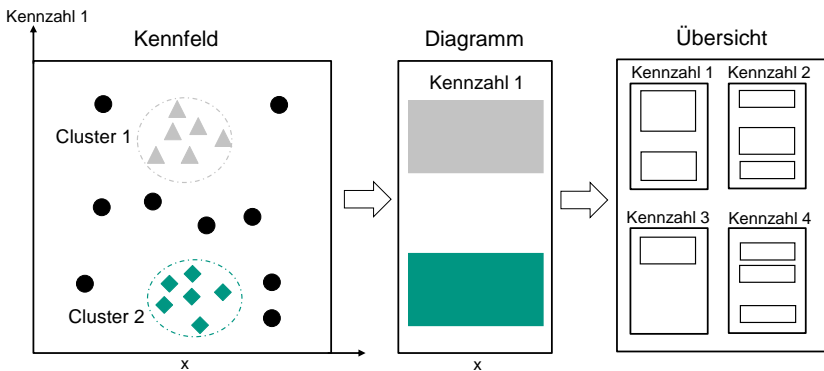


Abbildung 6.15: Veranschaulichung der Visualisierung

6.5.8 Verifikation und Validierung

Die abschließende Validierung besteht aus der Interpretation und Evaluation der visualisierten Daten durch den Anwendenden. Durch die Korrelationsanalyse, Clustering und Visualisierung werden die Informationen des zu untersuchenden Vorgangs wie eine Schaltung oder ein Motorwiederstart durch die Vernetzung der Kennzahlen in Wissen überführt.

Mit dem gewonnenen Wissen werden die Validierungshypothesen der Validierungsziele anhand der Kriterien verifiziert (Verifikation). Gleichzeitig wird der Validierungsbedarf abgeglichen und gegebenenfalls angepasst.

Zur Interpretation der Darstellung ist Systemwissen und Systemverständnis entscheidend. Die gewonnenen Erkenntnisse können genutzt werden, um Handlungsempfehlungen für die Entwicklung zu geben.

Können diese Fragestellungen nicht oder nur unvollständig beantwortet werden, so beginnt eine Iteration des Vorgehens (vgl. Abbildung 6.12). Auf Grundlage der Validierungsergebnisse werden das Zielsystem, Daten und Kennzahlen iterativ angepasst und erweitert und die nächsten Methodenschritte wiederum darauf aufbauend durchgeführt. Können die Fragestellungen beantwortet werden, so ist das Vorgehen abgeschlossen.

6.5.9 Anwendung und Übertragbarkeit

Die Methode Messdatenanalyse mit Cluster-Algorithmen eignet sich, um zeitreihenbasierte Signalverläufen mit Hilfe von Kennzahlen zu charakterisieren, diese Kennzahlen auf Korrelationen zu untersuchen und basierend auf diesen Kennzahlen Cluster in Fahrzeugmessdaten zu identifizieren. Dadurch trägt diese Methode im Rahmen der Validierung zur Analyse bei. Die Erkenntnisse ergänzen und erweitern das Zielsystem der Getriebeentwicklung.

Grundlage hierfür bilden Testergebnisse in Form von Messdaten. Im Kontext der Funktionsentwicklung paralleler hybrider Fahrzeuggetriebe eignen sich Fahrzeug-, Prüfstands- oder Simulationsdaten. Die Rechenzeit zur Auswertung der Daten hängt stark von der Datenmenge, deren zeitlicher Auflösung und der vorhandenen Rechenleistung ab. In der Auswertung von Entwicklungsdaten von etwa zehn Fahrzeugen liegt die Rechendauer im Minutenbereich. Bei größeren Datensätzen kann die Auswertung auch Stunden bis Tage andauern.

Dadurch, dass die Methode auf Daten aus verschiedenen Validierungsumgebungen angewendet werden kann, kann eine breite Anwendbarkeit zur Messdatenanalyse von parallelen hybriden Fahrzeuggetrieben angenommen werden.

Die Erkenntnisse aus der Analyse von Fahrzeugmessdaten aus der sich in der Entwicklung befindlichen Systemgeneration G_n können für Handlungsempfehlungen der Generation G_n , aber auch für die darauffolgende Generation G_{n+1} und für die bereits am Markt eingeführte Generation G_{n-1} genutzt werden (vgl. Abbildung 6.16). Die Übertragbarkeit auf die unterschiedlichen Generationen muss entsprechend geprüft werden.

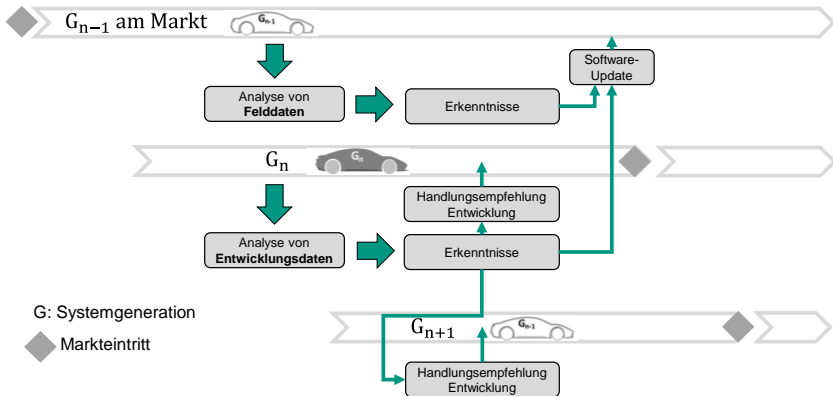


Abbildung 6.16: Datenanalyse im Kontext der SGE-Systemgenerationsentwicklung nach Albers. Die Erkenntnisse aus der Analyse von Feld- und Entwicklungsdaten können für verschiedene Systemgenerationen genutzt werden.

6.6 Zwischenfazit

In Kapitel 6 wurden die ML-Methoden und deren Anwendung im Kontext der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben beschrieben, um Forschungsfrage II (FF-II) zu beantworten.

Forschungsfrage II (FF-II)

Wie müssen die Methoden des maschinellen Lernens gestaltet werden, um Entwickelnde bei der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen?

Die Methode überwachtes Lernen wurde für virtuelle Sensoren, die Methode verstärkendes Lernen für Steuerung-Optimierungen, und die Methode nicht überwachtes Lernen für Messdatenanalysen genutzt. Die bereits existierenden ML-Methoden aus dem Stand der Forschung wurden mit konkreten Vorgehensmodellen und einer Beschreibung der Auswahl der komplexen Algorithmen und Optimierungsverfahren ergänzt.

Virtuelle Sensoren mit Künstlichen Neuronalen Netzen. Virtuelle Sensoren bilden die Schnittstelle zwischen realen physikalischen Größen der Hardware und virtuellen Größen der Getriebesoftware. Diese können auch als virtuelles Koppelsystem im Kontext des IPEK-XiL-Ansatzes verstanden werden. Im Zielsystem werden zunächst die relevanten Fahrmanöver und Umgebungsbedingungen sowie Modelleigenschaften wie zeitliche Auflösung und Dynamik des Modells, Modellparameteranzahl aufgrund gegebener Speicheranforderungen und Kriterien zur Bewertung der Modellgüte definiert. Wichtig ist hierbei, dass das Zielsystem iterativ anhand der Validierungsergebnisse angepasst und erweitert wird. Die Modelleingangsgrößen werden anhand des physikalischen Zusammenhangs zwischen Eingangsgrößen und Modellzielgröße abgeleitet. Um die Daten zu erzeugen, werden die Fahrmanöver in definierten Validierungsumgebungen durchgeführt. Liegen bereits Daten vor, so muss geprüft werden, ob diese den Fahrmanövern und einer korrekten Abbildung der Validierungsumgebung entsprechen, bevor diese genutzt werden können.

Bei den automatisierten Prozessschritten der Datenaufbereitung und Modellerstellung wird aufgezeigt, welche Parameter und Optimierungsverfahren speziell für virtuelle Sensoren sinnvoll sind. Erklärbarkeitsansätze unterstützen die Validierung im Umgang mit KNN als Black-Box-Modelle. Die Methoden Sensitivitätsanalyse und LIME-Analyse werden vorgeschlagen. Die abschließende Modellvalidierung wird anhand des Testdatensatzes durchgeführt. Erfüllt das Modell die Anforderungen des Zielsystems nicht, so wird die Methode iterativ durchgeführt. Eine Erweiterung der Daten durch weitere Fahrmanöver oder das Anpassen der Modelleingangsgrößen sind Maßnahmen zur Modelloptimierung. Zur Ableitung dieser Maßnahmen können die Erklärbarkeitsansätze unterstützen. Zusätzlich ergeben die Schritte Abgleich mit Validierungsdatensatz, Erklärbarkeitsansätze und Abgleich mit Testdatensatz die Validierung der Methode.

Steuerungs-Optimierung mit verstärkendem Lernen. Hierfür ist die Definition der Belohnungsfunktion entscheidend, weshalb eine allgemeine Berechnungsformel vorgestellt wird. Für die Wahl des Aktions- und Zustandsraums müssen die physikalischen Zusammenhänge herangezogen werden. Durch Anpassung der Belohnungsfunktion, des Aktions- und Zustandsraums sowie der Trainingsparameter in Abhängigkeit des Trainingsergebnisses erhält die Methode einen iterativen Charakter. Diese Iterationen sind wichtig, um Erfahrungen über das Agentenverhalten zu gewinnen und damit eine Belohnungsfunktion mit schneller Konvergenz und hoher Zielerreichung einstellen zu können.

Die Trainingsumgebung wird aus Kosten- und Sicherheitsgründen rein virtuell aufgebaut. Für eine gute Übertragbarkeit auf weiteren Validierungsebenen ist für

diese eine möglichst hohe Modellgüte anzustreben, bei gleichzeitig akzeptabler Rechenzeit. Auch die Validierungsumgebung muss bedarfsgerecht definiert werden. Der IPEK-XiL-Ansatz unterstützt den ziel- und bedarfsgerechten Aufbau der Trainings- und Validierungsumgebung. Die Trainingsergebnisse aus der virtuellen Simulationsumgebung werden anschließend in einer gemischt physisch-virtuellen oder rein physischen Validierungsumgebung validiert. Des Weiteren ermöglicht verstärkendes Lernen im Vergleich zu anderen Optimierungsverfahren eine parameterfreie Optimierung. Dadurch besteht die Möglichkeit, dass neue erfolgreiche Ansteuerungsstrategien abgeleitet werden können.

Messdatenanalyse mit Cluster-Algorithmen. Zur Validierung der Produktfunktionen werden anhand der Validierungsziele Hypothesen und Kriterien abgeleitet, um diese anhand definierter Testfälle zu verifizieren. Die Messdatenanalyse unterstützt dabei die Auswertung und Interpretation von Testergebnissen. Die Daten werden nach dem Wissensmanagement von North in Informationen und anschließend in Wissen überführt. Bei dieser Methode stehen die Datenspezifikation und die Definition von Kennzahlen im Vordergrund. Bei der Datenspezifikation müssen die relevanten Signale ermittelt werden. Die Ereigniserkennung kann über Trigger-Signale, Grenzwerte oder überwachte Lernansätze umgesetzt werden. Die Definition von Kennzahlen macht die Zeitreihen auswertbar. Diese werden anwendungsfallsspezifisch abgeleitet und können im Laufe der Analyse erweitert und angepasst werden. Die Korrelationsanalyse, das Clustering und die Visualisierung werden automatisiert durchgeführt, sodass Anwendende den Fokus auf die Interpretation und Evaluation der Ergebnisse und somit den Erkenntnisgewinn und das Ableiten von Handlungsempfehlungen setzen können. Die Handlungsempfehlungen entstehen dabei durch die Verifikation der Validierungsziele und den Abgleich mit den Validierungsbedarfen.

Bei der Anwendung der Methoden muss zwischen dem Entwickelnden eines späteren Tools zur Methodennutzung und dem Anwendenden, dem Getriebeentwickelnden, unterschieden werden. Zur Erstellung eines Tools ist großes Expertenwissen im Bereich ML notwendig. Dagegen kann durch die Automatisierung der gängigen ML-Schritte wie Datenaufbereitung, Modellerstellung, Training eines Agenten oder Clustering das Vorwissen für den Anwender der Methoden reduziert werden. Die Automatisierung bietet für alle drei vorgestellten Methoden den großen Vorteil, dass sich die Entwicklungsarbeit dann auf die manuellen Schritte der Methode fokussieren kann. Außerdem sind die automatisierbaren Schritte sehr gut skalierbar. Datensätze können einfach ausgetauscht und skaliert werden. Programmiersprachen wie Python, R, C++ und Matlab bieten anhand vordefinierter Bibliotheken eine große Funktionalität und reduzieren gleichzeitig den Implementierungsaufwand.

7 Anwendung der Methoden

Kapitel 7 behandelt die Anwendung der in Kapitel 6 beschriebenen ML-Methoden auf drei praxisrelevante Entwicklungsaufgaben innerhalb der Forschungsumgebung. Dabei sollen die Methoden evaluiert und die dritte Forschungsfrage beantwortet werden.

Forschungsfrage III (FF-III)

Welche messbaren Mehrwerte und Potentiale ergeben sich durch die Methoden des maschinellen Lernens?

Abbildung 7.1 zeigt die zeitliche Einordnung der Entwicklungsaufgaben in den Entwicklungsprozess anhand drei unterschiedlicher Systemgenerationen. Alle Generationen befinden sich in der Entwicklung. Generation $G_n^{\{konventionell\}}$ beschreibt einen konventionellen Antrieb, wohingegen $G_n^{\{hybrid\}}$ und $G_{n+1}^{\{hybrid\}}$ eine hybride Antriebsvariante darstellen.

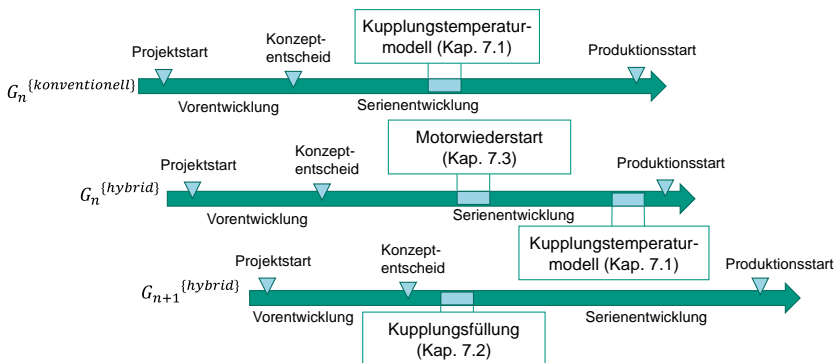


Abbildung 7.1: Einordnung der Entwicklungsaufgaben in den Entwicklungsprozess von drei verschiedenen Systemgenerationen

Bei der ersten Entwicklungsaufgabe (Kapitel 7.1) wird ein Kupplungstemperaturmodell entwickelt, das die höchste mittlere

Stahllamellentemperatur der Fahrkupplung K1 eines hybriden DKG eindimensional abbildet. Aktuell verwendete Temperaturmodelle weisen eine hohe Temperaturabweichung gegenüber den Messwerten von in einem Versuchsfahrzeug verbauten Temperatursensoren auf. Die Methode aus Kapitel 6.3 wird hierbei zur Modellierung eines virtuellen Temperatursensors genutzt. Die Methode wurde, angewendet auf ein Kupplungstemperaturmodell eines konventionellen Getriebes, in der Arbeit von Schönstein (2021)¹ behandelt und von Schuchter, Schönstein, Bause und Albers (2022) veröffentlicht. Die Ergebnisse der Übertragung auf weitere Kupplungen ist unter anderem in Altenburg, Schuchter, Bause und Albers (2023) zu finden.

Entwicklungsaufgabe zwei (Kapitel 7.2) beschäftigt sich mit der Kupplungsfüllung der Fahrkupplungen beim rein elektrischen Fahrmodus. Die Füllparameter werden im aktuellen Entwicklungsprozess aufwendig händisch ermittelt, um eine geringe Füllzeit bei gleichzeitig geringem Druckeinbruch zu ermöglichen. Druckeinbrüche können dabei zu Komforteinbußen führen. Die Methode verstärkendes Lernen wird als Optimierungswerkzeug angewendet.

Entwicklungsaufgabe drei (Kapitel 7.3) beschäftigt sich mit der Datenanalyse der Funktion hybrider Motorwiederstart auf Basis von Messungen aus Entwicklungsfahrzeugen. Für die Messdatenanalyse wird ein Cluster-Algorithmus genutzt, um Motorzustart-Cluster in den Daten zu ermitteln. Die Methode wurde von Schuchter, Bause und Albers (2022) veröffentlicht.

Abschließend gibt Kapitel 7.4 ein Zwischenfazit.

Die einzelnen Methoden werden nach den in Kapitel 6 beschriebenen Schritten durchgeführt. Das Vorgehen ist dabei stark iterativ. Aus Gründen der Übersichtlichkeit wird deshalb nur die finale Umsetzung der einzelnen Schritte aufgezeigt und erläutert. Zur Bewertung der Ergebnisse werden die in Kapitel 6.2 entwickelten messbaren Erfolgskriterien angewendet.

7.1 Kupplungstemperaturmodell

Kupplungstemperaturmodelle werden benötigt, um beim Überschreiten von kritischen Temperaturen die Kupplung zu öffnen und das tribologische System vor

¹ Abschlussarbeit (unveröffentlicht)

Schaden zu schützen. Zur Temperaturbestimmung werden Berechnungsmodelle auf dem Getriebesteuergerät verwendet, da aus Kosten- und Bauraumgründen keine Tempertursensoren in den Serienfahrzeugkupplungen verbaut werden können. Neben dem Bauteilschutz hat die Kupplungstemperatur wiederum Einfluss auf den Kühllölvolumenstrom sowie die Schaltstrategie. Wird eine Mehrfachrückschaltung bei hoher Kupplungstemperatur angefordert, so wird diese nicht ausgeführt.

In diesem Anwendungsbeispiel wird ein eindimensionales Lamellentemperatur-Modell der Kupplung K1 einer nasslaufenden Doppelkupplung in einem hybriden DKG betrachtet. Die Kupplung beinhaltet mehrere Stahl- und Belagslamellen mit einem Papierreibbelag. Die Temperaturmessungen stammen aus einem im Versuchsfahrzeug verbauten Telemetriegetriebe, das Temperatursensoren in den Stahllamellen 1, 3 und 5 verbaut hat. Die höchste Stahllamellentemperatur wird an der Fahrkupplung K1 an der Lamelle 3 gemessen und soll deshalb abgebildet werden (vgl. Abbildung 7.2).

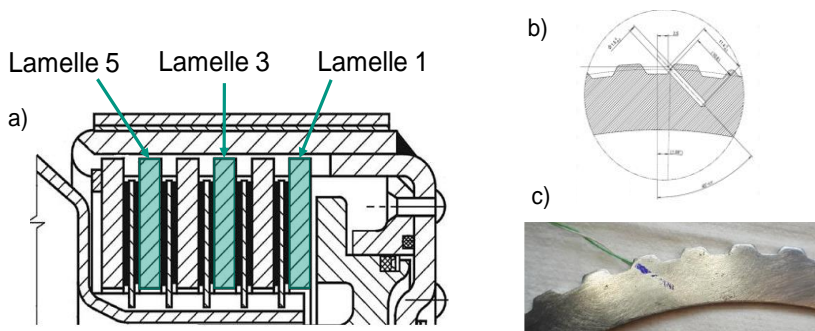


Abbildung 7.2: a) Kupplungsquerschnitt (Eigene Darstellung nach Naunheimer et al., 2019, b) Bohrungsquerschnitt Temperatursensor in Stahllamelle c) Temperatursensor und Stahllamelle

Beim Eintrag einer hohen Reibleistung kann es zum Überhitzen der Kupplung kommen. Dabei befindet sich die Kupplung in einem hohen Schlupf und wird gleichzeitig mit einem hohen Reibmoment belastet.

Die dafür relevanten Lastfälle sind Anfahren am Hang mit maximalem Fahrzeuggewicht sowie Mehrfachrückschaltungen z.B. eine Schaltung vom achten in den vierten Gang (8-4 Schaltung) mit einem Gangeinlegefehler. Ein Gangeinlegefehler tritt auf, wenn der Gang nicht eingelegt werden kann und ein zweiter Einlegevorgang notwendig ist. Dadurch kommt es zu hohen Schlupfzeiten und damit zu einer hohen Kupplungstemperatur.

Durch die Elektrifizierung des Antriebs ist auch rein elektrisches und hybrides Anfahren am Hang möglich. Die Drehmomente für das hybride Anfahren können auf den VM und die EM aufgeteilt werden, sodass bei gleichem Fahrmanöver im Vergleich zum konventionellen Antrieb ein anderer Betriebspunkt der VM gewählt wird. Liegt dieser zum Beispiel in einem niedrigeren Drehzahlbereich, liegt auch an der Fahrkupplung ein kleinerer Schlupf an. Dies erschwert die Modellbildung im Vergleich zum konventionellen Antrieb. Gleichzeitig muss auch beim Anfahren am Hang mit geringerer Zuladung und unterschiedlichen Steigungswinkeln sowie bei Mehrfach-Rückschaltungen ohne Gangeinlegefehler die Temperatur genau bestimmt werden.

Für diese Modelle werden deshalb eine hohe Genauigkeit und gleichzeitig eine hohe Robustheit bei verschiedenen Fahrmanövern gefordert.

Haupteinflussgrößen auf die Temperatur sind die eingetragene Reibleistung und das Abkühlverhalten. Die Reibleistung wird maßgeblich von der Reibungszahl bestimmt. Die Reibungszahl ist bei den Reibpaarungen, die in Automatikgetrieben verwendet werden, abhängig von vielen Parametern wie unter anderem Reibmaterial, Stahllamelle (Oberflächenbearbeitungen, Wärmebehandlungsverfahren, Nitrieren, Gestaltung Mitnehmer), Gleitgeschwindigkeit, Anpressdruck, Temperatur und Kühlöl. Während Änderungen von Temperatur, Anpressdruck usw. meist nur die Höhe des übertragbaren Drehmoments beeinflussen und bei der Steuerung kompensiert werden können, sind Reibungszahländerungen in Abhängigkeit von der Gleitgeschwindigkeit für den dynamischen Ablauf des Reibvorgangs maßgebend (vgl. Kapitel 2.1.2). Durch hohe Differenzdrehzahlen und Einspritzen des Kühlöls zwischen die Lamellen und der Nutenform kann es zu einer turbulenten Strömung kommen. Es ist sehr schwierig die Wärmeübertragungsvorgänge dabei genau zu beschreiben (Bernhardt, 2017, S. 96–123). Außerdem gibt es eine Vielzahl von variierenden Betriebsbedingungen im Fahrzeug. Wegen dieser Vielzahl von Einflussgrößen ist es aufwendig ein genaues und robustes Temperaturmodell der Kupplung zu bestimmen.

Aufgrund der beschriebenen Komplexität weisen aktuell verwendete physikalisch modellierte Modelle, nachfolgend als Referenzmodell bezeichnet, Abweichungen zur Messung auf. Die Modelle müssen deshalb nochmals im Fahrzeug kalibriert werden. Da es eine Vielzahl verschiedener Fahrmanöver gibt, ist es für Entwickelnde sehr zeitaufwendig einen optimalen Parametersatz für alle Fahrmanöver zu identifizieren.

Nachfolgend werden die in Kapitel 6.3 beschriebenen Schritte der Methode Modellierung von virtuellen Sensoren mit KNN auf diese Entwicklungsaufgabe

angewendet, um ein datenbasiertes Kupplungstemperaturmodell zu generieren (vgl. Abbildung 7.3). Um zusätzlich die Übertragbarkeit der Methode aufzuzeigen, wird die Methode anschließend auch auf das Temperaturmodell der Trennkupplung K0 ($G_n^{\{hybrid\}}$), sowie auf das Modell der Kupplung K1 des konventionellen Getriebes ($G_n^{\{konventionell\}}$) angewendet.

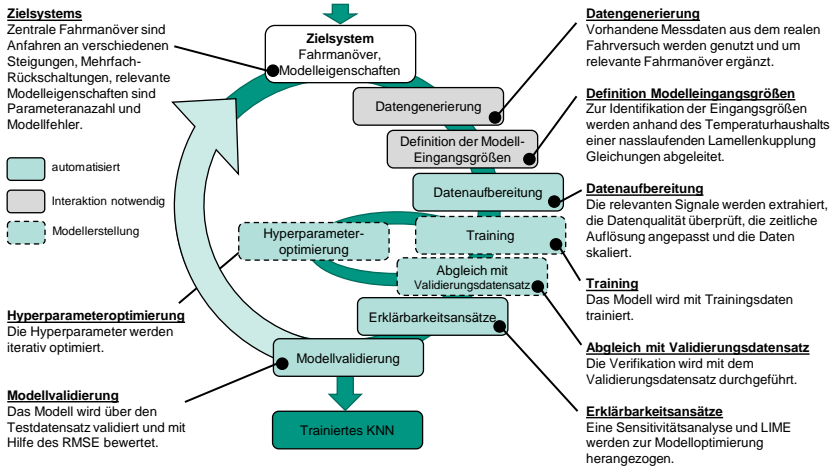


Abbildung 7.3: Vorgehen Modellierung von virtuellen Sensoren mit KNN angewendet auf Kupplungstemperaturmodelle

7.1.1 Zielsystem

Das Kupplungstemperaturmodell bildet die gemessene Stahlamellentemperatur der Lamelle 3 der Kupplung K1 ab und berücksichtigt dabei die temperaturkritischen Fahrmanöver aus Tabelle 7.1. Diese sind das Anfahren am Hang mit maximaler Fahrzeugzuladung mit Steigungen von 10 %, 15 %, 20 % und 30 % jeweils vorwärts und rückwärts sowie jeweils hybrid und reinelektrisch angefahren, und die Mehrfachrückschaltung (MFR) vom achten in den vierten Gang mit einem Gangeinlegefehler. Anfahren bedeutet dabei, dass das Fahrzeug mit einer Geschwindigkeit von kleiner 5 km/h mehrere Sekunden am Hang gefahren wird. Bei allen weiteren Schaltungen liegen die Temperaturen in niedrigen Bereichen, weshalb diese nicht berücksichtigt werden.

Ein Großteil der Fahrmanöver wurde zur Validierung des Referenzmodells bereits mit einem Versuchsfahrzeug mit Telemetriegetriebe im Fahrversuch in einer rein physischen Testumgebung vermessen. Da das Versuchsfahrzeug bereits aufgebaut ist, wurden weitere Messungen gefahren, um die Fahrmanöver zu vervollständigen. Durch die Verwendung einer rein physischen Testumgebung werden Fahrmanöver und das Systemverhalten mit einem sehr hohen Produktreifeegrad abgebildet. Von Nachteil ist die geringere Reproduzierbarkeit der Fahrmanöver. Alle Anfahrvorgänge werden mit maximalem Fahrzeuggewicht durchgeführt, um ein maximales Kupplungsmoment abzubilden. Um ein robustes Verhalten des KNN zu erreichen, sind mehrere unterschiedliche Messungen der gleichen Lastfälle in den Trainingsdaten enthalten. Eine Übersicht der Lastfälle und deren Häufigkeit ist in Tabelle 7.1 dargestellt.

Tabelle 7.1 Fahrmanöver Trainingsdatensatz

| | Steigung | Fahrmodus | Fahrrichtung | Häufigkeit |
|---|-----------------|------------------|---------------------|-------------------|
| Anfahren | 10% | elektrisch | vorwärts | 1x |
| | 10% | hybrid | vorwärts | 2x |
| | 15% | elektrisch | vorwärts | 3x |
| | 15% | elektrisch | rückwärts | 3x |
| | 20% | elektrisch | vorwärts | 3x |
| | 20% | elektrisch | rückwärts | 3x |
| | 20% | hybrid | vorwärts | 3x |
| | 20% | hybrid | rückwärts | 3x |
| | 30% | elektrisch | vorwärts | 3x |
| | 30% | elektrisch | rückwärts | 3x |
| | 30% | hybrid | vorwärts | 3x |
| | 30% | hybrid | rückwärts | 3x |
| Mehrfachrückschaltung 8-4 mit Gangeinlegefehler | | | | 4x |
| Mehrfachrückschaltung 7-3 mit Gangeinlegefehler | | | | 2x |

Da die Temperaturmesstelemetrie anfällig für Schäden ist, existieren keine Dauerlaufdaten der Kupplungstemperatur. Da nur ein Fahrzeug mit Messtechnik ausgestattet ist, können der Einfluss von Alterung, Verschleiß oder Fertigungstoleranzen über die Verwendung von Messungen aus verschiedenen Fahrzeugen mit unterschiedlichem Gebrauchstand nicht berücksichtigt werden.

Da das Modell später auf dem Getriebesteuergerät integriert werden soll, muss der Speicherbedarf des KNN geringgehalten werden. Dies führt zu einer begrenzten Parameteranzahl und damit begrenzter Größe des Netzes. Das aktuell genutzte Referenzmodell besitzt etwa 1000 Parameter, was deshalb als maximale Parameteranzahl für das KNN herangezogen wird. Die obere Grenze der zeitlichen Auflösung der Temperaturberechnung ist aufgrund von Systemanforderungen 200ms. Als Bewertungskriterium der Modellgüte wird der Root-Mean-Square-Error (RMSE) zwischen dem verwendeten Modell und der Messung herangezogen. Dieser wird aus der Wurzel der Summe der quadratischen Abweichung zwischen gemessener Temperatur T_i und Modelltemperatur $T_{KNN,i}$ über den gesamten Datensatz berechnet, der anschließend gemittelt wird. Je niedriger der RMSE, desto besser stimmt das Modell mit der Messung überein. Durch das Fehlerquadrat werden große Abweichungen stärker gewichtet, weshalb dieser Ansatz genutzt wird.

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=0}^N (T_{KNN,i} - T_i)^2} \quad 7.1$$

Der RMSE wird anschließend auf den RMSE des Referenzmodells normiert. Somit bedeutet ein normierter RMSE von größer 1 eine geringere Modellgüte als das Referenzmodell, und ein normierter RMSE von kleiner 1 eine höhere Modellgüte als das Referenzmodell.

7.1.2 Datengenerierung

Ein Großteil der Daten von Fahrmanövern im Versuchsfahrzeug am Steigungshügel sowie in der Ebene lagen bereits vor. Einzelne Fahrmanöver wurden durchfahren, um den Datensatz zu vervollständigen. Die Gesamtlänge des Trainingsdatensatzes beträgt insgesamt 80 Minuten. 12,7 % des gesamten Trainingsdatensatzes werden als Validierungsdatensatz verwendet. Darin sind alle Lastfälle einfach enthalten. Mit diesem wird die Leistungsfähigkeit des KNN während des Trainings bestimmt. Außerdem werden mit diesem anschließend die Hyperparameter optimiert.

Für den Testdatensatz werden ebenfalls die Lastfälle aus Tabelle 7.1 ausgewählt. Die Länge des Testdatensatzes beträgt 14min. Der Temperaturverlauf des

Trainingsdatensatzes mit Beschriftung der Lastfälle ist in Abbildung 7.4 dargestellt. Somit ergibt sich eine Aufteilung der gesamten Messdaten von 72,4 % für den Trainingsdatensatz, 12,7 % für den Validierungsdatensatz und 14,9 % für den Testdatensatz.

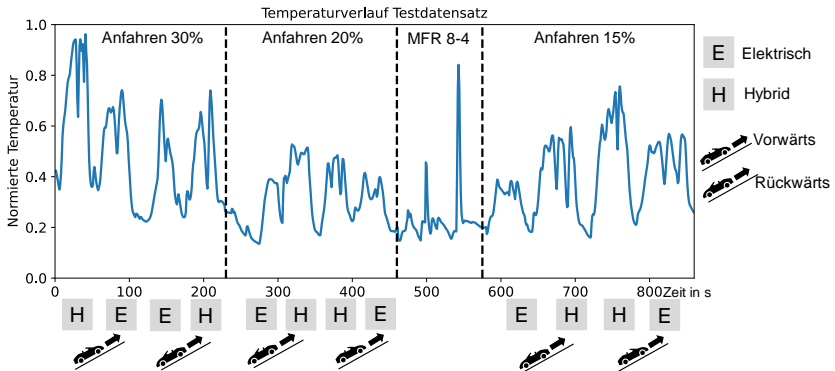


Abbildung 7.4: Normierter Temperaturverlauf des Testdatensatzes mit zugehörigen Lastfällen. Es werden für den Lastfall Anfahren für verschiedene Steigungen je vier Fahrzustände durchfahren, aufgebaut aus der Kombination hybrider Fahrbetrieb (H), elektrischer Fahrbetrieb (E) und vorwärts und rückwärts.

7.1.3 Definition der Modelleingangsgrößen

Die Eingangsparameter werden aus den physikalischen Zusammenhängen während dem Aufheizen und Abkühlen der Kupplung im Fahrbetrieb abgeleitet. Die Gleichungen hierfür wurden bereits in Kapitel 2.1.2 beschrieben und werden noch einmal aufgegriffen.

Für das Aufheizen der Kupplung ist die Reibleistung P_R maßgebend. Diese berechnet sich nach Gleichung 2.2 unter der Annahme eines konstanten Reibmoments und Winkelgeschwindigkeiten aus dem Produkt des Kupplungsmoments M_K und der Differenzwinkelgeschwindigkeit $\Delta\omega$. Die Differenzwinkelgeschwindigkeit ist proportional zur Differenzdrehzahl Δn zwischen Stahl- und Belagslamelle.

$$P_V = P_R = M_K(\omega_2(t) - \omega_1(t)) = 2\pi\Delta n M_K \quad 7.2$$

Die Normalkraft an der Kupplung ergibt sich aus dem Anpressdruck p und der Kolbenfläche A_K . Setzt man dies in Gleichung 2.1 ein, so ergibt sich für das Reibmoment M_K :

$$M_K = \mu r_m p A_K z \quad 7.3$$

Dabei steht z für die Anzahl der Lamellen, r_m für den mittleren Reibradius und μ für die Reibungszahl. Der Anpressdruck p und die Differenzdrehzahl Δn können über Sensoren genau gemessen werden und werden deshalb aus Eingangsgrößen ausgewählt.

Ein Teil der Wärmemenge wird in der Kupplung gespeichert. Gleichzeitig wird ein Wärmestrom der Kupplung an das umströmende Kühlöl und das Gehäuse abgegeben. Einfluss auf den durch Konvektion abgeleiteten Wärmestrom $\dot{Q}_{\dot{o}l}$ hat neben der zu berechnenden Lamellentemperatur T_K auch die Temperatur des vorbeiströmenden Öls $T_{\dot{o}l}$ (Ölsumpftemperatur) und die an der Wärmeübertragung beteiligte Fläche A_W . Die Wärmeübergangszahl α_K wird von der Fluidgeschwindigkeit maßgeblich beeinflusst.

$$\dot{Q}_{\dot{o}l} = \alpha_K A_W (T_K - T_{\dot{o}l}) \quad 7.4$$

Die Fluidgeschwindigkeit hängt vom Ölvolumenstrom $q_{\dot{o}l}$ und der Absolutdrehzahl der Kupplung (Getriebeeingangsdrehzahl) n_{in} ab. Die Umgebungstemperatur des Fahrzeugs beeinflusst wiederum die Öltemperatur. Aus der Korrelation zwischen Differenzdrehzahl und Getriebeeingangsdrehzahl kann ein Rückschluss auf den Ist-Gang des Fahrzeugs geschlossen werden. Der an das Gehäuse abgeleitete Wärmestrom kann durch keine zusätzliche Sensorgröße abgebildet werden. Die Größen Ölsumpftemperatur $T_{\dot{o}l}$, Getriebeeingangsdrehzahl n_{in} können ebenfalls über Sensoren gemessen werden. Der Ölvolumenstrom $q_{\dot{o}l}$ wird aus Kennfeldern der Bilanz Pumpenfördermenge, Schmierölverteilung bei den entsprechenden Schmierdrücken und der Leckage berechnet.

Somit ergeben sich als Eingangsgrößen für das KNN die Differenzdrehzahl der Kupplung Δn , die Getriebeeingangsdrehzahl n_{in} , der Anpressdruck p , der Ölvolumenstrom $q_{öl}$ und die Ölsumpftemperatur $T_{öl}$ (vgl. Abbildung 7.5). Während der Schlupfphase der Kupplung kann eine Unabhängigkeit dieser Größen voneinander angenommen werden. Die Architektur des KNN in Abbildung 7.5 ist nur beispielhaft dargestellt.

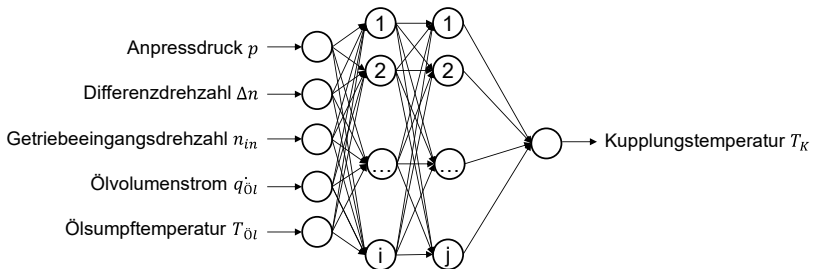


Abbildung 7.5: Eingangs- und Ausgangsgrößen des KNN

7.1.4 Datenaufbereitung

Die Modelleingangsgrößen sowie die Lamellentemperatur werden alle direkt über Drehzahl-, Druck- oder Temperatursensoren im Fahrzeug gemessen und sind aus Fahrzeugmessdaten vorhanden. Der Ölvolumenstrom wird in der Getriebe-Software berechnet.

Die Datenaufbereitung unterteilt sich in fünf Schritte:

- Extrahieren der relevanten Signale aus den Messdaten,
- Zusammenfügen zu einem Gesamtdatensatz,
- Prüfen des Datensatzes auf fehlende Werte und Ausreißer,
- Anpassen der zeitlichen Auflösung und
- Skalieren der Daten.

Im ersten Schritt erfolgt die Extraktion der relevanten Signale aus dem Gesamtdatensatz aller Messungen. Die im MF4-Format vorliegenden Messdaten werden mittels der Python-Bibliothek `asammdf` importiert und auf die relevanten Signale hin gefiltert. Anschließend werden die Signale zu einem Gesamtdatensatz zusammengefügt.

Um eine hohe Datenqualität sicherzustellen, werden die Messdaten auf fehlende Werte und Ausreißer geprüft. Eine Analyse ergibt, dass keine fehlenden Werte und keine Ausreißer vorhanden sind. Dies bestätigt die Annahme aus Kapitel 6.3.4, dass Signale aus dem Getriebesteuergerät eine hohe Datenqualität besitzen.

Die zeitliche Auflösung der Signale beträgt 5ms. Da ein rekurrentes neuronales Netz verwendet wird, welches vergangene Zeitschritte zur Temperaturberechnung nutzt, und gleichzeitig eine Größenhistorie von mehreren Sekunden aufgrund der Systemdynamik sinnvoll erscheint, müssten bei einer Auflösung von 5ms für die Berücksichtigung einer Sekunde 200 Zeitschritte genutzt werden. Dies würde zu einem großen Netz und hohen Trainings- und Rechenzeiten führen, weshalb die Auflösung der Messgrößen reduziert werden muss. Die zeitliche Auflösung des Modells wird deshalb auf die maximale Auflösung von 200ms anhand der Systemanforderungen festgelegt.

Anschließend wird der Datensatz gefiltert. Der Fehler zwischen den Messdaten mit einer Auflösung von 5ms und gefilterten 200ms liegt bei deutlich unter einem Prozent. Die Reduzierung der Auflösung führt somit zu keinem Informationsverlust. Abschließend werden die Messdaten skaliert. Hierfür wird der RobustScaler genutzt. Dieser hat den Vorteil, dass große Werte in den Daten nicht in die Berechnung des empirischen Mittels und die Standardabweichung eingehen, und somit eine homogenere Datenverteilung erzielt wird.

7.1.5 Modellerstellung

Training. Das Training des KNN, die Validierung mit dem Testdatensatz und die Optimierung der Netzwerk- und Hyperparameter bilden den iterativen Prozess der Modellentwicklung. Das Abbruchkriterium des Trainings ist so definiert, dass nach fünf nachfolgenden Trainingsepochen ohne Verbesserung des Validierungsfehlers das Training beendet wird.

Abgleich mit Validierungsdatensatz. Nach dem Training wird das KNN auf den Validierungsdatensatz angewendet und der RSME zur Temperaturmessung berechnet. Anhand dieser Größe werden die Leistungsfähigkeit des KNN bewertet und die optimalen Parameter ermittelt.

Hyperparameteroptimierung. Einige Parameter können aufgrund der guten Ergebnisse aus der Literatur für viele Anwendungsfälle übernommen werden. Als Neuronentyp werden LSTM verwendet, als Optimierungsalgorithmus der Backpropagation-Ansatzes Adam und als Metrik zur Fehlerbestimmung zum Validierungsdatensatz der RMSE (vgl. Kapitel 2.3.1).

Die zu optimierenden Hyperparameter sind die Aktivierungsfunktion, die Netzwerkarchitektur des KNN, die Sequenzlänge und die Stapelgröße. Für die Netzwerkarchitektur werden mindestens zwei bis maximal drei Schichten verwendet mit jeweils bis zu 8 Neuronen. Die möglichen Parameterkombinationen sind in Tabelle 7.2 dargestellt.

Tabelle 7.2 Hyperparameterraum

| Parameter | Parameterraum | ausgewählter Wert |
|---------------------------|---------------------------|-------------------|
| Aktivierungsfunktion | [tanh, relu, sigmoid] | tanh |
| Anzahl Neuronen Schicht 1 | [4, 8] | 4 |
| Anzahl Neuronen Schicht 2 | [4, 8] | 4 |
| Anzahl Neuronen Schicht 3 | [0, 4, 8] | 0 |
| Sequenzlänge | [5, 10, 15, 20, 25, 30] | 30 |
| Stapelgröße | [64, 128, 256, 512, 1024] | 512 |

Insgesamt ergeben sich 1.080 Parameterkombinationen. Eine so große Anzahl an möglichen Parameterkombinationen führt bei einer Berechnung jeder Variante (gridSearch) zu einer hohen Berechnungsdauer. Deshalb wird eine Random-Search verwendet. Es werden 100 Parameterkombinationen zufällig erzeugt und trainiert. Die Python-Bibliothek ray-tune ermöglicht eine einfache Implementierung der Optimierung.

Abbildung 7.6 zeigt die Verteilung der normierten Modellfehler für jede Parameterkombination der Hyperparameteroptimierung. Die gewählten Netzwerkarchitekturen mit weniger als 1000 Modellparametern haben nur geringen Einfluss auf den Modellfehler. Dagegen zeigt sich, dass eine zunehmende Anzahl an Zeitschritten den Modellfehler deutlich verringert. Modelle, die 20, 25 und 30 vorangehende Zeitschritte für die Vorhersage nutzen, können den Modellfehler im Vergleich zu Modellen, die nur 5 Zeitschritte berücksichtigen um über 20% verringern.

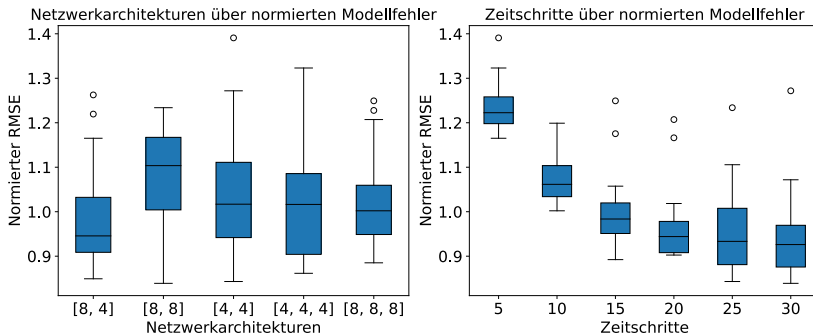


Abbildung 7.6: Ergebnisse der Hyperparameteroptimierung. Die Netzwerkarchitektur hat nur geringen Einfluss auf den Modellfehler, mit zunehmender Anzahl an Zeitschritten verringert sich der Modellfehler deutlich.

Die besten Modellparameter der Optimierung sind die Aktivierungsfunktion Tangenshyperbolicus, ein Netzwerk mit zwei Schichten mit 4 und 4 Knoten, eine Stapelgröße von 512 und eine Sequenzlänge von 30. Dieses Netzwerk besitzt 309 Parameter und liegt somit deutlich unter der Parameteranzahl des Referenzmodells von etwa 1000.

7.1.6 Erklärbarkeitsansätze

Nachfolgend werden eine Sensitivitätsanalyse und der LIME-Ansatz angewendet, um Verständnis über das Modellverhalten zu generieren, was wiederum zur Modelloptimierung genutzt werden kann.

Sensitivitätsanalyse. Für die Sensitivitätsanalyse werden die Eingangsgrößen des Testdatensatzes nacheinander mit Faktoren 0,8, 0,9, 1,1 und 1,2 multipliziert, während die anderen Eingangsgrößen konstant gehalten werden und dafür eine Vorhersage des KNN berechnet wird. Wird z.B. die Differenzdrehzahl mit Faktor 1,1 multipliziert, so ist zu erwarten, dass auch die Temperaturvorhersage größer ist als die Vorhersage mit der tatsächlichen Differenzdrehzahl. Dies wird für alle fünf Eingangsgrößen durchgeführt. Für die Differenzdrehzahl kann die Temperaturvorhersage direkt mit der Berechnung des Referenzmodells verglichen werden, um eine Vergleichbarkeit des Einflusses zu ermöglichen. Die anderen Eingangsgrößen gehen nicht direkt in das Referenzmodell ein, weshalb eine direkte Vergleichbarkeit nicht gegeben ist und deshalb nur die qualitative Veränderung für das KNN betrachtet wird.

Für die Differenzdrehzahl ergibt sich ein nachvollziehbares Verhalten. Wird die Differenzdrehzahl erhöht, so steigt die vorhergesagte Temperatur, wird sie reduziert, so sinkt die vorhergesagte Temperatur. Hierbei ist ein lineares Verhalten erkennbar. Wird der veränderte Testdatensatz für das Referenzmodell verwendet, so zeigt sich ein ähnliches Verhalten (vgl. Abbildung 7.7).

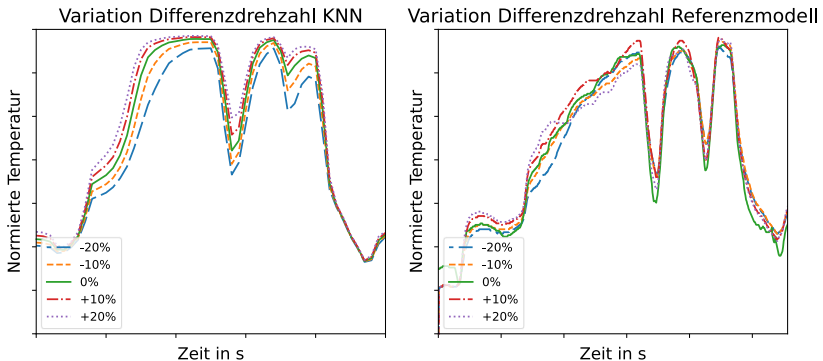


Abbildung 7.7: Ausschnitt der Sensitivitätsanalyse der Differenzdrehzahl. Die vorhergesagte Temperatur verhält sich proportional zur Differenzdrehzahl, dies entspricht ungefähr dem Verhalten des Referenzmodells.

Die Variation einer Eingangsgröße um bis zu 20 % führt auch für den Anpressdruck, die Getriebeeingangs-drehzahl und den Kühlöl-volumenstrom zu einem aus physikalischer Sicht nachvollziehbarem Verhalten des KNN (vgl. Abbildung 7.8). Der Einfluss der Ölumpftemperatur ist im gezeigten Ausschnitt sehr gering. Dies liegt daran, dass die prozentuale Veränderung der Ölumpftemperatur nur zu einer geringen absoluten Temperaturänderung führt, die somit nur einen sehr geringen Einfluss auf die Kuplungstemperatur hat.

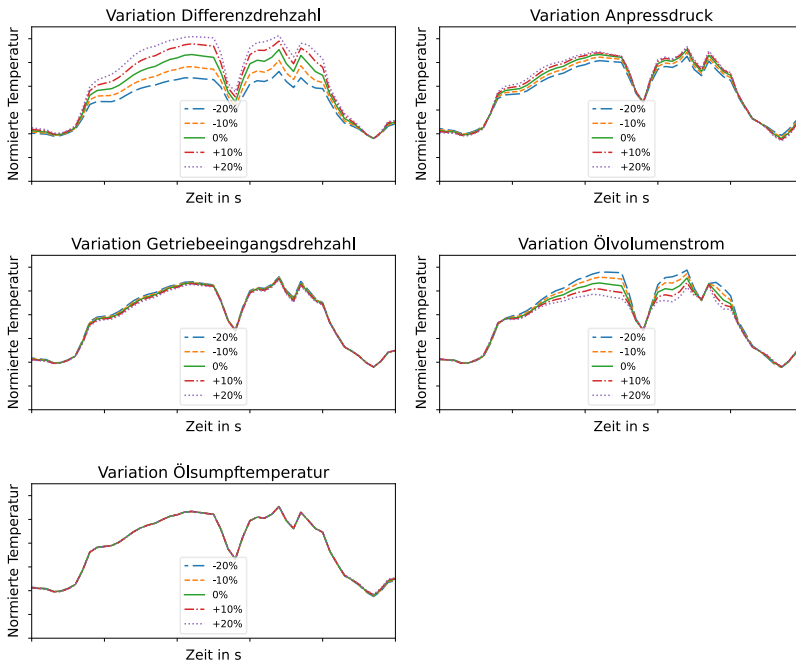


Abbildung 7.8: Ausschnitt der Sensitivitätsanalyse des KNN für alle fünf Eingangsgrößen. Das Modellverhalten ist bei einer Variation der Eingangsgrößen um bis zu 20 % nachvollziehbar.

Aus der Sensitivitätsanalyse lässt sich schlussfolgern, dass das KNN das physikalische Verhalten nachvollziehbar abbildet. Für die Variation einzelner Eingangsgrößen des Testdatensatzes von -20 % bis +20 % liefert das KNN plausible Ergebnisse.

LIME-Analyse. Die bereits in Kapitel 2.3.1 beschriebene LIME-Analyse erstellt ein lokales Ersatzmodell des KNN zu einem definierten Zeitpunkt, um den Einfluss der Eingangsgrößen auf die Vorhersage erklärbar zu machen. Der LIME-Ansatz wird auf den Testdatensatz angewendet. Für jeden fünften Zeitschritt wird ein Ersatzmodell berechnet. Da das Modell fünf Eingangsgrößen besitzt und 30 vergangenen Zeitschritte genutzt werden, ergeben sich insgesamt 150 Eingangsvariablen. Um eine Übersichtlichkeit zu gewährleisten, wird nach der Berechnung der LIME-Matrix jeweils der Einfluss der Zeitschritt t-0 bis t-14 und t-15 bis t-29 aufaddiert. Dadurch wird die Darstellung der Eingangsgrößen auf zehn reduziert. Abbildung 7.9 zeigt das Ergebnis der LIME-Analyse.

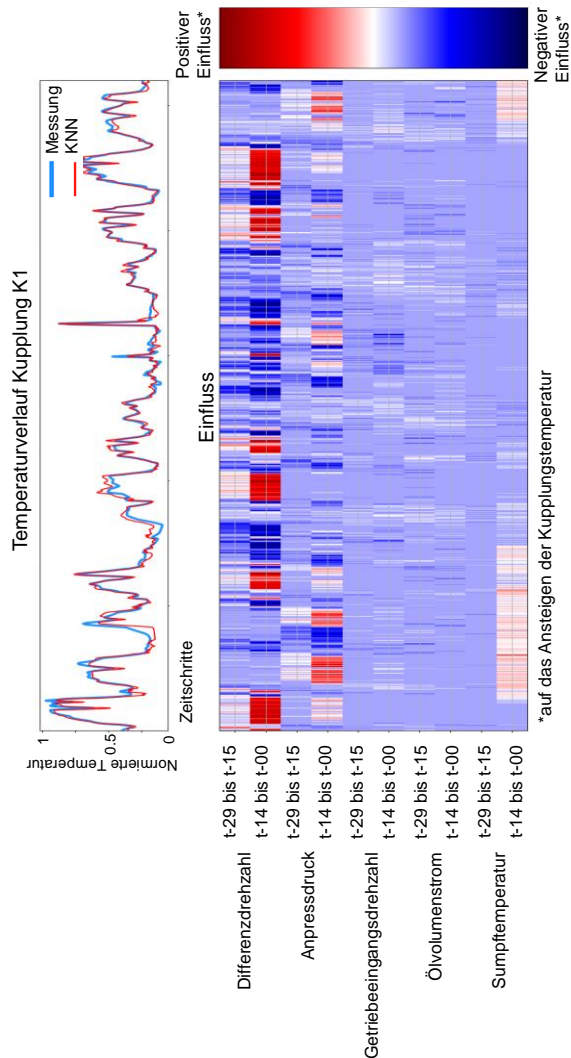


Abbildung 7.9: Die LIME-Analyse zeigt den Einfluss der Modelleingangsgrößen auf die Kupplungstemperatur für jeden Zeitschritt. Rot bedeutet, dass eine Zunahme der Größe zu einem Ansteigen der Temperatur führt, dunkles Blau entspricht analog einer Reduzierung der Temperatur.

Der Temperaturverlauf ist oberhalb dargestellt, um einen Zusammenhang mit diesem erkennbar zu machen. Der obere rote Wert der Farbskala beschreibt einen positiven Temperatureinfluss der Eingangsvariable bei deren Zunahme, der untere blaue Wert einen negativen Temperatureinfluss der Eingangsvariable bei deren Abnahme.

Betrachtet man das Aufheizen bei hybrider Fahrt, so haben alle Zeitschritte der Differenzdrehzahl und die Zeitschritte $t-14$ bis $t-0$ des Anpressdrucks einen positiven Einfluss auf die Temperatur.

Betrachtet man das Aufheizen bei elektrischer Fahrt, so haben hauptsächlich die Zeitschritte $t-14$ bis $t-0$ des Anpressdrucks einen positiven Einfluss auf die Temperatur.

Beim Abkühlen der Kupplung hat der fehlende Anpressdruck aller Zeitschritte einen negativen Einfluss auf die Temperatur. Ölvolumenstrom und Ölumpftemperatur haben ebenfalls einen negativen Einfluss.

Zusammenfassend zeigt das Diagramm, dass Differenzdrehzahl, Anpressdruck und Getriebeeingangsdrehzahl bei den Aufheizvorgängen einen positiven Beitrag zur Temperatur liefern. Ölvolumenstrom und Getriebeeingangsdrehzahl haben den stärksten Einfluss bei Abkühlvorgängen. Nach einigen Anfahrvorgängen steigt die Ölumpftemperatur, weshalb diese gestiegene Temperatur zum weiteren Ansteigen der Kupplungstemperatur positiv beiträgt, da dann weniger Wärme abgeführt werden kann. Außerdem führen kleine Werte des Anpressdrucks und damit resultierender geringer Reibleistung zu einer Abkühlung. Dieses Verhalten ist aus physikalischer Sicht nachvollziehbar.

Das KNN lernt die Unterscheidung von hybridem Anfahren und elektrischem Anfahren durch Verwendung des Anpressdrucks, die Differenzdrehzahl wird dabei vernachlässigt.

Bei der Auswertung des LIME-Ansatzes ist zu beachten, dass die Bildung eines Ersatzmodells zu Abweichungen im Vergleich zum KNN führt. Außerdem entsteht durch die Berechnung für Zeitschritte mit einer Länge von einer Sekunde sowie das Aufaddieren der Einflüsse der Zeitschritte ebenfalls ein Fehler. Dadurch ist eine quantitative Auswertung nicht möglich, weshalb in diesem Beispiel nur eine qualitative Betrachtung durchgeführt wird.

7.1.7 Modellvalidierung

Mit den ermittelten Parametern werden zehn Modelle trainiert und auf den Testdatensatz angewendet. Der Mittelwert des normierten RMSE ist um 15,5 %

geringer als das bisher verwendete Referenzmodell. Im Einzelfall wird eine Verbesserung um 21,8 % erzielt. Die Standardabweichung liegt bei 0,044.

Abbildung 7.10 zeigt die Temperaturverläufe der Messung, des KNN und des Referenzmodells für das Fahrmanöver „Anfahren 30 %“ aus dem Testdatensatz. Die Abweichungen zwischen Messung und KNN sowie zwischen Messung und Referenzmodell sind farblich hervorgehoben.

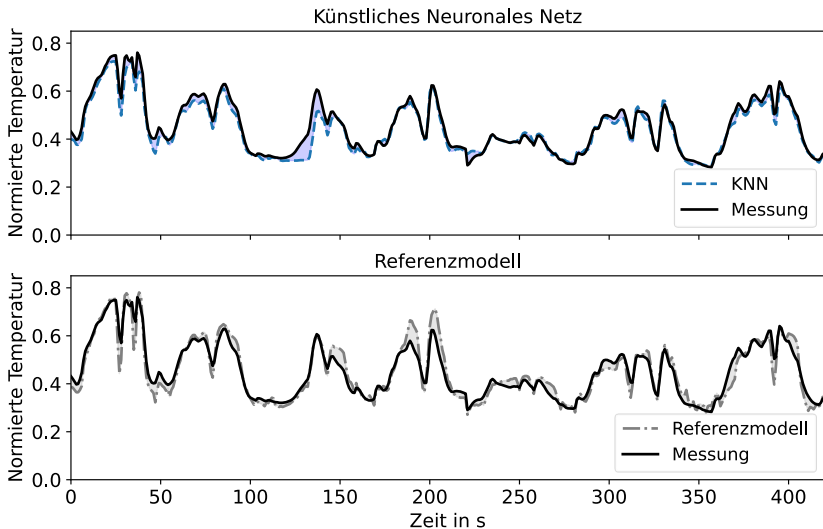


Abbildung 7.10: Vergleich der Kupplungstemperaturmodelle zu realen Messdaten für Fahrmanöver „Anfahren 30 %“ und „Anfahren 20 %“. Das KNN erzielt eine höhere Modellgüte wie das Referenzmodell.

Die Vorhersage des KNN bildet den zeitliche Temperaturverlauf der Messungen gut ab. Jedoch kommt es teilweise zum Unterschreiten der Temperatur speziell bei den elektrischen Anfahren. Das Referenzmodell hingegen liegt größtenteils über der gemessenen Temperatur.

Betrachtet man den kompletten Testdatensatz, so kann das KNN die Temperatur für das Fahrmanöver „Anfahren 20 %“ sehr gut vorhersagen, der RSME ist um 45% geringer als das Referenzmodell. Das Fahrmanöver „8-4 MFR“ wird ebenfalls durch das KNN mit einem 60% geringeren Fehler abgebildet. Im Fahrmanöver „Anfahren 15 %“ überschätzt das KNN die gemessene Temperatur.

7.1.8 Anwendung auf Trennkupplung K0

Die in Kapitel 6 beschriebene Vorgehensweise zur Modellierung eines KNN wird im Folgenden auf die Trennkupplung K0 des Hybridgetriebes angewendet.

Der RMSE wird als Bewertungsgröße herangezogen. Als Eingangsgrößen für das Modell wurden zunächst die Differenzdrehzahl, der Anpressdruck, die Getriebeeingangsdrehzahl, der Ölvolumenstrom und die Ölsumpftemperatur verwendet. Anhand der LIME-Analyse konnte festgestellt werden, dass die Getriebeeingangsdrehzahl nur sehr geringen Einfluss auf die Kupplungstemperatur hat, weshalb diese nachfolgend nicht mehr verwendet wird. Als Trainings- und Testdaten werden Fahrzeugmessungen verwendet, die Zustarts im Fahrbetrieb sowie wiederholte aktiv provozierte Volllastzustarts am Hang beinhalten. Aufgrund der Homogenität der Fahrmanöver werden je 15 % des gesamten Datensatzes als Validierungs- und Testdatensatz verwendet. Die Datensätze werden analog nach Abschnitt 7.1.4 aufbereitet sowie die Modell- und Trainingsparameter optimiert. Die optimale Netzwerkstruktur sind zwei Schichten mit 8 und 4 Knoten, was in einer Modellgröße von 661 Parameter resultiert. Abbildung 7.11 zeigt die Temperaturverläufe der Messung, des KNN und des Referenzmodells für den Testdatensatz.

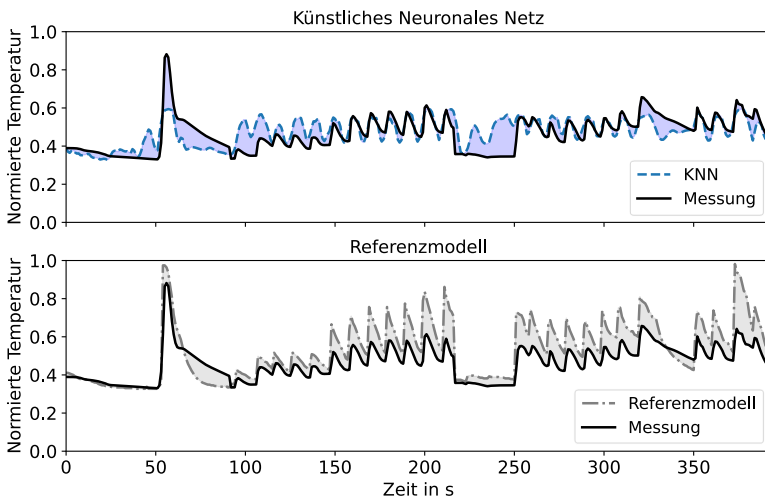


Abbildung 7.11: Vergleich der Kupplungstemperaturmodelle zu realen Messdaten. Das KNN erzielt eine höhere Modellgüte wie das Referenzmodell.

Grundsätzlich ist festzuhalten, dass das aktuell verwendete Referenzmodell die Kupplungstemperatur deutlich überschätzt. Die Verwendung des KNN kann den RMSE im Vergleich zum Referenzmodell im Mittelwert über 10 Modelle um 22,5 % und im Einzelfall um bis zu 28,5 % reduzieren. Die Standardabweichung liegt bei 0,06. Das KNN unterschätzt die hohe Temperaturspitze deutlich. Dies ist darauf zurückzuführen, dass dieses Fahrmanöver nur einmal im Trainingsdatensatz vorhanden ist.

7.1.9 Anwendung auf Kupplung K1 konventionelles Getriebe

Die in Kapitel 6 beschriebene Vorgehensweise zur Modellierung eines KNN wird im Folgenden auf die Trennkupplung K1 im konventionellen Getriebe angewendet. Der RMSE wird als Bewertungsgröße herangezogen. Als Eingangsgrößen für das Modell werden die Differenzdrehzahl, der Ölvolumenstrom und die Ölsumpftemperatur verwendet. Als Trainings- und Testdaten werden analog zur Kupplung K1 im Hybridgetriebe Fahrzeugmessungen der Fahrmanöver nach Tabelle 7.3 verwendet.

Tabelle 7.3 Fahrmanöver Trainingsdatensatz K1 konventionelles Getriebe

| | Steigung | Fahrmodus | Fahrrichtung | Häufigkeit |
|---|-----------------|-----------------------|---------------------|-------------------|
| Anfahren | 20% | verbrennungsmotorisch | vorwärts | 1x |
| | 20% | verbrennungsmotorisch | rückwärts | 1x |
| | 30% | verbrennungsmotorisch | vorwärts | 2x |
| | 30% | verbrennungsmotorisch | rückwärts | 4x |
| Mehrfachrückschaltung 8-4 mit Gangeinlegefehler | | | | 3x |

Der Validierungs- und Testdatensatz besteht je aus den drei Lastfällen „Anfahren 30 %“ vorwärts und rückwärts sowie einer 8-4 MFR unterschiedlicher Messungen. Die Datensätze werden analog nach Abschnitt 7.1.4 aufbereitet sowie die Modell- und Trainingsparameter optimiert. Die optimale Netzwerkstruktur sind zwei Schichten mit 8 und 4 Knoten, was in einer Modellgröße von 661 Parameter resultiert.

Abbildung 7.12 zeigt die Temperaturverläufe der Messung, des KNN und des Referenzmodells für den Testdatensatz. Für den gesamten Testdatensatz reduziert das KNN den RMSE im Vergleich zum Referenzmodell über 10 Modelle um 18,5 % und im Einzelfall um bis zu 33,1 %. Die Standardabweichung liegt bei 0,16.

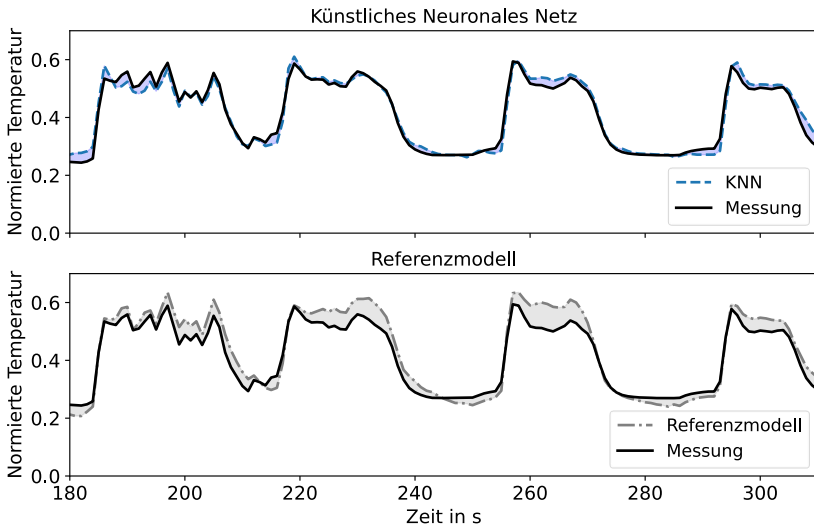


Abbildung 7.12: Vergleich der Kupplungstemperaturmodelle zu realen Messdaten für das Fahrmanöver „30 % Anfahren“. Das KNN erzielt eine höhere Modellgüte als das Referenzmodell.

7.2 Kupplungsfüllung

Das in Kapitel 5.2.1 vorgestellte elektrifizierte Doppelkupplungsgetriebe besitzt drei nasslaufende Mehrscheiben-Lamellenkupplungen. Vor der Betätigung einer Kupplung findet eine Kupplungsfüllung statt. Dabei wird der Raum hinter dem Kolben mit Öl gefüllt. Ziel ist es, den Kolben zu verschieben, bis dieser den Touch-Point der Kupplung erreicht (vgl. Abbildung 7.13). Am Touch-Point liegen die Lamellen der Kupplung an, es wird jedoch noch kein Drehmoment übertragen.

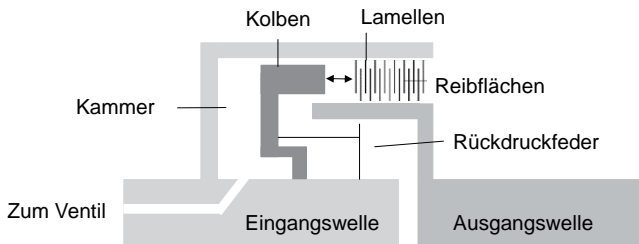


Abbildung 7.13: Schematische Darstellung Lamellenkupplung (Eigene Darstellung nach Pinte, Depraetere, Symens, Swevers & Sas, 2010).

Es existiert ein Zielkonflikt zwischen einer geringen Füllzeit, um unter anderem Schaltzeiten zu reduzieren, und einem möglichst geringen Systemdruckeinbruch, um andere Hydraulikverbraucher nicht zu beeinflussen. In einem konventionellen Antriebsstrang wird dabei eine Hydraulikpumpe eingesetzt, die mechanisch über einen Kettentrieb mit dem VM verbunden ist. Dadurch ist die Förderleistung motordrehzahlabhängig. In elektrifizierten Antriebssystemen muss je nach Anbindung eine elektrische Pumpe verwendet werden, um auch beim rein elektrischen Fahrbetrieb die Energieversorgung der Pumpe sicherzustellen. Außerdem bietet sie den Vorteil, dass sie unabhängig von der Motordrehzahl betrieben werden kann und ihr Energieverbrauch geringer als bei herkömmlichen Pumpen ist.

Um diesen geringen Energieverbrauch zu ermöglichen, stellt die Pumpe zu jeder Zeit einen möglichst geringen Ölvolumenstrom. Durch die Begrenzung des Ölvolumenstroms wird die Robustheit des Systems gegen Druckeinbrüche gemindert. Dadurch ist es für Entwickelnde schwierig für dieses Hydrauliksystem eine geeignete Ansteuerung zu identifizieren. Viele Versuche sind hier notwendig.

Erschwerend kommt hinzu, dass das Hydrauliksystem, welches alle drei Kupplungen ansteuert, aus Kostengründen kein Arbeitsdruckregelventil besitzt. Mit

Hilfe einer Druckwaage wird von der Software der Arbeitsdruck derjenigen Kupplung bestimmt, welche den höchsten Druck anfordert. Ein zusätzlicher Druckoffset wird dabei auf den bestimmenden Kupplungsdruck addiert. Im hybriden Fahrmodus kann durch das Überanpressen der K0 der Systemdruck eingestellt werden. Im elektrischen Fahrmodus hingegen wird der Systemdruck rein von der belegten Fahrkupplung bestimmt. Somit gibt es für eine Kupplungsfüllung nicht nur einen Systemdruck, bei dem die Füllung stattfindet, sondern ein breites Druckband. Für jeden Systemdruck muss deshalb eine passende Ansteuerung entwickelt werden. Damit steigt der Arbeitsaufwand um ein Vielfaches.

Zusätzlich besitzt das Hydrauliksystem und das daraus resultierende Hydraulikverhalten für das PE-Team einen hohen Neuheitsgrad. Erfahrungswerte aus vorhergehenden Systemgenerationen können nur teilweise übertragen werden. Der Fokus der Optimierung liegt auf dem ersten Teil der Füllung, bei dem der Druckeinbruch zustande kommt. Deshalb wird das Über- oder Unterfüllen der Kupplung vorerst nicht betrachtet.

Basierend auf der Methode aus Kap 6.4 ist das Vorgehen für den Anwendungsfall Kupplungsfüllung in Abbildung 7.14 dargestellt.

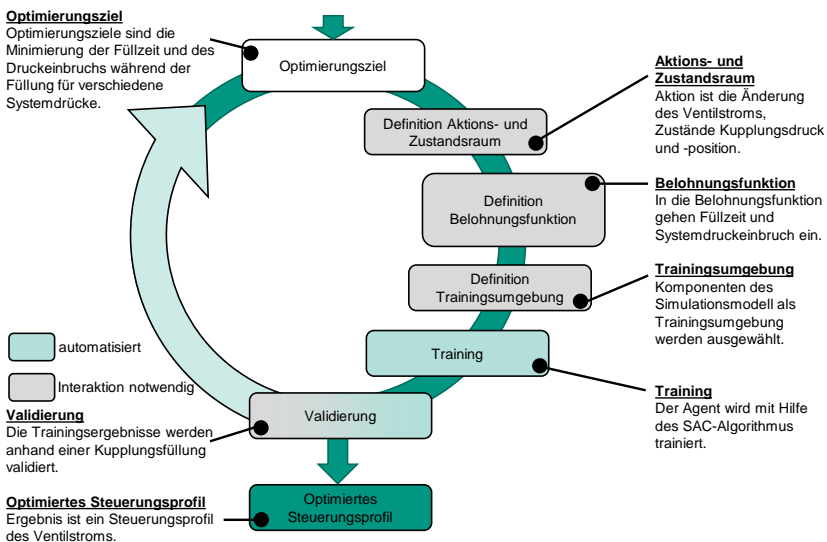


Abbildung 7.14: Vorgehen Steuerungs-Optimierung mit verstärkendem Lernen angewendet auf eine Kupplungsfüllung

7.2.1 Optimierungsziel

Der hier beschriebene Anwendungsfall bezieht sich auf das in Kapitel 5.2.1 vorgestellte elektrifizierte Doppelkupplungsgetriebe in P2-Anordnung. Das Vorgehen wird anhand der Füllung der Kupplung K2 beschrieben.

Für das Training wird folgendes Fahrmanöver definiert. Das Fahrzeug ist im rein elektrischen Fahrbetrieb. Die Kupplung K0 ist dabei geöffnet. Kupplung K1 ist unter Last, während die Kupplung K2 gefüllt wird. Die Optimierung erfolgt für eine konstante Öltemperatur und einem Ölluftgehalt von 3 %. Durch die Last der K1 stellt sich ein Systemdruck p_{sys} ein. Für diesen Zustand werden ein erstes Training durchgeführt und die Ergebnisse gezeigt. Anschließend wird die Optimierung für den Druckbereich von $0,5 * p_{sys}$ und $1,33 * p_{sys}$ durchgeführt. Die Hydraulikpumpe wird auf eine vordefinierte Drehzahl eingestellt. Der erste Ventilstrom durch den Agenten wird nach Erreichen der maximalen Pumpendrehzahl gestellt. Für die Validierung wird das gleiche Fahrmanöver genutzt, jedoch ohne Explorationsanteil des Agenten.

Als Optimierungsziel werden die Füllzeit und der Druckeinbruch minimiert und sollen dabei unterhalb eines Maximalwertes liegen. Die erreichten Werte werden dabei auf diese Maximalwerte normiert. Der maximal zulässige Druckeinbruch liegt bei $0,1 * p_{sys}$.

7.2.2 Definition Aktions- und Zustandsraum

Der Zustand s_t der Kupplungsfüllung wird über Kupplungsdruck p_{K2} und die berechnete Kupplungsposition pos_{K2} beschrieben.

$$s_t = [p_{K2}, \quad pos_{K2}] \quad 7.5$$

Die Aktion des Agenten a_t ist die Ansteuerung des Ventilstroms der Kupplung K2 $c_{Ventil\ K2}$.

$$a_t = [c_{Ventil\ K2}] \quad 7.6$$

Erreicht die Kupplung die berechnete Soll-Position, so wird der Eingriff des Agenten beendet und der Ventilstrom wird auf einen konstanten Wert vorgegeben.

7.2.3 Definition Belohnungsfunktion und Algorithmus

Um eine Minimierung der Zielgrößen in einer maximierten Belohnungsfunktion darzustellen, wird ein negatives Vorzeichen eingeführt, sodass eine Minimierung der Terme eine Maximierung der Belohnung zur Folge hat. Die Zielkriterien einer geringen Füllzeit und eines geringen Systemdruckeinbruchs fließen direkt ein.

Um die Füllzeit in der Belohnungsfunktion zu berücksichtigen, wird die Füllzeit $t_{Füll}$ als quadratischer Term eingeführt. Um den Systemdruckeinbruch zu minimieren, wird die quadratische Abweichung zum Sollsystemdruck $(p_{SysSoll} - p_{Sysist})^2$ verwendet.

Zusätzlich werden die beiden Terme auf den jeweiligen Zielwert normiert und anschließend gewichtet. Druckeinbruch und Füllzeit werden mit den Gewichtungsfaktoren c_1 und c_2 von je 0,5 gleich gewichtet.

$$r = -\left(\frac{c_1}{t_{Füllmax}^2} * (t_{Füll})^2 + \frac{c_2}{p_{Sysmax}^2} * (p_{SysSoll} - p_{Sysist})^2\right) \quad 7.7$$

Als Randbedingungen soll eine Lösung einen maximalen Systemdruckeinbruch p_{Sysmax} und eine maximale Füllzeit $t_{Füllmax}$ nicht übersteigen. Diese Randbedingungen werden über Bestrafungs-Terme berücksichtigt. Liegt die Füllzeit über dem Maximalwert, beträgt die Bestrafung -1 pro Zeitschritt. Übersteigt der Systemdruckeinbruch den Maximalwert, wird ebenfalls eine Bestrafung von -1 auf die Belohnungsfunktion addiert.

7.2.4 Definition Trainingsumgebung

Um ein schnelles und kostengünstiges Training des Agenten zu ermöglichen, ist die Trainingsumgebung, die Kupplungshydraulik, komplett virtuell aufgebaut. Abbildung 7.15 stellt die Validierungsumgebung des Trainingsprozesses anhand des IPEK-XiL-Ansatzes dar. Die Kupplungshydraulik ist das System-under-Investigation. Das System-in-Development ist die Ansteuerung des Magnet-Regel-Ventils der jeweiligen Fahrkupplung als Teilsystem der Betriebsstrategie.

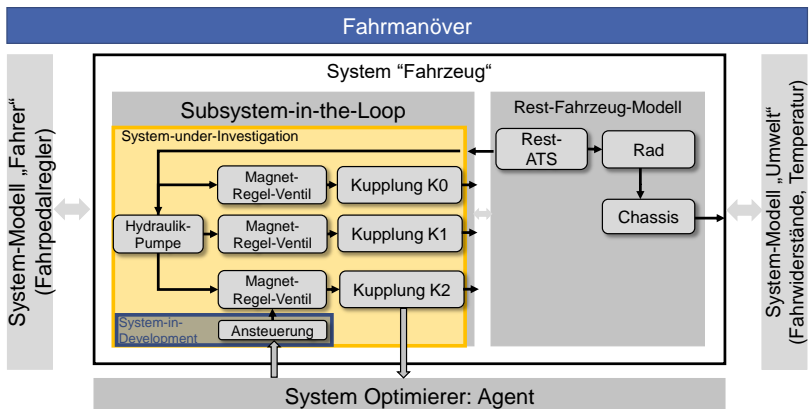


Abbildung 7.15: Virtuelle Validierungsumgebung der Kupplungsfüllung im Trainingsprozess anhand des IPEK-XiL-Ansatzes

Für das Training werden das Antriebsstrang-, Fahrer- und Umgebungsmodell stark vereinfacht dargestellt. Sie stellen lediglich die Initialbedingungen, zu denen die Füllung stattfindet. Aufgrund der geringen Füllzeit $\ll 1s$ kann die Fahrsituation und der Zustand des Gesamtantriebsstrangs während der Füllung als stationär angenommen werden. Es werden deshalb die Fahrzeuggeschwindigkeit, Gaspedalwinkel, Getriebeeingangsdrehzahl und Fahrmodus als konstant angenommen. Optional kann ein dynamisches Fahrzeugmodell angebunden werden, um z.B. einen Fahrzeugruck bei Systemdruckeinbruch zu bestimmen. Allerdings ist für diese Betrachtung der Systemdruckeinbruch als Bewertungsgröße ausreichend.

Das Hydraulikmodell wurde im Rahmen dieser Arbeit nicht aufgebaut, sondern von der Mercedes-Benz Group AG übernommen. Das Hydrauliksystem besteht aus der elektrischen Pumpe, dem Hydrauliköl, den Hydraulikleitungen, den drei Magnet-Regel-Ventilen der Kupplungen sowie den drei Kupplungen K1, K2, K0 (vgl. Abbildung 7.16). Mit Hilfe einer Druckwaage wird der Arbeitsdruck eingestellt. Der Kolbenweg wird in der Simulation über eine Kupplungskennlinie ermittelt, die die Kolbenkraft über den Kolbenweg abbildet. Dadurch kann der Kupplungsdruck in den Kolbenweg umgerechnet werden. Die Kupplungskennlinie wurde anhand einer FEM-Berechnung erstellt, die Kanalführung mittels CFD-Simulationen und anschließend in ein 1D-Modell umgewandelt. Das Hydraulikmodell ist valide für einen Temperaturbereich des Hydrauliköls von $-30^{\circ}C$ bis $90^{\circ}C$. Es ist außerdem in der Lage Schwingungen im Hydrauliksystem abzubilden. Ergeben sich im Laufe des Entwicklungsprozesses Änderungen der Hardware, so kann das Modell entsprechend neu

parametriert werden. Insgesamt besitzt das Modell eine hohe Abbildungsgüte. In der Validierung am realen System in Kap. 7.2.6 liegt der Fehler für den Druckeinbruch bei einem Systemdruck p_{sys} bei 1,5 % - 6 % des Systemdrucks.

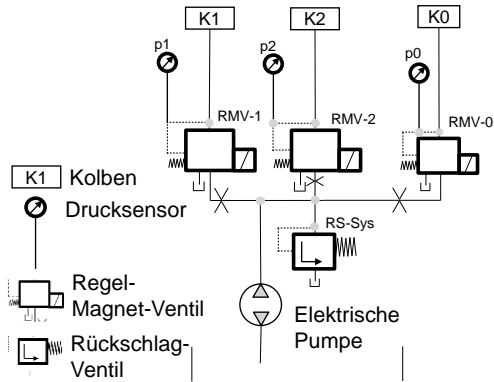


Abbildung 7.16: Hydraulischschaltplan der Getriebehydraulik

Das Simulationsmodell wird in ein Functional Mock-up Unit Format (FMU) konvertiert und mit der Python-Umgebung verknüpft. Die FMU ermöglicht eine einfache Anbindung an die Python-Umgebung sowie eine kurze Simulationszeit. Es wird keine zusätzliche Simulationssoftware benötigt.

Aufgrund des stationären Fahrzustands können direkt die Modelleingangsgrößen an das Kupplungshydraulikmodell aufgetragen werden. Das Soll-Kupplungsmoment der K1 wird über den Kupplungs-Solldruck in einen Soll-Ventilstrom umgerechnet. Die Soll-Drehzahl der Hydraulikpumpe wird ebenfalls vorgegeben. Aufgrund der rein elektrischen Fahrt ist der Ventilstrom der K0 gleich null. Der initiale Systemdruck kann über den Ventilstrom der belasteten Kupplung eingestellt werden.

7.2.5 Training

Als Algorithmus wird der SAC-Algorithmus verwendet. Dieser hat für eine Vielzahl an Steuerungs- und Regelungsproblemen eine sehr gute Performance gezeigt (vgl. Kapitel 2.3.3). Ein weiterer Vorteil des SAC-Algorithmus ist dessen Robustheit gegenüber Änderungen der Hyperparameter. Somit können die Default-Parameter des Algorithmus genutzt werden.

Für das Training wird nur die Füllung simuliert. Dafür wird das Modell zunächst auf den Start-Zustand der Füllung initialisiert. Dieser Zustand wird gespeichert und für jede Trainingsepisode neu geladen.

Im Trainingsverlauf führt der Agent zufällige Aktionen aus und strebt eine Maximierung der Belohnung an. Alle 25 Episoden wird der Agent getestet. Dabei führt er eine Füllung ohne Explorationsanteil aus. Verbessert der Agent die Belohnung aus dem vorherigen Testdurchlauf, so wird dieser Agent abgespeichert. Eine Episode entspricht dabei einer Kupplungsfüllung

Zur Untersuchung der Robustheit des Trainings werden je zehn Agenten für 1.000 Episoden für die drei Systemdrücke trainiert. Nach etwa 625 Episoden konvergieren alle Agenten und damit nach etwa 25 Testepisoden. Für den Systemdruck von p_{sys} wird die durchschnittliche Belohnung für die Testepisoden von anfänglich -440 auf -220 maximiert (vgl. Abbildung 7.17).

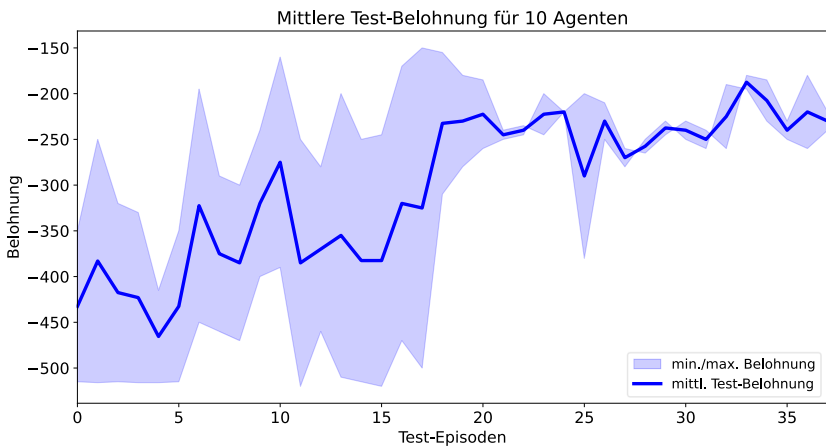


Abbildung 7.17: Verlauf der mittleren Belohnung über Test-Episoden während des Trainingsvorgangs für den Systemdruck p_{sys}

7.2.6 Validierung

Der mit dem verstärkenden Lernen ermittelte Ventilstromverlauf wird mit einer Referenzansteuerung und dem Optimierungsergebnis eines Genetischen Algorithmus verglichen, der vier Füllparameter optimiert (vgl. Abbildung 7.18). Die

Referenzansteuerung entspricht der ersten Applikation eines Entwickelnden und basiert auf Erfahrungswerten.

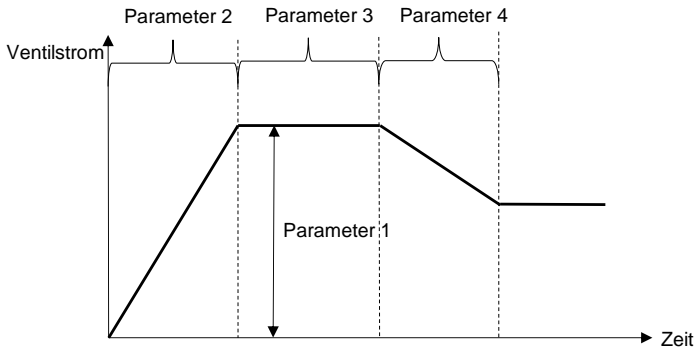


Abbildung 7.18: Parametrisierung des Ventilstroms

Der Genetische Algorithmus wird mithilfe der Python-Bibliothek PyGAD implementiert (Gad, 2021). Dieser verwendet eine Population von 20, 30 Generationen, zwei Nachkommen und einen Mutationsgrad von 10 %. Der Genetische Algorithmus erzeugt im Laufe der Optimierung eine Pareto-Front an Lösungen. Für den Vergleich wird die Lösung mit der maximal zulässigen Füllzeit und dem dabei geringsten Druckeinbruch gewählt.

Abbildung 7.19 zeigt die Optimierungsergebnisse des besten Agenten für einen Systemdruck von p_{sys} im Vergleich zur Referenzansteuerung sowie zum Ergebnis des Genetischen Algorithmus. Der Ventilstrom ist das optimierte Steuersignal. Die Kupplungsposition und der Systemdruck stellen die Optimierungsziele Füllzeit und Systemdruckabfall dar.

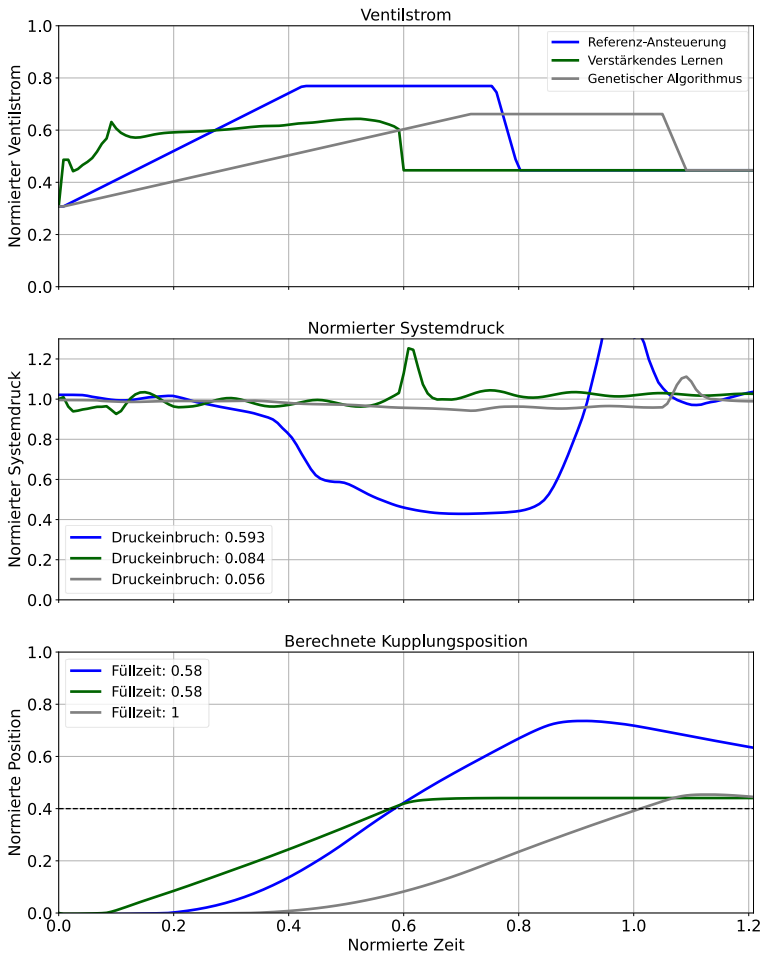


Abbildung 7.19: Vergleich der Referenzansteuerung, Verstärkendes Lernen und Genetischer Algorithmus für eine Füllung der Kupplung K2 bei einem Systemdruck p_{sys} in der Trainingsumgebung

Die optimierte Steuerung des verstärkenden Lernens erzielt die gleiche Füllzeit wie die Referenzansteuerung. Gleichzeitig wird der Systemdruckabfall um 86 % deutlich reduziert. Der Ansatz des verstärkenden Lernens zeigt ein robustes Verhalten. Die zehn Agenten konvergieren auf eine Streuung der Füllzeit von 8 %, des maximalen Druckeinbruchs von 7 %, sowie auf eine Belohnung von -210 bis -190. Der

Genetische Algorithmus erzielt bei einer Füllzeit von 1,01 eine Reduzierung des Druckeinbruchs um 91 %. Gleichzeitig benötigt der Genetische Algorithmus nur 60 Füllungssimulationen, wohingegen das verstärkende Lernen 625 benötigt (vgl. Tabelle 7.4).

Trotz der Parameterfreiheit des verstärkenden Lernens ergibt sich insgesamt ein ähnlicher Verlauf im Vergleich zur parameterbasierten Ansteuerung. Der größte Unterschied ergibt sich in der Stromrampe. Das verstärkende Lernen erhöht den Strom nichtlinear auf das Zielniveau, sondern sofort auf etwa 0,6 des normierten Stromes, senkt diesen kurz wieder ab und erhöht dann auf das Zielniveau. Dadurch wird die Füllzeit reduziert, der Druckeinbruch dagegen geringgehalten.

Tabelle 7.4 Vergleich Simulationsergebnisse Ansteuerung Kupplungsfüllung

| System- druck | Kriterien | Referenz | Verstärken des Lernen | Genetischer Algorithmus |
|------------------|-------------------------------|-------------|--------------------------|----------------------------|
| $0,5 * p_{sys}$ | Norm. Füllzeit | 0,62 | 0,58 | 0,99 |
| | Norm. Druckeinbruch | 0,173 | 0,128 | 0,06 |
| p_{sys} | Norm. Füllzeit | 0,58 | 0,58 | 0,99 |
| | Norm. Druckeinbruch | 0,593 | 0,084 | 0,056 |
| $1,33 * p_{sys}$ | Norm. Füllzeit | 0,64 | 0,58 | 1,01 |
| | Norm. Druckeinbruch | 0,51 | 0,119 | 0,045 |
| | mittl. Anzahl an Simulationen | - | 625 | 60 |

Für die Systemdrücke $0,5 * p_{sys}$ und $1,33 * p_{sys}$ ergibt sich ein ähnliches Verhalten. Die Füllzeit des verstärkenden Lernens liegt bei 0,58 der maximalen Füllzeit, der Druckeinbruch für Systemdrücke von $0,5 * p_{sys}$ und $1,33 * p_{sys}$ bei $0,128 * p_{sys}$ und $0,119 * p_{sys}$. Der Genetische Algorithmus kann den Druckeinbruch jeweils auf etwa $0,05 * p_{sys}$ reduzieren, die Füllzeit liegt dafür beim Maximalwert.

Anschließend werden die Optimierungsergebnisse in einer rein physischen Validierungsumgebung in einem Versuchsfahrzeug validiert. Die Füllung wird dabei im Stand ausgeführt. Die ermittelten Füllparameter werden in der Fahrzeugsoftware implementiert und im Fahrzeug validiert. Der komplexe Ventilstromverlauf des verstärkenden Lernens kann im Steuergerät nicht direkt abgebildet werden. Über die vier Parameter analog zum Genetischen Algorithmus (vgl. Abbildung 7.18) kann allerdings ein sehr ähnlicher Verlauf umgesetzt werden.

Abbildung 7.20 zeigt die Messergebnisse des verstärkenden Lernens und des Genetischen Algorithmus für einen Startsystemdruck von p_{sys} im Vergleich. Im Fahrzeug konnte der Systemdruck mit etwa $0,96 * p_{sys}$ nicht exakt auf p_{sys} eingestellt werden, ist für die Validierung der Simulationsergebnisse allerdings ausreichend. Um einen aussagekräftigen Vergleich zu erhalten, werden jeweils drei Füllungen mit den gleichen Parametern durchgeführt. Die Füllzeit im Fahrzeug wird über das Erreichen des Touch-Point-Drucks bestimmt, da im Fahrzeug keine Kupplungsposition zur Verfügung steht. Die Referenzansteuerung wurde aufgrund des schlechten Ergebnisses in der Simulation im Fahrzeug nicht validiert.

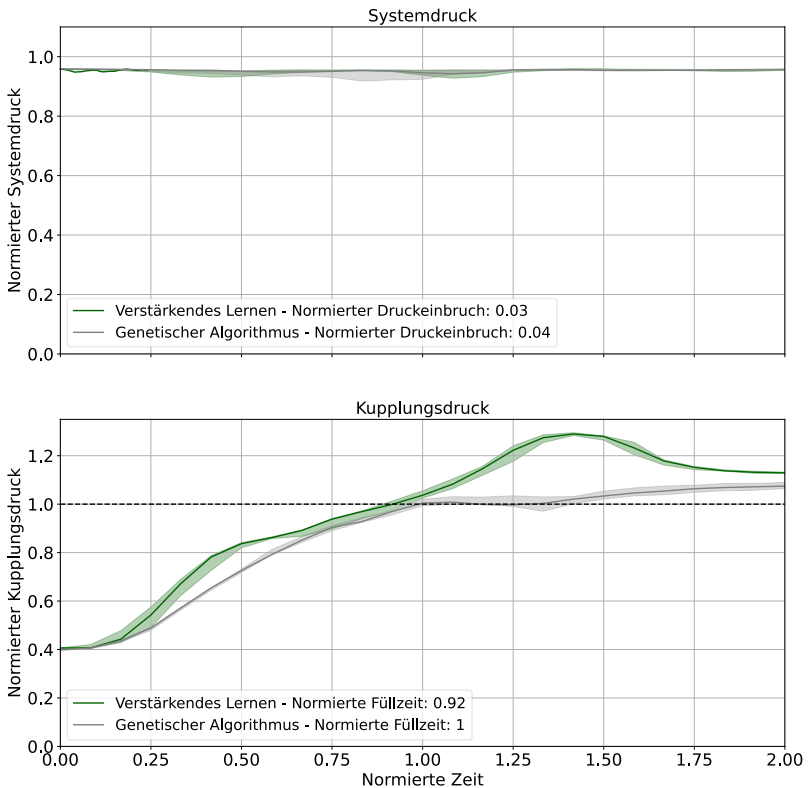


Abbildung 7.20: Fahrzeugmessung des Systemdrucks und der Füllzeit zur Validierung der Optimierungsergebnisse des Verstärkenden Lernens und des Genetischen Algorithmus

Die Lösung des verstärkenden Lernens mit einem Systemdruckeinbruch von $0,031 * p_{sys}$ sowie die Lösung des Genetischen Algorithmus mit $0,04 * p_{sys}$ liegen deutlich unter dem maximal zulässigen Druckeinbruch von $0,1 * p_{sys}$. Die maximale Füllzeit wird von beiden Ergebnissen eingehalten (vgl. Tabelle 7.5). Somit stimmen die Validierungsmessungen gut mit der Simulation überein. Die Systemdruckeinbrüche weichen um 1,5 % bis 6 % des Systemdrucks von der Simulation ab. Die Füllzeiten weichen um 33% ab, was aber ausreichend ist, da dies im niedrigen Millisekundenbereich liegt. Eine Ursache dafür ist die Bestimmung des Füllendes in der Simulation über die Positionsberechnung, wohingegen in der Fahrzeugmessung das Füllende über das Erreichen des Touch-Point-Drucks bestimmt wird. Außerdem wurde die Ansteuerkurve für das verstärkende Lernen zur Applikation im Fahrzeug leicht geglättet. Für die Lösung des verstärkenden Lernens kommt es noch zu einer leichten Überfüllung der Kupplung, was aber zunächst nicht im Vordergrund der Optimierung steht.

Tabelle 7.5 Vergleich Validierungsergebnisse Ansteuerung Kupplungsfüllung

| Systemdruck | Kriterien | Verstärkendes Lernen | Genetischer Algorithmus |
|------------------|---------------------|----------------------|-------------------------|
| $0,5 * p_{sys}$ | Norm. Füllzeit | 0,75 | 0,92 |
| | Norm. Druckeinbruch | 0,029 | 0,018 |
| p_{sys} | Norm. Füllzeit | 0,92 | 1 |
| | Norm. Druckeinbruch | 0,031 | 0,04 |
| $1,33 * p_{sys}$ | Norm. Füllzeit | 0,79 | 1,09 |
| | Norm. Druckeinbruch | 0,038 | 0,021 |

7.3 Hybrider Motorwiederstart

Beim hybriden Motorwiederstart wird der VM beim Übergang vom rein elektrischen zum hybriden Fahren durch das Schließen der Trennkupplung K0 gestartet (vgl. Abbildung 7.22). Diese Funktion ist entscheidend für die Fahrbarkeit, da der Motorwiederstart für den Fahrer unbemerkt stattfinden soll.

Der Validierungsbedarf ergibt sich daraus, optimale Zustarts für ein maximales Kundenerlebnis zu schaffen. Falls unterschiedliche Cluster von Zustarts existieren, sollen diese idealerweise spezifisch appliziert werden. Das daraus abgeleitete Validierungsziel ist es, festzustellen, ob unterschiedliche Zustart-Cluster existieren und wie diese über messbare Größen charakterisiert sind.

Die in Kapitel 6.5 beschriebene Methode wird als datengetriebener Ansatz auf die Entwicklungsaufgabe angewendet, um datengestützte Aussagen zu ermöglichen und daraus Handlungsempfehlungen für die Entwicklung abzuleiten (vgl. Abbildung 7.21). Dabei wird das Potential der Methode aufgezeigt, Wissen aus Messdaten zu generieren.

Basierend auf dem Validierungsziel werden Hypothesen abgeleitet, die anhand von messbaren Kriterien mit Hilfe der Datenanalyse überprüft werden.

Hypothese 1: *Es existieren bisher nicht bekannte Korrelationen in den Messdaten.*

Kriterien: *Mindestens eine unbekannte Korrelation zwischen relevanten Messsignalen des Motorzustart wird in den Messdaten erkannt.*

Hypothese 2: *Es existieren verschiedene Zustart-Cluster.*

Kriterien: *Mindestens zwei Zustart-Cluster werden anhand von Fahrzeuggeschwindigkeit, Gaspedalwinkel und Zustartzeit nachgewiesen.*

Hypothese 3: *Es existieren Optimierungspotentiale bei den Zustarts.*

Kriterien: *Eine Maßnahme zur Verbesserung relevanter Zustarteigenschaften wie z.B. der Zustartzeit kann aufgezeigt werden.*

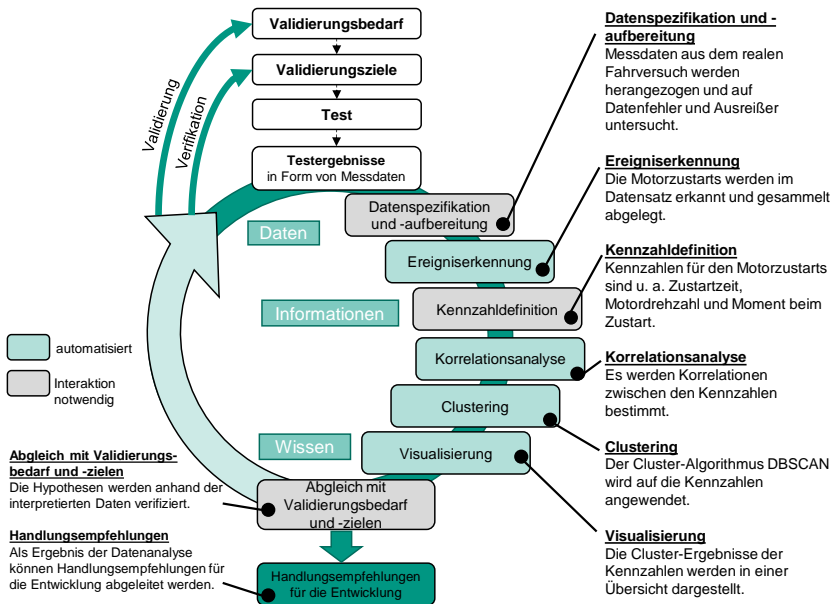


Abbildung 7.21: Vorgehen Messdatenanalyse mit Cluster-Algorithmen, angewendet auf die Funktion Motorzustart

7.3.1 Testergebnisse

Als Fahrmanöver der Testfälle ist ein breites Bild über verschiedene Fahrsituationen, Geschwindigkeitsbereiche, Fahrer und Fahrzeuge notwendig. Als Validierungsumgebung wird aufgrund der Analyse hinsichtlich der Applikation der reale Fahrversuch ausgewählt. Es liegen bereits Fahrzeugmessdaten aus Versuchsfahrten vor, die diese Anforderungen erfüllen. Über 1.000 Signale des Getriebesteuergeräts in unterschiedlicher zeitlicher Auflösung bei jedem Motorwiederstart und dies für verschiedene Fahrzeuge führen zu einer großen Datenmenge.

7.3.2 Datenspezifikation und -aufbereitung

Die Datenbasis bilden die vorhandenen Fahrzeugmessdaten. Für den Motorwiederstart werden die in Abbildung 7.22 dargestellten Signale aus den Gesamtmessdaten ausgewählt.

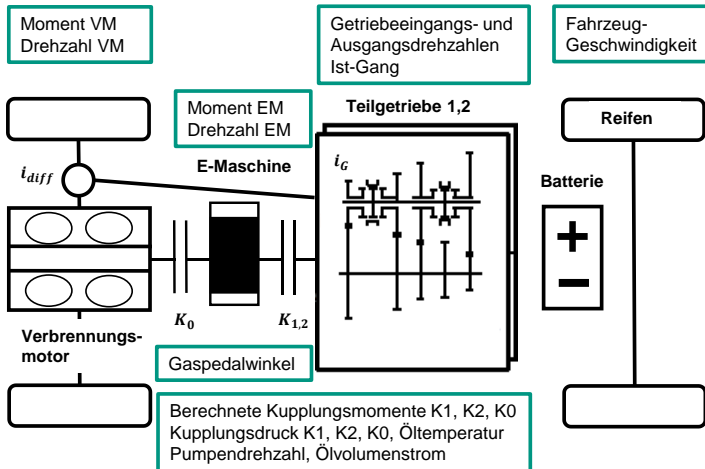


Abbildung 7.22: Schematische Darstellung des Antriebsstrangs und relevante Messsignale

Zusätzlich wird die Last- und Geschwindigkeitsverteilung der Daten betrachtet, um später Cluster-Effekte in der Auswertung aufgrund der Häufigkeitsverteilung der Messdaten vorzubeugen (vgl. Abbildung 7.23).

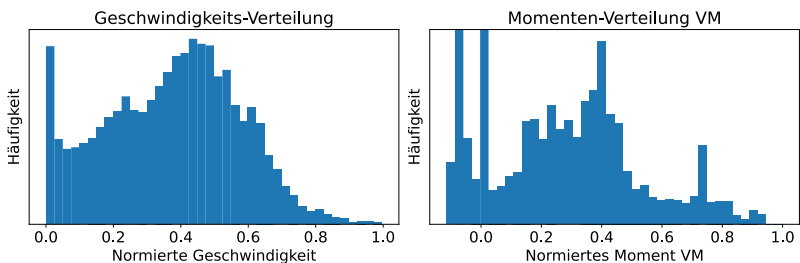


Abbildung 7.23: Geschwindigkeit- und Lastverteilung des untersuchten Datensatzes

Die Geschwindigkeit von 0,4 bis 0,6 der normierten Geschwindigkeit ist in den Daten überproportional vertreten. Die Daten zeigen auch eine Lastverteilung über das komplette VM-Moment. Die dominierende Last liegt zwischen 10 % und 50 % der maximalen Last.

Um eine hohe Datenqualität sicherzustellen, werden die Messdaten auf fehlende Werte und Ausreißer geprüft. Eine Analyse ergibt, dass keine fehlenden Werte und keine Ausreißer vorhanden sind. Dies bestätigt wiederum die Annahme aus Kapitel 6.3.4, dass Signale aus dem Getriebesteuergerät eine hohe Datenqualität besitzen.

7.3.3 Ereigniserkennung

In einem zweiten Schritt filtert eine Ereigniserkennung Zeitausschnitte der Motorwiederstarts aus den Messdaten heraus. Die Erkennung der Zustarts erfolgt anhand der Kriterien VM-Drehzahl und VM-Drehmoment sowie der Fahrzeuggeschwindigkeit. Start-Stopp-Zustarts werden nicht berücksichtigt. Liegt somit eine steigende VM-Drehzahl beginnend ab 0 rpm bei gleichzeitig positivem Motormoment und einer Geschwindigkeit größer 0 km/h vor, so wird dies als Motorwiederstart detektiert. Für jedes Ereignis wird ein Zeitraum von fünf Sekunden vor dem Start und nach dem Start abgelegt. Die Messung einer ersten VM-Drehzahl größer null definiert den Startpunkt des Motorwiederstarts. Die Betriebspunkte von VM und EM sind die Soll-Betriebspunkte nach dem Neustart. Die Ereigniserkennung erkennt aus den Messdaten insgesamt 278 Motorwiederstarts.

7.3.4 Kennzahldefinition

In einem dritten Schritt werden Kennzahlen für jeden Motorwiederstart berechnet. Diese beschreiben das Betriebsverhalten des Antriebsstrangs und werden basierend auf Expertenwissen für die Funktion Motorwiederstart ermittelt. Wichtige Kennzahlen sind der Zielbetriebspunkt des VM und der EM, Geschwindigkeitsgradienten sowie die Fahrzeuggeschwindigkeit und der gewählte Gang. Abbildung 7.24 zeigt einen Teil der Kennzahlen als graphische Übersicht. Die Drehmomente und Gaspedalwinkel werden als das Maximum des zeitlichen Kenngrößenverlaufs in dem betrachteten Zeitabschnitt bestimmt. Der Drehzahlgradient des VM wird als mittlerer Gradient im Zeitbereich zwischen Motorstart und Erreichen der Zieldrehzahl berechnet (2). Der Ist-Gang und die Fahrzeuggeschwindigkeit werden zum Zeitpunkt des Motorstarts bestimmt (t_1), die Drehzahl des VM und der EM nach der Synchronisation der Fahrkupplung (t_3).

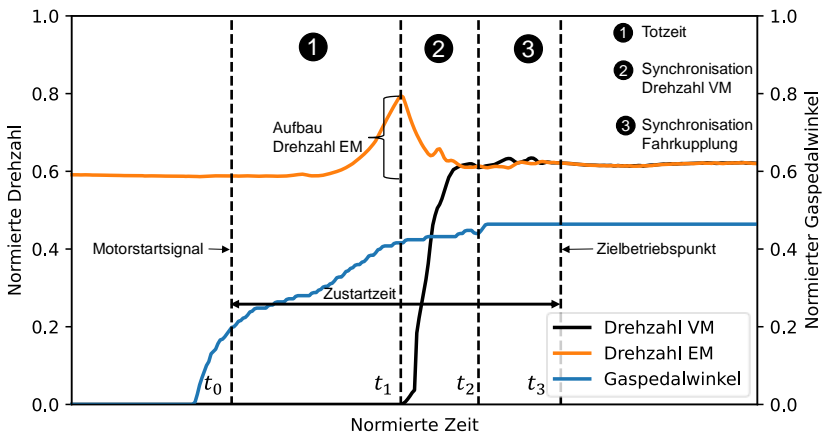


Abbildung 7.24: Zeitlicher Ablauf des Motorwiederstarts dargestellt über Drehzahl von EM und VM, sowie des Gaspedalwinkels.

Kundenbezogene Anforderungen sind die Motorwiederstartzeit und die Motorwiederstartstrategie. Komfort- und NVH-Kriterien werden in dieser Untersuchung nicht berücksichtigt.

7.3.5 Korrelationsanalyse

Im ersten Schritt der Datenanalyse werden Korrelationen der Kennzahlen ermittelt. Dabei wird die in Kapitel 2.3.2 beschriebene Korrelations-Matrix genutzt, die den linearen Zusammenhang der Kennzahlen beschreibt.

Die Korrelations-Matrix nach Bravais und Pearson ist in Abbildung 7.25 für die Kennzahlen dargestellt. Nachfolgend werden die wichtigsten Erkenntnisse beschrieben und ihre Plausibilität diskutiert.

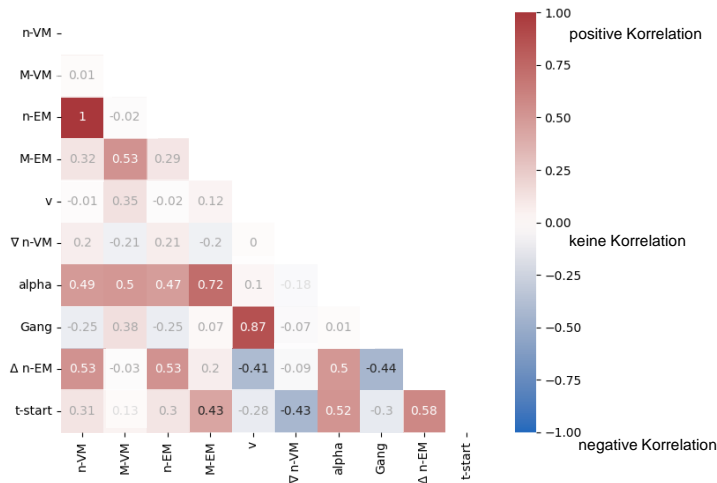


Abbildung 7.25: Korrelations-Matrix der zu untersuchenden Kennzahlen

Die VM-Zieldrehzahl n_{VM} und EM-Zieldrehzahl n_{EM} korrelieren mit einem Wert von 1, was einer genauen linearen Abhängigkeit entspricht. Dies ist zu erwarten und plausibel, da die Drehzahl des VM und der EM bei geschlossener Trennkupplung K0 durch die koaxiale Anordnung der EM identisch sein müssen.

Der Gaspedalwinkel $alpha$ beim Zustart korreliert positiv zu den Momenten M_{VM} , M_{EM} und Drehzahlen von VM und EM. Hohe Korrelationswerte zwischen 0,49 und 0,72 treten hier auf. Dies ist ebenfalls nachvollziehbar, dass der Gaspedalwinkel das Fahrerwunschmoment wiedergibt und somit direkt das Moment von VM und EM bestimmt. Zusätzlich treten hohe Momente auch in Kombination mit hohen Drehzahlen auf, um eine hohe Abtriebsleistung bereitzustellen.

Die Fahrzeuggeschwindigkeit v korreliert mit einem Wert von 0,87 zum Ist-Gang beim Zustart. Dies ist auch plausibel, da bei höheren Fahrzeuggeschwindigkeiten aufgrund von Drehzahlgrenzen und der Energieeffizienz höhere Gänge gefahren werden.

Die Motorzustartzeit t_{start} korreliert positiv zum Gaspedalwinkel, dem Moment der EM und der Drehzahldifferenz der EM Δn_{EM} . Dies ist unerwartet, da ein höherer Drehzahlaufbau der EM in einer kürzeren Startzeit resultieren sollte. Grundsätzlich sollte bei höherem Gaspedalwinkel auch eine geringere Motorstartzeit umgesetzt

werden, um dem Fahrerwunsch einer sofortigen Beschleunigung gerecht zu werden. Die Motorzustartzeit korreliert negativ zum Drehzahlgradienten des VM ∇n_{VM} beim Zustart, was der Koeffizient -0,43 belegt. Dies ist plausibel, da ein hoher Gradient die Motorstartzeit verkürzt. Zu weiteren Kennzahlen gibt es keine aussagekräftige Korrelation zum Drehzahlgradienten des VM.

7.3.6 Clustering

Wie in Kapitel 2.3.2 dargestellt, existiert eine Vielzahl an Cluster-Algorithmen. Für diesen Anwendungsfall gibt es zwei Anforderungen an den Cluster-Algorithmus. Da in den Daten unterschiedliche Formen von Clustern auftreten können, muss der Algorithmus diese erkennen. Dadurch entfallen partitionierende und hierarchische Ansätze. Wichtig ist auch, dass die Anzahl an Clustern nicht vorgegeben wird, sondern der Algorithmus diese selbst bestimmt. Der DBSCAN-Algorithmus erfüllt diese Anforderungen. Dieser verwendet die Parameter minPts und ϵ , um die Clusterdichte zu definieren. Nach Ester et al. (1996) ist der minimale minPts-Parameter als $2 * \text{dim}$ definiert. Da der zweidimensionale Raum genutzt wird, ist dieser Parameter somit vier. Um sehr kleine Cluster zu vermeiden, wird für minPts 15 gewählt, sodass die kleinste Clustergröße etwa 5,4 % des gesamten Datensatzes beträgt. Der Parameter ϵ hat einen wesentlichen Einfluss auf die Größe der Cluster und deren Anzahl. Wird er zu groß gewählt, werden in den Daten keine Cluster identifiziert (hoher Rauschanteil); wird er zu klein gewählt, dann ist es möglich, dass einzelne Cluster zu groß werden und alle Daten in ein Cluster fallen. Nach Schubert et al. (2017) sollte der Rauschanteil zwischen 1 % und 30 % liegen und das größte Cluster maximal 20 % bis 50 % der Datenpunkte enthalten. Um diese Anforderungen zu erfüllen, wird der optimale Parameter ϵ für jede Kennzahl einzeln berechnet. Vor dem Clustering wird der Datensatz mit dem RobustScaler skaliert (vgl. Kapitel 7.1.4). Als Beispiel zeigt Tabelle 7.6 das Rauschen und die Clustergröße für das VM-Kennfeld für verschiedene ϵ . Das optimale ϵ ist 0,20 mit einem Rauschen von 42 % und einem größten Cluster von 32 %. Das Rauschen kann hier nicht weiter reduziert werden, ohne das größte Cluster zu vergrößern. Daher wird für ϵ der Wert 0,20 gewählt.

Tabelle 7.6. Hyperparameterbestimmung DBSCAN

| ϵ | 0.15 | 0.175 | 0.20 | 0.225 | 0.25 |
|------------------|-------------|--------------|-------------|--------------|-------------|
| Rauschen VM | 60 % | 51 % | 42 % | 33 % | 29 % |
| Cluster Größe VM | 32 % | 34 % | 32 % | 57 % | 65 % |

Zunächst werden aus den Messdaten Motorstart-Cluster identifiziert. Eine sinnvolle Klassifizierung ergibt sich über die Eigenschaften Geschwindigkeit, Gaspedalwinkel und Zustartzeit. Die Anwendung des Cluster-Algorithmus führt zur Identifizierung von vier Clustern (vgl. Abbildung 7.26).

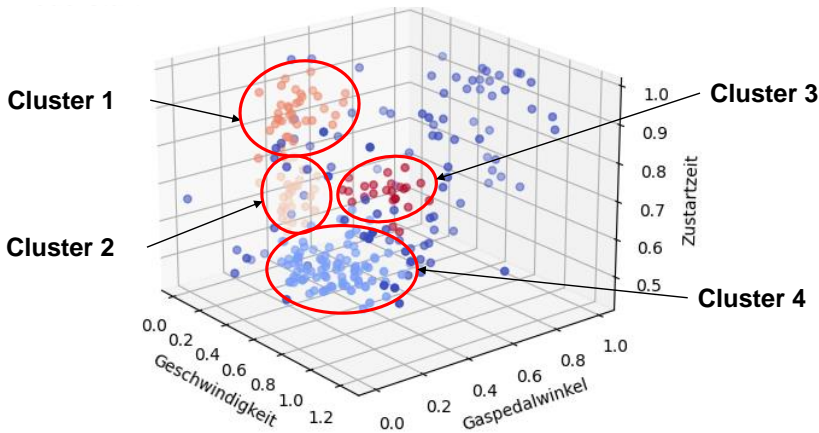


Abbildung 7.26: Darstellung von vier identifizierten Cluster des Motorzustarts über Geschwindigkeit, Gaspedalwinkel beim Zustart sowie Zustartzeit

Cluster 1 sind Zustarts bei kleinen Geschwindigkeiten und kleinem Gaspedalwinkel, die eine hohe Startzeit besitzen (langsame Zustarts). Gleichzeitig liegt Cluster 2 in dem gleichen Geschwindigkeits- und Gaspedalwinkelbereich, hat jedoch eine niedrigere Startzeit (v-min Zustarts). Cluster 3 liegt ebenfalls bei kleinen Geschwindigkeiten, aber großem Gaspedalwinkel und hat eine mittlere Zustartzeit (aggressive Zustarts). Cluster 4 sind Zustarts bei hohen Geschwindigkeiten, kleinem Gaspedalwinkel und geringen Zustartzeiten (v-max Zustarts). Die Benennung der Cluster ergibt sich aufgrund ihrer Eigenschaften.

Anschließend wird der DBSCAN-Algorithmus auf alle Kennzahlen angewendet. Diese werden im zweidimensionalen Datenraum gebildet. Dabei wird das VM-Moment und der VM-Drehzahlgradient über die VM-Drehzahl abgebildet. Das EM-Moment und der EM-Drehzahlgradient werden über die EM-Drehzahl abgebildet. Ist-Gang, Gaspedalwinkel und Geschwindigkeit werden im eindimensionalen Raum geclustert. Hier wäre auch eine einfache statische Auswertung möglich.

7.3.7 Visualisierung

Der Schritt Visualisierung stellt abschließend alle Kennzahlen automatisiert in einer Übersicht dar. Dafür können die Kennzahlen für ein bestimmtes Motorzustart-Cluster oder mehrere Cluster visualisiert werden. Abbildung 7.27 zeigt die Übersicht, die die vier Motorzustart-Cluster vergleicht.

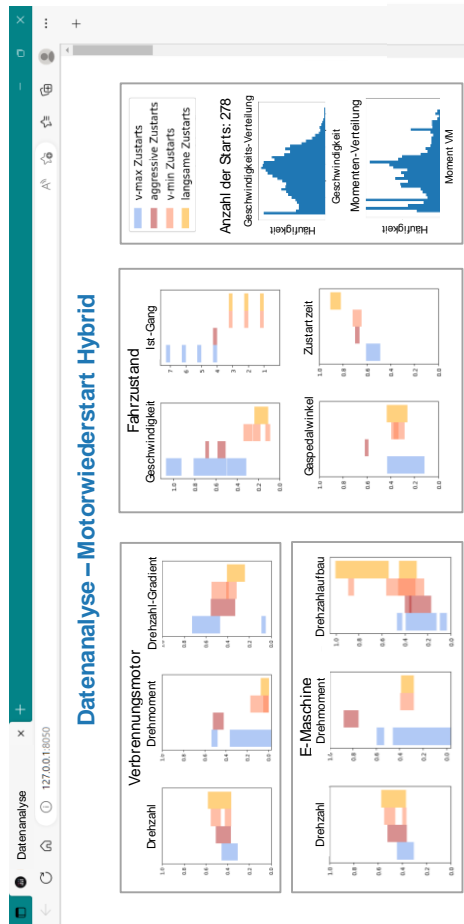


Abbildung 7.27: Vergleich der Motorzustart-Cluster in einer graphischen Übersicht

Um Auffälligkeiten gezielt analysieren zu können, ist neben der Darstellung in einer kompakten Übersicht auch die Darstellung einzelner Kennfelder möglich. Abbildung 7.28 zeigt die Kennfelder des VM und der EM ebenfalls mit bereits angepasster Cluster-Benennung.

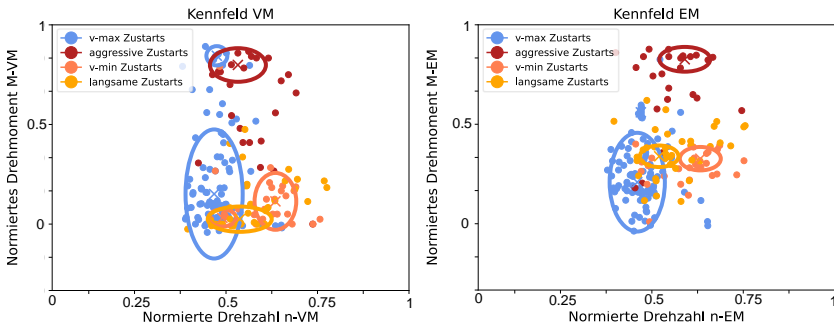


Abbildung 7.28: Kennfeld von VM und EM für die vier identifizierten Motorzustarts-Cluster

7.3.8 Verifikation und Validierung

In einem abschließenden Schritt werden die visualisierten Informationen interpretiert und validiert, indem ihre Plausibilität überprüft wird. Die validierten Ergebnisse werden dann zur Beantwortung der Analyse-Fragestellungen des Zielsystems genutzt.

Hypothese 1: *Es existieren bisher nicht bekannte Korrelationen in den Messdaten.*

Kriterium: *Mindestens eine unbekannte Korrelation zwischen relevanten Messsignalen des Motorzustarts wird in den Messdaten erkannt.*

Aufbauend auf der Korrelationsanalyse sind größtenteils alle Korrelationen plausibel, aus dem physikalischen Verständnis des Motorwiederstarts nachvollziehbar und bereits bekannt. Auffällig ist, dass der Gradient der VM-Drehzahl nur mit der Motorstartzeit und nicht mit weiteren Größen korreliert. Außerdem treten hohe Motorstartzeiten bei höherem EM-Drehzahlaufbau auf, was zu einem schnelleren Motorstart führen sollte. Dieses Phänomen gilt es näher zu untersuchen. Somit wird Hypothese 1 bestätigt.

Hypothese 2: *Es existieren verschiedene Zustarts-Cluster.*

Kriterium: *Mindestens zwei Zustart-Cluster werden anhand von Fahrzeuggeschwindigkeit, Gaspedalwinkel und Zustartzeit nachgewiesen.*

Anhand der Kennzahlen Motorstartzeit, Moment und Geschwindigkeit lassen sich vier Motorzustart-Cluster identifizieren. Die vier Zustart-Cluster erhalten aufgrund ihrer Eigenschaften nachfolgend eine neue Benennung.

„Aggressive Zustarts“ (Cluster 3) treten bei Gaspedalwinkel größer 50 % auf, damit auch bei Momenten des VM und der EM von größer 50 % bzw. 70 % und Geschwindigkeiten ab 50 km/h. Deren Motorstartzeit liegt im mittleren Bereich. Die Zielbetriebspunkte des VM und der EM sind dabei stark verstreut. Der Geschwindigkeitsbereich kann hierbei durch das Fahrprofil bestimmt worden sein. „Aggressive Zustarts“ können auch bei niedrigeren Geschwindigkeiten auftreten.

„v-max Zustart“ (Cluster 4) treten bei Gaspedalwinkel kleiner 40 %, Geschwindigkeiten größer 0,3 und im vierten bis siebten Gang auf. Die Momente von VM und EM sind entsprechend kleiner 50 %. Die Motorstartzeiten sind am geringsten von allen vier Clustern. Die Zielbetriebspunkte der VM und EM sind dabei recht stabil. Dies ermöglicht eine gute Applizierbarkeit dieser Motorstarts.

„Langsame Zustarts“ (Cluster 1) treten bei Gaspedalwinkeln kleiner 40 %, Geschwindigkeiten kleiner 0,2 und im ersten, zweiten und dritten Gang auf. Die Momente von VM und EM sind niedrig, die Drehzahlen dagegen höher.

Auffällig ist, dass diese Zustarts einen hohen Drehzahlaufbau der EM vor dem Motorstart aufweisen, der zu einem hohen Gradienten der VM-Drehzahl führen sollte, dieser jedoch von allen Motorstart-Clustern am niedrigsten ist. Schaut man nun genauer in einzelne Messungen dieser Motorstart-Cluster, so fällt auf, dass viele dieser Messungen eine Verzögerung beim Motorstart haben. Zunächst wird der VM gestartet und erreicht eine normierte Drehzahl von 0,2, erst mit einer Verzögerung wird die VM-Drehzahl erhöht (vgl. Abbildung 7.29).

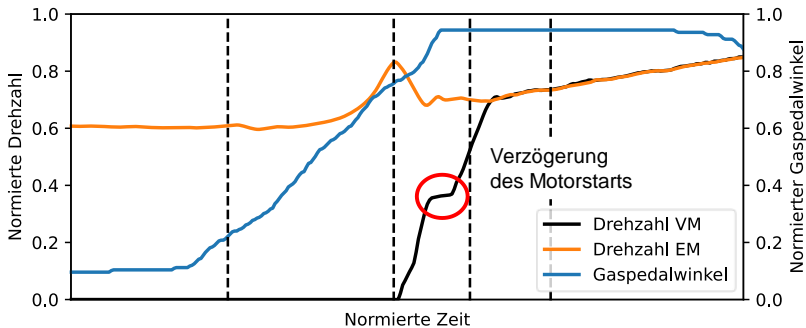


Abbildung 7.29: Analyse des Motorstart-Cluster „langsamer Zustand“. Anhand des Clustering kann eine Verzögerung des Motorstarts erkannt werden.

Nach Diskussion mit Experten konnte die Ursache identifiziert werden. Der Motorstart wird in zwei Phasen durchgeführt. Zunächst wird beim „Anreißen“ des VM mit dem K0-Moment gestartet, in der zweiten Phase erreicht der VM die Zieldrehzahl mit dem eigenen Moment. Der Grund für diese Haltephase ist eine nicht optimale Applikation des Zustarts, der dazu führt, dass beide Zustartsphasen nicht nahtlos ineinander übergehen. Dies muss überarbeitet werden. Somit konnte ein Optimierungspotential in der Applikation ermittelt werden.

„V-min Zustarts“ (Cluster 2) treten ebenfalls bei Gaspedalwinkel kleiner 40 %, Geschwindigkeiten kleiner 0,2 und im ersten, zweiten und dritten Gang auf. Die Momente von VM und EM sind niedrig, die Drehzahlen dagegen höher, wobei das VM-Moment nahe null ist.

Es ist gut zu erkennen, dass die „aggressiven Zustarts“ und „v-min Zustarts“ in einem gut definierten Betriebspunkt ausgeführt werden, wobei die „langsamen Zustarts“ und „v-max Zustarts“ stärker verstreut sind. Dies hängt auch damit zusammen, dass sich der Gaspedalwinkel und die Fahrzeuggeschwindigkeit indirekt in den Momenten und Drehzahlen widerspiegeln (vgl. Abbildung 7.28). Alle Motorstart-Cluster und ihre Eigenschaften sind plausibel nachvollziehbar.

Durch die Identifikation der vier Zustart-Cluster wird Hypothese 2 bestätigt.

Außerdem wird das Validierungsziel, die Identifikation und Klassifizierung unterschiedlicher Motorstart-Cluster, erfüllt.

Hypothese 3: *Es existieren Optimierungspotentiale bei den Zustarts.*

Kriterium: *Eine Maßnahme zur Verbesserung relevanter Zustarteigenschaften wie z.B. der Startzeit kann aufgezeigt werden.*

Die Motorstartzeit der „langsamen Starts“ und der „aggressiven Starts“ ist vergleichsweise hoch. Hier kann eine Fahrbarkeitsuntersuchung durchgeführt werden, um festzustellen, ob die Motorstartzeit auch für den Fahrer als zu hoch empfunden wird. Anhand der Kennzahlen lassen sich Ursachen identifizieren und Maßnahmen ableiten, um die Startzeit zu reduzieren. Für die „langsamen Starts“ muss die Applikation angepasst werden, um den Übergang der beiden Motorstartphasen zu verbessern. Für die „aggressiven Starts“ kann ein höherer Drehzahlaufbau der EM vor dem Motorstart die Motorstartzeit minimieren (vgl. Abbildung 7.26). Diese Erkenntnisse können wie bereits in Kapitel 6.5.9 beschrieben für die Entwicklung im Markt bestehender Fahrzeuge sowie für nachfolgende Systemgenerationen genutzt werden (vgl. Abbildung 6.16). Somit wird Hypothese 3 bestätigt.

7.4 Zwischenfazit

In Kapitel 7 wurden die in Kapitel 6 entwickelten Methoden auf aktuelle Entwicklungsaufgaben angewendet, um Forschungsfrage III (FF-III) zu beantworten.

Forschungsfrage III (F-III)

Welche messbaren Mehrwerte und Potentiale ergeben sich durch die Methoden des maschinellen Lernens?

Kupplungstemperaturmodell. Die Methode Modellierung von virtuellen Sensoren mit KNN kann erfolgreich für die Abbildung der Kupplungstemperatur der Fahrkupplung K1 eines hybriden Doppelkupplungsgetriebes eingesetzt werden. Die Modelle verringern den Modellfehler (RMSE) im Mittelwert um 15,5 % und im Einzelfall um bis zu 21,6 % im Vergleich zum bisher verwendeten physikalisch modellierten Temperaturmodell (Referenzmodell). Gleichzeitig verhält sich das Modell robust über eine Vielzahl von Fahrmanövern. Die Methode kann mit geringem Aufwand auf weitere Fahrzeugkupplungen und auch Systemgenerationen übertragen werden (vgl. Abbildung 7.30). Für die baugleiche Kupplung K1 kann für konventionelle Lastfälle der mittlere RMSE um 18,5 % verbessert werden. Für das

Temperaturmodell der Trennkupplung K0 wird die Modellgüte durch die Verwendung des KNN im Mittel um 22,5 % verbessert, jedoch weist es weiterhin eine große Abweichung zur Messung auf. Somit wird das Erfolgskriterium Steigerung der Modellgüte erfüllt. Auch die Parameteranzahl der Modelle liegt mit 309 Parametern deutlich unter der des bisher genutzten Referenzmodells mit 1000 Parametern, womit auch das Kriterium der Steuergerätfähigkeit erfüllt ist.

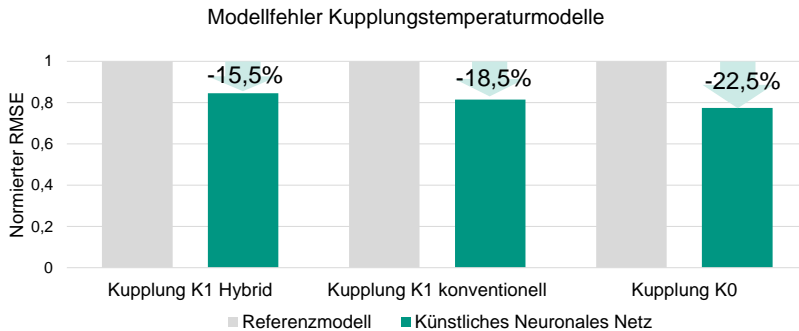


Abbildung 7.30: Durch die Verwendung eines KNN im Vergleich zum Referenzmodell verringert sich der Modellfehler für drei verschiedene Kupplungen und Antriebsstränge um bis zu 22,5%.

Ein geringerer Aufwand bei Erstellung und Übertragung des Modells auf weitere Fahrzeugkupplungen kann angenommen werden, konnte aber im Rahmen der Arbeit nicht quantitativ nachgewiesen werden.

Bei der hier gezeigten Entwicklungsaufgabe müssen nachfolgende Punkte beachtet werden. Der Kühllölvolumenstrom wird unter anderem durch die Kupplungstemperatur bestimmt. In den Fahrzeugmessdaten für das Training wird die Temperatur aus dem Referenzmodell berechnet, die somit den Kühllölvolumenstrom beeinflusst. Da das Referenzmodell von der gemessenen Temperatur abweicht, wurde z.B. bei einer zu hohen Temperaturberechnung ein höherer Kühllölvolumenstrom als notwendig gestellt. Dieses Verhalten lernt das KNN aus den Messdaten. Da die Temperaturabweichung des Referenzmodells allerdings gering ist, kann dieser Fehler vernachlässigt werden. Die Kupplungstemperaturmodelle der Fahrkupplung K1 für den Hybrid und den konventionellen Antrieb wurden nur mit den kritischen Lastfällen trainiert und damit validiert. Damit werden die temperaturkritischen Lastfälle, bei denen es zu Schäden an der Kupplung kommen kann, abgedeckt. Um ein vollständiges Modell zu erhalten, müssten zusätzlich Temperaturmessungen der Gangwechsel in das Training und die Validierung miteinfließen. Bei diesem Ansatz muss auch

berücksichtigt werden, dass das Modell mit Messdaten aus nur einem Fahrzeug mit einem bestimmten Verschleiß trainiert wurde. Deshalb sind der zukünftige Verschleiß und Alterungsprozesse in den Messdaten und somit in dem final trainierten Modell auf dem Steuergerät nicht berücksichtigt.

Folgende wichtige Erkenntnisse werden im Umgang mit der Methode gewonnen:

- **Die Automatisierung einzelner Methodenschritte ist ein großer Vorteil der ML-Ansätze.**
- **Die zeitliche Auflösung der Messdaten und die Anzahl der zurückliegenden Zeitschritte, die für eine Vorhersage genutzt werden, haben einen großen Einfluss auf die Abbildungsgüte.**
- **Erklärbarkeitsansätze sind eine gute Herangehensweise, um Einblicke in das Modell und Handlungsempfehlungen für die Modelloptimierung abzuleiten.**

Kupplungsfüllung. Die Methode Steuerungs-Optimierung mit verstärkendem Lernen wird auf die Füllung der Kupplung K2 eines hybriden Doppelkupplungsgetriebes angewendet. Es wird eine Ansteuerung ermittelt, die die maximal zulässige Füllzeit und Systemdruckeinbruch einhält und gleichzeitig den Systemdruckeinbruch in der Simulation für verschiedene Systemdrücke um 26 % bis 86 % reduziert. In Validierungsmessungen im Fahrzeug werden diese Optimierungsergebnisse bestätigt. Das Ergebnis ist hinsichtlich dieser Kriterien deutlich besser als eine auf Erfahrungswerten basierende Ansteuerungskurve. Somit wird das Erfolgskriterium Steigerung der Ansteuerungsgenauigkeit erfüllt.

Diese Lösung kann Entwickelnde nun als Startpunkt ihrer Auslegung verwenden und unterstützt sie somit mit der Komplexität in der Auslegung von Funktionen in hybriden Antriebssträngen umzugehen. Zusätzlich identifiziert der Agent eine neue Ansteuerung, indem er einen steilen initialen Stromgradienten nutzt und diesen wieder absenkt, was insgesamt zu einer verbesserten Ansteuerung führt. Der weitere Ventilstromverlauf ist ähnlich zum bisher genutzten parametrisierten Verlauf. Somit kann die Methode für den Entwickelnden ein Werkzeug darstellen, neue Ansteuerungen zu entwickeln. Ein Genetischer Algorithmus, der Parameter des Ventilstroms optimiert, kann den Druckeinbruch ebenfalls für verschiedene Systemdrücke um 65 % bis 91 % reduzieren. Jedoch ist die Füllzeit um etwa 36 % höher, aber noch innerhalb der Anforderungen (vgl. Abbildung 7.31). Die Anzahl der benötigten Simulationen liegt für den Genetischen Algorithmus um 70 % niedriger. Eine Erklärung für die höhere Anzahl an Trainingsdurchläufen liegt in dem hohen Bedarf des verstärkenden Lernens an Trainingsbeispielen, bis ein gutes Agentenverhalten vorliegt. Außerdem zeigt sich, dass eine Kupplungsfüllung als

Steuerungsproblem mit einem parameterbasierten Ansatz ebenfalls gut lösbar ist. Aufgrund der Skalierbarkeit von Trainingszeit z.B. über Rechenzentren ist dieser Nachteil jedoch vernachlässigbar.

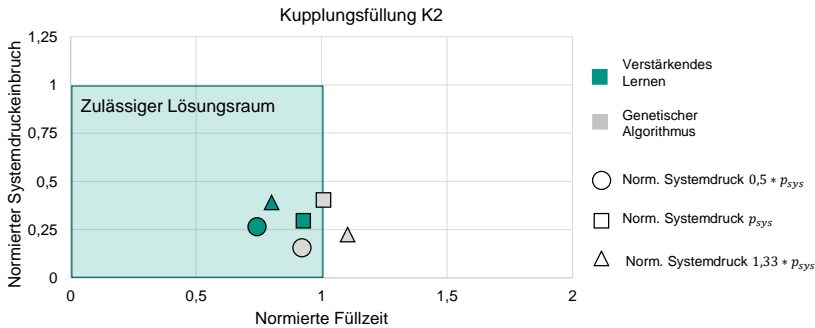


Abbildung 7.31: Ansteuerungen der Kupplungsfüllung im Vergleich über die Zielgrößen Systemdruckeinbruch und Füllzeit

Die Variablen der Belohnungsfunktion konnten ebenfalls eindeutig bestimmt werden, allerdings bedarf es einer iterativen Anpassung der Bestrafungsterme und Gewichtung, um eine Belohnungsfunktion mit gutem Konvergenzverhalten einzustellen. Bei der Bestimmung der Fitnessfunktion des Genetischen Algorithmus tritt dieser Mehraufwand ebenfalls auf. Die gute Übertragbarkeit der Trainingsergebnisse auf das Fahrzeug ist auf die hohe Modellgüte des Hydraulikmodells zurückzuführen. In weiteren Optimierungen können im Modell Parameter wie Temperatur und Luftgehalt des Öls verändert werden, sodass für verschiedene Umgebungsbedingungen jeweils eine optimale Ansteuerung ermittelt werden kann.

Als wichtige Erkenntnisse hat sich zur Ermöglichung einer guten Übertragbarkeit der Ergebnisse von der Simulation ins Fahrzeug ein Modell mit hoher Abbildungsgüte gezeigt.

Hybrider Motorwiederstart. Die in Kapitel 6.5 beschriebene Methode der Datenanalyse wird auf Fahrzeugmessdaten der Entwicklungsaufgabe Motorwiederstart in einem hybriden Doppelkupplungsgetriebe angewendet, um Hypothesen, abgeleitet aus den Validierungszielen, zu verifizieren, sowie den Validierungsbedarf abzugleichen. Die Korrelationsanalyse der Kennzahlen bestätigt das bisherige Systemverständnis. Mit dem Cluster-Ansatz können aus den 278 erkannten Motorstarts erfolgreich vier Motorzustart-Cluster „aggressiver Zustand“, „v-max Zustand“, „langsamer Zustand“, und „v-min Zustand“ erkannt und über Kennzahlen

charakterisiert werden. An der konkreten Aufgabe wurden ein Optimierungspotential der Motorzustartszeit identifiziert und die Ursache abgeleitet. Die Methode kann somit Auffälligkeiten in den Daten zeigen, die anschließend gezielt analysiert werden, woraus Optimierungspotentiale und Handlungsempfehlungen abgeleitet werden.

Grundsätzlich lassen sich die identifizierten Cluster auch mit Methoden der Statistik bestimmen. Cluster-Algorithmen vereinfachen jedoch den Umgang und können mit geringerem Vorwissen als statistische Methoden angewandt werden. Die Übersicht der Daten dient als eine qualitative Darstellung. Konkrete Werte sollten aus den Kennfeldern ermittelt werden. Zusätzlich sei hier nochmals auf den iterativen Charakter der Datenanalyseverfahren verwiesen. Neue Erkenntnisse fließen wiederum in das Zielsystem ein und führen zu einem neuen Analysevorgang. In diesem Anwendungsbeispiel wurde das Potential aufgezeigt, das Cluster-Algorithmen bei der Auswertung von Messdaten bieten können, womit das Erfolgskriterium erfüllt wird. Ein direkter Vergleich mit anderen Datenauswertungsmethoden wurde in diesem Fall nicht durchgeführt.

Folgende wichtige Erkenntnisse werden im Umgang mit der Methode gewonnen:

- **Der Validierungsbedarf sowie die Validierungsziele werden durch den Erkenntnisgewinn der Analyse iterativ angepasst und erweitert.**
- **Die Korrelationsanalyse eignet sich sehr gut, um neue Informationen aus dem System zu generieren und gleichzeitig das bisherige Verständnis datenbasiert zu bestätigen oder zu widerlegen.**
- **Auch in dieser Methode kann der Vorteil von ML-Ansätzen der Automatisierung für die Datenaufbereitung, Ereigniserkennung, Korrelationsanalyse, Clustering und Visualisierung genutzt werden.**

8 Studien zur Anwendbarkeit

Im vorherigen Kapitel wurde an jeweils einem Anwendungsbeispiel gezeigt, dass die ML-Methoden eine Verbesserung der in Kapitel 6.2 definierten Erfolgsfaktoren erzielen. Um sicherzustellen, dass auch die Anwendenden die Methoden nutzen können, wird in diesem Kapitel die Anwendbarkeit untersucht sowie Verbesserungspotentiale identifiziert. Damit wird Forschungsfrage IV (FF-IV) beantwortet.

Forschungsfrage IV (FF-IV)

Wie wird die Anwendbarkeit der entwickelten Methoden des maschinellen Lernens durch die Entwickelnden bewertet?

Mit den Methoden Ansteuerungsoptimierung mit verstärkendem Lernen und Messdatenanalyse mit Cluster-Algorithmen wird eine Evaluation in Form einer Befragungsstudie mit Getriebeentwickelnden (n=10, n=9) durchgeführt.

Für die Methode Modellierung von virtuellen Sensoren mit KNN wird ein Software-Tool erstellt, mit dem eine Anwenderstudie durchgeführt wird. Der Anwendende löst dabei eine Aufgabenstellung anhand einer Anleitung und dem Software-Tool. Neben Getriebeentwickelnden, den Ziel-Anwendenden (n=7), wird die Anwenderstudie auch mit einer Gruppe Entwickelnden mit ML-Erfahrung (n=8) durchgeführt. Damit wird untersucht, inwieweit beide Gruppen die Methode anwenden können und welche Unterschiede es zwischen diesen Gruppen gibt.

8.1 Design der Befragungsstudie

Ziel der Evaluation ist es, folgende Punkte der Anwendbarkeit zu untersuchen:

- Methodenverständnis
- Methodenakzeptanz
- Aufzeigen von Schwierigkeiten und Verbesserungspotentialen

In insgesamt zwei Workshops wurde jeweils eine Methode mit dem entsprechenden Anwendungsbeispiel aus Kapitel 7 vorgestellt. Ein Workshop dauerte etwa 60 Minuten. Zu jeder Methode gibt es einen Fragebogen¹ bestehend aus 23 Fragen, unterteilt in einen Frageblock je Methodenschritt². Die Teilnehmenden beantworteten direkt im Anschluss an die Vorstellung eines Methodenschritts den jeweiligen Teil des Fragebogens. Dabei wenden die Befragten die Methode nicht selbst an, sondern beantworten den Fragebogen auf Basis der Erklärung der Methode anhand des Anwendungsbeispiels.

Die Teilnehmenden sind Getriebeentwickelnde mit mehr als fünf Jahren Berufserfahrung in der Funktionsentwicklung und Applikation von Automatikgetrieben und stellen potenzielle Anwendende der Methode dar. Von den Teilnehmenden hatte lediglich eine Person bereits im Vorfeld Kontakt mit der Methode. Der Großteil der Teilnehmenden ist somit unbefangen.

8.2 Ergebnisse der Befragungsstudie

Die fünf Antwortmöglichkeiten der geschlossenen Fragen werden mit einem Punktesystem von eins bis fünf wie folgt ausgewertet:

- 1 = trifft gar nicht zu
- 2 = trifft eher nicht zu
- 3 = teils/teils
- 4 = trifft eher zu
- 5 = trifft voll zu

Zur Auswertung der Fragebögen wird für jede Frage der Minimal-, Maximal- und Mittelwert der Antworten berechnet. Die Evaluation wurde mit zehn bzw. neun

¹ Die Anwendungsfelder, sowie Vor- und Nachteile des Fragebogens als empirische Methode werden auf S.60 beschrieben.

² Die Fragebogenvorlagen der beiden Studien sind im Anhang auf S.XXV-XXVI beigefügt.

Befragten durchgeführt. Die wichtigsten Erkenntnisse werden nachfolgend erläutert³.

8.2.1 Steuerungs-Optimierung mit verstärkendem Lernen

Über alle 15 Fragen hinweg sind die Mittelwerte der Antworten größer als 3. Der niedrigste Mittelwert liegt bei 3,25. Durch die positive Formulierung der Fragen erhält die Methode in allen Aspekten mindestens teilweise Zustimmung. 9 der 15 Fragen haben einen Mittelwert von mindestens 4, was einer überwiegenden Zustimmung entspricht.

Alle Methodenschritte außer dem Abgleich des Optimierungsziels werden von den Befragten in Bezug auf Verständlichkeit im Durchschnitt mit über 4 und somit als verständlich eingeordnet. Die logische Verknüpfung der Methodenschritte stößt mit einem Wert von 4,4 ebenfalls auf breite Zustimmung. Somit ist das Methodenverständnis größtenteils erfüllt.

Die Befragten beantworten nach Vorstellung der Methode mit einem konkreten Anwendungsbeispiel die Frage, ob sie es sich zutrauen, den Methodenschritt auch auf ein anderes Anwendungsbeispiel anzuwenden, für drei der fünf Methodenschritte im Durchschnitt mit größer 4. Bei der Aufstellung einer Belohnungsfunktion sowie beim Abgleich des Optimierungsziels zeigt sich ein geteiltes Bild. Das Zutrauen fällt mit einem Durchschnittswert von 3,8 und 3,2 geringer aus.

Die Unterstützung des Umgangs mit Komplexität in der Getriebeentwicklung wird mit einem Wert von 3,7 teilweise bestätigt. Die grundsätzliche Anwendbarkeit der Methode im Kontext der Getriebeentwicklung wird mit 4,5 bestätigt.

Die offene Frage nach dem Methodenmehrwert beantwortet je ein Befragter mit dem Abgleich mehrerer Optimierungsziele, freier Optimierungsmöglichkeit hinsichtlich des Lösungsraums, Optimierung von Systemen mit komplexen Zusammenhängen, ohne die physikalischen Zusammenhänge bis ins kleinste Detail verstehen zu müssen, und Beschleunigung des Entwicklungsprozesses.

³ Die Zusammenfassung der Ergebnisse der 15 bzw. 16 geschlossenen Fragen für beide Methoden ist im Anhang auf S.XXVII-XXVIII enthalten. In den Abbildungen ist außerdem die Streuung der Antworten dargestellt.

Da der logische Aufbau der Methode, das Methodenverständnis, die grundsätzliche Anwendbarkeit größtenteils bestätigt und die wesentlichen Mehrwerte erkannt werden, kann von einer Akzeptanz der Methode ausgegangen werden.

Schwierigkeiten in der Anwendung der Methode sehen drei Befragte in der Verfügbarkeit von Modellen mit einer hohen Modellgüte, zwei Befragte in dem Zusammenhang zwischen der Bewertung Belohnungsfunktion und Änderung des Parameterraums, sowie ein Befragter in der finalen Übertragung der gewonnenen Erkenntnisse in die Applikationsumgebung.

Als Verbesserungspotential gaben zwei Befragte an, dass das gegeneinander Auftragen der beiden Zielgrößen z. B. in Form einer Pareto-Front das Verständnis verbessern würde. Ein Befragter schlägt vor, Gütekriterien an das Simulationsmodell zu definieren, um zu entscheiden, ob dieses für die Methode genutzt werden kann.

8.2.2 Messdatenanalyse mit Cluster-Algorithmen

Über alle 16 Fragen hinweg sind die Mittelwerte der Antworten größer als 3. Der niedrigste Mittelwert liegt bei 3,33. Durch die positive Formulierung der Fragen erhält die Methode in allen Aspekten mindestens teilweise Zustimmung. 8 der 16 Fragen haben einen Mittelwert von mindestens 4 was einer überwiegenden Zustimmung entspricht. Der logische Aufbau der Methode wird ebenfalls bestätigt.

Die Methodenschritte Datenspezifikation und Kennzahldefinition werden von den Befragten im Durchschnitt über 4 und somit als verständlich eingeordnet. Das Verständnis der Korrelationsanalyse sowie der Gesamtvisualisierung liegt mit Durchschnittswerten von 3,8 und 3,9 leicht darunter. Somit ist das Methodenverständnis größtenteils erfüllt. Extremwerte sowie Durchschnittswerte von Signalen werden als sinnvolle Kennzahlen bestätigt.

Die Befragten beantworten nach Vorstellung der Methode mit einem konkreten Anwendungsbeispiel die Frage, ob sie es sich zutrauen, die Methode auch auf ein anderes Anwendungsbeispiel anzuwenden für die ersten beiden Methodenschritte im Durchschnitt mit größer 4. Bei der Auswertung der Korrelationsanalyse sowie der Gesamtvisualisierung zeigt sich ein geteiltes Bild. Das Zutrauen fällt mit einem Durchschnittswert von 3,55 und 3,33 geringer aus. Gleichzeitig wird die Frage, ob die Korrelationsanalyse das Systemverständnis unterstützt, mit 4,11 und die Unterstützung der Gesamtvisualisierung Zusammenhänge im System zu erkennen, mit 3,88 bewertet. Der Mehrwert der beiden Schritte wird somit erkannt, jedoch die Anwendung noch nicht zugetraut.

Die Unterstützung im Umgang mit der Komplexität in der Getriebeentwicklung wird mit einem Wert von 3,63 teilweise bestätigt. Die grundsätzliche Anwendbarkeit der Methode im Kontext der Getriebeentwicklung wird mit 4,13 bestätigt.

Die offene Frage nach dem Methodenmehrwert beantwortet je ein Befragter mit der Automatisierung der Auswertung großer Datenmengen, der schnellen Visualisierung von Messdaten, Entdecken unbekannter Korrelationen sowie der Reduzierung von Analysezeiten.

Da der logische Aufbau der Methode, das Methodenverständnis, die grundsätzliche Anwendbarkeit größtenteils bestätigt und die wesentlichen Mehrwerte erkannt werden, kann auch hier von einer Akzeptanz der Methode ausgegangen werden.

Schwierigkeiten in der Anwendung der Methode sehen die Befragten in mehreren Punkten. Die genaue Kenntnis der Abhängigkeiten der ausgewerteten Prüfsignale untereinander ist Voraussetzung für eine sinnvolle Analyse der Ergebnisse durch den Nutzenden. Die Visualisierung ist so zu gestalten, dass brauchbare Rückschlüsse getroffen werden können. Je nach Anwendungsfall müssen genügend Messdaten ohne personenspezifisches Fahrverhalten vorliegen, damit eine statistische Auswertung möglich wird. Bei der Interpretation der Daten kann es schwierig sein, gut von schlecht zu unterscheiden. Jeder Punkt wurde jeweils einmal genannt.

Als Verbesserungspotential nennt ein Befragter die graphische Aufbereitung von Zusammenhängen in den Daten.

8.3 Design der Anwenderstudie

Für die Methode Modellierung von virtuellen Sensoren mit KNN wird eine Anwenderstudie durchgeführt. Ziel der Anwenderstudie ist es, folgende Punkte der Anwendbarkeit zu untersuchen:

- Methodenverständnis
- Methodenakzeptanz
- Aufzeigen von Schwierigkeiten und Verbesserungspotentialen
- Direkte Anwendung der Methode
- Übersichtlichkeit des Tools

Die einzelnen Schritte der Methode wurden in einem Software-Tool umgesetzt, mit dem die Anwenderstudie durchgeführt wird.

Die Anwendenden führen die Methode an dem in Kapitel 7 beschriebenen Anwendungsfall Kupplung K1 Hybridgetriebe mit Hilfe einer Aufgabenstellung und einer Anleitung⁴ selbst durch. Die Anleitung beinhaltet keine spezifischen Aussagen oder Empfehlungen bezüglich der Beantwortung der Aufgabenstellung. Zu Beginn werden die Grundlagen des überwachten Lernens vorgestellt.

In dem Software-Tool sind die Schritte Messdatenaufbereitung und die Modellerstellung automatisiert. Eine Hyperparameteroptimierung wird aus Zeitgründen nicht durchgeführt, Modellparameter werden vorgegeben. Weitere Vereinfachungen der Aufgabenstellung, um die Methode durchführen zu können, sind eine allgemeine Beschreibung der Fahrmanöver und keine selbstständige Einteilung der Daten in Training und Testdaten.

Die Anleitung beinhaltet einen Fragebogen⁵ mit insgesamt 37 Fragen, die jeweils direkt nach der Durchführung einer Teilaufgabe beantwortet werden⁶. Davon sind 23 geschlossene Fragen und 14 offene Fragen. Die Lösung der Aufgabenstellung und Beantwortung der Fragen findet in schriftlicher Form und in Anwesenheit eines Interviewenden statt. Der Interviewende ist dabei kein Übermittler von Fragen, sondern nur eine Ansprechperson, falls Unklarheiten in der Aufgabenstellung entstehen oder es Probleme mit dem Software-Tool gibt. Der Interviewende verhält sich damit neutral bis weich. Die Studie dauerte jeweils etwa 60 Minuten.

Das Vorgehen zum Aufbau, Durchführung und Auswertung der Anwenderstudie orientiert sich am Leitfaden zur Anwendung empirischer Forschungsmethoden in der nutzerzentrierten Produktentwicklung nach Fotler, Germann, Gröbe-Boxdorfer, Engeln und Matthiesen (2021).

Die Anwenderstudie wird mit folgenden zwei Gruppen durchgeführt:

- Getriebeentwickelnde (n=7): Diese sind potentielle Anwendende der Methoden.

⁴ Die Anleitung und Aufgabenstellung ist im Anhang auf S.XXIX beigelegt.

⁵ Die Anwendungsfelder, sowie Vor- und Nachteile des Fragebogens als empirische Methode werden auf S.60 beschrieben.

⁶ Die Fragebogenvorlage der Studie ist im Anhang auf S.XLIV beigelegt.

- Entwickelnde mit ML-Erfahrung (n=8): Diese Gruppe besitzt eine Ingenieurs-Ausbildung und hat ML-Methoden bereits in anderen technischen Projekten eingesetzt.

Alle Befragten waren noch nicht in Kontakt mit der Methode.

Abbildung 8.1 zeigt einen Ausschnitt des Software-Tools mit Erklärungen der einzelnen Eingabe- und Ausgabefelder.

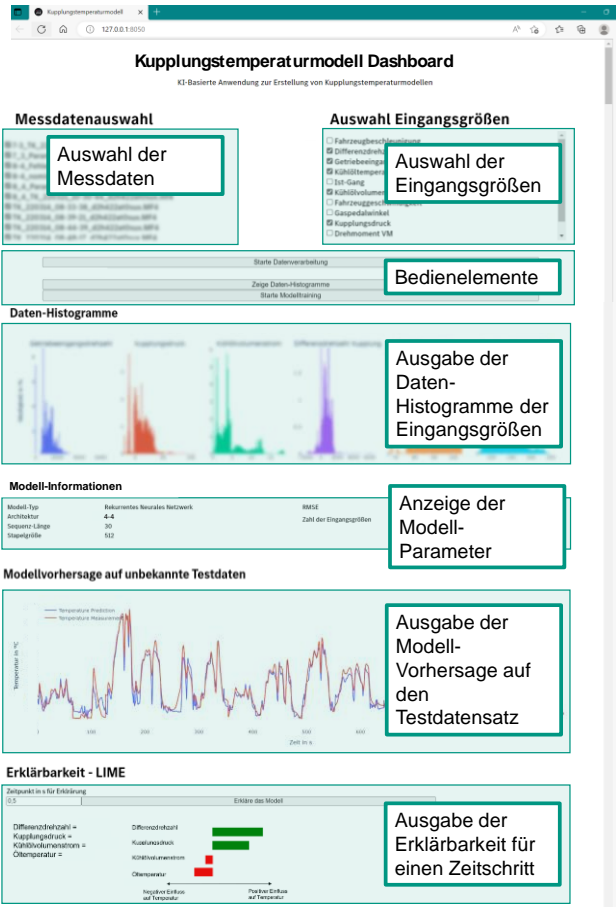


Abbildung 8.1: Ausschnitt aus dem Software-Tool der Anwenderstudie und Beschreibung der Abschnitte

8.4 Ergebnisse der Anwenderstudie

Zu Beginn der Studie wird zunächst der Kenntnisstand der Befragten in den Bereichen Fahrzeugkupplungen, Wärmehaushalt und ML zwischen „keine“ bis „sehr gut“ abgefragt (vgl. Abbildung 8.2). Die Gruppe der Entwickelnden mit ML-Erfahrung besitzt gute bis sehr gute Kenntnisse im Bereich maschinelles Lernen, wobei die Gruppe der Getriebeentwickelnden nur geringe Kenntnisse besitzt. Dies bestätigt die Einteilung der Befragten in die jeweiligen Gruppen. Im Bereich Fahrzeugkupplungen und Wärmehaushalt liegen die Kenntnisse der Getriebeentwickelnden deutlich höher.

Insgesamt erzeugen die Getriebeentwickelnden anhand der Anleitung und Aufgabenstellung in einer ersten Iteration mit dem Software-Tool Modelle mit einem um 5,3 % verbesserten RMSE im Vergleich zum physikalischen Modell, die Entwickelnden mit ML-Erfahrung um 2,5 %. Somit können beide Gruppen Modelle mit gleicher bzw. leicht höherer Modellgüte als das bisherige Modell erzeugen. Auch die ML-Entwickelnden mit einem geringen Vorwissen im Bereich Kupplungen und Wärmehaushalt erzeugen ähnlich gute Modelle wie die Getriebeentwickelnden. Hierbei muss allerdings beachtet werden, dass aus Machbarkeitsgründen die Messdaten bereits vorgegeben waren und nur die Auswahl der Modelleingangsgrößen Einfluss auf die Modellgüte hat.

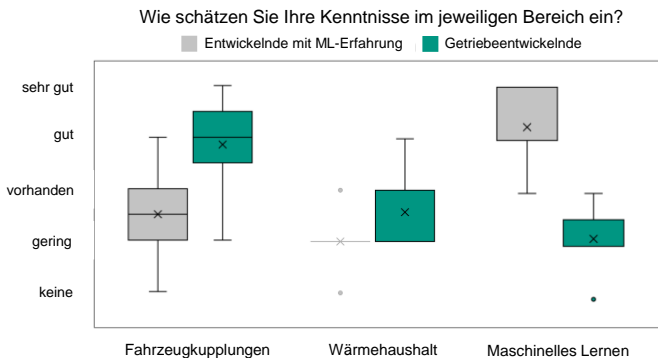


Abbildung 8.2 Kenntnisstand beider Untersuchungsgruppen

Die Ergebnisse der geschlossenen Fragen der beiden Gruppen zeigt Abbildung 8.3 im Vergleich. Diese sind dabei in 15 Fragen zur Methode mit den Antwortmöglichkeiten „trifft nicht zu“ bis „trifft voll zu“, und in 7 Fragen zur Ein- und Ausgabe-Maske des Tools aufgeteilt.

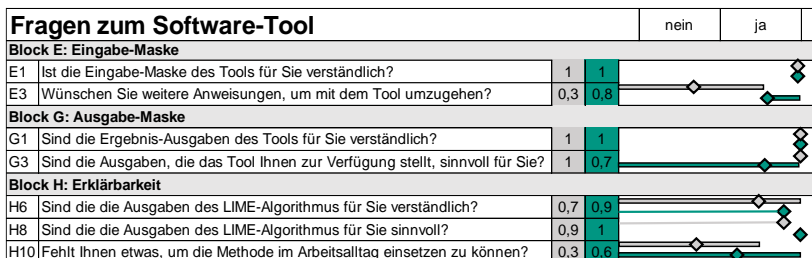
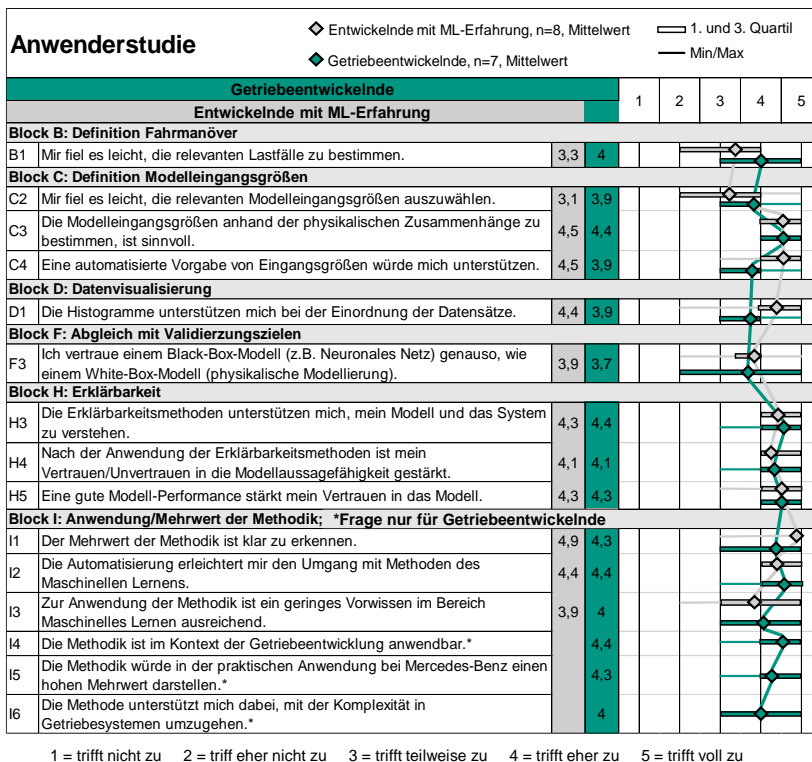


Abbildung 8.3 Ergebnisse Fragebogen zur Anwendbarkeit, beide Gruppen im Vergleich

Über alle 15 Fragen zur Methode sind die Mittelwerte der Antworten größer als 3. Der niedrigste Mittelwert liegt bei 3,13. Durch die positive Formulierung der Fragen erhält die Methode in allen Aspekten mindestens teilweise Zustimmung. 10 der 15

Fragen haben einen Mittelwert von mindestens 4, was einer überwiegenden Zustimmung entspricht.

Eine einfache Bestimmung der relevanten Fahrmanöver sowie der Modelleingangsgrößen stimmen beide Gruppen nur teilweise bis eher zu, wobei die Getriebeentwickelnden mit einem Mittelwert von 4 eher zustimmen.

Beide Gruppen sehen es als sinnvoll an, die Modelleingangsgrößen über die physikalischen Zusammenhänge zu bestimmen (Mittelwert 4,43 und 4,5). Gleichzeitig würde eine automatisierte Vorgabe der Eingangsgrößen beide Gruppen unterstützen, wobei die ML-Entwickelnden einen Mittelwert größer 4 angeben.

In den Histogrammen sehen beide Gruppen ebenfalls eine Unterstützung, wobei die Antwort der Gruppe ML höher liegt (Mittelwert 4,38).

Gleichem Vertrauen in ein Black-Box-Modell wie in ein physikalisch modelliertes White-Box-Modell stimmen beide Gruppen mit einem Mittelwert von 3,71 und 3,88 teilweise bis eher zu.

Den Fragen zur Erklärbarkeit stimmen beide Gruppen mit Mittelwerten größer 4 zu. Dazu gehören die Unterstützung durch Erklärbarkeitsmethoden im Verständnis des Modells und betrachteten Systems, sowie der Bestärkung des Vertrauens oder Misstrauen in die Modellaussagefähigkeit. Das Vertrauen in ein Modell wird bei beiden Gruppen durch eine gute Modellperformance gestärkt.

Beide Gruppen erkennen den Mehrwert der Methode und sehen in der Automatisierung einzelner Methodenschritte einen erleichterten Umgang mit ML-Methoden. Die Frage, ob zur Anwendung der Methode ein geringes Vorwissen im Bereich ML ausreichend ist, beantworten die Getriebeentwickelnden mit einem Mittelwert von 4.

Die Anwendbarkeit der Methode wird von den Getriebeentwickelnden klar bestätigt (Mittelwert 4,43). Einen hohen Mehrwert der praktischen Anwendung im Unternehmen stimmen die Getriebeentwickelnden ebenfalls zu (Mittelwert 4,29). Eine Unterstützung mit dem Umgang der Komplexität in Getriebesystemen wird abschließend bestätigt (Mittelwert 4).

Neben den geschlossenen Fragen zur Methode werden 7 ja/nein-Fragen und 6 offene Fragen zur Ein- und Ausgabe-Maske des Software-Tools, sowie 3 offene Fragen zu Stärken der Methode, Verbesserungspotentiale, und Schwierigkeiten bei der Methodenanwendung gestellt.

Alle Befragten verstehen die Eingabe-Maske, wobei sich ein Großteil der Getriebeentwickelnden weitere Anweisungen wünscht, um mit dem Tool umzugehen. Als weitere Informationen zum Verständnis der Eingabe-Maske und Anweisungen zum Umgang mit dem Tool gaben zwei Getriebeentwickelnde an, dass eine geführte Bedienung in Form einer Schrittanleitung auf der Oberfläche hilfreich wäre. Ein Entwickler mit ML-Erfahrung schlug Erklärungsfenster für die Modellparameter vor. Außerdem gab je ein ML-Entwickler an, dass Anweisungen zur Messdatenauswahl sowie Hinweise zur Auswahl der Modelleingangsparameter unterstützen würden.

Für fast alle Teilnehmenden beider Gruppen ist die Ausgabe-Maske verständlich und deren Ausgaben sinnvoll. Für die Ausgaben der Erklärbarkeitsmethode trifft dies ebenfalls zu. Bei der Frage, ob zum Einsatz des Tools im Arbeitsalltag etwas fehlt, gaben die Getriebeentwickelnden ein geteiltes Meinungsbild ab, vor allem die Ausgabe-Maske ist noch zu verbessern.

Ein Hauptpunkt zur besseren Anwendbarkeit der Ausgabe-Maske von Seiten dreier Getriebeentwickelnde ist, dass die Eingangsgrößen des Testdatensatzes in einem einzigen Graphen dargestellt werden sollten, damit Zusammenhänge zwischen den Größen sofort erkannt werden und somit die Ergebnisse des Modells besser plausibilisiert werden können. Auch die Aufteilung in Trainings-, Validierungs- und Testdaten muss deutlicher gemacht werden. Diesen Punkt gab auch ein ML-Entwickler an. Zwei ML-Entwickler nannten auch die Ausgabe einer globalen Wichtigkeit der Eingangsgrößen als hilfreich. Ein anderer ML-Entwickler gibt den Wunsch an, mehrere Testdatensätze auswählen zu können.

Speziell bei der Erklärbarkeitsausgabe ist für einen Getriebeentwickelnden die Farbgebung zunächst nicht verständlich, da ein Einfluss auf das Ansteigen der Temperatur grün hinterlegt ist und ein Abkühlen rot. Die Erwartungshaltung war umgekehrt. Bei einer LIME-Analyse für einen Anwendungsfall mit vielen notwendigen Signalen wird eine Verifikation der Ergebnisse zu einem Zeitpunkt recht schwierig. Für einen ML-Entwickler ist die Angabe der Intervalle verwirrend, da es ist nicht direkt ersichtlich ist, wie sich der Einfluss ändert, wenn man außerhalb der Intervalle liegt. Auch die Benennung der Zeitintervalle mit $t-1$ war für einen weiteren ML-Entwickler nicht direkt verständlich.

Die Stärken der Methode sehen drei Getriebeentwickelnde in der Steigerung der Modellgüte. Außerdem können damit sehr komplexe Zusammenhänge abgebildet werden, was mit einem physikalischen Modell nicht möglich ist, und eine Kenntnis der Regelstrecke ist nicht zwingend notwendig, um ein System zu beschreiben. Drei Getriebeentwickelnde sehen die Stärke in der Reduzierung von Zeit und Kosten, da

weniger Iterationsschritte benötigt werden und die Komplexität an der Applikation der physikalischen Modelle vereinfacht wird. Ein Getriebeentwickler nennt die Möglichkeit eine große Anzahl an Messdaten zu analysieren und idealerweise Eingangsgrößen zu finden, die relevant sind, aber bisher nicht im Fokus stehen.

Auch drei der ML-Entwickler sehen die Stärke in der höheren Modellgüte. Zwei ML-Entwickler nennen den Vorteil der Modellgrundlage auf Fahrzeugmessdaten, was zu einer hohen Modellgüte im Serienfahrzeug führt. Drei nennen den hohen Automatisierungsgrad. Weitere Vorteile sind, dass kein oder nur wenig Kupplungs- und Getriebeexpertenwissen notwendig ist, sowie die Einflussfaktoren auf die Temperatur gut nachvollziehbar sind. Allerdings muss beachtet werden, dass bei Programmfehlern Expertenwissen zur Fehlerbehebung notwendig ist.

Vier der sieben Getriebeentwickler sehen die größte Schwierigkeit zur Anwendung der Methode darin, dass bei einer Änderung der Hardware neue Trainingsdaten erzeugt werden müssen. Zwei Entwickler nennen die Überprüfung der relevanten Fahrmanöver auf Vollständigkeit. Zwei weitere Entwickler nennen die Validierung des Black-Box-Modells. Vor allem bei sehr komplexen physikalischen Phänomenen sollte ein hohes technisches Hintergrundwissen vorhanden sein. Ein Entwickler nennt die hohe Komplexität der Erklärung, wenn ein Modell mit vielen Eingangsgrößen erklärt wird. Außerdem muss für die konstruktive Auslegung der Kupplung ein physikalisches Modell vorhanden sein, da erst nach dem Aufbau eines ersten Kupplungssystems Fahrzeugmessungen gemacht werden können.

Zwei ML-Entwickler sehen die Akzeptanz der Methode und des Black-Box-Modells bei KI-fremden Personen als Schwierigkeit an. Auch die Validierung in Form eines Standardprozesses für Absicherung von Automotive KI und dem Umgang mit Ausreißern und außergewöhnlichen Eingangssignalen wird als Schwierigkeit genannt. Zudem sollte auch die Auswahl der Eingangsparameter unterstützt werden.

Für die Verbesserung der Methode schlagen zwei Getriebeentwickler die Auswahl und Gewichtung der Messdaten vor, da falsche oder einseitige Messdaten zu schlechten Ergebnissen führen können. Außerdem soll für die Eingangsgrößen die direkte Rückmeldung gegeben werden, welche Signale den größten Effekt auf die Zielgröße haben. Ein ML-Entwickler schlägt zur Verbesserung eine Erklärung der KI-Qualitätskriterien für KI-fremde Personen vor, ein weiterer eine Prüfung, ob für Vorhersagebereiche Trainingsdaten vorliegen. Um Ergebnisse und Daten einfach weiternutzen zu können, wird deren Exportierbarkeit in andere Formate vorgeschlagen.

8.5 Zwischenfazit

In Kapitel 8 wurden Studien zur Anwendbarkeit durchgeführt, um Forschungsfrage IV (FF-IV) zu beantworten.

Forschungsfrage IV (FF-IV)

Wie wird die Anwendbarkeit der entwickelten Methoden des maschinellen Lernens durch die Entwickelnden bewertet?

Mit den Methoden Ansteuerungsoptimierung mit verstärkendem Lernen und Datenanalyse mit Cluster-Algorithmen wird eine Evaluation in Form einer Befragungsstudie mit Getriebeentwickelnden ($n=10$, $n=9$) durchgeführt.

Die Methode der Ansteuerungs-Optimierung erfährt hinsichtlich Logik und Verständnis eine breite Zustimmung. Das Verständnis und das Zutrauen einer eigenen Anwendung fällt für den Methodenschritt Definition Optimierungsziel geringer aus. Hier geben die Befragten auch an, dass das gegeneinander Auftragen der beiden Zielgrößen in der späteren Auswertung, z. B. in Form einer Pareto-Front, das Verständnis verbessern würde. In der Definition der Belohnungsfunktion liegt ebenfalls geringes Zutrauen vor. Schwierigkeiten sehen die Befragten in der Verfügbarkeit eines Simulationsmodells mit geeigneter Abbildungsgüte, weshalb für dessen Auswahl ein Gütekriterium vorgeschlagen wird.

Auch die Methode der Messdatenanalyse erfährt hinsichtlich Logik und Verständnis eine breite Zustimmung. Lediglich in den beiden Methodenschritten Korrelationsanalyse und Gesamtvisualisierung sehen die Befragten den Mehrwert, trauen sich aber die Anwendung nicht zu und das Verständnis fällt etwas geringer aus. Dies kann darin begründet sein, dass die beiden Methodenschritte im gezeigten Anwendungsbeispiel eine mit den anderen Methodenschritten vergleichbare hohe Informationsdichte und somit Komplexität aufweisen. Hier könnten weitere Anwendungsbeispiele und Übungen unterstützen. Außerdem können eine gute graphische Aufbereitung und Darstellung der Zusammenhänge in den Abbildungen die Auswertung vereinfachen, was als Verbesserungspotential genannt wird.

Für die Methode Modellierung von virtuellen Sensoren mit KNN wurde eine Anwenderstudie anhand eines Software-Tools sowie einer Aufgabenstellung und einer Anleitung durchgeführt. Neben Getriebeentwickelnden ($n=7$) nehmen an der

Anwenderstudie auch eine Gruppe von Entwickelnden mit ML-Erfahrung (n=8) teil. Beide Gruppen konnten anhand der Anleitung und Aufgabenstellung mit dem Software-Tool Modelle mit höherer Abbildungsgüte als ein physikalisches Modell erzeugen. Die Anwendbarkeit der Methode wird von den Getriebeentwickelnden klar bestätigt. Einem hohen Mehrwert der praktischen Anwendung im Unternehmen, sowie einer Unterstützung mit dem Umgang der Komplexität in Getriebesystemen stimmen die Entwickelnden ebenfalls zu. Die Unterstützung durch Erklärbarkeitsmethoden zum Verständnis des Modells und des betrachteten Systems, die Bestärkung des Vertrauens oder Misstrauens in die Modellaussagefähigkeit, sowie die Stärkung in das Modellvertrauen durch eine hohe Modellgüte wird von beiden Gruppen bestätigt. Beide Gruppen sehen in einer automatisierten Eingangsgrößenauswahl eine Unterstützung. Die Getriebeentwickelnden sehen die größten Schwierigkeiten in der Anwendung der Methode in einer erneuten Generierung von Trainingsdaten bei einer Änderung der Hardware, Überprüfung der relevanten Fahrmanöver auf Vollständigkeit und umfassende Validierung des Black-Box-Modells. Auch die ML-Entwickelnden sehen Schwierigkeiten in der umfassenden Validierung des Modells. Verbesserungspotentiale sehen die Getriebeentwickelnden in der Auswahl und Gewichtung der Trainingsdaten.

Bei den Ergebnissen der Anwenderstudie muss beachtet werden, dass diese nur mit Getriebeentwickelnden sowie Entwickelnden mit ML-Erfahrung der Mercedes-Benz Group AG durchgeführt wurde. Eine Übertragbarkeit der Ergebnisse auf andere Unternehmen ist denkbar, muss aber aufgrund von unternehmensspezifischen Faktoren und Marktbedingungen kritisch geprüft werden.

9 Zusammenfassung und Ausblick

Dieses Kapitel fasst die Ergebnisse der Forschungsarbeit zusammen und gibt einen Ausblick auf Anknüpfungspunkte für weitere Arbeiten. Diese Arbeiten können das geschaffene Wissen als Ausgangsbasis nutzen und die gezeigten Ansätze erweitern.

9.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde der Einsatz von ML-Methoden in hybriden Fahrzeuggetrieben am Beispiel des Doppelkupplungsgetriebes untersucht. Ziel war es, Entwickelnde mit dem Einsatz dieser Methoden in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen.

Um diese Zielsetzung zu erreichen, wurden vier Forschungsfragen formuliert:

- *Welche Herausforderungen existieren in der Entwicklung von Steuerungs- und Regelungsalgorithmen von parallelen hybriden Fahrzeuggetrieben?*
- *Wie müssen die Methoden des maschinellen Lernens gestaltet werden, um Entwickelnde bei der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben zu unterstützen?*
- *Welche messbaren Mehrwerte und Potentiale ergeben sich durch die Methoden des maschinellen Lernens?*
- *Wie wird die Anwendbarkeit der entwickelten Methoden des maschinellen Lernens durch Entwickelnde bewertet?*

Um die erste Forschungsfrage zu beantworten, wurden die Herausforderungen in der Funktionsentwicklung von parallelen hybriden Fahrzeuggetrieben anhand zweier Methoden abgeleitet: leitfadengeführte Experteninterviews mit 14 Getriebeentwickelnden und einer Systemanalyse unterstützt durch den C&C²-Ansatz. Die Herausforderungen umfassen die Fahrfunktionen Motorwiederstart, hybrides Anfahren sowie die hohe Komplexität des Antriebsstrangs, die zu einer komplexen Auslegung der Steuerung und Regelung und Modellierung von virtuellen Sensoren führt.

Bei den Interviewergebnissen muss beachtet werden, dass diese ausschließlich innerhalb des Unternehmens durchgeführt wurden. Außerdem entwickeln die Befragten Doppelkupplungsgetriebe, weshalb sich ihre Aussagen auf dieses System beziehen. Eine Übertragbarkeit der Ergebnisse auf andere Unternehmen ist denkbar, muss aber aufgrund von unternehmensspezifischen Faktoren und Marktbedingungen kritisch geprüft werden. Die Systemanalyse wurde an dem Beispiel eines hybriden Doppelkupplungsgetriebes des Unternehmens durchgeführt. In weiteren oder unterschiedlichen Bauformen eines parallelen Hybridgetriebes können weitere Herausforderungen auftreten.

Zur Beantwortung der zweiten Forschungsfrage wurden die drei zentralen ML-Methoden: überwachtes Lernen, nicht überwachtes Lernen und verstärkendes Lernen an den Kontext der Funktionsentwicklung paralleler hybrider Fahrzeuggetriebe adaptiert und daraus drei konkrete Vorgehensmodelle abgeleitet. Es wurden dabei außerdem eine Beschreibung der Auswahl komplexer Algorithmen und Optimierungsverfahren zur Umsetzung der Methoden ergänzt. Zu den drei Methoden gehören: Virtuelle Sensoren mit KNN, Steuerungs-Optimierung mit verstärkendem Lernen und Messdatenanalyse mit Cluster-Algorithmen.

Zur Evaluation der Methoden wurden diese auf praxisrelevante Entwicklungsaufgaben in der Forschungsumgebung angewendet, um Forschungsfrage drei zu beantworten.

Mit Hilfe der Methode Modellierung von virtuellen Sensoren mit KNN wurde ein Kupplungstemperaturmodell der Fahrkupplung K1 eines hybriden Getriebes auf Basis von Fahrzeugmessdaten modelliert. Die Abbildungsgüte konnte im Vergleich zum aktuell verwendeten physikalisch modellierten Modell im Mittelwert um 15,5 % verbessert werden. Erklärbarkeitsansätze unterstützen Entwickelnde in der Modellierung dieser Black-Box-Modelle. Die Anwendung der Methode auf das Temperaturmodell der Trennkupplung K0 und der Kupplung K1 aus dem konventionellen Getriebe erzielten ebenfalls eine Verbesserung von 22,5% und 18,5 %. Die Methode konnte damit auf weitere Systemgenerationen übertragen werden. Damit wird das Erfolgskriterium Steigerung der Modellgüte erfüllt. Auch die Anzahl der Parameter der Modelle liegt in der Größenordnung des physikalischen Modells, womit auch das Kriterium der Steuergerätefähigkeit hinsichtlich der Speicheranforderung erfüllt ist.

Die Methode Steuerungs-Optimierung mit verstärkendem Lernen wurde auf den Vorgang der Kupplungsfüllung angewendet. Die Zielgröße Druckeinbruch konnte mit einer parameterfreien Ansteuerung um bis zu 86 % bei gleichbleibender Füllzeit im Vergleich zur Ansteuerung eines Entwickelnden reduziert werden. Ein

Genetischer Algorithmus, der Parameter der Ansteuerung optimiert, konnte den Druckeinbruch um bis zu 91 % reduzieren. Die Optimierungsergebnisse wurden abschließend erfolgreich im Fahrzeug validiert. Auch hier wird damit das entsprechende Erfolgskriterium Steigerung der Ansteuerungsgenauigkeit erfüllt.

Die Methode Messdatenanalyse mit Cluster-Algorithmen wurde auf Fahrzeugmessdaten zur Klassifizierung des hybriden Motorwiederstarts angewendet. Eine vorgelagerte Korrelationsanalyse der Kennzahlen bestätigt das bisherige Funktionsverständnis. Mit dem Cluster-Ansatz konnten aus den 278 detektierten Motorstarts vier Motorzustart-Cluster erfolgreich erkannt und über Kennzahlen charakterisiert werden. Durch den Vergleich der Cluster und deren Charakteristiken konnten Verbesserungspotentiale der Applikation aufgezeigt werden. In diesem Anwendungsbeispiel wurde das Potential aufgezeigt, das Cluster-Algorithmen bei der Auswertung von Messdaten bieten können, womit das Erfolgskriterium erfüllt wird.

Alle drei Methoden bieten den Vorteil der Automatisierung, Skalierung und einfacher Implementierung bei gleichzeitig hoher Funktionalität durch vordefinierte Bibliotheken der Programmiersprachen Python (oder auch R, C++ oder Matlab). Durch die Automatisierung kann sich die Entwicklungsarbeit auf die manuellen Schritte der Methoden fokussieren.

Die ML-Methoden wurden erfolgreich auf je eine Entwicklungsaufgabe innerhalb der Forschungsumgebung angewendet. Die vergleichbar erfolgreiche Anwendung auf eine Vielzahl von Aufgabenstellungen muss noch gezeigt werden. Es kann aber angenommen werden, dass die adaptierten Vorgehensmodelle eine geeignete Grundlage für die Übertragbarkeit auf weitere Anwendungsfälle bilden. Die potentiellen Erfolgskriterien der Reduzierung von Entwicklungszeiten und –kosten wurden in dieser Arbeit nicht untersucht. Weitere Forschungsarbeiten könnten diese Aspekte untersuchen.

Zur Beantwortung der vierten und letzten Forschungsfrage wurden drei Studien zur Anwendbarkeit der entwickelten Methoden in der Forschungsumgebung durchgeführt. In einer Evaluation in Form einer Befragungsstudie mit zehn bzw. neun Getriebeentwickelnden mit den Methoden Ansteuerungsoptimierung mit Verstärkendem Lernen und Datenanalyse mit Cluster-Algorithmen werden für die beiden Methoden das Methodenverständnis und die Methodenakzeptanz bestätigt. Für die Methode Modellierung von virtuellen Sensoren mit KNN wurde eine Anwenderstudie anhand eines Software-Tools sowie einer Aufgabenstellung und einer Anleitung mit sieben Getriebeentwickelnden sowie acht Entwickelnden mit ML-Erfahrung durchgeführt. Auch hier wurden das Methodenverständnis und die

Methodenakzeptanz bestätigt und einzelne Verbesserungspotentiale aufgezeigt. Entwickelnde mit ML-Erfahrung wie auch Getriebeentwickelnde konnten anhand des Software-Tools Kupplungstemperaturmodelle mit hoher Modellgüte erzeugen.

Die Anzahl an Teilnehmenden an den Studien von sieben bis zehn Personen je Methode und Gruppe ist verhältnismäßig gering. Außerdem wurden in der Anwenderstudie mit Software-Tool einzelne Vereinfachungen getroffen, um die Methode in einem überschaubaren zeitlichen Rahmen durchführen zu können. Für die initiale Untersuchung der Anwendbarkeit war dies jedoch ausreichend. Weitere Studien mit einer größeren Teilnehmerzahl sowie eine vollständige Durchführung aller Methodenschritte können hieran anknüpfen.

9.2 **Ausblick**

Die Ergebnisse dieser Arbeit eröffnen fünf große Themenblöcke als Anknüpfungspunkte für weitere Forschungsarbeiten:

- Anwendung der gezeigten Methoden auf weitere Beispiele in hybriden und elektrischen Antriebssträngen,
- Untersuchung der Entwicklungszeiten und –kosten,
- Integration von Neuronalen Netzen auf Getriebesteuergeräten,
- Validierung und Verifikation von ML-Modellen und
- Sicherheitskonzepte für den Einsatz von ML-Modellen.

In dieser Arbeit wurden die drei entwickelten Methoden auf jeweils eine Entwicklungsaufgabe eines hybriden Doppelkupplungsgetriebes angewendet. Die Übertragung dieser Methoden auf weitere Anwendungsfälle in hybriden Getrieben sowie die Erweiterung auf elektrische Antriebssysteme gilt es durchzuführen. Virtuelle Sensoren für Druck- oder Drehmomentenberechnungen sind denkbar. Der Einsatz eines trainierten Agenten auf einem Getriebesteuergerät hat das Potential den Applikationsaufwand zu reduzieren. Altenburg forscht am IPEK hierbei an geeigneten Validierungsumgebungen. Die Analyse von Entwicklungsdaten kann auf die Analyse von Felddaten ausgeweitet werden.

Ein interessanter Aspekt durch den Einsatz von ML-Methoden ist eine mögliche Reduzierung der Entwicklungszeiten und -kosten aufgrund des datenbasierten Charakters und hohen Automatisierungsgrad der Methoden. Diese Ergebnisse sind auch von hoher Relevanz, um den Einsatz von ML-Methoden für das Management mit unternehmerischen Kennwerten bewerten zu können.

Eine Aufgabe der Entwicklung ist die Integration von Neuronalen Netzen auf dem Getriebesteuergerät. Speicherbedarf und Rechenzeiten sind hierbei wichtige Größen. Da Neuronale Netze grundsätzlich Matrix-Multiplikationen sind, ist die Integrationsfähigkeit anzunehmen, aber in diesem konkreten Anwendungsfall noch nachzuweisen.

ML-Algorithmen, die direkt im Fahrzeug auf einem Getriebesteuergerät eingesetzt werden, müssen auch die Anforderungen an Funktionaler Sicherheit nach DIN 26262 erfüllen. In dieser Arbeit betrifft dies die beschriebenen virtuellen Sensoren. Aufgrund der Black-Box-Eigenschaften dieser Modelle sind erste Diskrepanzen zu erkennen. Aktuelle Normen wie die IEC 61508 schließen eine Verwendung nicht aus, gleichzeitig wird diese auch nicht empfohlen (Maack, Bertovic & Radtke, 2020, S. 84). Es werden Sicherheitskonzepte benötigt, die diese Vereinbarkeit ermöglichen.

Zur Validierung und Verifikation der Black-Box-Modelle wurden neben der Vorhersage auf Testdaten eine Sensitivitätsanalyse und eine LIME-Analyse unterstützend eingesetzt. Dies sind erste Schritte, mit diesem Verhalten umzugehen. Das Forschungsfeld der „Explainable AI“ beinhaltet noch weitere Methoden, mit denen das intransparente Verhalten von virtuellen Sensoren untersucht werden kann.

Literaturverzeichnis

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C. et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.
- Albers, A. (2010). Five hypotheses about engineering processes and their consequences. *Proceedings of the TMCE, Vol. 12*, 343–356.
- Albers, A., Alink, T., Matthiesen, S. & Thau, S. (2008). Support of system analyses and improvement in industrial design through the contact & channel model. *DS 48: Proceedings DESIGN 2008, the 10th International Design Conference*.
- Albers, A., Behrendt, M., Klingler, S. & Matros, K. (2016). *Verifikation und Validierung im Produktentstehungsprozess*. München: Carl Hanser Verlag.
- Albers, A. & Braun, A. (2011). A generalised framework to compass and to support complex product engineering processes. *International Journal of Product Development*, 15(1-3), 6–25.
- Albers, A., Burkardt, N. & Duser, T. (2006). Competence-profile oriented education with the Karlsruhe education model for product development (KaLeP). *World Transactions on Engineering and Technology Education*, 5(2), 271.
- Albers, A., Bursac, N. & Wintergerst, E. (2015). Produktgenerationsentwicklung - Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In *Stuttgarter Symposium für Produktentwicklung (SSP) : Stuttgart, 19. Juni 2015 ; Hrsg.: H. Binz* (S. 1–10). Fraunhofer Verlag.
- Albers, A., Deters, L., Feldhusen, J., Leidich, E., Linke, H., Poll, G. et al. (2018). *Konstruktionselemente des Maschinenbaus 2: Grundlagen von Maschinenelementen für Antriebsaufgaben*. Berlin: Springer.
- Albers, A. & Düser, T. (2010). Implementation of a Vehicle-in-the-Loop Development and Validation Platform. In *Proceedings of the 33rd FISITA World Automotive Congress* (S. 1–10).

- Albers, A., Ebel, B. & Lohmeyer, Q. (2012). Systems of objectives in complex product development. In *Proceedings of TMCE* (S. 7–11).
- Albers, A., Ebertz, J., Rapp, S., Heimicke, J., Kürten, C., Zimmermann, V. et al. (2020). *Produktgeneration 1 im Modell der PGE-Produktgenerationsentwicklung: Verständnis, Zusammenhänge und Auswirkungen in der Produktentwicklung*. Karlsruher Institut für Technologie (KIT).
- Albers, A., Fahl, J., Hirschter, T., Endl, M., Ewert, R. & Rapp, S. (2020). Model of PGE–Product Generation Engineering by the Example of Autonomous Driving. *Procedia CIRP*, 91, 665–677.
- Albers, A., Haug, F., Heitger, N., Arslan, M., Rapp, S. & Bursac, N. (2016). Produktgenerationsentwicklung-Praxisbedarf und Fallbeispiel in der automobilen Produktentwicklung. In *12. Symposium für Vorausschau (SVT)*. Heinz-Nixdorf-Institut.
- Albers, A., Hirschter, T., Fahl, J., Wöhrle, G., Reinemann, J., Rapp, S. et al. (2020). Generic reference product model for specifying complex products by the example of the automotive industry. *Digital Proceedings of TMCE*, 353–370.
- Albers, A., Matros, K., Behrendt, M. & Jetzinger, H. (2015). Das Pull-Prinzip der Validierung–Ein Referenzmodell zur effizienten Integration von Validierungsaktivitäten in den Produktentstehungsprozess. *VDI Konstruktion*, 6, 74–81.
- Albers, A. & Rapp, S. (2021). Model of SGE: system generation engineering as basis for structured planning and management of development. In *Design Methodology for Future Products: Data Driven, Agile and Flexible* (S. 27–46). Springer.
- Albers, A., Rapp, S., Fahl, J., Hirschter, T., Revfi, S., Schulz, M. et al. (2020). Proposing a generalized description of variations in different types of systems by the model of PGE–Product Generation Engineering. In *Proceedings of the Design Society: DESIGN Conference* (Bd. 1, S. 2235–2244). Cambridge University Press.

- Albers, A., Rapp, S., Spadinger, M., Richter, T., Birk, C., Marthaler, F. et al. (2019). The reference system in the model of PGE: proposing a generalized description of reference products and their interrelations. In *Proceedings of the design society: international conference on engineering design* (Bd. 1, S. 1693–1702). Cambridge University Press.
- Albers, A. & Wintergerst, E. (2014). The Contact and Channel Approach (C&C 2-A): Relating a system's physical structure to its functionality. In *An anthology of theories and models of design: Philosophy, approaches and empirical explorations* (S. 151–171). Springer.
- Altenburg, S., Bause, K. & Albers, A. (2022). Validation Environment for Deep Reinforcement Learning Based Gear Shift Controllers. In *22. Internationales Stuttgarter Symposium* (S. 354–368). Springer.
- Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E. et al. (2019). Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (S. 291–300). IEEE.
- Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. (2017). Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 34(6), 26–38. <https://doi.org/10.1109/MSP.2017.2743240>
- Bai, S., Kolter, J. Z. & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- Basiewicz, M. (2020). Ein Beitrag zur Validierung nasslaufender Lamellenpakete aus Anfahrerelementen von Fahrzeugen im Betriebszustand „geregelter Dauerschlupf“. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK- Institut für Produktentwicklung* (Bd. 128). ISSN: 1615-8113. Karlsruhe: Karlsruher Institut für Technologie (KIT).
- Bernhardt, J. (2017). Entwicklung von Friktionssystemen am Beispiel einer nasslaufenden Lamellenkupplung. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 102). ISSN: 1615-8113. Karlsruhe: Karlsruher Institut für Technologie (KIT).

- Birkhold, J.-M. (2013). Komfortobjektivierung und funktionale Bewertung als Methoden zur Unterstützung der Entwicklung des Wiederstartsystems in parallelen Hybridantrieben. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 60). ISSN: 1615-8113. Karlsruhe.
- Biswas, A., Anselma, P. G. & Emadi, A. (2019). Real-Time Optimal Energy Management of Electrified Powertrains with Reinforcement Learning. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)* (S. 1–6).
- Blessing, L. T. M. & Chakrabarti, A. (2009). *DRM: A design research methodology*. London: Springer. https://doi.org/10.1007/978-1-84882-587-1_2
- Bock, F., Sippl, C., Siegl, S. & German, R. (2019). Status report on automotive software development. In *Automotive Systems and Software Engineering* (S. 29–57). Springer.
- Bovee, K. & Rizzoni, G. (2016). Model-based torque shaping for smooth acceleration response in hybrid electric vehicles. *IFAC-PapersOnLine*, 49(11), 525–532.
- Braun, J., Künnemann, S., Rich, B., Mißler-Behr, M. & Woll, R. (2020). Veränderung von Kundenanforderungen in der Automobilindustrie—Analyse von Experteninterviews. In *Datengetriebenes Qualitätsmanagement* (S. 144–161). Springer.
- Cabrera Cano, M. (2017). *Neuronale Netze mit externen Laguerre-Filtern zur automatischen numerischen Vereinfachung von Getriebemodellen* (Bd. 52). Karlsruhe: KIT Scientific Publishing.
- Cao, M., Wang, K. W., Fujii, Y. & Tobler, W. E. (2004). Advanced hybrid neural network automotive friction component model for powertrain system dynamic analysis. Part 1: model development. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 218(8), 831–843.
- Cavina, N., Olivi, D., Corti, E., Poggio, L. & Marcigliano, F. (2012). Development of a dual clutch transmission model for real-time applications. *IFAC Proceedings Volumes*, 45(30), 440–447.

- Chapman, P., Clinton, J., Khabaza, T., Reinartz, T. & Wirth, R. (1999). The CRISP-DM process model. *The CRIP-DM Consortium*, 310, 91.
- Ćirović, V., Smiljanić, D. & Aleksendrić, D. (2014). Neuro-genetic optimization of disc brake performance at elevated temperatures. *FME Transactions*, 42(2), 142–149.
- Dörr, M., Ries, M., Gwosch, T. & Matthiesen, S. (2019). Recognizing product application based on integrated consumer grade sensors: a case study with handheld power tools. *Procedia CIRP*, 84, 798–803.
- Duboue, P. (2020). *The art of feature engineering: essentials for machine learning*. Cambridge University Press. <https://doi.org/10.1017/9781108671682>
- Ebel, B. (2015). Modellierung von Zielsystemen in der interdisziplinären Produktentstehung. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 85). ISSN: 1615-8113. Karlsruhe: Karlsruher Institut für Technologie (KIT).
- Ehrlenspiel, K. (2009). *Integrierte Produktentwicklung*. München: Carl Hanser Verlag.
- Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96-Proceedings* (Bd. 96, S. 226–231).
- Ester, M. & Sander, J. (2013). *Knowledge discovery in databases: Techniken und Anwendungen*. Berlin: Springer-Verlag.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37.
- Fischer, R., Küçükay, F., Jürgens, G. & Pollak, B. (2016). *Das Getriebebuch* (2. Aufl.). Wiesbaden: Springer. <https://doi.org/10.1007/978-3-658-13104-3>
- Förster, D., Inderka, R. B. & Gauterin, F. (2020). Data-Driven Identification of Characteristic Real-Driving Cycles Based on k-Means Clustering and Mixed-Integer Optimization. *IEEE Transactions on Vehicular Technology*, 69(3), 2398–2410. <https://doi.org/10.1109/TVT.2019.2963272>

- Fotler, D., Germann, R., Gröbe-Boxdorfer, B., Engeln, W. & Matthiesen, S. (2021). *Leitfaden zur Anwendung empirischer Forschungsmethoden in der nutzerzentrierten Produktentwicklung : Forschung mit und an Menschen in den Ingenieurwissenschaften*. Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000132655>
- Gad, A. F. (2021). Pygad: An intuitive genetic algorithm python library. *arXiv preprint arXiv:2106.06158*.
- Gaiselmann, G., Altenburg, S., Studer, S. & Peters, S. (2022). Deep reinforcement learning for gearshift controllers in automatic transmissions. *Array*, 15, 100235.
- Geering, H. P. (2013). *Regelungstechnik: Mathematische Grundlagen, Entwurfsmethoden, Beispiele*. Berlin: Springer-Verlag.
- Haarnoja, T., Tang, H., Abbeel, P. & Levine, S. Reinforcement learning with deep energy-based policies. In *International conference on machine learning* (S. 1352–1361). PMLR.
- Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (S. 1861–1870). PMLR.
- Harold, C. K. D., Prakash, S. & Hofman, T. Powertrain Control for Hybrid-Electric Vehicles Using Supervised Machine Learning. In *Vehicles* (Bd. 2, S. 267–286).
- Heyden, M., Wilwer, J., Fouché, E., Thoma, S., Matthiesen, S. & Gwosch, T. (2022). Tandem Outlier Detectors for Decentralized Data. In *Proceedings of the 34th International Conference on Scientific and Statistical Database Management* (S. 1–4).
- Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. In *Neural Computation* (Bd. 9, S. 1735–1780). MIT press.
- Hofmann, P. (2014). *Hybridfahrzeuge: ein alternatives Antriebssystem für die Zukunft* (2. Aufl.). Wien: Springer-Verlag.
- Hrisca, D. (2024). *ASAM MDF*. Zugriff am 02.10.2024. Verfügbar unter: <https://github.com/danielhrisca/asammdf>

- Isermann, R. (2007). *Mechatronische Systeme: Grundlagen*. Berlin: Springer-Verlag.
- Isermann, R. (2008). *Kompensationsregler*. Berlin: Springer-Verlag.
- Jung, S., Choi, S. B., Ko, Y., Kim, J. & Lee, H. (2019). Pressure control of an electro-hydraulic actuated clutch via novel hysteresis model. *Control engineering practice*, 91, 104112.
- Kawakami, T., Ide, T., Moriyama, E., Hoki, K. & Muramatsu, M. (2020). Development of Artificial Intelligence to Classify Quality of Transmission Shift Control Using Deep Convolutional Neural Networks. *IEEE Transactions on Vehicular Technology*.
- Kirchner, E. (2007). *Leistungsübertragung in Fahrzeuggetrieben: Grundlagen der Auslegung, Entwicklung und Validierung von Fahrzeuggetrieben und deren Komponenten*. Berlin: Springer.
- Kistner, B., Sanzenbacher, S., Munier, J. & Fietkau, P. (2019). Die digitale Antriebsentwicklung der Zukunft: ganzheitlich, systematisch und kundenzentriert. In *Experten-Forum Powertrain: Simulation und Test 2019* (S. 1–14). Springer.
- Klotzbach, S. & Henrichfreise, H. (2002). Ein nichtlineares Reibmodell für die numerische Simulation reibungsbehafteter mechatronischer Systeme. In *Symposium Simulationstechnik ASIM* (Bd. 10).
- Kniel, J. (2017). Potentiale der Temperaturverteilungsmessung zur Analyse trockenlaufender Friktionssysteme auf Basis neuer Messtechnik. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 109). ISSN: 1615-8113. Karlsruhe: Karlsruher Institut für Technologie (KIT).
- Kosfeld, R., Eckey, H. F. & Türc, M. (2016). Korrelationsanalyse. In *Deskriptive Statistik* (S. 199–224). Springer.
- Krätschmer, A., Lutchen, R. & Reuss, H.-C. (Hrsg.). (2021). *AI-Based Diagnostic Tool for Offline Evaluation of Measurement Data on Test Benches*: Springer.

- Kreutzer, R. T. & Sirrenberg, M. (2019). Grundlagen und Treiber der Künstlichen Intelligenz. In *Künstliche Intelligenz verstehen* (S. 73–105). Springer.
- Lampe, A., Serway Roland, Siestrup, L. G. & Gühmann, C. (2019). Artificial Intelligence in Transmission Control. Clutch Engagement with Reinforcement Learning. *VDI-Berichte*, (2354), 113–132.
- Lemieux, G. & Lewis, D. (2004). *Design of interconnection networks for programmable logic* (Bd. 22). New York: Springer-Verlag.
- Lerspalungsanti, S. (2010). Ein Beitrag zur Modellierung des menschlichen Komfortempfindens und Beurteilung der NVH-Eigenschaften in der Antriebsstrangentwicklung auf Basis von Künstlichen Neuronalen Netzen. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 43). ISSN: 1615-8113. Karlsruhe.
- Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E. & Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y. et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Lin, X., Wang, Y., Bogdan, P., Chang, N. & Pedram, M. (2014). Reinforcement learning based power management for hybrid electric vehicles. In *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (S. 33–38).
- Liu, H. [Huan], Motoda, H., Setiono, R. & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. In *Feature selection in data mining* (S. 4–13). PMLR.
- Liu, T., Tang, X., Wang, H., Yu, H. & Hu, X. (2019). Adaptive hierarchical energy management design for a plug-in hybrid electric vehicle. *IEEE Transactions on Vehicular Technology*, 68(12), 11513–11522.
- Maack, S., Bertovic, M. & Radtke, M. (2020). *Deutsche Normungsroadmap künstliche Intelligenz*. Berlin: Deutsches Institut für Normung (DIN).

- Maisch, M., Hörz, M. & Jesser, D. (2019). The 8G-DCT Plug-in Hybrid Transmission for the Mercedes-Benz Compact Car Family. In *CTI SYMPOSIUM 2019* (S. 337–345). Springer.
- Mandel, C., Wolter, K., Bause, K., Behrendt, M., Hanf, M. & Albers, A. (2020). Model-Based Systems Engineering methods to support the reuse of knowledge within the development of validation environments. In *2020 IEEE International Systems Conference (SysCon)* (S. 1–7). IEEE.
- Margolis, G. B., Yang, G., Paigwar, K., Chen, T. & Agrawal, P. (2022). Rapid Locomotion via Reinforcement Learning. *arXiv preprint arXiv:2205.02824*.
- Marler, R. T. & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6), 369–395.
- Marxen, L. (2014). A Framework for Design Support Development based on the integrated Product Engineering Model iPeM. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 74). ISSN: 1615-8113. Karlsruhe.
- Mayring, P. (2016). *Einführung in die qualitative Sozialforschung* (6. Aufl.). eine Anleitung zu qualitativem Denken. Weinheim, Basel: Beltz.
- McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. In *Bulletin of Mathematical Biophysics* (Bd. 5, S. 115–133).
- McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H. et al. (2020). International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788), 89–94.
- Mildenberger, O. (1992). *Entwurf analoger und digitaler Filter*. Braunschweig/Wiesbaden: Vieweg.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. et al. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Moor, J. (2006). The Dartmouth College artificial intelligence conference: The next fifty years. *AI magazine*, 27(4), 87.

- Naser, M. Z. & Alavi, A. H. (2021). Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. *Architecture, Structures and Construction*, 1–19.
- Naunheimer, H., Bertsche, B., Ryborz, J., Novak, W. & Fellows, G. L. (2019). *Fahrzeuggetriebe. Grundlagen, Auswahl, Auslegung und Konstruktion* (3. Aufl.). Berlin: Springer-Vieweg.
- North, K. (2016). *Wissensorientierte Unternehmensführung: Wissensmanagement gestalten*. Wiesbaden: Springer.
- Ott, S. *Antriebsstrangprüfstand*. Zugriff am 25.09.2024. Verfügbar unter: <https://www.ipek.kit.edu/177.php#gallery>
- Pearson, K. (1896). VII. Mathematical contributions to the theory of evolution.—III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187), 253–318.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. et al. *Overview of clustering methods*. Zugriff am 23.09.2022. Verfügbar unter: <https://scikit-learn.org/stable/modules/clustering.html>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. et al. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825–2830.
- Pinner, T. (2017). Ein Beitrag zu Entwicklung von Koppelsystemen für die Validierung im Kontext des X-in-the-loop-Frameworks am Beispiel eines Schaltroboters. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 104). ISSN: 1615-8113. Karlsruhe.
- Pinte, G., Depraetere, B., Symens, W., Swevers, J. & Sas, P. (2010). Iterative learning control for the filling of wet clutches. *Mechanical systems and signal processing*, 24(7), 1924–1937.
- Pollie, R. (2021). Nanosheet Chips Poised to Rescue Moore's Law. *Engineering*, 7(12), 1655–1656.

- Proff, H. (2020). Automobilindustrie im Umbruch. In *Neue Dimensionen der Mobilität* (S. 49–63). Springer.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M. & Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268), 1–8.
- Reif, K. (2007). *Automobilelektronik* (Bd. 2). Wiesbaden: Vieweg.
- Reif, K., Noreikat, K. E. & Borgeest, K. (2012). *Kraftfahrzeug-Hybridantriebe: Grundlagen, Komponenten, Systeme, Anwendungen*. Wiesbaden: Springer Vieweg.
- Reuss, H.-C., Liao, J., Gitt, C. & Schröder, J. (2019). Effiziente automatische Applikation in der Rapid-Prototyping-Softwareentwicklung. In *ATZ elektronik* (Bd. 14, S. 54–59).
- Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (S. 1135–1144).
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Russell, S. & Norvig, P. (2012). *Artificial intelligence: a modern approach* (3. Aufl.). München: Pearson.
- Schmidt, G. (1991). *Einfache nichtlineare Regelungen*. Berlin: Springer.
- Schröter, J. (2013). Das erweiterte X-in-the-Loop-Framework zur durchgängigen Integration von Optimierungsverfahren in den Produktentwicklungsprozess am Beispiel der Entwicklung energieeffizienter Fahrzeuge. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 64). ISSN: 1615-8113. Karlsruhe.
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P. & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3), 1–21.

- Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A. et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419), 1140–1144.
- Sommer Obando, H. (2016). Reinforcement Learning Framework for the self-learning Suppression of Clutch Judder in automotive Drive Trains. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung*. ISSN: 1615-8113. Karlsruhe: Karlsruher Institut für Technologie (KIT).
- Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S. et al. (2021). Towards CRISP-ML (Q): a machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction*, 3(2), 392–413.
- Sutton, R. S. & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT press.
- Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge: MIT press.
- Tenzer, F. *Volumen der jährlich generierten/replizierten digitalen Datenmenge weltweit in den Jahren 2012 und 2020 und Prognose für 2025*. Zugriff am 25.09.2024. Verfügbar unter: <https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/>
- Tewiele, S., Driesch, P., Weber, T. & Schramm, D. (2018). Clustering of real BEV driving data with subsequent driving cycle construction using Markov chains. In *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium* (S. 1–6).
- Tröster, P. M., Klotz, T., Rapp, S., Ott, S. & Albers, A. (2021). Modellierung einer Einscheibentrockenkupplung mithilfe des C&C²-Ansatzes zur Identifikation von

- kritischen Gestalt-Funktion-Zusammenhängen bezüglich des Schwingungsphänomens zwangserregtes Kupplungsrupfen. In *Forschung im Ingenieurwesen* (Bd. 85, S. 881–894).
- Tröster, P. M., Prager, F., Schmidt, D. & Albers, A. (2023). Modeling of Embodiment Function Relations using the C&C2-Approach. In *Stuttgarter Symposium für Produktentwicklung SSP 2023, Stuttgart, 25th May 2023* (S. 175–186). Fraunhofer IAO.
- Vadamalu, R. & Beidl, C. (2016). MPC for Active Torsional Vibration Reduction of Hybrid Electric Powertrains. *IFAC-PapersOnLine*, 49, 756–761. <https://doi.org/10.1016/j.ifacol.2016.08.110>
- VDI, 2206. *VDI-Richtlinie 2206*. Berlin: Beuth Verlag.
- Veith, J.-M., Krüger, B., Albers, A. & Behrendt, M. (2018). Torque prediction in vehicle drive trains based on bus signals using artificial neural networks. In VDI Wissensforum GmbH (Hrsg.), *Drivev – Drivetrain For Vehicles 2018* (1. Aufl., 413-424). Düsseldorf: VDI Verlag. <https://doi.org/10.51202/9783181023280-I-413>
- Völz, A. (2016). *Modellprädiktive Regelung nichtlinearer Systeme mit Unsicherheiten*. Wiesbaden: Springer Vieweg.
- Wang, J., Hou, X. [Xianjun], Du, C., Xu, H. & Zhou, Q. (2020). A Moment-of-Inertia-Driven Engine Start-Up Method Based on Adaptive Model Predictive Control for Hybrid Electric Vehicles With Drivability Optimization. *IEEE Access*, 8, 133063–133075.
- Wirth, R. & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Bd. 1, S. 29–39). Manchester.
- ECE, R83 (15.02.2019). *UN/ECE R 83*.
- Wolter, K., Ruoff, S., Lutz, S., Bause, K. & Ott, S. (2018). Potential in customer-specific manual transmission development through the use of neural networks. In *12th International Conference & Expo: Automotive Transmissions, HEV and EV Drives (CTI Symposium USA 2018), May 14-17, 2018, Novi, MI*.

- Yang, N., Han, L., Xiang, C., Liu, H. [H.] & Hou, X. [X.]. (2021). Energy management for a hybrid electric vehicle based on blended reinforcement learning with backward focusing and prioritized sweeping. *IEEE Transactions on Vehicular Technology*, 1. <https://doi.org/10.1109/TVT.2021.3064407>
- Yu, T. & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.
- Zeng, R., Xue, L. & Xiao, J. (2019). Pressure Control Method of an Electro-hydraulic Actuated Clutch Considering Hysteresis. *Journal of Engineering Science & Technology Review*, 12(5).
- Zhang, J. M., Harman, M., Ma, L. & Liu, Y. (2020). Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*.
- Zhang, X., Ebner, T., Arntz, M., Ramsauer, A. & Küçükay, F. (2021). New Solution Supporting Efficient Vehicle Calibration Using Objective Driveability Evaluation and AI. 21. *Internationales Stuttgarter Symposium: Automobil-und Motorentechnik*, 278–293.

Studentische Abschlussarbeiten, die im Kontext dieser Dissertation am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT) vom Autor co-betreut wurden:

- Schönstein, M. (2021). Temperaturmodellierung einer Doppelkupplung mit Künstlichen Neuronalen Netzen. Unveröffentlichte Masterarbeit.

Veröffentlichungen, die unter Mitautorenschaft des Autors / der Autorin dieser Forschungsarbeit entstanden sind:

Altenburg, S., Schuchter, F., Bause, K. & Albers, A. (2023). Nutzung von KI-Methoden für die Kupplungsentwicklung in automobilen Antriebssträngen. In *Forschung im Ingenieurwesen* (S. 1–9). Springer.

Schuchter, F., Bause, K. & Albers, A. (2021). Artificial intelligence in hybrid vehicle transmission control-literature review and research methodology. In *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)* (S. 303–307). IEEE.

Schuchter, F., Bause, K. & Albers, A. (2022). Intelligent Data Analytics with Artificial Intelligence for Hybrid Engine Restart. In *22. Internationales Stuttgarter Symposium* (S. 61–72). Springer.

Schuchter, F., Schönstein, M., Bause, K. & Albers, A. (2022). Temperature Modelling of a Dual Clutch with Artificial Neural Networks. In VDI Wissensforum GmbH (Hrsg.), *Dritev 2022* (S. 335–346). Düsseldorf: VDI Verlag. <https://doi.org/10.51202/9783181024010-335>

Glossar

Reibungszahl

Der Begriff Reibungszahl beschreibt für Tribosysteme mit heterogenen Systemeigenschaften den Proportionalitätsfaktor, der bei Relativbewegung der in Kontakt befindlichen Oberflächen das Verhältnis von Normal- und Tangentialkraft beschreibt. Aufgrund der heterogenen Systemeigenschaften ist die Reibungszahl in der Regel eine Funktion in Abhängigkeit von beispielsweise Ort, Zeit, Geschwindigkeit, Flächenpressung oder Temperatur. (Bernhardt, 2017; Kniel, 2017)

Anhang

Literaturliste ML in hybriden Fahrzeuggetrieben

| Nr. | Titel | Erstautor | Jahr | Thematik | Methode | Datenbank |
|-----|--|---|------|----------------------|--------------------------|-------------|
| 1 | Concurrent Optimization for Parameters of Powertrain and Control System of Hybrid Electric Vehicle Based on Multi-Objective Genetic Algorithms | Fang, L.; Qin, S. | 2006 | Energy Management | Genetic Algorithm | IEEE Xplore |
| 2 | Dynamic Programming Algorithm for minimizing operating cost of a PEM fuel cell vehicle | Xu, L.; Ouyang, M.; Yang F. | 2012 | Energy Management | Dynamic Programming | IEEE Xplore |
| 3 | Fuzzy Logic Controller Design for Retrofitted Three-wheeled Hybrid Electric Vehicle | Sudhakara n, S.; Indulal, S. | 2020 | powertrain control | Fuzzy | IEEE Xplore |
| 4 | Neuro-genetic energy management for hybrid fuel cell power train | Mohammad ian, M.; Bathaee, S. | 2004 | Energy Management | Neural Network | IEEE Xplore |
| 5 | Automotive Gear-Shifting Decision Making Based on Neural Network Computation Model | Tan, J.; Yin X.; Yin L.; Zhao, L. | 2007 | Shfit Decision | Neural Network | IEEE Xplore |
| 6 | A four-step method to design an energy management strategy for hybrid vehicles | Zhu, Y. | 2004 | Energy Management | Fuzzy/ Genetic Algorithm | IEEE Xplore |
| 7 | Sensor health monitoring using simple data driven approaches | Mathew, V. | 2016 | Sensor Health | Supervised Learning | IEEE Xplore |
| 8 | On the road friction recognition based on the driving wheels deceleration | Jin, H.; Zhou, M. | 2014 | Friction Recognition | Supervised Learning | IEEE Xplore |
| 9 | Neuro fuzzy transmission control for automobile with variable loads | Hayashi, K.; Shimizu, Y.; Dote, Y. | 1995 | Shift Scheduling | Neural Network | IEEE Xplore |
| 10 | From Optimal to Real-Time Control of a Mechanical Hybrid Powertrain | van Berkel, K.; Titualer, R.; Hofmann, T. | 2014 | Energy Management | Dynamic Programming | IEEE Xplore |
| 11 | Decision Forest for Root Cause Analysis of Intermittent Faults | Singh, S.; Subramani a, H.; Holland, S. | 2012 | Intermittent faults | Decision Tree | IEEE Xplore |
| 12 | Ecological Adaptive Cruise Controller for a Parallel Hybrid Electric Vehicle | Frezza, G.; Evangelou, S. | 2020 | Cruise Control | MPC | IEEE Xplore |
| 13 | A Fleet-Based Machine Learning Approach for Automatic Detection of Deviations between Measurements and Reality | Pfeiffer, J.; Wolf, P.; Pereira, R. | 2019 | Deviations | Machine Learning | IEEE Xplore |
| 14 | Time Series Anomaly Detection Based on GAN | Sun, Y.; Yu W.; Chen Y. | 2019 | Anomaly Detection | Deep Learning | IEEE Xplore |

| | | | | | | |
|----|--|--|------|---------------------|---------------------|----------------|
| 15 | Drive Cycle Identification and Energy Demand Estimation for Refuse-Collecting Vehicles | Soriano, F.; Moreno-Eguilaz, M. | 2014 | Energy Management | Machine Learning | IEEE Xplore |
| 16 | Real-Time Model Predictive Control for a Parallel Hybrid Electric Vehicle using Outer Approximation and Semi-Convex Cut Generation | de Mauri, M.; Gillis, J.; Pipellers, G. | 2020 | Energy Management | MPC | IEEE Xplore |
| 17 | Nonlinear Model Predictive Control for Series-Parallel Hybrid Electric Buses | Liu, B.; Wang T.; Wang H. | 2019 | Buses | MPC | IEEE Xplore |
| 18 | Using Balanced Random Forests on Load Spectrum Data for Classifying Component Failures of a Hybrid Electric Vehicle Fleet | Bergemir, P.; Nitsche, C.; Nonnast, J. | 2014 | Component Failures | Data Mining | IEEE Xplore |
| 19 | Prediction of vehicle driving conditions with incorporation of stochastic forecasting and machine learning and a case study in energy management of plug-in hybrid electric vehicles | Liu, Y.; Li, J.; Gao, J. | 2021 | Energy Management | Machine Learning | Science Direct |
| 20 | Genetic Algorithm Fuzzy Logic Energy Management Strategy for Fuel Cell Hybrid Vehicle | Hankache, W.; Caux, S.; Hissel, D. | 2009 | Energy Management | Fuzzy/Genetic | Science Direct |
| 21 | Prediction-based optimal power management in a fuel cell/battery plug-in hybrid vehicle | Bubna, P.; Brunner, D.; Advani, A. | 2010 | Energy Management | Machine Learning | Science Direct |
| 22 | Multi-objective genetic optimization of the fuel cell hybrid vehicle supervisory system: Fuzzy logic and operating mode control strategies | Ahmadi, S. Bathaee, S. | 2015 | Energy Management | Genetic Algorithm | Science Direct |
| 23 | A novel cascade approach to control variables optimisation for advanced series-parallel hybrid electric vehicle power-train | Cipek, M.; Kasac, J. Pavkovic | 2020 | Energy Management | dynamic programming | Science Direct |
| 24 | Efficient experimental energy management operating for FC/battery/SC vehicles via hybrid Artificial Neural Networks-Passivity Based Control | Benmouna, A.; Becherif, M. | 2021 | Energy Management | Machine Learning | Science Direct |
| 25 | Active damping of driveline vibration in power-split hybrid vehicles based on model reference control | Liu, H.; Zhang, X., Chen, Y. | 2019 | Vibration Reduction | MPC | Science Direct |
| 26 | Optimal control of a parallel hybrid vehicle equipped with a dual electrical storage system | Fontaine, C.; Delprat, S.; Paganelli, S. | 2013 | Energy Management | Genetic Algorithm | Science Direct |
| 27 | Model-based design validation for advanced energy management strategies for electrified hybrid power trains using innovative vehicle hardware in the loop (VHIL) approach | Mayyas, A.; Kumar, S.; Pisu, P. | 2017 | Energy Management | MPC | Science Direct |

| | | | | | | |
|----|--|---|------|----------------------|---------------------|----------------|
| 28 | Power management and control of strategies for a hybrid vehicle with a dual mode power split transmission | Ren, Q.; Crolla, D.; Wheatley, A. | 2007 | Energy Management | Dynamic Programming | Science Direct |
| 29 | A comprehensive observer-based fault isolation method with application to a hydraulic power train | Pröll, S.; Jarmolowit h, F.; Lunze, J. | 2016 | Hydraulic Powertrain | Machine Learning | Science Direct |
| 30 | Variable horizon MPC for energy management on dual planetary gear hybrid electric vehicle | Li, M.; He, H.; Yan, M. | 2018 | Energy Management | MPC | Science Direct |
| 31 | Energy management of hybrid electric bus based on deep reinforcement learning in continuous state and action space | Tan, H.;Zhang, H.; Peng, J. | 2019 | Energy Management | Machine Learning | Science Direct |
| 32 | Guided model predictive control for connected vehicles with hybrid energy systems | Min, Q.; Li, J.; Liu, B. | 2021 | Connected Vehicles | MPC | Science Direct |
| 33 | Optimal strategies of energy management integrated with transmission control for a hybrid electric vehicle using dynamic particle swarm optimization | Chen, S.; Wu, C.; Hung, Y. | 2018 | Energy Management | Genetic Algorithm | Science Direct |
| 34 | Performance optimization of mild hybrid passenger vehicle by dual control strategy for city driving cycle | Shanu, P.; Senthilkumar, S. | 2021 | Energy Management | Dynamic Programming | Science Direct |
| 35 | Wavelet-transform-based power management of hybrid vehicles with multiple on-board energy sources including fuel cell, battery and ultracapacitor | Zhang, X.; Mi, C. Marhur, A. | 2008 | Energy Management | wavelet-algorithm | Science Direct |
| 36 | Rule-corrected energy management strategy for hybrid electric vehicles based on operation-mode prediction | Liu, Y., Gao, J.; Qin, D. | 2018 | Energy Management | Machine Learning | Science Direct |
| 37 | Hierarchical and Adaptive Neuro-Fuzzy Control for Intelligent Energy Management in Hybrid Electric Vehicles | Kamal, E.; Adouane, L. Abdrakhmanov, R. | 2017 | Energy Management | Fuzzy | Science Direct |
| 38 | Model-Based Torque Shaping for Smooth Acceleration Response in Hybrid Electric Vehicles | Bovee, K.; Rizzoni, G. | 2016 | Torque Shaping | MPC | Science Direct |
| 39 | Optimal Vehicle Following Strategy for Joint Velocity and Energy Management Control of Series Hybrid Electric Vehicles | Pan, X.; Chen, B.; Evangelou, S. | 2020 | Energy Management | MPC | Science Direct |
| 40 | Energy management and gear shifting control for a hybridized vehicle to minimize gas emissions, energy consumption and battery aging | Eckert, J.; da Silva, S. | 2021 | Energy Management | Genetic Algorithm | Science Direct |
| 41 | Optimization & validation of Intelligent Energy Management System for pseudo dynamic predictive regulation of plug-in hybrid electric vehicle as donor clients | Chacko, P.; Schidanandam, M. | 2020 | Energy Management | Machine Learning | Science Direct |

| | | | | | | |
|----|--|--------------------------------------|------|-------------------|---------------------|----------------|
| 42 | Implementing an MPC Algorithm in a Vehicle with a Hybrid Powertrain using Telematics as a Sensor for Powertrain Control | Finkeldei, E.; Back, M. | 2004 | Energy Management | MPC | Science Direct |
| 43 | Parameter Optimization of Dual Clutch Transmission for an Axle-split Hybrid Electric Vehicle | Cao, X.; Du, C.; Yan, F. | 2019 | Energy Management | Swarm Optimization | Science Direct |
| 44 | Improving fuel economy and performance of a fuel-cell hybrid electric vehicle (fuel-cell, battery, and ultra-capacitor) using optimized energy management strategy | Ahmadi, S. Bathaee, S. | 2018 | Energy Management | fuzzy/genetic | Science Direct |
| 45 | Modeling, Control, and Powermanagement Optimization for an Emulated Multi-Source Hybrid Drivetrain | Moulik, B.; Söffker, D. | 2015 | Energy Management | MPC | Science Direct |
| 46 | On quantifying the utility of look-ahead data for energy management | Hegde, B.; Rajendra, A.; Ahmed, Q. | 2018 | Energy Management | Machine Learning | Science Direct |
| 47 | Data-driven reinforcement-learning-based hierarchical energy management strategy for fuel cell/battery/ultracapacitor hybrid electric vehicles | Sun, H.; Fu, Z.; Tao, F. | 2020 | Energy Management | Deep Learning | Science Direct |
| 48 | An Agent based Power Flow Control for Hybrid Electric Vehicles | Ippoliot, L.; Loia, V.; Siano, P. | 2004 | Energy Management | Deep Learning | Science Direct |
| 49 | Power management strategy for vehicular-applied hybrid fuel cell/battery power system | Li, X.; Xu, L.; Hua, J. | 2009 | Energy Management | fuzzy | Science Direct |
| 50 | Robust energy management of plug-in hybrid electric bus considering the uncertainties of driving cycles and vehicle mass | Shangguan, J.; Guo, H.; Yue, M. | 2020 | Energy Management | Dynamic Programming | Science Direct |
| 51 | Using Stochastic Dynamic Programming for look-ahead control of a Wheel Loader Diesel Electric Transmission | Nilsson, T.; Fröberg, A.; Aslund, J. | 2014 | Engine Speed | Dynamic Programming | Science Direct |
| 52 | Effects of Time Horizon on Model Predictive Control for Hybrid Electric Vehicles | Rezaei, A.; Burl, J. | 2015 | Energy Management | MPC | Science Direct |
| 53 | A fuzzy logic based supervisory controller for an FC/UC hybrid vehicular power system | Eren, Y.; Erdinc, O.; Gorgun, H. | 2009 | Energy Management | fuzzy | Science Direct |
| 54 | Power management optimization in plug-in hybrid electric vehicles subject to uncertain driving cycles | Zhang, H.; Qin, Y.; Li, X. | 2020 | Energy Management | Dynamic Programming | Science Direct |
| 55 | Plug-In Hybrid Electric Bus Energy Management Based on Stochastic Model Predictive Control | Xie, S.; Peng, J.; He, H. | 2017 | Energy Management | MPC | Science Direct |
| 56 | Stochastic self-optimizing power management for fuel cell hybrid scooters of different sized components | Lee, C.; Lin, W. | 2015 | Energy Management | Machine Learning | Science Direct |
| 57 | Real-time predictive control of hybrid fuel cell drive trains | Bartholomaeus, R.; Fischer, A. | 2007 | Energy Management | MPC | Science Direct |

| | | | | | | |
|----|--|---|------|-------------------------|---------------------------------------|----------------|
| 58 | Optimal Power Management for Vehicle Hybrid Electric System using Look-ahead Route Information | Bin, Y.; Reama, A.; Cela, A. | 2010 | Energy Management | Dynamic Programming | Science Direct |
| 59 | Fuzzy-tuned model predictive control for dynamic eco-driving on hilly roads | Bakibillah, A.; Kamal, M.; Tan, C. | 2021 | Energy Management | MPC | Science Direct |
| 60 | Towards a generic control-oriented model for HEV predictive energy management | Stroe, N.; Colin, G. | 2016 | Energy Management | MPC | Science Direct |
| 61 | Power flow management with predictive capabilities for a hybrid fuel cell vehicle | De Vito, D. | 2007 | Energy Management | Machine Learning | Science Direct |
| 62 | Applying forward dynamic programming to combined energy and thermal management optimization of hybrid electric vehicles | Engbroks, L.; Görke, D.; Schmiedler, S. | 2018 | Energy Management | dynamic programming | Science Direct |
| 63 | MPC for Active Torsional Vibration Reduction of Hybrid Electric Powertrains | Vadamalu, R.; Beidl, C. | 2016 | Vibration Reduction | MPC | Science Direct |
| 64 | A Comparison Study of Energy Management for A Plug-in Serial Hybrid Electric Vehicle | Liu, W.; He, H.; Wang, Z. | 2016 | Energy Management | optimal single point start-stop | Science Direct |
| 65 | Design and Control Co-Optimization for Hybrid Powertrains: Development of Dedicated Optimal Energy Management Strategy | Zhao, J.; Sciarretta, A. | 2016 | Energy Management | Pontryagin's Minimum Principle | Science Direct |
| 66 | Real-time predictive control strategy for a plug-in hybrid electric powertrain | Taghavipour, A.; Azad, N.; McPhee, J. | 2015 | Energy Management | MPC | Science Direct |
| 67 | An Intelligent Energy Management Strategy for Hybrid Vehicle with irrational actions using Twin Delayed Deep Deterministic Policy Gradient | Liu, Z.; Zhou, Q.; Li, Y. | 2021 | Energy Management | DRL | Science Direct |
| 68 | Critical review on structural architecture, energy control strategies and development process towards optimal energy management in hybrid vehicles | Saiteja, P.; Ashok, B. | 2022 | Energy Management | Vergleich vieler Methoden | Science Direct |
| 69 | Comparative study on fuel saving potential of series-parallel hybrid transmission and series hybrid transmission | Xu, X.; Zhao, J. | 2022 | Energy Management | DP | Science Direct |
| 70 | A Comparative Study of H_∞ and Model Predictive Control Approach to Eco-Driving | Chen, D.; Kim, Y. | 2021 | Energy Management | H_∞ und MPC | Science Direct |
| 71 | Optimization of gear shifting strategy for hybrid electric vehicles on the process of braking | Jian-xin, P.; Yu-hui, H.; Tian-xiao, Y. | 2021 | gear shift optimization | Multi-Island Genetic Algorithm (MIGA) | Science Direct |
| 72 | Modeling and control of a hybrid electric vehicle to optimize system performance for fuel efficiency | Saju, C.; Michael, P.; Jarin, T. | 2022 | Energy Management | Fuzzy | Science Direct |

Leitfaden Experteninterview

Experteninterview Dissertation Florian Schuchter (IPEK in Kooperation mit der Mercedes-Benz Group AG)

Einstiegsfrage

In welchen Bereichen haben Sie bisher in der Entwicklung von Getriebesoftware Erfahrungen gesammelt und in welchen Bereichen waren Sie davor aktiv?

Fragenblock

- Worin liegen in der Entwicklung von Steuerungs- und Regelungsalgorithmen von Doppelkupplungsgetrieben die größten technischen Herausforderungen?
- Zu den genannten Herausforderungen:
Was sind die Ursachen?

Was sind die Folgen? (z.B. Mehraufwand, Qualitätseinbußen, ...)

Wie häufig treten diese Herausforderungen auf?

Fragebogen Steuerungs-Optimierung

| Befragungsstudie - Optimierung | | Trifft gar nicht zu | Trifft eher nicht zu | Trifft teilwei se zu | Trifft eher zu | Trifft voll zu |
|--|---|------------------------------|-------------------------------|----------------------------|----------------------|-------------------|
| Block A: Aufbau/Struktur gesamte Methodik | | | | | | |
| A1 | Die Schritte der Methodik sind logisch miteinander verknüpft | | | | | |
| Block B: Definition Optimierungsziel | | | | | | |
| B1 | Die Definition des Optimierungsziels ist für mich verständlich. | | | | | |
| | Ich traue mir zu, ein Optimierungsziel für einen anderen | | | | | |
| B2 | Anwendungsfall festzulegen. | | | | | |
| | Falls geringes Zutreffen der Frage B2: | | | | | |
| B3 | Wieso trauen Sie es sich nicht zu? | | | | | |
| Block C: Definition Aktions- und Zustandsraum | | | | | | |
| C1 | Die Definition des Aktions- und Zustandsraum ist für mich verständlich. | | | | | |
| | Ich traue mir zu, einen Aktions- und Zustandsraum für einen anderen | | | | | |
| C2 | Anwendungsfall festzulegen. | | | | | |
| | Falls geringes Zutreffen der Frage C2: | | | | | |
| C3 | Wieso trauen Sie es sich nicht zu? | | | | | |
| Block D: Definition Belohnungsfunktion | | | | | | |
| D1 | Die Definition der Belohnungsfunktion ist für mich verständlich. | | | | | |
| | Ich traue mir zu, eine Belohnungsfunktion für einen anderen | | | | | |
| D2 | Anwendungsfall zu definieren. | | | | | |
| | Falls geringes Zutreffen der Frage D2: | | | | | |
| D3 | Wieso trauen Sie es sich nicht zu? | | | | | |
| Block E: Trainingsumgebung | | | | | | |
| E1 | Die Auswahl eines geeigneten Simulationsmodells ist für mich verständlich. | | | | | |
| | Ich traue mir zu, eine Simulationsumgebung für einen anderen | | | | | |
| E2 | Anwendungsfall zu definieren. | | | | | |
| | Falls geringes Zutreffen der Frage E2: | | | | | |
| E3 | Wieso trauen Sie es sich nicht zu? | | | | | |
| Block F: Abgleich Optimierungsziel | | | | | | |
| F1 | Der Abgleich des Optimierungsergebnisses mit den Optimierungszielen ist für mich verständlich. | | | | | |
| | Ich traue mir zu, den Abgleich zwischen Optimierungsziel und Ergebnis für einen anderen Anwendungsfall durchzuführen. | | | | | |
| F2 | Falls geringes Zutreffen der Frage F2: | | | | | |
| F3 | Wieso trauen Sie es sich nicht zu? | | | | | |
| Block G: Anwendung/Mehrwert der Methodik | | | | | | |
| G1 | Die Methodik ist im Kontext der Getriebeentwicklung anwendbar. | | | | | |
| G2 | Der Mehrwert der Methodik ist klar zu erkennen. | | | | | |
| | Die Methodik unterstützt mich dabei, mit der Komplexität in | | | | | |
| G4 | Getriebesystemen umzugehen. | | | | | |
| G5 | Die Methodik unterstützt mich bei der Auslegung einer Ansteuerung. | | | | | |
| Block I: Offene Fragen | | | | | | |
| I1 | Wo liegen die Stärken der Methodik? | | | | | |
| I2 | Was müsste an der Methodik verbessert werden? | | | | | |
| | In welchem Schritt sehe ich die größte Schwierigkeit zur Anwendung | | | | | |
| I3 | der Methodik? | | | | | |

Fragebogen Datenanalyse

| Befragungsstudie - Datenanalyse | | Trifft gar nicht zu | Trifft eher nicht zu | Trifft teilwei se zu | Trifft eher zu | Trifft voll zu |
|--|---|------------------------------|-------------------------------|----------------------------|----------------------|-------------------|
| Block A: Aufbau/Struktur gesamte Methodik | | | | | | |
| A1 | Die Schritte der Methodik sind logisch miteinander verknüpft | | | | | |
| Block B: Datenspezifikation und -aufbereitung | | | | | | |
| B1 | Die Definition von einzelnen Signalen zur Auswertung ist für mich verständlich. | | | | | |
| B2 | Ich traue mir zu, Signale für einen anderen Anwendungsfall zu definieren. | | | | | |
| B3 | Falls geringes Zutreffen der Frage B2: Wieso trauen Sie es sich nicht zu? | | | | | |
| Block C: Kennzahldefinition | | | | | | |
| C1 | Die Definition von Kennzahlen ist für mich verständlich. | | | | | |
| C2 | Minima, Maxima, Gradienten und Durchschnittswerte sind geeignete Kennzahlen. | | | | | |
| C3 | Ich traue mir zu, selbst Kennwerte für einen anderen Anwendungsfall zu definieren. | | | | | |
| C4 | Falls geringes Zutreffen der Frage C2: Wieso trauen Sie es sich nicht zu? | | | | | |
| Block D: Verifikation und Validierung | | | | | | |
| D1 | Die Korrelationsanalyse unterstützt mich, mein System besser zu verstehen. | | | | | |
| D2 | Die Auswertung einer Korrelationsanalyse ist für mich verständlich. | | | | | |
| D3 | Ich traue mir zu, eine Korrelationsanalyse für einen anderen Anwendungsfall auszuwerten. | | | | | |
| D4 | Falls geringes Zutreffen der Frage D3: Wieso trauen Sie es sich nicht zu? | | | | | |
| D5 | Die Auswertung der Gesamtvisualisierung ist für mich verständlich. | | | | | |
| D6 | Die Visualisierung unterstützt mich, Zusammenhänge im System zu erkennen. | | | | | |
| D7 | Ich traue mir zu, eine Gesamtvisualisierung für einen anderen Anwendungsfall auszuwerten. | | | | | |
| D8 | Falls geringes Zutreffen der Frage D7: Wieso trauen Sie es sich nicht zu? | | | | | |
| Block E: Anwendung/Mehrwert der Methodik | | | | | | |
| E1 | Die Methodik ist im Kontext der Getriebeentwicklung anwendbar. | | | | | |
| E2 | Der Mehrwert der Methodik ist klar zu erkennen. | | | | | |
| E3 | Die Methodik unterstützt mich dabei, mit der Komplexität in Getriebesystemen umzugehen. | | | | | |
| E4 | Die Methodik unterstützt mich bei der Datenanalyse. | | | | | |
| Block F: Offene Fragen | | | | | | |
| F1 | Wo liegen die Stärken der Methodik? | | | | | |
| F2 | Was müsste an der Methodik verbessert werden? | | | | | |
| F3 | In welchem Schritt sehe ich die größte Schwierigkeit zur Anwendung der Methodik? | | | | | |

Ergebnisse Fragebogen Steuerungs-Optimierung

| Befragungsstudie - Optimierung Ergebnisse - Anzahl der Befragten, n=10 | | 1 | 2 | 3 | 4 | 5 |
|---|---|-----|---|---|---|---|
| ◇ Mittelwert ■ 1. und 3. Quartil — Min/Max | | | | | | |
| Getriebeentwickelnde | | | | | | |
| Block A: Aufbau/Struktur gesamte Methodik | | | | | | |
| A1 | Die Schritte der Methodik sind logisch miteinander verknüpft | 4,4 | | | | |
| Block B: Definition Optimierungsziel | | | | | | |
| B1 | Die Definition des Optimierungsziels ist für mich verständlich. | 4,5 | | | | |
| B2 | Ich traue mir zu, ein Optimierungsziel für einen anderen Anwendungsfall festzulegen. | 4 | | | | |
| Block C: Definition Aktions- und Zustandsraum | | | | | | |
| C1 | Die Definition des Aktions- und Zustandsraum ist für mich verständlich. | 4,5 | | | | |
| C2 | Ich traue mir zu, einen Aktions- und Zustandsraum für einen anderen Anwendungsfall festzulegen. | 4,1 | | | | |
| Block D: Definition Aktions- und Zustandsraum | | | | | | |
| D1 | Die Definition der Belohnungsfunktion ist für mich verständlich. | 4,3 | | | | |
| D2 | Ich traue mir zu, eine Belohnungsfunktion für einen anderen Anwendungsfall zu definieren. | 3,8 | | | | |
| Block E: Definition Aktions- und Zustandsraum | | | | | | |
| E1 | Die Auswahl eines geeigneten Simulationsmodells ist für mich verständlich. | 4,6 | | | | |
| E2 | Ich traue mir zu, eine Simulationsumgebung für einen anderen Anwendungsfall zu definieren. | 4 | | | | |
| Block F: Definition Aktions- und Zustandsraum | | | | | | |
| F1 | Der Abgleich des Optimierungsergebnisses mit den Optimierungszielen ist für mich verständlich. | 3,7 | | | | |
| F2 | Ich traue mir zu, den Abgleich zwischen Optimierungsziel und Ergebnis für einen anderen Anwendungsfall durchzuführen. | 3,3 | | | | |
| Block G: Definition Aktions- und Zustandsraum | | | | | | |
| G1 | Die Methodik ist im Kontext der Getriebeentwicklung anwendbar. | 4,5 | | | | |
| G2 | Der Mehrwert der Methodik ist klar zu erkennen. | 3,9 | | | | |
| G4 | Die Methodik unterstützt mich dabei, mit der Komplexität in Getriebesystemen umzugehen. | 3,7 | | | | |
| G5 | Die Methodik unterstützt mich bei der Auslegung einer Ansteuerung. | 3,8 | | | | |

1 = trifft nicht zu 2 = trifft eher nicht zu 3 = trifft teilweise zu 4 = trifft eher zu 5 = trifft voll zu

Ergebnisse Fragebogen Datenanalyse

| Befragungsstudie - Datenanalyse Ergebnisse - Anzahl der Befragten, n=9 | | 1 | 2 | 3 | 4 | 5 |
|---|---|-----|---|---|---|---|
| Getriebeentwickelnde | | | | | | |
| Block A: Aufbau/Struktur gesamte Methodik | | | | | | |
| A1 | Die Schritte der Methodik sind logisch miteinander verknüpft | 4,3 | | | | |
| Block B: Datenspezifikation und -aufbereitung | | | | | | |
| B1 | Die Definition von einzelnen Signalen zur Auswertung ist für mich verständlich. | 4,4 | | | | |
| B2 | Ich traue mir zu, Signale für einen anderen Anwendungsfall zu definieren. | 4,2 | | | | |
| Block C: Kennzahldefinition | | | | | | |
| C1 | Die Definition von Kennzahlen ist für mich verständlich. | 4,7 | | | | |
| C2 | Minima, Maxima, Gradienten und Durchschnittswerte sind geeignete Kennzahlen. | 4,3 | | | | |
| C3 | Ich traue mir zu, selbst Kennwerte für einen anderen Anwendungsfall zu definieren. | 4,2 | | | | |
| Block D: Verifikation und Validierung | | | | | | |
| D1 | Die Korrelationsanalyse unterstützt mich, mein System besser zu verstehen. | 4,1 | | | | |
| D2 | Die Auswertung einer Korrelationsanalyse ist für mich verständlich. | 3,8 | | | | |
| D3 | Ich traue mir zu, eine Korrelationsanalyse für einen anderen Anwendungsfall auszuwerten. | 3,6 | | | | |
| D5 | Die Auswertung der Gesamtvisualisierung ist für mich verständlich. | 3,9 | | | | |
| D6 | Die Visualisierung unterstützt mich, Zusammenhänge im System zu erkennen. | 3,8 | | | | |
| D7 | Ich traue mir zu, eine Gesamtvisualisierung für einen anderen Anwendungsfall auszuwerten. | 3,3 | | | | |
| Block E: Anwendung/Mehrwert der Methodik | | | | | | |
| E1 | Die Methodik ist im Kontext der Getriebeentwicklung anwendbar. | 4,1 | | | | |
| E2 | Der Mehrwert der Methodik ist klar zu erkennen. | 3,8 | | | | |
| E3 | Die Methodik unterstützt mich dabei, mit der Komplexität in Getriebesystemen umzugehen. | 3,6 | | | | |
| E4 | Die Methodik unterstützt mich bei der Datenanalyse. | 3,9 | | | | |

1 = trifft nicht zu 2 = trifft eher nicht zu 3 = trifft teilweise zu 4 = trifft eher zu 5 = trifft voll zu

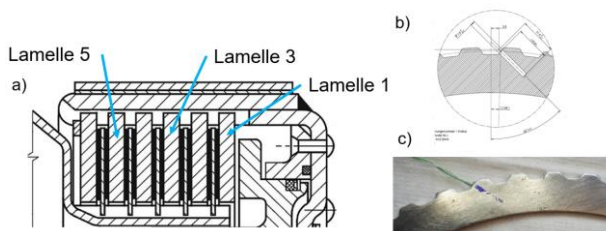
Aufgabenstellung und Anleitung Anwenderstudie Modellierung

Anwenderstudie Kupplungstemperaturmodell

– Aufgabenstellung und Anleitung

Kupplungstemperaturmodelle werden benötigt, um beim Überschreiten von kritischen Temperaturen die Kupplung zu öffnen und den Kupplungsbelag vor Schaden zu schützen. Zur Temperaturbestimmung werden Berechnungsmodelle auf dem Getriebesteuergerät verwendet, da aus Kosten- und Bauraumgründen keine Temperatursensoren in den Serienfahrzeugkupplungen verbaut werden können.

In diesem Anwendungsbeispiel wird ein eindimensionales Lamellentemperaturmodell der Kupplung K1 einer nasslaufenden Doppelkupplung in einem hybriden Doppelkupplungsgetriebe betrachtet. Es handelt sich um einen P2-Hybrid, bei dem die E-Maschine vor dem Getriebe angeordnet ist. Über eine Trennkupplung K0 kann der Verbrennungsmotor an den Getriebeeingang zugeschaltet werden. Die Kupplung beinhaltet mehrere Stahl- und Belagslamellen mit einem Papierreibbelag. Die Temperaturmessungen stammen aus einem im Versuchsfahrzeug verbauten Telemetriegetriebe, das Temperatursensoren in den Stahllamellen verbaut hat. Die höchste Stahllamellentemperatur wird abgebildet.



a) Kupplungsquerschnitt (Eigene Darstellung nach Naunheimer et al., 2019),
 b) Bohrungsquerschnitt Temperatursensor in Stahllamelle c)
 Temperatursensor und Stahllamelle

Abbildung 1

Der Temperaturhaushalt ist sehr komplex, weshalb aktuell verwendete physikalisch modellierte Modelle große Abweichungen zur Messung zeigen. Die Modelle müssen deshalb nochmals im Fahrzeug kalibriert werden. Da es verschiedene Fahrmanöver gibt, die das Modell gut abbilden soll, ist es für den Entwickler sehr zeitaufwendig einen optimalen Parametersatz für alle Fahrmanöver zu identifizieren.

Der Einsatz von Künstlicher Intelligenz ermöglicht es, Modelle anhand von Messdaten zu trainieren (vgl. Abbildung 2).

Das in dieser Aufgabe verwendete Modell ist ein Rekurrentes Neuronales Netz. Dieses nutzt nicht nur den aktuellen, sondern auch zurückliegende Zeitschritte für die Vorhersage. Das Modell lernt dabei aus Messdaten, die aus den Eingangsgrößen für das Modell und der Zielgröße bestehen. Beim Modelltraining werden die Parameter des Neuronalen Netzes so verändert, das der Fehler zwischen Modellvorhersage und gemessener Zielgröße minimiert wird. Die Modell-Performance wird anschließend auf einen für das Modell unbekannten Datensatz getestet.

Das fertig trainierte Modell kann nun anhand der Eingangsgröße die Zielgröße vorhersagen.

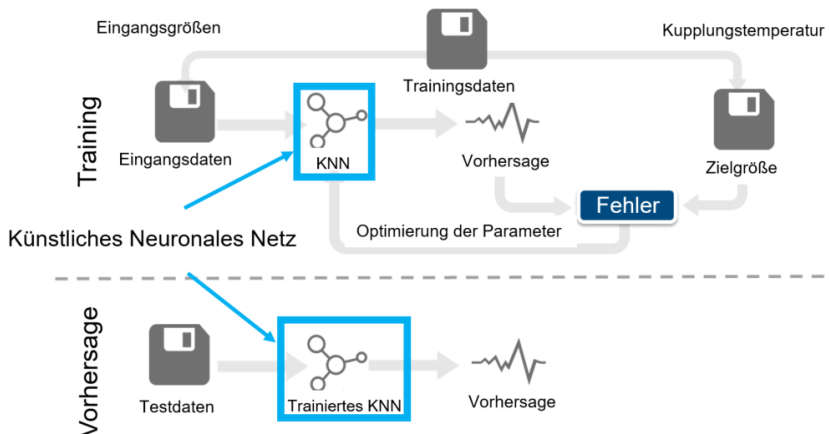


Abbildung 2

Abbildung 3 zeigt die einzelnen Schritte der Methode. Diese werden anhand der Aufgabenstellung nachfolgend durchlaufen.

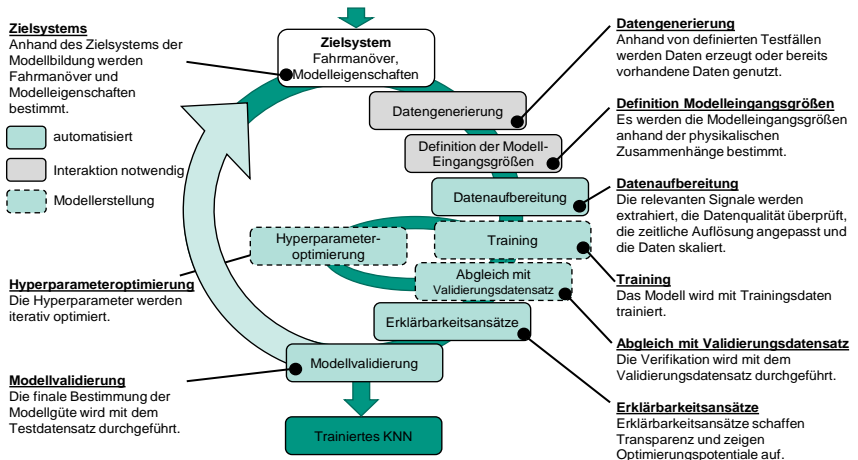


Abbildung 3

Gesamtaufgabe:

Erstellen Sie ein Modell, das anhand von Signalen in der Fahrzeugsoftware die Kupplungstemperatur der Kupplung K1 berechnet. Ziel ist es, ein Modell mit einem Root-Mean-Squared-Error von maximal 10 K im Vergleich zu einer Fahrzeugmessung für beide Fahrbetriebe zu erreichen. Die nachfolgenden Aufgabenschritte werden Sie durch die Umsetzung der Aufgabe führen.

Aufgabe 0 - Kenntnisstand

0.1 Wie schätzen Sie Ihre Kenntnisse im Bereich Fahrzeugkupplungen ein?

☐ keine ☐ gering ☐ vorhanden ☐ gut ☐ sehr gut

0.2 Wie schätzen Sie Ihre Kenntnisse im Bereich Wärmehaushalt ein?

☐keine ☐gering ☐vorhanden ☐gut ☐sehr gut

0.3 Wie schätzen Sie Ihre Kenntnisse im Bereich Maschinelles Lernen ein?

☐keine ☐gering ☐vorhanden ☐gut ☐sehr gut

Aufgabe 1 – Auswahl der Fahrmanöver für das Modelltraining

Für das Training des Modells müssen zunächst Fahrmanöver bestimmt werden, mit deren Messdaten dann das Modell trainiert wird.

Die Kupplung K1 ist beim Zuschalten aller ungeraden Gänge belastet. Aus Temperatursicht ist sie bei Anfahrvorgängen, wie Anfahren am Hang mit Steigung sowie Mehrfachrückschaltungen z.B. eine Schaltung vom achten in den vierten Gang (8-4 Schaltung) thermisch stark belastet. Beim Hybrid ist rein elektrisches und hybrides Anfahren möglich. Am Hang kann außerdem vorwärts oder rückwärts angefahren werden.

1.1 Mit welchen Fahrmanövern würden Sie das Modell trainieren?

1.2 Mir fiel es leicht, die relevanten Fahrmanöver zu bestimmen.

☐Trifft gar nicht zu ☐Trifft eher nicht zu ☐Trifft teilweise zu ☐Trifft eher zu ☐Trifft voll zu

Aufgabe 2 – Auswahl der Modelleingangsgrößen

Zum Verständnis des Kupplungssystems und dessen Wärmehaushalts ist in Abbildung 4 ein Ausschnitt einer Lamellenkupplung dargestellt.

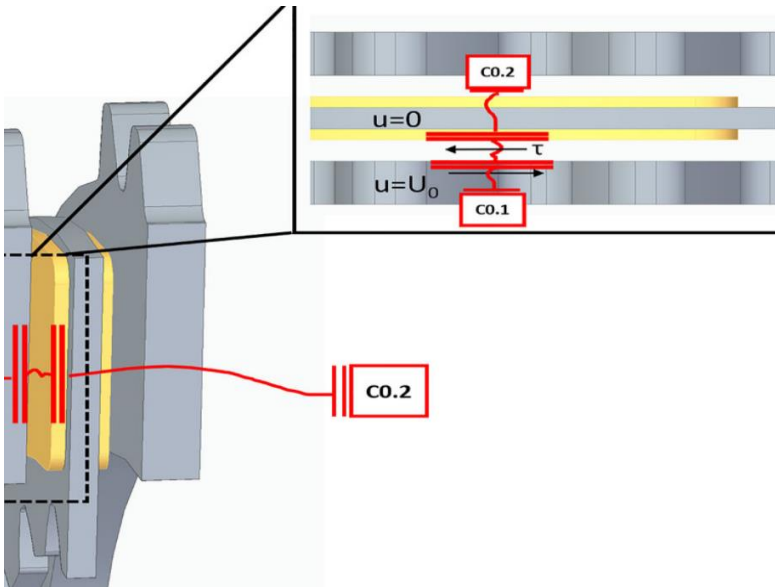


Abbildung 4

Um ein Drehmoment zu übertragen, werden die Lamellen über die Kupplungshydraulik gegeneinander angepresst.

Befindet sich die Kupplung im Schlupf, so entsteht eine Reibleistung nach Formel 1 aus dem Produkt von Differenzdrehzahl Δn und Reibmoment M_K .

$$P_V = M_K(\omega_2(t) - \omega_1(t)) = 2\pi\Delta n M_K$$

1

Das Reibmoment wird nach Formel 2 aus dem Produkt von Reibungszahl μ , Kupplungsdruck p , der Reibfläche pro Lamelle A_K , mittlerem Reibradius r_m und Anzahl der Lamellen z berechnet

$$M_K = \mu r_m p A_K z \quad 2$$

Um die Funktionsfähigkeit der Kupplung zu gewährleisten und die Kupplung zu kühlen, wird ein Kühlöl zwischen die Lamellen eingespritzt. Darüber wird ein Großteil der Wärme abgeführt. Ein weiterer Anteil der Wärme wird über Wärmeleitung der Lamellen und des Gehäuses abgeführt. Die abgeführte Wärmeleistung an das Öl berechnet sich aus dem Wärmeübergangskoeffizient α_K , der Kühlfläche A_W und der Temperaturdifferenz zwischen Lamelle T_K und Öl $T_{öl}$.

$$\dot{Q} = \alpha_K A_W (T_K - T_{öl}) \quad 3$$

2.1 Bestimmen Sie anhand des physikalischen Verständnisses die relevanten Einflussgrößen auf die die Kupplungstemperatur, die als Eingangsgrößen für das Modell dienen. Wählen Sie diese dabei aus der Liste im Tool aus.

Auswahl Eingangsgrößen

- ☐ Fahrzeugbeschleunigung
- ☐ Differenzdrehzahl Kupplung
- ☐ Getriebeeingangs-drehzahl
- ☐ Kühlöltemperatur
- ☐ Ist-Gang
- ☐ Kühlölvolumenstrom
- ☐ Fahrzeuggeschwindigkeit
- ☐ Gaspedalwinkel
- ☐ Kupplungsdruck
- ☐ Drehmoment Verbrennungsmotor
- ☐ Drehmoment E-Maschine

Abbildung 5

| |
|--|
| |
|--|

Kreuzen Sie Ihre Antwort an.

2.2 Mir fiel es leicht, die relevanten Modelleingangsgrößen auszuwählen.

| | | | | |
|---|--|---|--|--|
| <input type="checkbox"/> Trifft gar nicht zu | <input type="checkbox"/> Trifft eher nicht zu | <input type="checkbox"/> Trifft teilweise zu | <input type="checkbox"/> Trifft eher zu | <input type="checkbox"/> Trifft voll zu |
|---|--|---|--|--|

2.3 Die Modelleingangsgrößen anhand der physikalischen Zusammenhänge zu bestimmen, ist sinnvoll.

| | | | | |
|---|--|---|--|--|
| <input type="checkbox"/> Trifft gar nicht zu | <input type="checkbox"/> Trifft eher nicht zu | <input type="checkbox"/> Trifft teilweise zu | <input type="checkbox"/> Trifft eher zu | <input type="checkbox"/> Trifft voll zu |
|---|--|---|--|--|

2.4 Eine automatisierte Vorgabe von Eingangsgrößen würde mich unterstützen

| | | | | |
|---|--|---|--|--|
| <input type="checkbox"/> Trifft gar nicht zu | <input type="checkbox"/> Trifft eher nicht zu | <input type="checkbox"/> Trifft teilweise zu | <input type="checkbox"/> Trifft eher zu | <input type="checkbox"/> Trifft voll zu |
|---|--|---|--|--|

Nachdem Sie die relevanten Eingangsgrößen ausgewählt haben, drücken Sie auf den Button „Starte Datenverarbeitung“. Dadurch werden die Daten für das Modelltraining verarbeitet und vorbereitet.

Drücken Sie anschließend auf den Button „Zeige Daten-Histogramme“. Dies erzeugt Histogramme, Häufigkeitsverteilungen der Daten über ihren Wertebereich. Es veranschaulicht somit, wie eine Modelleingangsgröße über den Trainingsdaten verteilt ist.

2.5 Die Histogramme unterstützen mich, die Datengrundlage zu verstehen.

☐ Trifft gar
nicht zu

☐ Trifft eher nicht
zu

☐ Trifft teilweise
zu

☐ Trifft eher
zu

☐ Trifft voll
zu

2.6 Ist die Eingabe-Maske des Tools für Sie verständlich?

☐ Ja

☐ Nein, ich benötige weitere Informationen

*Welche Informationen

2.7 Wünschen Sie weitere Anweisungen, um mit dem Tool umzugehen?

☐ Ich kann die Eingabemaske ohne Probleme bedienen und benötige keine Anweisungen

☐ Ich benötige Anweisungen zum Bedienen der Eingabemaske*

*Welche Anweisungen benötigen Sie?

Aufgabe 3 – Modelltraining und Evaluation

Starte Sie das Modelltraining, indem Sie auf den Button „Starte Modelltraining“ drücken.

Nach wenigen Minuten erhalten Sie die Ausgabe des Trainings in einer Abbildung. Darin ist die Temperaturmessung aus dem Testdatensatz, und im Vergleich dazu die Vorhersage des Modells ausgegeben. Die Modellkennzahlen sind darüber dargestellt.

3.1 Welchen Root-Mean-Squared-Error (RMSE)* in K erzielt ihr Modell?

*Der RMSE beschreibt die Wurzel aus dem Mittelwert der Fehlerquadrate zwischen Modell und Messung

3.2 Erfüllt das Modell damit die Anforderungen? (Anforderung: $RMSE \leq 10\text{ K}$)

3.3 Ich vertraue einem mit messdatentrainierten Modell z.B. Neuronales Netz genauso, wie einem auf physikalischen Gleichungen aufgebauten Modell.

☐ Trifft gar nicht zu ☐ Trifft eher nicht zu ☐ Trifft teilweise zu ☐ Trifft eher zu ☐ Trifft voll zu

3.4 Sind die Ergebnis-Ausgaben des Tools für Sie verständlich?

☐ Ja, ich kann alle Ausgaben nachvollziehen

☐ Ja, ich kann die meisten Ausgaben nachvollziehen, aber benötige Erklärungen zu gewissen Ausgaben*

☐ Nein, ich kann die Ausgaben nicht nachvollziehen und benötige Hilfe zum Verstehen dieser*

*Welche Informationen sind in der Ausgabe nicht verständlich und benötigen weiterer Erklärungen?

3.5 Sind die Ausgaben, die das Tool Ihnen zur Verfügung stellt, sinnvoll für Sie?

☐ Ja

☐ Nein*

*Welche Angaben erscheinen Ihnen nicht sinnvoll und weshalb?

3.6 Hätten Sie gerne weitere Ausgaben?

Modellexport und Einsatz in der Getriebe-Software:

Das trainierte Modell kann aus dem Tool exportiert werden. Eine automatisierte Toolkette wandelt das Modell dann in das lauffähiges Matlab-Simulink-Modell um und kann dann in die Getriebesteuergeräte-Software übernommen werden.

Aufgabe 4 – Erklärbarkeit

Das bisher verwendete Neuronale Netze zur Modellierung der Kupplungstemperatur ist ein Black-Box-Modell. Das heißt, es ist nicht direkt ersichtlich, wie das Modell die Temperatur berechnet, oder sich über verschiedene Eingangsgrößen-Bereiche verhält. Um das Modell erklärbar zu machen, gibt es verschiedene Ansätze.

Der hier genutzte LIME-Algorithmus (Local Interpretable Model-Agnostic Explanations) erstellt ein lokales lineares Ersatzmodell des Neuronalen Netzes zu einem konkreten Zeitpunkt der Vorhersage. Es wird dargestellt, welche Modelleingangsgrößen in welchem Wertebereich einen positiven oder negativen Einfluss zu diesem Zeitpunkt auf die Temperatur haben. Abbildung 6 erklärt die Ergebnisse einer lokalen Erklärung des LIME-Algorithmus.

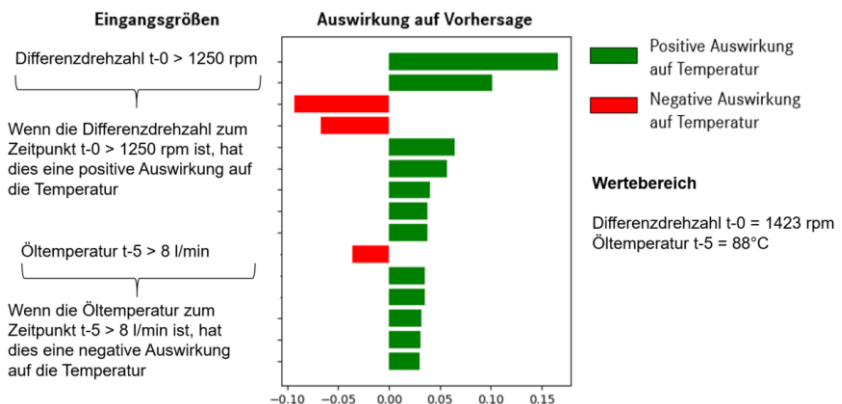


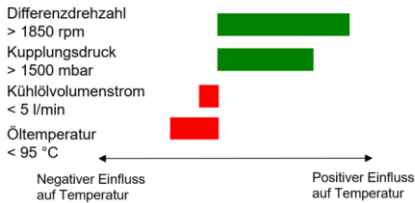
Abbildung 6

Abbildung 7 zeigt zwei Schaubilder. Zum Zeitpunkt $t=150$ s heizt die Kupplung auf, zum Zeitpunkt $t=183$ s kühlt sie ab.

Aufheizung (Zeitpunkt $t_1 = 150s$)

Wertebereich

Differenzdrehzahl = 1723 rpm
Kupplungsdruck = 2488 mbar
Kühlölvolumenstrom = 3,4 l/min
Öltemperatur = 86 °C



Abkühlung (Zeitpunkt $t_2 = 183s$)

Wertebereich

Differenzdrehzahl = 30 rpm
Kupplungsdruck = 1130 mbar
Kühlölvolumenstrom = 12,2 l/min
Öltemperatur = 97 °C

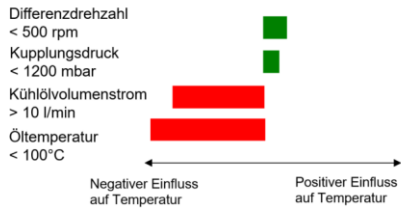


Abbildung 7

4.1 Welche Eingangsgrößen haben zum Zeitpunkt $t = 150s$ den größten Einfluss? Deckt sich dies mit ihren Erwartungen?

4.2 Welche Eingangsgrößen haben zum Zeitpunkt $t = 183s$ den größten Einfluss? Deckt sich dies mit ihren Erwartungen?

4.3 Die Erklärbarkeitsmethode unterstützt mich, mein Modell und das System zu verstehen.

- ☐ Trifft gar nicht zu
 ☐ Trifft eher nicht zu
 ☐ Trifft teilweise zu
 ☐ Trifft eher zu
 ☐ Trifft voll zu

4.4 Nach der Anwendung der Erklärbarkeitsmethode ist mein Vertrauen/Unvertrauen in die Modellaussagefähigkeit gestärkt.

☐ Trifft gar
nicht zu

☐ Trifft eher nicht
zu

☐ Trifft teilweise
zu

☐ Trifft eher
zu

☐ Trifft voll
zu

4.5 Eine hohe Modellgüte stärkt mein Vertrauen in das Modell.

☐ Trifft gar
nicht zu

☐ Trifft eher nicht
zu

☐ Trifft teilweise
zu

☐ Trifft eher
zu

☐ Trifft voll
zu

4.6 Sind die die Ausgaben des LIME-Algorithmus für Sie verständlich?

☐ Ja, ich kann die Darstellung nachvollziehen

☐ Nein, ich kann die Darstellung nicht nachvollziehen und benötige Hilfe zum Verstehen dieser*

*Welche Informationen sind in der Ausgabe nicht verständlich und benötigen weiterer Erklärungen?

4.7 Sind die die Ausgaben des LIME-Algorithmus für Sie sinnvoll?

☐ Ja

☐ Nein*

*Welche Angaben erscheinen Ihnen nicht sinnvoll und weshalb?

4.8 Fehlt Ihnen etwas, um die Methode im Arbeitsalltag einsetzen zu können?

☐ Ja*

☐ Nein

*Was fehlt Ihnen?

5. Allgemeine Fragen:

| | Trifft gar nicht zu | Trifft eher nicht zu | Trifft teilweise zu | Trifft eher zu | Trifft voll zu |
|---|------------------------------|-------------------------------|---------------------------|----------------------|-------------------|
| Der Mehrwert der Methodik ist klar zu erkennen. | | | | | |
| Die Automatisierung erleichtert mir den Umgang mit Methoden des Maschinellen Lernens. | | | | | |
| Zur Anwendung der Methodik ist ein geringes Vorwissen im Bereich Maschinelles Lernen ausreichend. | | | | | |
| Die Methodik ist im Kontext der Getriebeentwicklung anwendbar.* | | | | | |
| Die Methodik würde in der praktischen Anwendung bei Mercedes-Benz einen hohen Mehrwert darstellen.* | | | | | |
| Die Methode unterstützt mich dabei, mit der Komplexität in Getriebesystemen umzugehen.* | | | | | |

*nur von Getriebeentwicklern auszufüllen

Wo liegen die Stärken der Methodik (KI-Modell statt physikalisches Modell)?

Was müsste an der Methodik verbessert werden?

In welchem Schritt sehe ich die größte Schwierigkeit zur Anwendung der Methodik?

Fragebogen Anwenderstudie Modellierung

| Anwenderstudie | | keine | gering | vor- handen | gut | sehr gut |
|--|--|------------------------------|-------------------------------|---------------------------|----------------------|----------------------|
| Block A: Kenntnisse | | | | | | |
| A1 | Wie schätzen Sie Ihre Kenntnisse im Bereich Fahrzeugkupplungen ein? | | | | | |
| A2 | Wie schätzen Sie Ihre Kenntnisse im Bereich Wärmehaushalt ein? | | | | | |
| A3 | Wie schätzen Sie Ihre Kenntnisse im Bereich Maschinelles Lernen ein? | | | | | |
| | | Trifft gar nicht zu | Trifft eher nicht zu | Trifft teilweise zu | Trifft eher zu | Trifft voll zu |
| Block B: Definition Fahrmanöver | | | | | | |
| B1 | Mit welchen Fahrmanövern würden Sie das Modell trainieren? | | | | | |
| B2 | Mir fiel es leicht, die relevanten Lastfälle zu bestimmen. | | | | | |
| Block C: Definition Modelleingangsgrößen | | | | | | |
| C1 | Mit welchen Modelleingangsgrößen würden Sie das Modell trainieren? | | | | | |
| C2 | Mir fiel es leicht, die relevanten Modelleingangsgrößen auszuwählen. | | | | | |
| C3 | Die Modelleingangsgrößen anhand der physikalischen Zusammenhänge zu bestimmen, ist sinnvoll. | | | | | |
| C4 | Eine automatisierte Vorgabe von Eingangsgrößen würde mich unterstützen | | | | | |
| Block D: Datensätze | | | | | | |
| D1 | Die Histogramme unterstützen mich bei der Einordnung der Datensätze. | | | | | |
| Block E: Eingabe-Maske | | | | | | |
| E1 | Ist die Eingabe-Maske des Tools für Sie verständlich? (0 = nein, 1 = ja) | | | | | |
| E2 | Ich benötige weitere Informationen | | | | | |
| E3 | Wünschen Sie weitere Anweisungen, um mit dem Tool umzugehen? (0 = nein, 1 = ja) | | | | | |
| E4 | Ich benötige weitere Anweisungen | | | | | |
| Block F: Abgleich mit Validierungszielen | | | | | | |
| F1 | Welchen Root-Mean-Squared-Error (RMSE)* erzielt ihr Modell? | | | | | |
| F2 | Erfüllt das Modell damit die Anforderungen? (0 = nein, 1 = ja) | | | | | |
| F3 | Ich vertraue einem Black-Box-Modell (z.B. Neuronales Netz) genauso, wie einem White-Box-Modell (physikalische Modellierung). | | | | | |
| Block G: Ausgabe-Maske | | | | | | |
| G1 | Sind die Ergebnis-Ausgaben des Tools für Sie verständlich? (0 = nein, 1 = ja) | | | | | |
| G2 | Ich benötige weitere Informationen | | | | | |
| G3 | Sind die Ausgaben, die das Tool Ihnen zur Verfügung stellt, sinnvoll für Sie? (0 = nein, 1 = ja) | | | | | |
| G4 | Welche Angaben erscheinen Ihnen nicht sinnvoll und weshalb? | | | | | |
| G5 | Hätten Sie gerne weitere Ausgaben? | | | | | |
| Block H: Erklärbarkeit | | | | | | |
| H1 | Welche Eingangsgrößen haben zum Zeitpunkt t = 150s den größten Einfluss? Deckt sich dies mit Ihren Erwartungen? | | | | | |
| H2 | Welche Eingangsgrößen haben zum Zeitpunkt t = 183s den größten Einfluss? Deckt sich dies mit Ihren Erwartungen? | | | | | |
| H3 | Die Erklärbarkeitsmethoden unterstützen mich, mein Modell und das System zu verstehen. | | | | | |
| H4 | Nach der Anwendung der Erklärbarkeitsmethoden ist mein Vertrauen/Unvertrauen in die Modellaussagefähigkeit gestärkt. | | | | | |
| H5 | Eine gute Modell-Performance stärkt mein Vertrauen in das Modell. | | | | | |
| H6 | Sind die die Ausgaben des LIME-Algorithmus für Sie verständlich? | | | | | |
| H7 | Welche Informationen sind in der Ausgabe nicht verständlich und benötigen weiterer Erklärungen? | | | | | |
| H8 | Sind die die Ausgaben des LIME-Algorithmus für Sie sinnvoll? | | | | | |
| H9 | Welche Angaben erscheinen Ihnen nicht sinnvoll und weshalb? | | | | | |
| H10 | Fehlt Ihnen etwas, um die Methode im Arbeitsalltag einsetzen zu können? | | | | | |
| H11 | Was fehlt Ihnen? | | | | | |
| Block I: Anwendung/Mehrwert der Methodik; *Frage nur für Getriebeentwickler | | | | | | |
| I1 | Der Mehrwert der Methodik ist klar zu erkennen. | | | | | |
| I2 | Die Automatisierung erleichtert mir den Umgang mit Methoden des Maschinellen Lernens. | | | | | |
| I3 | Zur Anwendung der Methodik ist ein geringes Vorwissen im Bereich Maschinelles Lernen ausreichend. | | | | | |
| I4 | Die Methodik ist im Kontext der Getriebeentwicklung anwendbar.* | | | | | |
| I5 | Die Methodik würde in der praktischen Anwendung bei Mercedes-Benz einen hohen Mehrwert darstellen.* | | | | | |
| I6 | Die Methode unterstützt mich dabei, mit der Komplexität in Getriebesystemen umzugehen.* | | | | | |
| Block J: Offene Fragen | | | | | | |
| J1 | Wo liegen die Stärken der Methodik? | | | | | |
| J2 | Was müsste an der Methodik verbessert werden? | | | | | |
| J3 | In welchem Schritt sehe ich die größte Schwierigkeit zur Anwendung der Methodik? | | | | | |