

Frank-Florian Steege

**Einsatz adaptiver Lernverfahren zur Regelung
industrieller Verbrennungsprozesse**

Einsatz adaptiver Lernverfahren zur Regelung industrieller Verbrennungsprozesse

Frank-Florian Steege



Universitätsverlag Ilmenau
2015

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 7. April 2014

1. Gutachter: Prof. Dr.-Ing. Horst-Michael Groß
(Technische Universität Ilmenau)

2. Gutachter: Prof. Dr.-Ing. Markus Krabbes
(HTWK Leipzig)

3. Gutachter: Prof. Dr. rer. nat. Martin Riedmiller
(Albert-Ludwigs-Universität Freiburg)

Tag der Verteidigung: 29. Januar 2015

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-86360-114-0 (Druckausgabe)

URN [urn:nbn:de:gbv:ilm1-2015000055](http://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2015000055)

Titelfoto: photocase.com

Kurzbeschreibung

In dieser Arbeit wird dargestellt, wie lernfähige adaptive Prozessmodelle zum Verbessern der Prozessführung in großen industriellen Verbrennungsanlagen beitragen können. Dazu werden die wesentlichen Bestandteile eines solchen Verbrennungsprozesses dargestellt, und es wird erläutert, warum konventionelle Regelungsverfahren (z.B. PID-Regler) Teile dieses Prozesses nicht oder nur unzureichend beherrschen können. Lernfähige Modelle sind prinzipiell in der Lage eine bessere Prozessführung zu gewährleisten. Ebenso wird gezeigt, welche Probleme die Standard-Lernverfahren mit den Daten realer Prozesse haben. Im Hauptteil der Arbeit werden drei Verbesserungsansätze erarbeitet:

1. Der Einfluss von verrauschten Daten auf den Lernvorgang. Um trotz solcher Daten gute Modelle zu erhalten, werden sowohl verrauschte Punkte (Ausreißer) aus dem Datensatz entfernt als auch robuste Trainingsverfahren eingesetzt.
2. Die Anpassung eines lernfähigen Modells an Veränderungen des Prozesses über langen Zeiträumen. Es wird gezeigt, dass ein Adaptieren des Modells zu Vorteilen gegenüber einem statischen Modell führt. Auf Verbrennungsdaten werden die besten Ergebnisse mit der gewichteten Kombination von adaptiven Modellen in Ensembles erreicht.
3. Die funktionale Sicherheit der Stellstrategie eines lernfähigen Modells. Lernfähige Standardverfahren scheitern oft an der Verteilung der Daten in bereits geregelten Prozessen. Zur Vermeidung solcher Fehler werden mehrere sich ergänzende Ansätze entwickelt. Bedeutsam ist dabei das Einfügen von Expertenwissen in den Lernprozess. Durch Abgleich mit bereits vorhandenen Daten der Anlage kann gezielt Wissen für unbekannte Bereiche integriert und die Zuverlässigkeit lernfähiger Regler erhöht werden.

Das wesentliche Herausstellungsmerkmal dieser Arbeit im Vergleich zu anderen Arbeiten auf dem Gebiet der lernfähigen Prozessmodelle liegt im unmittelbaren praktischen Bezug zu Verbrennungsprozessen. Die vorgestellten Verfahren wurden auf verschiedenen Müllverbrennungsanlagen und Zementwerken über mehrere Wochen und Monate erprobt. Es werden Lösungsmöglichkeiten aufgezeigt, die es ermöglichen, lernfähige Systeme über lange Zeiträume aktiv Stellgrößen regeln zu lassen und so eine erhebliche Verbesserung der Prozessführung zu erreichen.

Abstract

In this thesis the contribution of self-learning adaptive process models to a stable operation of industrial combustion plants is discussed. Therefore, the main parts of a combustion process are described. It is explained why conventional controllers (e.g. PID controllers) are not able to control certain parts of this process satisfactorily whereas self-learning models can achieve better results in principle. However, main problems of standard learning algorithms applied to industrial combustion data are pointed out. Therefore, three improvements are presented in the main part of the thesis:

1. The negative effects of noisy process data on the learning process. Outliers are removed and robust training algorithms are used to obtain accurate process models despite this noisy training data.
2. The long-term adaptation of a process model on a changing process. The performance of an adapted model exceeds the performance of a static model. The best results on combustion process data are achieved with a weighted combination of different models within an ensemble.
3. The functional safety of the control strategy of a self-learning model. Standard self-learning approaches often fail to learn an accurate strategy because of the distribution of training data in controlled processes. To avoid this error complementary approaches are developed. Most important is the insertion of expert knowledge into the training data. Information about unknown process situations can be gained by combining expert knowledge and process data. Thus, the functional safety of self-learning models can be increased.

The main contribution of this thesis compared to other publication on self-learning models is the direct reference to practical industrial combustion applications. The developed algorithms are tested at different waste combustion plants and cement plants for time periods of several weeks and months. Solutions are presented which enable self-learning systems to control actuators in industrial plants over long time ranges with a considerable improvement of process stability.

Danksagung

Diese Arbeit wäre nicht entstanden ohne die Unterstützung vieler Menschen, bei denen ich mich an dieser Stelle bedanken möchte.

An erster Stelle möchte ich mich bei meinen Eltern, Sabine und Stephan Steege, bedanken. Vieles von dem was ich bin, verdanke ich ihnen, und ohne ihre Unterstützung wäre weder mein Studium noch diese Arbeit möglich gewesen.

Ebenso großer Dank gilt meinem Doktorvater Prof. Dr. Horst-Michael Groß. Ohne sein Vorbild während des Studiums hätte ich nie den Entschluss gefasst, diese Arbeit zu schreiben, und unsere sehr positiven Gespräche halfen mir sehr dabei, sie so zu Ende zu bringen, wie sie hier jetzt vorliegt.

Weiterhin danke ich sehr herzlich Dr. Volker Stephan für seine Betreuung der Arbeit bei Powitec. Seine vielen Anregungen halfen beim Entstehen der Arbeit genau so wie seine Unterstützung bei der Durchführung der vielen Versuche und sein unermüdlicher Einsatz zum Abwenden von störenden Einflüssen.

Ebenso möchte ich Franz Wintrich danken. Ohne seinen leidenschaftlichen Einsatz für die Powitec GmbH und ohne sein großartiges Streben, neue Wege zu gehen, hätte diese Arbeit keine Grundlage gehabt und hätte nicht durchgeführt werden können.

Ganz besonderer Dank gilt auch den vielen Operatoren und Produktionsleitern in den untersuchten Anlagen, die bereit waren, immer wieder unsere Testfahrten durchzuführen, und die einen Teil des Risikos tragen mussten. Fortschritt kann nur geschehen, wenn es Menschen gibt, die bereit sind, etwas Neues auszuprobieren. Diese Menschen in den Anlagen haben entscheidend dazu beigetragen, dieser Arbeit den Praxisbezug zu geben.

Ich möchte ebenfalls meinen Kollegen Andre Hartmann, Christian Diez und Thomas Herre danken. Oft haben Sie mich unterstützt und einen Teil der Arbeit übernommen, der anfiel, während ich Versuche für die Doktorarbeit durchführte. Ich danke ebenso Christoph Möller und Jens Reinsberger für ihre Ratschläge zur Gestaltung der Arbeit und Anna-Maria Rönsch für ihre graphische Unterstützung.

Weiterhin gilt mein Dank Dr. Erik Schaffernicht für die zahlreichen Anregungen, Gespräche und gemeinsamen Veröffentlichungen sowie den vielen Studenten, namentlich Christoph Ebert, Markus Eisenbach, Juliane Goldhahn, Hermann Heiber, Danny Irmer, Robert Kaltenhäuser und Paul Rudi Torke, die unter meiner Betreuung Studienarbeiten angefertigt haben.

Ganz besonders möchte ich mich bei meiner Freundin Polina für ihre Unterstützung und für die Sonne in meinem Leben bedanken.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Anspruch der Arbeit	4
1.2. Gliederung	6
2. Besonderheiten industrieller Verbrennungsprozesse	9
2.1. Müllverbrennungsanlagen	11
2.1.1. Aufbau des Verbrennungsraumes	15
2.1.2. Konventionelle Regelung der Verbrennung	19
2.1.3. Nachteile der konventionellen Regelung	25
2.2. Zementwerke	38
2.3. Durchführung von Testreihen in Anlagen	42
2.4. Fazit	44
3. Selbstlernende Verfahren zur Regelung von Prozessen	45
3.1. Einordnung lernfähiger und adaptiver Verfahren	47
3.2. Anwendungsgebiete in der Prozessführung	49
3.3. Adaptive Verfahren	52
3.3.1. Anpassung der Reglerparameter	52
3.3.2. Anpassung der Reglereingabe	56
3.4. Lernfähige Verfahren	60
3.4.1. Multi-Layer Perceptron	63
3.4.2. Neural Fitted Q-Iteration	66
3.4.3. Lineare Regression	70
3.5. Probleme beim Training selbstlernender Modelle	71
3.5.1. Verrauschte Messdaten	72
3.5.2. Veränderliche Prozessbedingungen	73

3.5.3.	Sicherheitskriterien und Daten geregelter Prozesse	75
3.6.	Fazit	76
4.	Umgang mit schlechter Datenqualität	77
4.1.	Verrauschte Daten	78
4.2.	Abschätzung der möglichen Approximationsgüte	82
4.2.1.	Der Gammatest zur Abschätzung des Rauschens	85
4.2.2.	Versuche mit künstlichen Daten	85
4.2.3.	Versuche mit realen Daten	90
4.3.	Auswirkungen von Rauschen auf den Lernvorgang	94
4.3.1.	Datenauswahl unter Beachtung von Rauschen	97
4.3.2.	Verfahren zur Ausreißererkenntnis	101
4.3.3.	Robuste Trainingsverfahren	111
4.3.4.	Kombination robuster Trainings und Datenvorverarbeitung	116
4.4.	Fazit	119
5.	Einfluss veränderlicher Systembedingungen	121
5.1.	Adaptivität - Überblick	122
5.2.	TestszENARIO	129
5.3.	Adaptivität bei Verwendung eines einzelnen Modells	134
5.3.1.	Datenakkumulation	134
5.3.2.	Bereichsauswahl	137
5.3.3.	Bereichsauswahl mit Kontrolle	139
5.3.4.	Ergebnisse auf künstlichen Daten	142
5.3.5.	Ergebnisse auf realen Daten aus Zementwerken	147
5.4.	Adaptivität bei Verwendung eines Ensembles von Netzen	151
5.4.1.	Ensemble Selection	152
5.4.2.	Ensemble Weighting	155
5.4.3.	Ergebnisse auf künstlichen Daten	157
5.4.4.	Ergebnisse auf realen Daten aus Zementwerken	158
5.5.	Übertragbarkeit auf Müllverbrennungsanlagen	162
5.6.	Fazit	164

6. Funktionale Sicherheit des lernenden Systems	167
6.1. Prozessregelung und funktionale Sicherheit	169
6.1.1. Prozessmodell als Eingabe des Reglers	175
6.1.2. Lernfähige Regler	177
6.2. Daten geregelter Prozesse und Expertenwissen	179
6.2.1. Lernen auf Daten eines geregelten Prozesses . . .	180
6.2.2. Sichere Strategie am Cart-Pole-Simulator	189
6.2.3. Sichere Strategie in einer MVA	196
6.2.4. Sichere Strategie in einem Zementwerk	202
6.3. Seltene Zustände und Expertenwissen	208
6.4. Untersuchung der Regelstrategie eines Modells	214
6.4.1. Einfluss der Stellgröße auf Zielgrößen	216
6.4.2. Wirkrichtung der Stellgröße	219
6.5. Ensembles	222
6.6. Fazit	227
7. Praktische Untersuchung der selbstlernenden Regelung	229
7.1. Model-Predictive Control und Softsensor	232
7.1.1. Prozessmodelle ohne direkten Regler	232
7.1.2. Prozessmodelle mit direktem konv. Regler	234
7.2. Ersetzung konventioneller Regler durch NFQ-Regler . . .	238
7.2.1. Praktische Untersuchungen MVA 1 - Zuteiler . . .	240
7.2.2. Praktische Untersuchungen MVA 1 - Rost	241
7.2.3. Praktische Untersuchungen MVA 1 - Zuteiler und Rost	243
7.2.4. Praktische Untersuchungen Zement - SNCR	245
7.3. Parametrisierungsaufwand selbstlernender System	248
7.3.1. Adaptive Modelle mit externer Stellstrategie . .	249
7.3.2. Adaptive Modelle mit interner Stellstrategie . . .	252
7.3.3. Hinweise zur Methodenauswahl	254
7.4. Fazit	256
8. Zusammenfassung	259

9. Ausblick	265
9.1. Parametrisierung und Überwachung mittels HMI	266
9.1.1. Online Priorisierung von Regelgrößen	266
9.1.2. Darstellung aktiver und inaktiver Modelle	269
9.1.3. Eingabemöglichkeit für Expertenwissen	269
9.2. Übertragbarkeit auf andere Basismodelle	273
9.2.1. Lineare Regressionsmodelle	273
9.2.2. Bayes'sche Modelle	274
9.2.3. Clusteranalyse	275
A. Merkmalsauswahl	277
A.1. Totzeit und Verzögerung	279
A.2. Mutual Information	282
A.3. Residual Mutual Information	285
A.4. Automatische Problem Dekomposition	288
B. Detaillierte Ergebnisse und Verfahrensparameter	293
B.1. Müllverbrennungsanlagen und Zementwerke	293
B.2. Cartpole-Versuchsumgebung	294
B.2.1. Physik und Parameter des Simulators	294
B.2.2. Versuche zum Lernen auf geregelten Daten	299
B.3. Umgang mit schlechter Datenqualität	301
B.3.1. Gammatest-Pseudocode	301
B.3.2. Abschätzung des Rauschens für verschiedene Datensätze	302
B.3.3. Auswirkungen zusätzlicher Eingaben	304
B.3.4. Auswirkungen der Datensatzgröße	305
B.3.5. Filterung von Outliern	306
B.4. Veränderliche Prozesse	310
B.5. Sicherheit und geregelte Prozesse	316
B.5.1. Parameter und Aufbau der Simulation	316
B.5.2. Erweiterte Versuchsergebnisse	317
C. Verwandte Untersuchungen an Verbrennungsprozessen	325

Literaturverzeichnis	331
Symbolverzeichnis	349
Abkürzungsverzeichnis	349

1. Einleitung

Die Nutzung des Feuers und die Kontrolle von Verbrennungsprozessen haben seit je her eine große Bedeutung für die Entwicklung der Menschheit. Die Entwicklung von Öfen zum Brennen von Keramik, die Metallverarbeitung in Schmelzen, die Erfindung der Dampfmaschine und deren Nutzung zur Erzeugung von Antriebskraft oder Energie gelten als wichtige Schritte der Entwicklung zur modernen Gesellschaft. Alle drei Beispiele haben eine Gemeinsamkeit: das gewünschte Ergebnis wird nur erzeugt, wenn der Verbrennungsprozess innerhalb bestimmter Grenzen abläuft. Werden diese Grenzen verlassen und das Feuer wird zu heiß oder zu kalt, ist das entstehende Produkt von geringer Qualität oder sogar unbrauchbar.

Es ist deswegen von großer Bedeutung, den Verbrennungsprozess möglichst kontrolliert und gleichmäßig ablaufen zu lassen. Im Zuge der Industrialisierung und der Weiterentwicklung der Wissenschaft wurde es möglich, immer mehr Informationen über einen laufenden Verbrennungsprozess zu erhalten. Durch widerstandsfähige Messgeräte können Temperatur und Druck innerhalb eines Ofens gemessen werden. Gleichzeitig wurde es möglich, durch Stellgrößen einen größeren Einfluss auf den Ablauf der Verbrennung zu gewinnen. Durch diese technischen Fortschritte konnten immer größere und komplexere Verbrennungsvorgänge konzipiert und industriell genutzt werden.

Ein weiterer wichtiger Schritt in der Kontrolle des Verbrennungsprozesses war die Automatisierung, die Nutzung elektrischer Schaltkreise und schließlich die Digitalisierung. Während bis dahin noch ein Mensch manuell auf eine Veränderung eines Messwertes reagieren musste, konnten nun Messwerte und Stellgrößen verknüpft werden.

Wird beispielsweise die Temperatur in einem Ofen zu hoch, so nutzt eine analoge oder digitale Schaltung direkt die Abweichung der Temperatur vom Sollwert und berechnet daraus die Korrektur einer Stellgröße (zum Beispiel die Menge der zugeführten Kühlluft). Die Korrektur der Stellgröße wird dann ebenfalls maschinell sofort durchgeführt. Durch die rasante Entwicklung der digitalen Rechentechnik konnten immer komplexere Schaltungen für einen solchen Kreis aus Messgröße und Regelgröße entwickelt werden.

Die dazu gehörige Wissenschaft, die Regelungstechnik als Teilgebiet der Automatisierungstechnik, entwickelte sich ebenfalls schnell fort und füllt heute dicke Lehrbücher.

Durch diese Entwicklungen wurde es möglich viele Verbrennungsprozesse äußerst präzise zu regeln und Abweichungen der gewünschten Temperatur oder Produktgüte auf ein Minimum zu reduzieren. Beispiele für solch präzise Verbrennungsvorgänge sind unter anderem die Herstellung von Keramik oder die Verbrennung in einem Fahrzeugmotor. Gleichzeitig gibt es weiterhin Verbrennungsprozesse, bei denen auch heute noch starke Schwankungen der Qualität oder der Temperaturen auftreten. Beispiele dafür sind die Verbrennungsprozesse in Müllverbrennungsanlagen oder Zementwerken.

Die Ursachen für die großen Unterschiede in der Qualität der Regelung liegen dabei vor allem in den verwendeten Brennstoffen. Besonders auffällig wird dieser Unterschied bei Müllverbrennungsanlagen. Aufgrund gesetzlicher Vorgaben für die Entsorgung von Abfall wird in Müllverbrennungsanlagen nicht nur normaler Hausmüll verbrannt, sondern auch Industrieabfälle, Bauschutt und zum Teil sogar nicht weiter verwertbare Schlacken aus Kläranlagen. Je nachdem, welche Müllsorte gerade verbrannt wird, ändern sich die Eigenschaften und der Ablauf des Verbrennungsvorganges. Die Einstellung eines Reglers für eine Stellgröße, die in einer solchen Anlage für eine konstante Temperatur im Feuerraum sorgen soll, muss also ständig an die jeweilige Müllzusammensetzung angepasst werden.

Hinzu kommt, dass sich die richtigen Einstellungen für die ideale Verbrennung eines Brennstoffes im Laufe der Betriebszeit ändern. An den Wänden des VerbrennungsOfens setzen sich Schlacken fest. Anlagenelemente, die hohen Temperaturen und mechanischen Belastungen ausgesetzt sind, verschleifen mit der Zeit. Dadurch wird es schwierig, eine allgemeine Regel zu definieren wie eine bestimmte Stellgröße zu stellen ist, da dies nicht nur von der aktuellen Verbrennungssituation sondern auch vom Brennstoff und der aktuellen Verschleißsituation der Anlage abhängt.



Abbildung 1.1. – Verbrennung von Müll auf dem Vorschubrost einer Müllverbrennungsanlage. Quelle: [Poulsen, 2007]

Für ein solches Anwendungsfeld, den Einsatz von Reglern in veränderlichen und teilweise unbekanntem Umgebungen, hat sich in den letzten Jahrzehnten ein eigenes Forschungsgebiet ergeben. Von besonderem Interesse sind hier adaptive Regler, deren Parameter sich an veränderte Systembedingungen anpassen können, bzw. lernfähige Systeme, die vorhandene Prozessdaten analysieren und daraus eine Regelstrategie ableiten. Beispielsweise existieren Verfahren, die innerhalb von 20 Minuten eigenständig lernen, ein Auto in der Mitte einer Fahrbahnspur zu halten [Riedmiller et al., 2007a] oder einen Stab auf einem beweglichen Wagen zu balancieren [Riedmiller, 2005a].

In dieser Arbeit wird untersucht, inwiefern sich solche adaptiven und lernfähigen Systeme auf schwierige Verbrennungsprozesse wie in Müllverbrennungsanlagen oder Zementwerken anwenden lassen. Aus der Vielzahl an lernfähigen Systemen werden die Vielversprechenden ausgewählt und auf ihre Eignung bei der Regelung eines industriellen Verbrennungsprozesses getestet. Im Vordergrund steht dabei die Praxistauglichkeit und die Zuverlässigkeit der Verfahren in sicherheitskritischen Bereichen. Der Schwerpunkt der Arbeit liegt deshalb vor allem auf der Analyse und der Anpassung unterschiedlicher Verfahren an die Daten von realen Verbrennungsprozessen sowie auf dem praktischen Einsatz der Verfahren bei der Regelung dieser Prozesse.

1.1. Anspruch der Arbeit

Diese Arbeit soll einen praktischen Beitrag zur Automatisierung komplexer Verbrennungsprozesse leisten. Der Schwerpunkt liegt dabei in der Integration selbstlernender Komponenten in die Feuerleistungsführung großindustrieller Verbrennungssysteme. Es wurden gezielt Müllverbrennungsanlagen und Zementwerke als Anwendungen ausgesucht, da Brennstoffwechsel und Prozessveränderungen eine konventionelle Regelung dieser Prozesse erschweren und sie interessant für die Anwendung adaptiver und lernfähiger Regelungssysteme machen.

In dieser Arbeit wird analysiert, welche Vorteile der Einsatz lernfähiger Systeme bei der Regelung solcher Prozesse bringen kann. Es wird allerdings auch erläutert, warum bekannte lernfähige Verfahren auf Daten von industriellen Verbrennungsprozessen oft scheitern. Für jeden der drei unterschiedlichen Bereiche Messrauschen, Prozessveränderung und funktionale Sicherheit wird gezeigt, wie die Merkmale des Verbrennungsprozesses zu Nachteilen bei der Anwendung bekannter Lernalgorithmen führen.

Um diese Nachteile auszugleichen, werden Modifikationen der bekannten Verfahren oder komplett neue Ansätze vorgestellt, mit denen auch auf den anspruchsvollen Daten industrieller Verbrennungsprozesse ein Einsatz lernfähiger Methoden möglich ist:

- Anhand mehrerer Versuche wird gezeigt, wie sich verrauschte Daten auf die Qualität von Prozessmodellen auswirken. Für die Auswahl und Weiterverarbeitung von sehr verrauschten Daten wird ein neuer Algorithmus vorgestellt, der vergleichbaren Algorithmen für den behandelten Anwendungsbereich überlegen ist.
- Für die Veränderung von Prozessen und die Anpassung von Prozessmodellen wird eine Systematisierung bekannter Verfahren vorgenommen. Die vorgestellten Verfahren werden anschließend an die Aufgabenstellung dieser Arbeit angepaßt und auf langen Datenbereichen von bis zu zwei Jahren getestet. Durch die Kombination verschiedener Verfahren kann eine deutliche Verbesserung gegenüber den Ausgangsverfahren erzielt werden.
- Die funktionale Sicherheit der lernfähigen Regler spielt für die praktische Anwendbarkeit eine sehr große Rolle. Es wird begründet, warum sich viele Trainingsverfahren nicht für den Einsatz in Feuerleistungsführungen eignen. Gleichzeitig werden neue Algorithmen zum Training, zur Analyse und zur Kontrolle von lernfähigen Modellen vorgestellt, die ein höheres Maß an funktionaler Sicherheit bieten.

Algorithmen aus allen drei Bereichen werden anschließend verwendet, um lernfähige Regler zu erstellen, die in mehreren Müllverbrennungsanlagen und Zementwerken im täglichen Betrieb eingesetzt werden. Durch Versuche über mehrere Wochen wird gezeigt, dass diese Regler zu einer Verbesserung der Prozessführung für die jeweiligen Ziele führen.

1.2. Gliederung

Diese Arbeit wurde mit besonderem Augenmerk auf die praktische Anwendbarkeit der vorgestellten Verfahren für industrielle Verbrennungsprozesse erstellt. In Kapitel 2 werden deshalb zuerst die Funktionsweise und die Besonderheiten eines industriellen Verbrennungsprozesses am Beispiel einer Müllverbrennungsanlage erläutert. Es wird weiterhin darauf eingegangen, warum konventionelle Regelungssysteme Teile des Prozesses nicht zufriedenstellend regeln können.

In Kapitel 3 werden die Grundlagen der in dieser Arbeit benutzten lernfähigen Verfahren erläutert. Es wird darauf eingegangen, warum selbstlernende Verfahren grundsätzlich konventionellen Regelungen überlegen sind und warum andererseits viele bekannte Lernverfahren bisher nicht zum Einsatz kommen konnten. Kapitel 3 legt damit die Basis für alle weiteren Kapitel. Die wesentlichen Besonderheiten und Probleme, die in Kapitel 3 vorgestellt werden, sind:

- verrauschte Messdaten,
- zeitlich veränderliche Prozessvariablen/Prozessabläufe
- und funktionale Sicherheitsbedingungen.

Jedes der drei folgenden Kapitel 4-6 behandelt Lösungen eines dieser drei Problemfelder, die häufig bei der Konzeption einer selbstlernenden Regelung für einen Verbrennungsprozess auftauchen. Alle drei Kapitel sind so aufgebaut, dass sie in Verbindung mit Kapitel 3 gelesen werden können ohne notwendigerweise die anderen beiden Kapitel ebenfalls zu lesen. Falls ein Anwender also nur mit einem der genannten Probleme konfrontiert ist, kann er sich auf das jeweilige Kapitel konzentrieren.

In Kapitel 7 wird dann die Zusammenführung aller Verfahren, die in Kapitel 4-6 vorgestellt wurden, zu einem Gesamtsystem für die selbstlernende Regelung eines Verbrennungsprozesses kombiniert, und es werden die Ergebnisse diese Systems im Einsatz in verschiedenen Anlagen präsentiert.

Die Arbeit schließt mit einer Zusammenfassung in Kapitel 8 und einem Ausblick auf mögliche Erweiterungen in Kapitel 9.

In Abbildung 1.2 ist eine grafische Darstellung der Kapitel dieser Arbeit, symbolisch in der Form einer industriellen Verbrennungsanlage, abgebildet. Zu Beginn jedes Kapitels wird erneut diese Grafik erscheinen, wobei das aktuelle Kapitel hervorgehoben wird um dem Leser als Orientierung zu dienen.

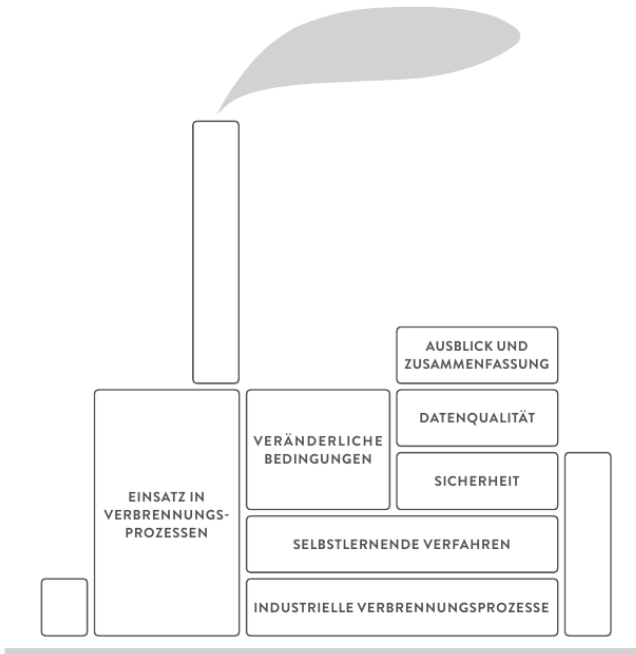
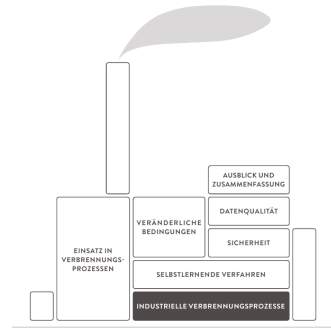


Abbildung 1.2. – Grafische Darstellung der Gliederung der Arbeit. Zuerst werden die Grundlagen industrieller Verbrennungsprozesse und lernfähiger Verfahren zu deren Kontrolle vorgestellt. Beim Einsatz dieser Verfahren auf den Daten realer Anlagen ergeben sich die drei Problemfelder: veränderliche Daten, unzureichende Datenqualität und funktionale Sicherheit. Lösungen zu jedem Bereich werden in einem eigenen Kapitel erläutert. Alle Verfahren werden anschließend in realen Anlagen eingesetzt und die Ergebnisse bei der Prozessführung dargestellt. Zuletzt folgt eine Zusammenfassung der Arbeit und ein Ausblick auf mögliche Erweiterungen.

2. Besonderheiten industrieller Verbrennungsprozesse



In der Einleitung wurde erläutert, dass diese Arbeit mit besonderem Fokus auf automatisierten Lernverfahren in industriellen Verbrennungsprozessen erstellt wurde. In diesem Kapitel werden deshalb zuerst die Besonderheiten und Anforderungen der Regelung eines industriellen Verbrennungsprozesses am Beispiel einer Müllverbrennungsanlage (MVA) erläutert. Es sollen dabei nicht technische oder mathematische Details im Vordergrund stehen. Schwerpunkt ist stattdessen zu erklären, welche Abläufe in einer großen Verbrennungsanlage stattfinden und inwiefern selbstlernende Systeme dabei von Nutzen sein können.

Zuerst wird dazu in Abschnitt 2.1 der prinzipielle Aufbau einer MVA mit den einzelnen großen Funktionsgruppen erläutert. In Abschnitt 2.1.1 werden anschließend der Verbrennungsraum, der für diese Arbeit von wesentlichem Interesse ist, sowie die Stellgrößen, die zur Beeinflussung der Verbrennung zur Verfügung stehen, näher betrachtet.

In Abschnitt 2.1.2 wird dargelegt, wie eine konventionelle Regelung eines Verbrennungsraumes aufgebaut ist. Eine solche konventionelle Regelung hat systembedingt einige Nachteile. Da diese Nachteile die wesentliche Motivation darstellen, eine komplexere selbstlernende Regelung zu verwenden, werden sie in einem eigenen Abschnitt 2.1.3 erläutert.

Da in dieser Arbeit nicht nur Müllverbrennungsanlagen als Anwendungsgebiet genutzt werden, sondern auch Prozesse in Zementwerken, wird in Abschnitt 2.2 auf die wesentlichen Unterschiede zu den Regelungssystemen in Zementwerken eingegangen.

Das Kapitel wird beschlossen mit einem Fazit in Abschnitt 2.4.

2.1. Müllverbrennungsanlagen

In diesem Abschnitt wird der generelle Aufbau einer Müllverbrennungsanlage beschrieben. Es wird auf die wesentlichen Ziele der Müllverbrennung und die daraus abgeleiteten Baugruppen eingegangen. Der Fokus dieses Abschnitts liegt nicht auf den komplexen technischen Details einzelner Baugruppen, da diese sich je nach Anlagentyp unterscheiden können¹. Stattdessen soll ein Überblick über den Ablauf der Müllverbrennung und die dabei zu beachtenden Ziele und Randbedingungen gegeben werden.

Aktuelle Müllverbrennungsanlagen können in drei Typen unterteilt werden: Rostfeuerungs-Anlagen, Pyrolyseanlagen und Sondermüllverbrennungsanlagen. Für normalen Hausmüll werden hauptsächlich Rostfeuerungs-Anlagen genutzt. Eine moderne Müllverbrennungsanlage soll durch ihren Betrieb drei wesentliche Ziele erfüllen:

1. Der angelieferte Müll soll verbrannt werden, wodurch sein Volumen reduziert wird. In Deutschland muss der Anteil des Unverbrannten in der Schlacke unter 3% reduziert werden². Die entstehende Schlacke kann je nach Zusammensetzung weiter verwendet werden (beispielsweise im Straßenbau) oder wird auf Deponien gelagert.
2. Die durch die Verbrennung entstehende Wärme soll mit einem möglichst hohen Wirkungsgrad weiter genutzt werden. Meistens geschieht dies durch die Erzeugung von Dampf, der im weiteren zur Stromerzeugung oder zur Produktion von Heißwasser für Fernwärme verwendet wird.

¹Eine umfassende Darstellung sämtlicher Anlagentypen und Baugruppen sowie ein Verzeichnis (fast) aller Müllverbrennungsanlagen in Deutschland findet sich in [Thome-Kozmiensky, 1994]

²Siehe 3.Allgemeine Verwaltungsvorschrift zum Abfallgesetz (TA Siedlungsabfall (TASi), Bundesgesetzblatt, 1992

3. Bei der Verbrennung des Mülls sollen so wenig wie möglich giftige oder umweltschädliche Gase entstehen. Für Anlagen in Deutschland sind die Grenzwerte für Abgase wie CO , NO_x , SO_2 und andere Schadstoffe in der Bundesimmissionsschutzverordnung³ festgeschrieben.

Für den Anlagenbetreiber besteht zusätzlich noch das Ziel, die Anlage möglichst kostengünstig mit wenigen Wartungen und Reparaturen zu betreiben.

Aus diesen Zielen leitet sich der prinzipielle Aufbau einer Müllverbrennungsanlage ab. Sie besteht stets aus einem Müllbunker, einem Verbrennungskessel und einer Rauchgasreinigung. Die Größe und der genaue Aufbau der einzelnen Komponenten kann von Anlage zu Anlage variieren, vorhanden sind alle drei Komponenten aber immer. Alle drei Komponenten werden im Folgenden einzeln beschrieben. Der Aufbau ist außerdem in Grafik 2.1 dargestellt.

Für die Versuche in dieser Arbeit wurden drei verschiedene Müllverbrennungsanlagen und sieben Zementwerke genutzt. Im Hauptteil der Arbeit werden die Anlagen durch ein Kürzel (z.B. MVA1 und Zem3) dargestellt. In Anhang B.1 finden sich detailliertere Angaben zu den einzelnen Anlagen.

Müllbunker Der Müllbunker dient zur Aufnahme und Zwischenlagerung des angelieferten Mülls. Je nach Größe der MVA hat der Müllbunker eine Kapazität von $3.000m^3$ bis $25.000m^3$. An der Decke des Müllbunkers sind Kräne befestigt, mit denen der Müll in den Einfülltrichter des Kessels befördert werden kann. Die Müllkräne werden außerdem benutzt, um den angelieferten Müll im Bunker zu vermischen und gegebenenfalls in den installierten Sperrmüllscheren zu zerkleinern. Bei den meisten Anlagen erfolgt die Anlieferung von Müll nur von Montag bis Freitag. Am Wochenende muss der Betrieb der Verbrennungskessel durch den bereits im Müllbunker gelagerten Müll geschehen.

³Siehe 17. Bundesimmissionsschutzgesetz (BImSchG) (1990/1999).

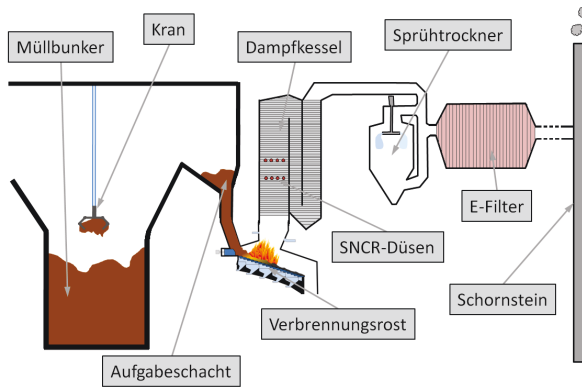


Abbildung 2.1. – Querschnitt der Elemente einer Müllverbrennungsanlage mit Müllbunker, Verbrennungskessel und Rauchgasreinigung. Oftmals ist die Rauchgasreinigung noch wesentlich komplexer als hier dargestellt.

Verbrennungskessel Im Verbrennungskessel wird der durch den Einfülltrichter angekommene Müll auf einem Rost- oder Walzensystem verbrannt. Viele Müllverbrennungsanlagen verfügen nicht nur über einen, sondern mehrere Verbrennungskessel, die gleichzeitig betrieben werden. Der Müll gelangt durch einen beweglichen Stößel oder ein Förderband auf den Verbrennungsgasrost, wo er unter Zugabe von Luft zu Schlacke verbrennt. Das System des Verbrennungskessels und der darin ablaufenden Müllverbrennung ist für diese Arbeit von besonderem Interesse und wird ausführlich in Abschnitt 2.1.1 beschrieben.

Rauchgasreinigung Die Rauchgasreinigung dient dazu sicherzustellen, dass nur ein geringer Prozentsatz der giftigen oder umweltschädlichen Gase, die während der Verbrennung entstehen, in die Umwelt gelangt. Die Rauchgasreinigung nimmt meist den größten Teil des Gebäudes einer Müllverbrennungsanlage ein und ist häufig auch der kostenintensivste Teil der Anlage.

In Deutschland gelten für die Abgase aus Müllverbrennungsanlagen die Emissionsgrenzwerte der 17. Verordnung zum BImSchG. Darin festgelegt sind Grenzwerte für Schwermetalle wie Blei und Quecksilber, Staub, Kohlenmonoxid sowie Stick- und Schwefeloxide. Um diese Giftstoffe zu entfernen, durchläuft das Rauchgas viele verschiedene Stationen. Bereits im Verbrennungskessel wird durch Eindüsung von Sekundärluft Kohlenmonoxid (CO) nachverbrannt und durch Eindüsung von Ammoniakwasser die Entstehung von Stickoxiden (NO_x) verhindert. Nach Verlassen des Kessels werden in der Nachbehandlung der Rauchgase als erstes mittels eines Sprühtrockners und eines Elektrofilters Staub- und Metallpartikel aus dem Gas entfernt. Dann wird das Rauchgas im Quench durch Wassereindüsung abgekühlt und dabei von Chlorwasserstoff (HCl) befreit. Anschließend wird durch eine Wäsche mit Natronlauge Schwefeldioxid (SO_2) abgeschieden. In der letzten Stufe der Reinigung werden die Rauchgase durch Gewebefilter (Flugstrom-Adsorber) von Dioxinen und Furanen befreit. Die so gereinigten Rauchgase verlassen schließlich durch den Schornstein die Anlage⁴.

In dieser Arbeit wird das System der Rauchgasreinigung im Folgenden nur eine geringe Bedeutung haben. Die hier behandelte Feuerleistungsregelung hat nur Zugriff auf Stellgrößen, die sich im Bereich des Kessels befinden, und nicht in der nachfolgenden Reinigungsanlage. Die entstehenden Abgase sind allerdings trotzdem von großer Bedeutung, da bereits während der Verbrennung möglichst wenige Schadstoffe erzeugt werden sollen, um die Kosten der nachfolgenden Rauchgasreinigung gering zu halten.

⁴Je nach Anlage können teilweise andere Bauteile verwendet werden. Die Behandlung von NO_x ist z.B. sowohl durch eine SNCR- als auch durch eine SCR-Anlage möglich. Eine Übersicht der Alternativen findet sich in [Thome-Kozmiensky, 1994]. Der hier dargestellte Ablauf trat allerdings bei den in dieser Arbeit untersuchten Anlagen am häufigsten auf.

2.1.1. Aufbau des Verbrennungsraumes

In diesem Abschnitt wird der Aufbau des Verbrennungskessels einer Müllverbrennungsanlage erläutert. Dieser Aufbau ist von besonderem Interesse, da im Rahmen dieser Arbeit Feuerleistungsführungen untersucht werden, deren Stellensignale direkt auf Komponenten des Verbrennungskessels zugreifen. In Grafik 2.2 ist der Querschnitt eines Verbrennungskessels einer MVA des Rostfeuerungstyps zu sehen.

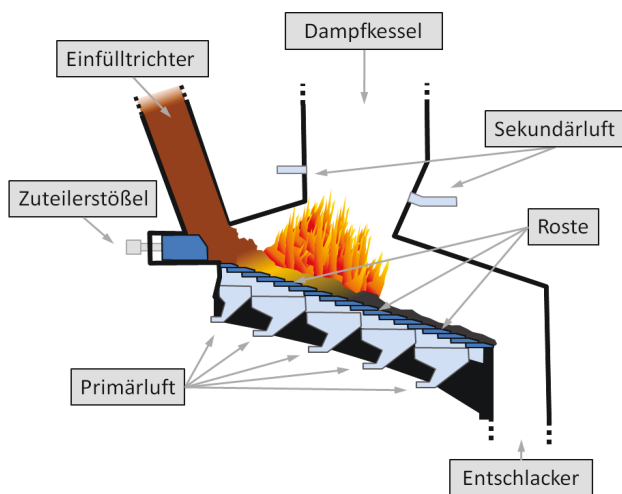


Abbildung 2.2. – Querschnitt des Verbrennungsrostes einer MVA. Die für die Feuerleistungsregelung wichtigen Stellgrößen sind gekennzeichnet.

Durch den Einfülltrichter wird der Müll auf den Verbrennungsrost gegeben. Am unteren Ende des Trichters ist entweder ein Zuteilerstößel oder ein Förderband angebracht. Der Stößel wird durch einen hydraulischen Antrieb bewegt. Mit jeder Vorwärtsbewegung des Stößels wird Müll aus dem Trichter auf den Verbrennungsrost geschoben.

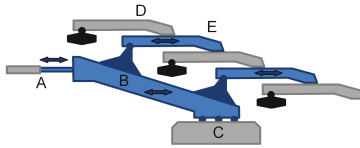
Die Geschwindigkeit des Stößels oder des Förderbandes, und damit auch die Menge an Müll, die pro Zeiteinheit auf den Rost gelangt, ist durch die Feuerleistungsregelung einstellbar.

Nach dem Verlassen des Trichters gelangt der Müll auf den Verbrennungsrost. Es sind dabei Vorschubroste, Rückschubroste oder Walzenroste möglich. Bei einem Vorschubrost ist jedes zweite Rostelement beweglich und kann über die davor liegende Rostbahn geschoben werden. In Abbildung 2.3a ist dieses Prinzip des Vorschubrostes als Skizze zu sehen und in Abbildung 2.3b eine Aufnahme des Feuers auf dem Vorschubrost einer MVA. Die Geschwindigkeit, mit der die Roste sich vorwärts bewegen, ist durch die Feuerleistungsregelung einstellbar. Je nach Größe des Verbrennungskessels existieren 3-6 Rostzonen, für die sich die Geschwindigkeit der jeweiligen Rostelemente separat einstellen lässt. Je nach Anlagenbauweise werden die vordersten Rostzonen, auf denen häufig die höchsten Temperaturen herrschen, durch einen Wasserkreislauf gekühlt.

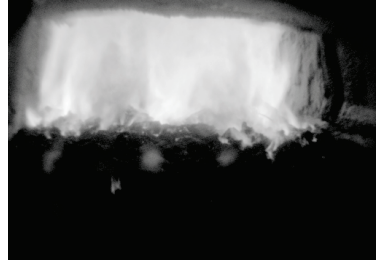
Damit der Müll auf dem Rost verbrennt, ist die ständige Zufuhr von Luft erforderlich. Die Luft, die direkt durch die Rostelemente in die Verbrennungszone geblasen wird, bezeichnet man im Bereich der Müllverbrennung als Primärluft. Für jede der Rostzonen lässt sich eine eigene Primärluftmenge einstellen. Die Verbrennung des Mülls auf dem Rost kann dabei laut [Wolf, 2005] und [Beckmann and Spiegel, 2006] in vier Phasen eingeteilt werden:

1. Trocknung des Mülls
2. Entgasung und Pyrolyse
3. Hauptverbrennung
4. Nachverbrennung

In Abbildung 2.4 sind diese vier Phasen dargestellt. Aufgabe der Feuerleistungsregelung ist es, entsprechend dieser Phasen jeweils eine möglichst günstige Primärluftmenge in jeder der Rostzonen



(a) Querschnitt



(b) Blick auf das Müllfeuer aus Richtung des Naßentschlackers

Abbildung 2.3. – Funktionsweise eines Vorschubrostes: Eine Antriebsstange (A) bewegt den Rostschlitten (B) der auf einem Lauflager (C) aufliegt und auf dem die beweglichen Rostelemente (E) montiert sind. Zwischen den beweglichen Rostelementen sind unbewegliche Elemente (D). Alle beweglichen Teile sind blau eingefärbt. Rechts ist der Blick auf das Müllfeuer einer MVA mit Vorschubrost während des laufenden Betriebs aus Sicht der Wand am Schlackeabwurf zu sehen. Durch die Bewegung der Rostelemente wird der Müll in Richtung des Betrachters geschoben.

einzustellen. Dabei muss die Beschaffenheit des Mülls beachtet werden, da je nach Feuchte und Zusammensetzung die Größe und Dauer der einzelnen Verbrennungsphasen auf dem Rost variiert.

Zusätzlich zur Primärluft wird an den Seitenwänden und oberhalb der Verbrennungszone Luft eingedüst. Die Lufteindüsung an den Seitenwänden dient dazu Anbackungen des Mülls an den Wänden des Kessels zu verhindern. Oberhalb der Hauptverbrennungszone befindet sich die Sekundärlufteindüsung. Sie ist wichtig um die Nachverbrennung der entstehenden Gase zu gewährleisten, beispielsweise die Umwandlung von giftigem Kohlenmonoxid (CO) in ungefährlicheres Kohlendioxid (CO_2).

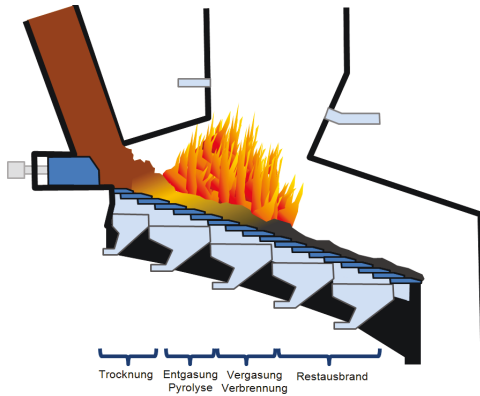


Abbildung 2.4. – Verbrennungszonen auf einem Rost

Häufig geschieht die Eindüsung der Sekundärluft an den Wänden am Übergang des Feuerraums in dem Dampfkessel, manchmal auch über einen eigens dafür installierten Prismabalken. Die Gesamtmenge und die Aufteilung der Sekundärluft auf die einzelnen Eindüsungspositionen ist durch die Feuerleistungsregelung einstellbar.

Die heißen Verbrennungsgase ziehen dann vom Verbrennungsraum in den Dampfkessel. Dieser ist mit Rohren ausgekleidet, in denen Wasser durch die Hitze der Verbrennungsgase in Dampf umgewandelt wird. Eine detaillierte Darstellung der Prozesse im Dampfkessel findet sich in [Effenberger, 1999]. Je nach Müllverbrennungsanlage wird der erzeugte Dampf durch eine Turbine und einen Generator in Strom umgewandelt, zur Erzeugung von Fernwärme genutzt oder durch Verbindungsrohre an einen Abnehmer geliefert.

Die bei der Verbrennung entstehende Schlacke sowie unverbrannte Bestandteile des Mülls (hauptsächlich Metalle) werden über den Rost weiter transportiert und fallen schließlich in den Nassentschlacker.

2.1.2. Konventionelle Regelung der Verbrennung

Im vorherigen Abschnitt wurde dargestellt, wie die Verbrennung des Mülls auf dem Verbrennungsrost einer typischen Müllverbrennungsanlage abläuft. In diesem Abschnitt wird nun erläutert, wie eine herkömmliche Regelung dieses Verbrennungsprozesses ohne selbstlernende Komponenten aufgebaut ist.

Die Regelung des Verbrennungsprozesses ist an erster Stelle abhängig von vorhandenen Bauteilen und dem Aufbau des Verbrennungskessels. In [Thome-Kozmiensky, 1994] sind die Kennzahlen und einige der Bauteile von über 50 Müllverbrennungsanlagen aufgeführt. Kaum eine Anlage ist dabei vollkommen baugleich mit einer anderen. Demzufolge sind auch die verwendeten Feuerleistungssysteme unterschiedlich.

Nach [Thome-Kozmiensky, 1994] lassen sich trotzdem zwei unterschiedliche Regelungsansätze unterscheiden:

- **Festlastregelung:** Es wird ein Sollwert für die zu produzierende Dampfmenge vorgegeben. Aus Heizwert und gewünschtem Luftüberschuß wird eine dafür nötige Verbrennungsluftmenge errechnet. Auf Schwankungen der produzierten Dampfmenge wird durch Änderung der Rost- und Zuteilergeschwindigkeit reagiert.
- **Verbrennungsluftregelung:** Es wird ein Sollwert für den Mülldurchsatz vorgegeben, der relativ konstant gehalten wird. Auf Änderungen der produzierten Dampfmenge wird mit einer Veränderung der Verbrennungsluftmenge reagiert.

Bei den in dieser Arbeit untersuchten Anlagen kam hauptsächlich das erste Konzept, die Festlastregelung, zum Einsatz.

Steht das grundsätzliche Konzept fest, so muss entschieden werden, wie eine Reglerstruktur zur Umsetzung dieses Konzeptes aufgebaut sein soll. Die wesentliche Frage dabei ist, welche Stellgrößen zur Führung welcher Zielgrößen genutzt werden sollen. Da kaum eine Anlage komplett baugleich mit einer anderen ist, ist dies ein schwieriger Prozess⁵. Oftmals wird vom Hersteller einer Anlage eine Reglerstruktur geliefert, welche aufgrund von Erfahrungswerten und Simulationen [Wolf, 2005, Görner et al., 1999, Auel and Kühl, 2007] aufgestellt wurde. Diese Regelstruktur wird dann häufig an die Bedingungen, Ziele und Besonderheiten der konkreten Anlage angepaßt. In [Leskens et al., 2010] oder [Thome-Kozmiensky, 1994] finden sich Beispiele für Regelkonzepte verschiedener Anlagen.

In Tabelle 2.1 ist der Einfluss der vorhandenen Stellgrößen auf die zu beachtenden Ziele für eine in dieser Arbeit untersuchte Müllverbrennungsanlage (MVA1) dargestellt. Die dafür nötigen Daten wurden durch Analyse der vorhandenen Reglerstruktur und Gespräche mit den Anlagenfahrern gewonnen.

⁵In Anhang A.4 ist ein Ansatz dargestellt, mit dem auf automatischem Wege eine solche Struktur gefunden werden kann.

	Dampfproduktion	Schlackequalität	Emissionen (CO)	Emissionen (NO_x)	Emissionen (O_2)
Primärluftmenge	mittel	mittel	gering	gering	mittel
Primärluftvert.	gering	mittel	mittel	mittel	mittel
Sekundärluft	gering	gering	groß	gering	groß
Rostgeschw.	groß	gering	gering	mittel	groß
Zuteilergeschw.	mittel	groß	groß	mittel	gering
NH_3-Eindüsung	gering	gering	gering	groß	gering

Tabelle 2.1. – Einfluss der Stellgrößen auf einzelne Ziele der Verbrennung bei einer MVA mit Vorschubrost und Festlastregelung in Nordrhein-Westfalen. Die verschiedenen Einflüsse wurden im Rahmen dieser Arbeit durch Datenanalyse, Experimente und Befragung der Operatoren vor Ort ermittelt.

Nachdem ermittelt wurde, welche Stellgrößen auf welche Zielgrößen wirken, wird eine Regelung für jede Stellgröße aufgestellt. Bei einer konventionellen Regelung kommen dabei als Elemente der Regelung im Wesentlichen Kennlinien und Proportional-Integral-Differential (PID)-Regler zum Einsatz ⁶.

⁶Alle in dieser Arbeit untersuchten Anlagen, unabhängig ob es sich um Müllverbrennungsanlagen oder Zementwerke handelte, verwendeten vor der Optimierung durch STEAG Powitec ein System auf Basis von Kennlinien und PID-Reglern. Auch die in [Leskens et al., 2010, Thome-Kozmiensky, 1994] geschilderten Systeme bestehen aus diesen Elementen. [Jain and De Silva, 1999] führt eine Studie in Raffinerien in Japan auf, bei denen 91,3% der Stellgrößen durch einfache oder erweiterte PID-Regler gesteuert werden.

Eine Kennlinie oder ein PID-Regler ist stets so aufgebaut, dass auf Basis eines Eingangssignals ein Ausgangssignal berechnet wird. Ein einzelnes Regelement dieser Art ist also ein Single-Input-Single-Output (SISO) System. Wirkt eine Stellgröße auf mehr als eine Zielgröße, so muss eine der Zielgrößen entweder ignoriert werden, oder es müssen mehrere Regelemente verwendet werden, deren Ausgaben am Ende zu einer Gesamtausgabe vereinigt werden.

Ein PID-Regler besteht aus drei Komponenten: einem **P**roportional-, einem **I**ntegral- und einem **D**ifferential-Anteil. Als Eingangssignal eines PID-Reglers wird die Abweichung $e(t) = w(t) - x(t)$ der Zielgröße $x(t)$ von ihrem Sollwert $w(t)$ verwendet. Die Reaktion $u(t)$ des PID-Reglers berechnet sich laut [Mann et al., 2005] dann wie folgt:

$$u(t) = K_P \left[e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau + T_V \frac{d}{dt} e(t) \right] \quad (2.1)$$

Durch die drei Parameter K_P (Verstärkung), T_N (Nachstellzeit) und T_V (Vorhaltzeit) lässt sich das Verhalten des PID-Reglers an unterschiedliche Regelstrecken anpassen. Wird T_V beispielsweise auf 0 gestellt, so entfällt der Differential-Anteil komplett.

In [Lutz and Wendt, 2002, Mann et al., 2005] sind ausführliche Beschreibungen zur Wirkungsweise von PID-Reglern und deren Konfiguration für verschiedene Regelstrecken dargestellt.

Eine Kennlinie definiert das Grundniveau einer Stellgröße. Im Unterschied zu einem PID-Regler wird sie meist als Steuerung und nicht als Regelung konzipiert. Das heißt der Eingangswert ist keine Regelabweichung die korrigiert werden soll, sondern er wird entweder von außen vorgegeben (z.B. die gewünschte Last) oder ist die einfache Messung einer Prozessgröße. In Abbildung 2.5 sind verschiedene Kennlinientypen dargestellt. Beim Einsatz in der Feuerleistungsführung werden meist lineare Kennlinien verwendet, deren Eingangswert entweder der Dampfleistungssollwert oder der Sollwert des Mülldurchsatzes ist⁷.

⁷Kennlinien in Zementwerken hingegen sind häufig vom gewählten Rezept und von der Rohmehlmenge abhängig.

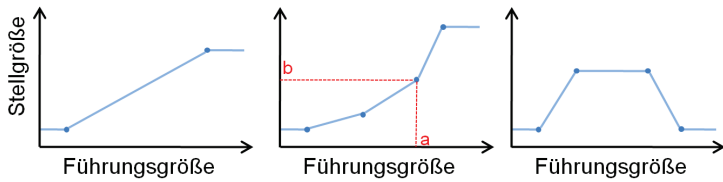
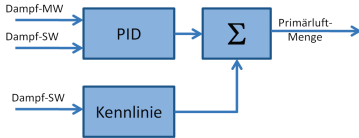


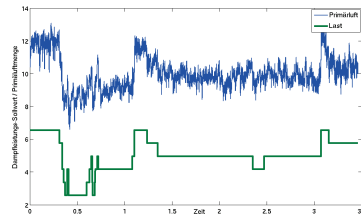
Abbildung 2.5. – Beispiele für Kennlinien: linear ansteigend, nichtlinear ansteigend, Peak. Für einen bestimmten Wert (a) der Führungsgröße wird ein Wert (b) der Stellgröße definiert. Zwischen den definierten Punkten wird linear interpoliert. Die Kennlinie berechnet dann die Ausgabe der Stellgröße für einen Wert der Führungsgröße.

Die erläuterten Elemente lassen sich so kombinieren, dass eine Regelstruktur für Prozessgrößen in der Feuerleistungsführung entsteht. In Abbildung 2.6 ist beispielhaft eine solche Struktur zum Regeln der Gesamt-Primärluft einer MVA in Abhängigkeit von der Dampfleistung dargestellt. Als Eingangsgrößen dienen dabei der gewünschte Dampfleistungssollwert und die aktuelle Dampfleistung. Abhängig vom Dampfleistungssollwert wird mit Hilfe einer Kennlinie das Grundniveau der Primärluftmenge bestimmt. Die Regelabweichung zwischen der aktuellen Dampfleistung und dem Sollwert wird berechnet und dient als Eingangsgröße eines PID-Reglers. Je nach aktueller Abweichung kann dieser PID-Regler einen bestimmten Betrag von der Kennlinie abziehen oder sie erhöhen.

Die Ausgaben des Kennlinienmoduls und des Reglers werden summiert und dann mittels der Prozesskopplung an die entsprechenden Stellgrößen übergeben, welche die Primärlüfter entsprechend stellen.



(a) Reglerstruktur



(b) Verlauf der Last und Primärluftmenge

Abbildung 2.6. – Beispiel für die Struktur zur Stellung einer Stellgröße mittels Kennlinie und PID-Regler. Ebenso dargestellt ist der Verlauf der Primärluftmenge in MVA1, die nach dem Prinzip Kennlinie+Regler gestellt wird. Der Last-Sollwert bestimmt das Niveau der Menge, der PID-Regler regelt den genauen Bedarf je nach Zustand der Anlage.

Dieses Beispiel ist natürlich stark vereinfacht. In der Realität muss außerdem der Druck im Kessel sowie die Auslastung des Saugzuges berücksichtigt werden, bevor die Primärluft erhöht wird. Weiterhin dürfen die meisten Stellgrößen nicht mit einem zu großen Gradienten gefahren werden, und es sind weitere Module notwendig, die die Gesamtluftmenge in Einzelmengen an den jeweiligen Eindüsungen umrechnen und entsprechende Klappenstellungen durchführen. Trotzdem wird durch dieses Beispiel die Funktionsweise einer konventionellen Regelung deutlich. Eine derartige Regelung kam in allen in dieser Arbeit untersuchten Anlagen vor.

Im nächsten Abschnitt wird untersucht, welche Probleme beim Einsatz einer solchen Regelung in der Feuerleistungsführung auftreten können, um anschließend zu erläutern, welchen Nutzen die Ersetzung oder Ergänzung einzelner Komponenten durch lernfähige Elemente hat.

2.1.3. Nachteile der konventionellen Regelung

Im vorherigen Abschnitt wurde dargestellt, wie eine konventionelle Feuerleistungsführung einer Müllverbrennungsanlage funktioniert. In diesem Abschnitt wird nun erläutert, welche Nachteile eine solche Regelung hat und warum es von Vorteil sein kann, stattdessen eine Feuerleistungsführung mit selbstlernenden Komponenten zu verwenden.

Die Probleme konventioneller Feuerleistungsführungen lassen sich dabei prinzipiell in drei Bereiche gliedern:

1. *Monotone Funktionalität des Reglers*: nichtmonotone Prozesszusammenhänge zwischen der Stellgröße und der zu beeinflussenden Zielgröße können nicht behandelt werden
2. *Mehrfachwirkung auf Zielgrößen*: Wirkung einer Stellgröße auf mehrere Zielgrößen, die sich durch einen einzelnen SISO-Regler nicht abbilden lässt
3. *Prozessveränderung*: Veränderungen des Prozesses durch andere Brennstoffe, Verschlackungen oder Verschleiß erfordern jeweils eigene Regelparameter für eine optimale Regelgüte

Auf jeden dieser drei Bereiche wird im Folgenden gesondert eingegangen.

Nichtmonotone Prozesszusammenhänge

Ein sehr häufig auftretendes Problem besteht darin, dass der Zusammenhang zwischen einer Stellgröße und der Zielgröße, auf die der zugehörige Regler wirken soll, zum einen nichtlinear und zum zweiten nichtmonoton ist. Ein PID-Regler ist prinzipiell für lineare Zusammenhänge ausgelegt und kann durch den I- und D-Anteil nur in geringem Maße auch auf Nichtlinearitäten eingehen⁸.

⁸Selbst wenn der Prozess linear ist, können durch Totzeiten und Verzögerungen Probleme mit PID-Reglern entstehen. In [Ingimundarson and Hägglund, 2001] wird gezeigt, dass ein PID-Regler Prozesse mit großer Verzögerung nur ungenügend regeln kann.

Noch bedeutender ist, dass der PID-Regler grundsätzlich nur für monotone Zusammenhänge eingesetzt werden kann. In Abbildung 2.7 ist der Unterschied zwischen monotonen und nichtmonotonen Zusammenhängen dargestellt.

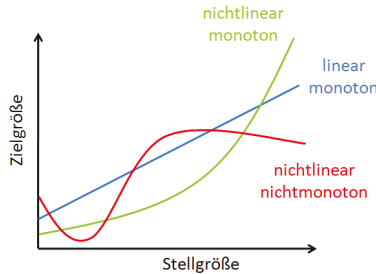


Abbildung 2.7. – Beispiele für den Verlauf eines linearen, eines nichtlinearen monotonen und eines nichtlinearen nichtmonotonen Zusammenhangs zwischen Stellgröße und Zielgröße.

Wie in Abschnitt 2.1.2 gezeigt, berechnet sich die Stellausgabe $u(t)$ eines PID-Reglers zum Zeitpunkt t nach der Formel:

$$u(t) = K_P \left[e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau + T_V \frac{d}{dt} e(t) \right] \quad (2.2)$$

wobei $e(t)$ der Fehler zwischen Sollwert und Istwert der zu regelnden Zielgröße ist. Wesentliche Bedeutung hat dabei der Parameter K_P , der die Richtung der Stellausgabe vorgibt. Ist K_P positiv, so wird auf einen negativen Regelfehler $e(t)$ mit einer Erhöhung der Stellgröße $u(t)$ reagiert. Ist K_P negativ, wird die Stellgröße bei gleichem Fehler stattdessen verringert.

Ist der Zusammenhang zwischen Stellgröße und Zielgröße nichtmonoton, so gibt es Stellbereiche, in denen ein positives K_P erforderlich ist und gleichzeitig auch Stellbereiche, in denen ein negatives K_P erforderlich wäre. Das lässt sich durch einen einzelnen, mit festen Parametern arbeitenden PID-Regler nicht realisieren.

Dass dieses Problem der nichtmonotonen Prozesszusammenhänge tatsächlich auftritt, ist in Abbildung 2.8 zu sehen. Dargestellt ist der Zusammenhang zwischen Veränderung der Rostgeschwindigkeit und Veränderung der Dampfleistung an einer realen Müllverbrennungsanlage.

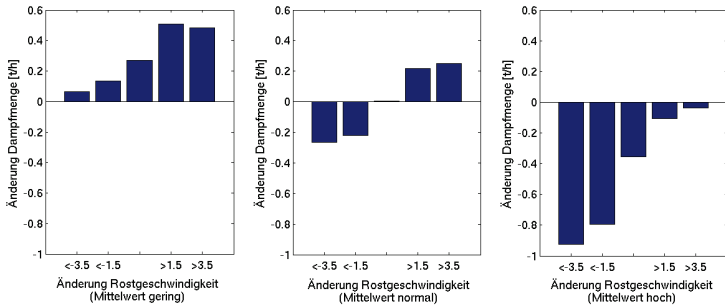


Abbildung 2.8. – Nichtmonotoner Prozesszusammenhang zwischen Rostgeschwindigkeit und Dampfleistung in MVA 1. Die Diagramme entstanden durch Auswertung der Betriebsdaten über einem Zeitraum von zwei Wochen. Im normalen Arbeitsbereich (Mitte) führt eine Erhöhung der Rostgeschwindigkeit zu einer Erhöhung der Dampfleistung. Eine Verringerung der Rostgeschwindigkeit führt zu einer verringerten Dampfleistung. Wird aber über einen Zeitraum von über 15 Minuten im Mittel eine hohe Rostgeschwindigkeit (rechts) gefahren, kommt es in der Folge zu einer Verringerung der Dampfleistung, unabhängig von der Rostgeschwindigkeitsänderung. Ebenso kommt es immer zu einem Ansteigen der Dampfleistung, falls über 15 Minuten im Mittel zu geringe Rostgeschwindigkeiten gefahren werden (links).

Wenn die Rostgeschwindigkeit sich im normalen Arbeitsbereich zwischen 10%-35% befindet, führt eine Erhöhung der Rostgeschwindigkeit zu einer Erhöhung der Dampfleistung. Ein PID-Regler mit einem positiven K_P kann dazu eingesetzt werden, bei einer zu hohen Dampfleistung die Rostgeschwindigkeit zu reduzieren und die Dampfleistung

wieder auf den Sollwert zu bringen. Ein Problem taucht auf, wenn der Müll schlecht brennt und die Dampfleistung längere Zeit unter den Sollwert sinkt. Die Rostgeschwindigkeit wird durch den Regler nun so lange erhöht, dass der Müll zu schnell auseinander gezogen wird und nicht mehr richtig zündet. In der Folge fällt die Dampfleistung noch mehr. In dieser Situation ist es notwendig, die Rostgeschwindigkeit trotz Abfalls der Dampfleistung wieder zu reduzieren, um das Zünden des Mülls zu gewährleisten. Der PID-Regler hingegen erhöht die Rostgeschwindigkeit aufgrund des positiven K_P weiter und verschlimmert so die Situation. Gibt es keine anderen Regler oder Prozesse, die die Aktion des PID-Reglers in so einem Fall begrenzen oder umkehren, so muss der menschliche Anlagenfahrer manuell eingreifen, da andernfalls ein Ausfall der Müllverbrennungsanlage droht. In Abbildung 2.9 ist ein solcher Fall zu sehen.

These 1 (PID-Regler und Nichtmonotone Prozesse)

Ein einzelner PID-Regler kann einen Prozess, dessen Charakteristik nichtmonoton ist, nicht optimal stellen. Der PID-Regler verstärkt in bestimmten Situationen sogar die Abweichung des Prozesses vom Sollwert.

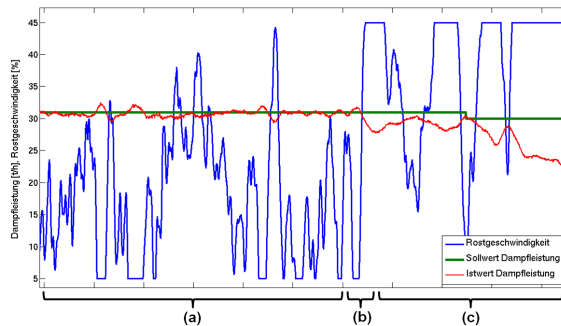


Abbildung 2.9. – Rückgang der Dampfleistung aufgrund falscher Wirkungsweise des Reglers der Rostgeschwindigkeit in MVA 1. Im Bereich (a) befindet sich die Rostgeschwindigkeit im normalen Arbeitsbereich. Im Bereich (b) kommt es aufgrund schlechteren Mülls zu einem Absinken der Dampfleistung. In Bereich (c) wird durch den Regler zu lange eine zu hohe Rostgeschwindigkeit gefahren, was zu einem weiteren Absinken der Dampfleistung führt, bis der automatische Regler durch einen Anlagenfahrer deaktiviert wird.

Nichtmonotone Zusammenhänge dieser Art gibt es nicht nur zwischen der Rostgeschwindigkeit und der Dampfleistung. In Tabelle 2.2 sind auch für andere Stellgrößen im Bereich der Müllverbrennung nichtmonotone Zusammenhänge aufgeführt.

Stellgröße	normaler Zusammenhang	Ausnahme
Zuteilerstößel	höhere Geschwindigkeit führt zu höherer Dampfleistung	wird zu schnell zu viel Müll aufgegeben, kommt es zur Überschüttung; das Feuer wird durch zu viel nachrückenden Müll erstickt
Primärluft	mehr Primärluft führt zu höherer Dampfleistung	zündet der Müll nicht richtig, führt mehr Primärluft zu einem Kaltblasen des Ofens, und die Dampfleistung sinkt
SNCR-Eindüsung	mehr NH_3 führt zu einer Reduktion von NO_x -Emissionen	ist die Temperatur im Dampfkessel zu hoch, reagiert NH_3 ebenfalls zu NO_x ; die Gesamtemissionen steigen also an

Tabelle 2.2. – Im Ergebnis ausführlicher eigener Analysen im Rahmen dieser Arbeit und Gesprächen mit Operatoren vor Ort in MVA1-MVA3 wurden diese nichtmonotonen Zusammenhänge bei Stellgrößen in der Müllverbrennung beobachtet.

Die Behandlung nichtmonotoner Zusammenhänge in konventionellen Prozessregelungen ist schwierig. Eine feste Begrenzung der Regler auf einen bestimmten Arbeitsbereich ist meist nicht sinnvoll, da die verhängnisvollen nichtmonotonen Zusammenhänge oft erst auftreten, wenn die Stellgröße längere Zeit in ungünstigen Bereichen liegt. Meistens wird versucht, durch zusätzliche Regler einen Übergang des Prozesses in einen solchen ungünstigen Bereich zu verhindern. Im oben genannten Beispiel der Regelung der Rostgeschwindigkeit wurde ein solcher zusätzlicher Regler eingebaut. Seine Zielgröße ist eine geringere Abweichung der mittleren Rostgeschwindigkeit von dem normalen Arbeitsbereich von 10%-30%. Befindet sich der Mittelwert der Rostgeschwindigkeit der letzten 15 Minuten oberhalb dieses Bereiches, so reduziert der Regler die Rostgeschwindigkeit.

Ähnliche Mechanismen können sich auch bei anderen Stellgrößen finden. Bei der Regelung der Zuteilergeschwindigkeit findet sich oft ein Regler der den Druck unter den ersten Rosten überwacht (falls eine derartige Messung vorhanden ist). Steigt dieser Druck, weil zu viel Müll auf den Rost befördert wird, so wird die Zuteilergeschwindigkeit verringert.

Diese zusätzlichen Regler und Ausnahmebedingungen machen es zwar möglich, auch nichtmonotone Prozesse zu regeln, haben aber mehrere Nachteile:

1. Die Regelung wird durch zusätzliche Elemente komplexer. Für jeden zusätzlichen Regler müssen ebenfalls Parameter definiert werden.
2. Durch die Kombination mehrerer Regler für eine Stellgröße können ungewollte Seiteneffekte entstehen. Einzelne Regler können sich in ihrer Wirkung aufheben oder in manchen Situationen verstärken.
3. Verändert sich der Prozess mit der Zeit (siehe Abschnitt 2.1.3), müssen die Parameter mehrerer Regler angepasst werden.

Als Alternative zu konventionellen Reglern werden in dieser Arbeit deshalb selbstlernende Systeme untersucht. In Kapitel 3 wird dargestellt, wie selbstlernende Systeme mit nichtmonotonen Zusammenhängen umgehen und weshalb sie konventionellen Reglern überlegen sind.

Mehrfachwirkungen

Ein normaler PID-Regler ist so ausgelegt, dass die Abweichung einer einzelnen Zielgröße von einem Sollwert durch die Veränderung einer einzelnen Stellgröße ausgeglichen wird. In einem komplexen Prozess wie der Abfallverbrennung besteht allerdings das Problem, dass eine Stellgröße häufig Einfluss auf zwei oder mehrere Zielgrößen hat⁹. Dies

⁹In Tabelle 2.1 wurde der Einfluss von Stellgrößen auf Zielgrößen dargestellt.

ist vor allem dann störend, wenn die Stellgröße auf verschiedene Ziele entgegengesetzt wirkt, so dass eine Veränderung der Stellgröße die Abweichung einer Zielgröße reduziert, aber gleichzeitig die Abweichung einer anderen Zielgröße erhöht. Das zeigt sich unter anderem bei der Zuteilersteuerung. Die Stellgröße ist hierbei die Geschwindigkeit, mit der der Müll auf den Rost aufgegeben wird. Je schneller der Zuteiler fährt, desto mehr Müll und damit auch Brennstoff gelangt auf den Rost.

Diese Veränderung der Brennstoffmenge hat Einfluss auf mehrere Zielgrößen:

- Je mehr Brennstoff auf den Rost gelangt, desto größer ist die potentiell erzeugbare Wärme und damit die generierte Dampfleistung.
- Gleichzeitig führt mehr Brennstoff zu höheren Temperaturen und damit vermehrt zur Bildung von NO_x -Emissionen.
- Je mehr Müll sich auf dem Rost befindet, desto schlechter ist die Durchmischung des Mülls. Damit steigt die Wahrscheinlichkeit, dass Teile des Mülls nicht verbrennen, und die Ausbrandqualität sinkt.
- Befindet sich zu wenig Müll auf dem Rost, dann besteht die Gefahr, dass das Feuer abreißt, was zu starker Bildung von CO -Emissionen führt.

Wie zu sehen ist, wirkt die Stellgröße auf verschiedene Ziele in unterschiedlicher Weise. Um eine hohe Dampfleistung zu generieren, müsste der Zuteiler sehr schnell gefahren werden, was aber gleichzeitig die Ausbrandqualität verschlechtert.

Soll eine konventionelle Regelung mit PID-Reglern für die Stellgröße Zuteiler konzipiert werden, muss man sich zwischen drei Varianten entscheiden:

1. Eine oder mehrere Zielgrößen werden nicht mit dieser Stellgröße geführt. Diese Zielgrößen müssen mittels anderer Stellgrößen geregelt werden.
2. Es werden mehrere Regler verwendet, von denen jeder für eine Zielgröße zuständig ist. Die Ausgaben der einzelnen Regler werden anschließend zum tatsächlichen Stellwert verrechnet.
3. Es wird eine kaskadierte Regelung konzipiert. Hierbei wird die Ausgabe eines Reglers verwendet um den Sollwert eines zweiten Reglers zu modifizieren.

In Abbildung 2.10a und 2.10b sind Variante 2 und 3 in einem Beispiel dargestellt. Es soll hierbei eine Kombination der Ziele Dampfleistung und Ausbrandqualität erfolgen. Im ersten Beispiel erhalten beide Regler jeweils einen Sollwert und einen Messwert aus der Prozesskopplung. Die Ausgaben beider Regler werden summiert zum Wert der Stellgröße.

Im zweiten Beispiel wird eine Kaskadenregelung verwendet. Die Ausgabe des Reglers für die Ausbrandqualität wird auf den Sollwert des Reglers für die Dampfmenge addiert. Ist der Ausbrand schlecht, wird so der Sollwert der Dampfproduktion abgesenkt, damit dieser Regler wiederum weniger Müll in den Kessel einbringt.

Jede dieser Varianten bringt eigene Probleme mit sich. In allen Fällen erhöht sich durch die steigende Anzahl der Regler auch die Anzahl der Parameter, die durch einen Experten eingestellt werden müssen.

Werden mehrere Regler durch Summation ihrer Ausgaben verknüpft, ergibt sich ein Problem durch die Verrechnung der Regler zu einem Stellsignal: dieses Stellsignal kann effektiv nur in einem bestimmten

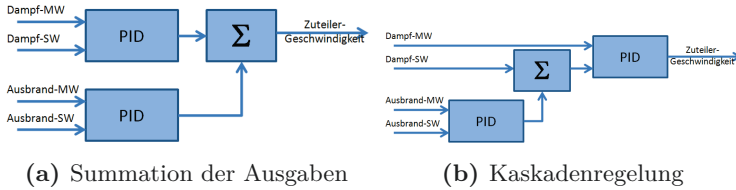


Abbildung 2.10. – Kombination von zwei Reglern.

Bereich arbeiten. Werden mehrere Regler kombiniert, von denen jeder den vollen Stellbereich ausschöpfen darf, so kann durch die Kombination der Ausgaben dieser Stellbereich überschritten werden. Eine Möglichkeit zur Lösung besteht darin, die Stellbereiche der Einzelregler so klein zu setzen, dass eine Unter- oder Überschreitung des Arbeitsbereiches durch Kombination der Ausgaben nicht möglich ist. Das führt allerdings dazu, dass die einzelnen Regler nur eine geringe Wirkung haben. Die zweite Möglichkeit besteht darin, den einzelnen Reglern größere Arbeitsbereiche einzugestehen. Dies kann dazu führen, dass die Summe der Reglerausgaben außerhalb des Arbeitsbereiches der Stellgröße ist und die Summe vorher auf die möglichen Grenzen begrenzt werden muss.

Wird eine Kaskadenregelung angewandt, besteht dieses Problem nicht. Um eine Kaskadenregelung aufzustellen ist allerdings sehr umfassendes Expertenwissen¹⁰ nötig, da abgeschätzt werden muss, wie die einzelnen Ziele in der Kaskade angeordnet werden. Durch die Kombination der Regler kann es außerdem zu unerwünschten Seiteneffekten kommen, insbesondere zu einer gegenseitigen Neutralisierung der einzelnen Regler oder einem Aufschwingen der Regelung.

¹⁰Siehe u.a. [Lutz and Wendt, 2002, Mann et al., 2005] unter den Stichworten “Kaskadenregelung“ und Mehrgrößenregelung“.

Lernfähige Systeme haben hingegen den Vorteil, dass sie auf abstrakteren Zuständen arbeiten, die sich aus mehr als einer Messgröße zusammensetzen können. Die Abhängigkeit einer Größe von mehreren anderen Größen kann somit gelernt werden und muss nicht beim Design der Regelung vorgegeben werden.

Veränderung des Prozesses

Eine wesentliche Besonderheit industrieller Verbrennungsprozesse besteht darin, dass sich der Prozess in seiner Charakteristik und Dynamik ständig verändert. Diese Veränderung ist auf unterschiedliche Brennstoffe zurückzuführen¹¹, auf Abnutzung und Verschleiß von Bauteilen und auf Schlacke, die sich im Laufe der Zeit an den Wänden des Verbrennungsofens absetzt. Diese Schlacke-Anbackungen wachsen und haben im Besonderen Einfluss auf:

- die Strömungsverhältnisse und die Geschwindigkeit der Abgase im Kessel
- die Wärmemenge, die an die Kesselwände abgegeben wird.

Durch diese Veränderungen werden weitere Prozessmerkmale beeinflusst. Wenn weniger Wärme an die Wände des Kessels abgegeben wird, haben die Abgase am Kesselende insgesamt eine höhere Temperatur, und weniger Wärme wird zur Erzeugung von Dampf genutzt. Der Gesamtwirkungsgrad des Kessels sinkt also. Soll weiterhin eine bestimmte Menge an Energie produziert werden, so muss mehr Brennstoff zur Erzeugung aufgewandt werden. In Abbildung 2.11 ist die Steigerung der Abgastemperatur in MVA 1 über einen Zeitraum von 45 Tagen nach einer Revision abgebildet.

Dadurch, dass der Querschnitt des Kessels durch die Verschlackungen kleiner wird, kann (bei konstanter Geschwindigkeit) weniger Rauchgas den Kessel passieren. Da trotzdem eine bestimmte Menge

¹¹Besonders im Bereich der Müllverbrennung unterscheiden sich die Abfälle teilweise erheblich.

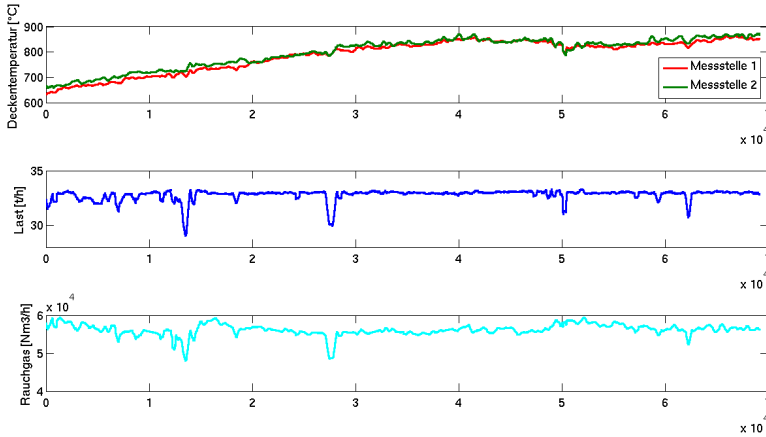


Abbildung 2.11. – Anstieg der Decktemperatur des Kessels in MVA 1 durch zunehmende Verschlackung der Kesselwände. Die Daten erstrecken sich über einen Zeitraum von 45 Tagen nach einer Revision. Es ist deutlich zu sehen, wie bei gleicher Last und fast gleich bleibender Rauchgasmenge die Decktemperatur von anfangs $650\text{ }^{\circ}\text{C}$ auf bis zu $850\text{ }^{\circ}\text{C}$ ansteigt.

an Rauchgas zur Erzeugung der benötigten Wärme durch den Kessel geführt werden muss, muss der dafür nötige Saugzug mit einer höheren Drehzahl arbeiten und das Gas schneller strömen lassen. Da sich mitunter auch an den Blättern des Saugzuges Verschlackungen bilden, wird der Saugzug so mit der Zeit immer mehr belastet.

Der Umstand, dass insgesamt mit zunehmenden Verschlackungen höhere Temperaturen gefahren werden müssen, um eine höhere Leistung zu erzielen, wirkt sich natürlich auch auf die erzeugten Emissionen aus. Falls zum Beispiel eine SNCR-Anlage zur Reduktion von NO_x im Abgasstrom verwendet wird, so verschiebt sich der ideale Punkt, an dem das Reduktionsgemisch in den Abgasstrom eingedüst

werden muss, mit der Zeit immer weiter nach oben. Stehen an diesen Stellen keine Eindüsungsmöglichkeiten zur Verfügung, so wird der NO_x-Gehalt im Rauchgas ansteigen.

These 2 (Veränderung von Verbrennungsprozessen)

Die Charakteristik eines industriellen Verbrennungsprozesses ändert sich mit zunehmender Zeitdauer. Durch Änderungen des Brennstoffs, Verschlackungen und Abnutzungen ändern sich sowohl die Wertebereiche der erhaltenen Messwerte als auch die Zusammenhänge zwischen den einzelnen Messgrößen.

Veränderungen ergeben sich in Verbrennungsanlagen nicht nur durch Verschlackungen an den Wänden. Einen großen Einfluss haben auch die verwendeten Brennstoffe. In Müllverbrennungsanlagen unterscheidet sich der Heizwert und die Körnung des Mülls, je nachdem wie groß die Anteile an Sperrmüll, Hausmüll und Industrieabfällen im angelieferten Müll sind¹². Die in Zementwerken für den Brenner verwendete Kohle ist abhängig vom Abbauort und unterscheidet sich in den Schwefel-, Kohlenstoff- und Asche-Anteilen¹³. Außerdem werden in Zementwerken je nach aktuellem Rezept andere Additive und unterschiedliche Mengen an Sekundärbrennstoffen verwendet [Alsop, 2005].

Alle genannten Faktoren führen dazu, dass es sehr schwer ist, eine langfristig optimale Einstellung der konventionellen Regler zu finden. In Müllverbrennungsanlagen gibt es für Regler, die davon besonders betroffen sind, manchmal mehrere Parametersätze, zwischen denen der Anlagenfahrer per Hand umschaltet. Dies setzt voraus, dass bekannt ist, wie die Zusammensetzung des Mülls ist und welcher Parametersatz für den aktuell verbrannten Müll am besten passt.

¹²Eine aussagekräftige Auflistung der möglichen Bestandteile von Müll und dessen Eigenschaften findet sich in [Thome-Kozmiensky, 1994] Kapitel 1.

¹³Beispiele der Kohlezusammensetzung verschiedener Abbauorte finden sich in [Zelkowsky, 1986, Effenberger, 1999].

2.2. Zementwerke

In diesem Abschnitt wird auf die Unterschiede der Prozessführung in Zementwerken gegenüber Müllverbrennungsanlagen eingegangen. Es sollen dabei nur die wesentlichen Charakteristiken des Prozesses erfasst werden. Eine umfangreichere Darstellung aller Aspekte der Zementherstellung findet sich in [Alsop, 2005, Holcim, 2004].

Wenn im folgenden von “Zement” gesprochen wird, so ist dabei Portland-Zement gemeint. Portland-Zement entsteht durch die Vermahlung von Gips, Anhydrit und Zementklinker. Der Klinker wird aus den Rohstoffen Kalkstein, Sand, Ton und Eisenerz sowie verschiedenen Beigaben durch Brennen in einem Zementofen hergestellt. Die Rohstoffe werden dabei als Rohmehl bezeichnet. Im Zementofen¹⁴ trennt sich der Kalk $CaCO_3$ in Calciumoxid CaO und Kohlendioxid CO_2 auf. Das Calciumoxid reagiert anschließend mit dem im Rohmehl enthaltenen Al_2O_3 , Fe_2O_3 und SiO_3 zu den für Zementklinker wichtigen Verbindungen Di-/Tricalciumsilikat Ca_2SiO_4/Ca_3SiO_5 , Tricalciumaluminat $Ca_3Al_2O_6$ und Tetracalciumaluminatferrit $Ca_4Al_2Fe_2O_{10}$. Außerdem verbleiben Reste von Kalk und Calciumoxid im Klinker, welche dann als “Freikalk” bezeichnet werden.

Der Anteil des Freikalks im Zementklinker wird auch als Freikalk-Wert oder FCaO-Wert bezeichnet und ist neben der Struktur und Mischung der Zementkristalle eines der wesentlichen Gütekriterien für den Zementklinker. Ist der Freikalk-Anteil zu gering, so gilt der Klinker als “überbrannt”, ist der Anteil zu hoch als “zu weich gebrannt”.

¹⁴Genau genommen beginnt die Reaktion bereits im Vorkalzinator (sofern vorhanden) und endet erst im Kühler.

Nachdem im Ofen der Klinker gebrannt wurde, wird er auf einem Kühler¹⁵ auf eine Temperatur von ca. 100°C gekühlt, bevor er anschließend unter Zusatz von Gips, Anhydrit und anderen Zuschlagsstoffen zu Zement gemahlen wird. In Abbildung 2.12 ist dieser Prozess der Zementherstellung schematisch dargestellt.

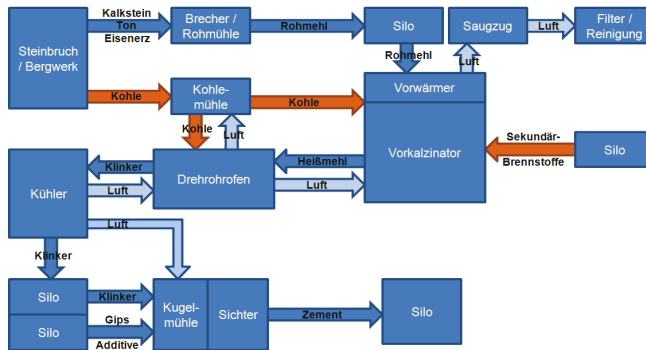


Abbildung 2.12. – Schema des Prozesses der Zementherstellung. Materialströme sind dunkelblau, Luftströme hellblau und Brennstoffströme orange dargestellt. Der dargestellte Aufbau ist charakteristisch für die meisten Zementwerke, wobei einzelne Elemente auch anders angeordnet sein können.

Für jede der einzelnen Stationen der Zementherstellung, von der Rohmehl-Mühle am Beginn bis zur Zement-Mühle am Ende, existieren Prozessleitsysteme und Reglerstrukturen. Im Rahmen dieser Arbeit soll aber vor allem der Ofen im Fokus stehen, in dem der Brennvorgang stattfindet. Dieser Ofen ist im allgemeinen ein Drehrohrofen, wie er in Abbildung 2.13 dargestellt ist.

¹⁵Dies ist meist ein Kühler mit beweglichen Rosten, ähnlich dem bei Müllverbrennungsanlagen beschriebenen Verbrennungsrost.

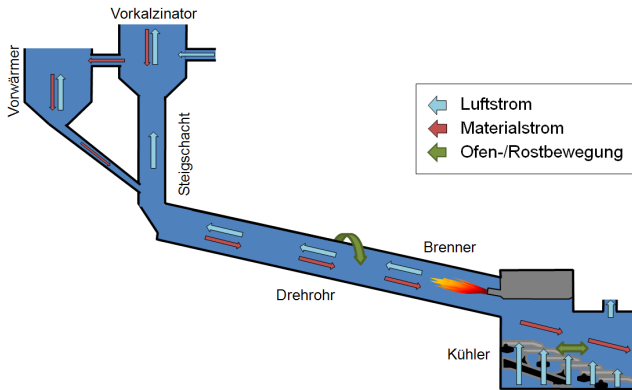


Abbildung 2.13. – Typischer Aufbau eines Drehrohrrohrs. Luft erhitzt sich im Kühler und am Brenner und wird durch das Drehrohr zum Kalzinator und den Vorwärmern gesaugt. Der Materialstrom erfolgt entgegengesetzt zum Luftstrom. Das Rohmehl wird in den Vorwärmern erhitzt, bis schließlich im Vorkalzinator die Aufspaltung des Kalks beginnt. In der Brennzone unterhalb des Kohlebrenners herrschen schließlich Temperaturen von bis zu 1500°C . Im Beispiel ist der Anfang eines Rostkühlers und eine Stufe eines Zyklon-Vorwärmers dargestellt.

Als wichtige Stellgrößen zur Beeinflussung des Prozesses im Ofen stehen meist zur Verfügung:

- die Umdrehungsgeschwindigkeit des Rohres, welche die Geschwindigkeit beeinflusst, mit der das Rohmehl durch das Rohr gleitet
- die Menge an Luft, die am Brenner (Primärluft), im Kühler (Sekundärluft) und am Vorkalzinator (Tertiärluft) eingedüst wird
- die aufgegebene Menge an Rohmehl und Sekundärbrennstoffen
- die Menge und Zusammensetzung an Brennstoffen, die vom Brenner genutzt wird

Die wichtigsten Ziele sind:

- eine hohe Menge an produziertem Klinker mit dem gewünschten Freikalk-Wert
- einen geringen Verbrauch an Brennstoffen und benötigter Energie
- die Vermeidung von großen Ansammlungen von Klinker im Vorwärmer, dem Rohr oder auf dem Kühler
- die Vermeidung von Emissionsüberschreitungen der Gase CO und NO_x und anderer schädlicher Gase

Im Unterschied zu der Regelung von Müllverbrennungsanlagen sind bei Zementwerken vor allem lange Totzeiten und Verzögerungen zwischen der Änderung einer Stellgröße und der Reaktion des Prozesses zu beachten. Vom Eintreten des Rohmehls in den Vorkalzinator bis zum Austritt des Klinkers aus dem Kühler kann je nach Werk bis zu eine Stunde vergehen.

Da der FCaO-Wert ein wichtiges Maß für die Bestimmung der Zementqualität ist, muss er einem Regelungssystem als Zielwert zur Verfügung gestellt werden. FCaO-Werte können allerdings nicht kontinuierlich gemessen werden, sondern müssen im Rahmen einer Laboranalyse bestimmt werden. Dazu wird in definierten Zeitabständen (meist 2-4 Stunden) eine Probe des Zementklinkers aus dem Kühler entnommen und durch ein Labor analysiert. Je nach Analysemethode ist das Ergebnis 0,5 bis 2 Stunden nach Entnahme der Probe bekannt. Es stehen also pro Tag nur 6 bis 12 Freikalkwerte zur Verfügung, die für das Training eines Lernverfahrens genutzt werden können.

Während in der Müllverbrennung meist kontinuierliche Messwerte mit geringen Verzögerungen zur Verfügung stehen, müssen Lernverfahren, die im Bereich der Zementherstellung eingesetzt werden, mit langen Verzögerungen und nur wenigen Beispieldaten auskommen.

2.3. Durchführung von Testreihen in Anlagen

Das Ziel dieser Arbeit ist die Verbesserung des Betriebes unterschiedlicher Müllverbrennungsanlagen und Zementwerke durch den Einsatz selbstlernender Regelkomponenten. Viele der durchgeführten Versuche erfolgen während des normalen Betriebs dieser Anlagen. Das ist das wesentliche Herausstellungsmerkmal dieser Arbeit und der wesentliche Unterschied zu vielen anderen Arbeiten auf dem Gebiet der neuronalen Regelung industrieller Prozesse. Die erzielten Ergebnisse entsprechen realen Verbesserungen. Die damit verbundene direkte Bindung an die Anwendung bringt aber auch Nachteile mit sich:

- Um ein Verfahren auf einer Anlage einzusetzen, muss es in die dortige Prozesssteuerung integriert werden. Es muss nachweislich sicher mit der Prozesskopplung kommunizieren und innerhalb der Bedingungen des Prozessleitsystems arbeiten. Neben der Konzeption des zu testenden Verfahrens müssen auch Nebenbedingungen zur Integration des Verfahrens in das jeweilige Leitsystem beachtet werden.
 - Versuche auf Anlagen unter realen Betriebsbedingungen lassen sich nie mit vollkommen gleichen Bedingungen wiederholen. Besonders bei Müllverbrennungsanlagen ändert sich der verwendete Brennstoff ständig. Um trotzdem zwei Regelungssysteme verlässlich vergleichen zu können müssen beide Systeme abwechselnd regeln und über längere Zeiträume verglichen werden. Das macht den Vergleich von Systemen sehr zeitaufwendig.
 - Bei realen Versuchen sind immer Sicherheits- und Wirtschaftlichkeitskriterien zu beachten. Sollte ein System funktionale Sicherheitskriterien verletzen, muss ein Versuch abgebrochen werden. Selbst wenn ein System nicht sicherheitskritische Bedingungen verletzt sondern nur teilweise schlechter läuft als ein anderes System, kann ein Versuchsprogramm über lange Zeiträume zu Einbußen führen.
-

Aufgrund dieser Besonderheiten ist es nicht möglich, sämtliche in Frage kommenden Systeme für die Regelung eines Verbrennungssystems im aktiven Betrieb einer realen Anlage zu testen. Es wurde deshalb in dieser Arbeit folgendes Vorgehen gewählt:

1. Es wird untersucht, welche Verfahren prinzipiell in Frage kommen um eine Verbesserung der Regelung des Verbrennungsprozesses zu erzielen.
2. Aus den verschiedenen Varianten eines Verfahrens werden die erfolgversprechenden herausgesucht. Diese Varianten werden an Simulatoren und mit realen Daten von Anlagen getestet, ohne dass das System direkt in der Anlage im Einsatz ist.
3. Das Verfahren, das auf diesen Offline-Tests am besten abschneidet, wird übertragen in die Software, die auf Anlagen eingesetzt wird, und ein erster Test auf der Anlage wird vorbereitet.
4. Wenn dieser Test erfolgreich verläuft und das System eine ähnliche oder bessere Leistung erzielt, wird mit dem Anlagenbetreiber ein längerer Vergleichstest vereinbart. Sollte auch dieser Test erfolgreich sein, wird das neue System dauerhaft in der Anlage eingesetzt.

In Kapitel 7 werden die Ergebnisse von fünf Versuchen beschrieben, die in dieser vierten Phase auf verschiedenen Anlagen erzielt wurden. Die dort beschriebenen Systeme, die im Rahmen dieser Arbeit entwickelt wurden, sind nach diesen erfolgreichen Tests im dauerhaften Einsatz auf den jeweiligen Anlagen und werden künftig auch auf weiteren Anlagen als Standardsystem eingesetzt.

2.4. Fazit

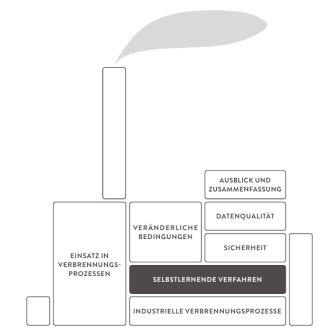
In diesem Kapitel wurde ein industrieller Verbrennungsprozess am Beispiel einer Müllverbrennungsanlage erläutert. Als wesentliche Ziele der Regelung des Verbrennungsprozesses gelten das vollständige Ausbrennen des Mülls, die Erzeugung von Dampf oder Wärme und die Einhaltung von Emissionsgrenzwerten. Es wurden die wesentlichen Stellgrößen benannt, mit denen Einfluss auf die Erfüllung dieser Ziele genommen werden kann.

Es wurde weiterhin dargestellt, wie eine konventionelle Regelung aufgebaut ist, die diese Stellgrößen ansteuert. Typische Elemente der Regelung wie PID-Regler und Kennlinien wurden vorgestellt, und anhand verschiedener Beispiele konnte gezeigt werden, dass eine konventionelle Regelung einige Prozesssituationen nicht optimal regeln kann. Als wesentliche Ursachen für ein Versagen des konventionellen Systems wurden nichtmonotone Prozesszusammenhänge, Mehrfachwirkungen und Veränderung des Verbrennungsprozesses aufgeführt.

Als weiteres Beispiel für Verbrennungsprozesse wurde der Aufbau eines Zementwerkes erklärt. Als wesentlicher Unterschied zur Müllverbrennungsanlage bei der Charakteristik der vorkommenden Messdaten wurden die langen Totzeiten und Verzögerungen und die nur selten verfügbaren Laborwerte des Freikalkwertes genannt.

Nachdem in diesem Kapitel die Eigenschaften der Verbrennungsprozesse und die Probleme der konventionellen Regler erläutert wurden, werden im nächsten Kapitel lernfähige und adaptive Regelungsverfahren vorgestellt, die helfen sollen die Nachteile der konventionellen Regelung auszugleichen.

Selbstlernende 3. Verfahren zur Regelung von Prozessen



In Kapitel 2 wurde am Beispiel einer Müllverbrennungsanlage erläutert, wie ein industrieller Verbrennungsprozess abläuft, und wie eine konventionelle Regelung aufgebaut ist, die einen solchen Prozess regeln kann. Es wurde ebenfalls erläutert, warum eine konventionelle Regelung Teile des Prozesses nicht optimal regeln kann, und welche Probleme sich deshalb bei der Konzeption einer solchen Regelung ergeben.

In diesem Kapitel werden die Grundlagen selbstlernender und adaptiver Verfahren vorgestellt, mit denen einige der Nachteile konventioneller Regelungen überwunden werden können. Zunächst wird in Abschnitt 3.1 ein Überblick gegeben, was unter dem Begriff “selbstlernend“ zu verstehen ist. Es wird erläutert, wie sich lernfähige Verfahren prinzipiell in die Regelung eines Prozesses integrieren lassen (Abschnitt 3.2) und was die Vorteile und Nachteile einzelner Ansätze sind.

Wurde entschieden, an welcher Stelle ein lernendes/adaptives Verfahren eingesetzt wird, so muss anschließend ein Lernalgorithmus ausgewählt werden. In Abschnitt 3.4 werden dazu die in dieser Arbeit angewandten Verfahren vorgestellt.

Wenn eine selbstlernende Regelung für einen Verbrennungsprozess aufgestellt werden soll, ergeben sich allerdings auch einige Probleme beim Training und der Anwendung der selbstlernenden Modelle. Die Ursachen dieser Probleme, die dazu führen können, dass sich einige bekannte Verfahren nicht anwenden lassen, werden in Abschnitt 3.5 erläutert.

Für jedes der dort vorgestellten Probleme wurden im Rahmen dieser Arbeit Maßnahmen entwickelt, mit denen sich negative Auswirkungen auf die Regelung eines Prozesses vermeiden oder zumindest abschwächen lassen. Die dazu konzipierten Verfahren stellen den Hauptteil dieser Arbeit dar und werden in den nachfolgenden Kapiteln 4-6 im einzelnen erläutert.

Abschnitt 3.6 beendet dieses Kapitel und gibt nochmals eine Zusammenfassung der Vor- und Nachteile der Anwendung Neuronaler Netze auf die Regelung von Verbrennungsprozessen.

3.1. Einordnung lernfähiger und adaptiver Verfahren

Die Begriffe “Lernen” und “Adaption” werden in vielen unterschiedlichen Bereichen verwendet. Selbst innerhalb der Informatik existiert eine so große Menge an “lernfähigen Systemen”, dass sie innerhalb dieser Arbeit nicht aufgeführt werden können.

In [Weller, 1985] wird Lernen beschrieben als “Aneignung bzw. Vervollständigung der Fähigkeit, aufgrund von Beobachtungen der Umgebung neue Erfahrungen anzueignen, um ... ein Ziel wiederholt optimal zu erreichen.”¹. [Zell, 1997] beschreibt Lernfähigkeit als Eigenschaft eines Systems aufgrund von Beispielen eine Aufgabe zu erfüllen, ohne dass die Lösung selbst programmiert werden muss. In [MacKay, 2003, Knauf, 1993] wird als Teil des Lernens die Inferenz oder induktive Inferenz beschrieben, die Fähigkeit aus Beispielen einer Wissensbasis Regeln oder Folgerungen abzuleiten.

Gemein ist allen genannten Quellen, dass durch Lernen aus Beispieldaten Wissen oder Funktionalität abgeleitet wird. Für den Zweck dieser Arbeit soll deshalb folgende Definition eines selbstlernenden Systems gelten:

Definition 3.1.1 (*Selbstlernendes System*)

Ein selbstlernendes System ist ein System, welches die Fähigkeit hat, aufgrund von Datenbeispielen selbstständig einen Zusammenhang oder eine Funktion zu erkennen und damit eine Aufgabe zu erfüllen.

¹ [Weller, 1985], S. 19

Für die Anwendung im Bereich der Feuerleistungsführung bestehen die Datenbeispiele aus den Werten, die durch die Messstellen der Anlage geliefert werden. Die Aufgabe des lernenden Systems besteht dann darin, aus diesen Werten auf den Zustand der Verbrennung zu schließen und ggf. eine Handlung abzuleiten, mit der dieser Zustand gemäß eines Zieles verbessert werden kann.

Ähnlich wie der Begriff des Lernens wird auch der Begriff der Adaption sehr vielfältig gebraucht. In [Si et al., 2004] wird Adaptivität im Zusammenhang mit Adaptive Dynamic Programming (ADP) definiert als die Fähigkeit eines Verfahrens bestimmte Variablen oder Parameter zu einem Optimum zu konvergieren. Eine solche Definition findet sich auch in [Weller, 1985]. In dieser Arbeit wird die folgende Definition für ein Adaptives System verwendet:

Definition 3.1.2 (*Adaptives System*)

Ein adaptives System hat die Fähigkeit seine interne Funktionalität anhand von Datenbeispielen selbstständig anzupassen.

Die Unterscheidung zwischen lernendem und adaptivem System geschieht in dieser Arbeit ganz bewusst und soll mit zwei Beispielen erläutert werden:

1. Ein menschlicher Experte definiert einen PID-Regler und verwendet anschließend einen Optimierungsalgorithmus, der in bestimmten Zeitabständen die Parameter des Reglers anpasst. Dieses System ist adaptiv, aber nicht selbstlernend. Es werden die Parameter des Reglers angepasst, die Reglerfunktion selbst wurde aber von einem menschlichen Experten definiert.
 2. Ein neuronales Modell wird auf Daten eines Prozesses trainiert. Anschließend wird durch Nutzung des Modells eine Stellgröße geregelt. Wird dieses Modell nicht mehr verändert, so ist das System selbstlernend, aber nicht adaptiv.
-

Ein System kann gleichzeitig adaptiv und lernfähig sein, falls es einen Zusammenhang selbständig ableiten kann und sich immer wieder an Veränderungen dieses Zusammenhanges anpasst.

Laut [Jain and De Silva, 1999], in dem eine ähnliche Einteilung in lernende und adaptive Verfahren getroffen wird, genügt für das Problem der veränderlichen Prozessbedingungen ein adaptives System. Liegt die Ursache des Fehlverhaltens konventioneller Regler hingegen in Nichtmonotonie und Nichtlinearität und sind die genauen Zusammenhänge unbekannt, so muss ein lernfähiges System verwendet werden.

3.2. Anwendungsgebiete in der Prozessführung

Soll eine konventionelle Regelung im Bereich der Feuerleistungsführung durch lernfähige oder adaptive Elemente ergänzt oder ersetzt werden, so stellt sich als erstes die Frage, ob durch das lernfähige System auch die grundlegende Struktur der Feuerleistungsregelung geändert werden soll. Prinzipiell wäre es möglich den bisherigen Aufbau der Reglerstruktur durch ein einziges monolithisches System zu ersetzen. Es gibt allerdings mehrere Gründe, die gegen dieses Vorgehen sprechen:

- Die Menge an Messgrößen und Stellgrößen in einer realen Anlage ist viel zu hoch. Je nach Anlage ist mit 5-15 Stellgrößen und 10-40 bedeutenden Messgrößen zu rechnen. Kaum ein datengetriebenes Lernverfahren kann in einem solch hochdimensionalen Raum zuverlässig arbeiten.
 - Die Kontrolle eines solchen monolithischen Systems auf Schwachstellen und sicherheitskritische Aspekte wäre kaum möglich.
 - Bestehendes Expertenwissen über die Regelung des Prozesses, die in der alten Reglerstruktur steckt, würde nicht genutzt.
 - Die Akzeptanz eines solchen Systems beim Anlagenbetreiber und den Anlagenfahrern ist nicht gegeben.
-

Aus diesen Gründen ist es erfolgversprechender, die grundlegende Reglerstruktur beizubehalten und nur einzelne Reglerelemente (PID-Regler und Kennlinien) durch lernfähige Elemente zu ersetzen oder zu ergänzen.

These 3 (*Weiterverwendung der alten Reglerstruktur*)

Die komplette Ersetzung eines vorhandenen Feuerleistungssystems durch ein monolithisches lernfähiges System ist nicht praktisch durchführbar, da die vorhandenen Risiken für den Anlagenbetrieb zu groß sind. Stattdessen sollen vorhandene Reglerstrukturen um lernfähige oder adaptive Komponenten erweitert werden.

Vor diesem Hintergrund soll im Folgenden dargestellt werden, welche adaptiven und/oder lernfähigen Verfahren sich mit einem bestehenden Reglerelement kombinieren lassen. Unterschieden werden die Verfahren in dieser Arbeit nach der Art der Veränderung, die im Zuge der Lernens bzw. der Adaption am Regler vorgenommen wird. Dabei treten vor allem drei grundsätzliche Ansätze auf:

1. **Änderung der Parameter des Reglers:** Die grundsätzliche Wirkungsweise des Reglers bleibt hierbei unverändert. Es werden nur (meistens innerhalb definierter Grenzen) bestimmte Parameter des Reglers verändert, die bestimmen, wie stark oder wie schnell der Regler auf eine Veränderung eines Zustandes reagiert. Am Beispiel eines PID-Reglers entspricht dies der Veränderung der Parameter K_P , T_N oder T_V , während die grundsätzliche Arbeitsweise des Reglers durch die Formel

$$u(t) = K_P \left[e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau + T_V \frac{d}{dt} e(t) \right]$$

nicht verändert wird. Dieses Verfahren ist in der Regel adaptiv, aber nicht selbstlernend.

2. **Änderung der Eingabe des Reglers:** Die Parameter und die Arbeitsweise des Reglers bleiben unverändert. Stattdessen wird die Eingabe des Reglers modifiziert. Dies ist beispielsweise der

Fall, wenn der Regler nicht ein Messsignal als Eingabe erhält, sondern eine errechnete Prognose. Wird die Berechnungsvorschrift dieser Prognose geändert, so ändert sich auch das Regelverhalten, obwohl der eigentliche Regler unverändert ist. Das Prognosemodell selbst kann adaptiv und/oder selbstlernend sein, die Wirkung auf die Stellstrategie ist durch den konstant bleibenden Regler nur adaptiv.

3. **Änderung der Reglerfunktion:** In diesem Fall wird die grundsätzliche Wirkungsweise des Reglers verändert. Diese Änderungen gehen über die in Punkt 1. genannten Parameteränderungen hinaus. Im Falle eines PID-Reglers ist dies zum Beispiel die Änderung der PID-Formel, indem statt des normalen Fehlers der quadratische Fehler genutzt wird oder die Art der Integration des Fehlersignals geändert wird. Die Modelländerung kann adaptiv und/oder selbstlernend sein.

In Abbildung 3.1 werden die unterschiedlichen Ansatzpunkte zur Anpassung eines Reglers² noch einmal dargestellt. In den folgenden Abschnitten werden Vor- und Nachteile jedes dieser Ansätze sowie unterschiedliche Verfahren zu ihrer Umsetzung erläutert.

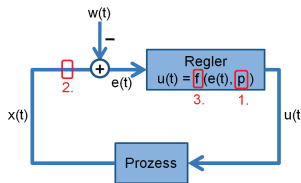


Abbildung 3.1. – Ansatzpunkte zur Anpassung eines Reglers mit lernfähigen oder adaptiven Komponenten. 1.: Änderung der Reglerparameter, 2.: Änderung der Eingangssignale, 3.: Änderung der Reglerfunktion

²Im Beispiel verwendet der Regler einen Prozesswert $x(t)$ und einen Sollwert $s(t)$ zur Berechnung einer Regeldifferenz $e(t)$, auf die der Regler mit der Stellgröße $u(t)$ reagiert.

3.3. Adaptive Verfahren

3.3.1. Anpassung der Reglerparameter

Die erste Möglichkeit zur Anpassung eines Reglers an einen sich ändernden Prozess besteht in der Änderung der Prozessparameter. Falls es sich um einen PID-Regler handelt, ist dies beispielsweise die Anpassung der Parameter K_P , T_N und T_V . Verfahren, die die Parameter eines konventionellen Reglers anpassen, lassen sich laut [Jain and De Silva, 1999] in drei Klassen einteilen:

1. Gain Scheduling erfordert, dass die optimalen Reglerparameter von einem beobachtbaren Arbeitspunkt abhängen und für jeden Arbeitspunkt bekannt sind. Die Umschaltung der Parameter erfolgt dann je nach dem Wert des Arbeitspunktes.
2. Model-Referenced Adaptive Control (MRAC) verwendet ein Referenzmodell des geregelten Prozesses. Die Abweichung zwischen Modellwert und realem Prozess wird genutzt um die Parameter des Reglers anzupassen.
3. Self-Tuning Regulator (STR) schätzen aus Reglerausgabe und Systemantwort die Systemdynamik und ändern basierend auf dieser Schätzung die Reglerparameter.

Gemeinsam ist allen Verfahren, dass Wissen über die grundsätzliche Systemdynamik erforderlich ist, um entweder das benötigte Referenzmodell oder das korrekte Schätzverfahren anzuwenden.

Ein bekanntes Verfahren zur Bestimmung der Reglerparameter, das der STR-Kategorie zugeordnet werden kann, ist beispielsweise das Ziegler-Nichols-Verfahren [Ziegler and Nichols, 1942]. Zur Anwendung des Verfahrens ist es erforderlich, die Sprungantwort der Zielgröße auf die Stellgröße zu ermitteln. Dazu wird ein Sprung, also eine möglichst große Änderung der Stellgröße, durchgeführt und anschließend die Reaktion der Zielgröße aufgezeichnet (siehe Abbildung 3.2).

Aus den dabei ermittelten Werten für Verzugszeit T_U , Ausgleichzeit T_G und Verstärkung K_S lassen sich die Parameter eines für diesen Prozess geeigneten PID-Reglers wie folgt bestimmen:

$$K_P = 1,2 \frac{T_G}{K_S T_U} \quad T_N = 2T_U \quad T_V = \frac{1}{2} T_U \quad (3.1)$$

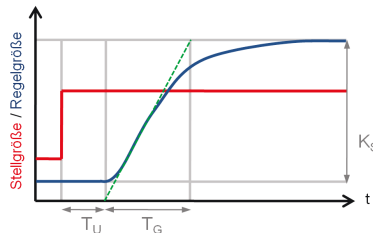


Abbildung 3.2. – Idealisierte Antwort einer Regelgröße (blau) auf den Sprung einer Stellgröße (rot). Die Wendetangente (grün) wird genutzt um die Verzugszeit/Totzeit T_U und die Ausgleichzeit/Verzögerung T_G zu bestimmen.

Ist es nicht möglich, die Parameter nach den oben genannten Methoden zu bestimmen, da das erforderliche Modellwissen nicht vorliegt und die Versuche zur Bestimmung der Modelle zu aufwendig oder gefährlich sind, können die Parameter des Reglers auch durch Optimierungsverfahren der Hillclimbing Kategorie verändert werden. Voraussetzung für ein solches Optimierungsverfahren ist dabei meist, dass eine Qualitätsmessung verfügbar ist, mit deren Hilfe sich die Güte der aktuellen Parameterkonstellation bewerten lässt. Ein einfaches Beispiel für einen Hillclimbing-Algorithmus unter Nutzung einer Gütefunktion ist dabei der Simplex-Algorithmus nach Nelder und Mead [Nelder and Mead, 1965].

Voraussetzung für die Anwendung dieses Algorithmus ist nur die Beobachtbarkeit der aktuellen Parameter und einer dazugehörigen Qualitätsmessung. In Abbildung 3.3 ist die Funktionsweise des Algorithmus an einem einfachen Beispiel mit zwei zu optimierenden Parametern dargestellt.

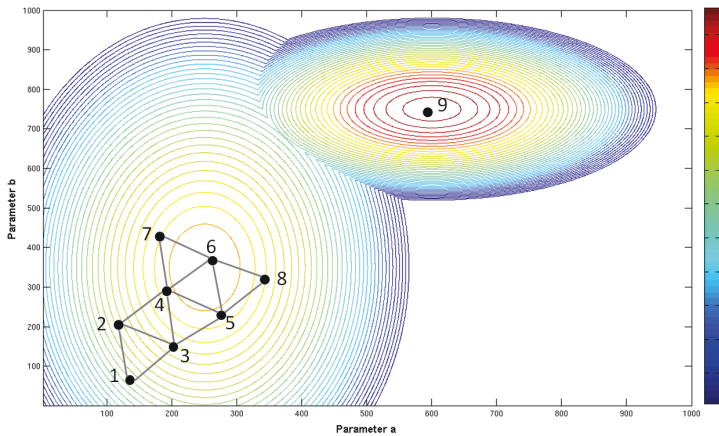


Abbildung 3.3. – Funktionsweise des Simplex-Algorithmus (ohne Kontraktion und Komprimierung) bei zwei zu optimierenden Parametern. Die Prozessgüte bei jeder Parameterkombination ist farblich dargestellt, wobei Rot einer hohen und Blau einer geringen Güte entspricht. Der Startpunkt des Algorithmus ist an Punkt 1, der Endpunkt an Punkt 6. Die optimale Kombination an Punkt 9 wird durch den Algorithmus nicht erreicht.

Optimierungsverfahren nach dem vorgestellten Prinzip haben den Vorteil, dass Änderungen der Parameter nur in kleinen Schritten erfolgen. Für sicherheitskritische Anwendungen ist dies vorteilhaft, da eine kleine Parameteränderung in eine falsche Richtung das Reglerverhalten meist nicht so stark beeinflusst, dass der Prozess dadurch instabil werden würde.

Gleichzeitig dauert es allerdings lange, bis der Regler an eine große Prozessänderung angepasst wird, und es besteht die Gefahr nur ein lokales Optimum zu finden. Bei Prozessen wie der Müllverbrennung, bei denen eine Schwankung des Prozesses aufgrund von inhomogenen Brennstoffen auftritt, muss die Qualitätsmessung für die Beurteilung der Parameter außerdem über einen langen Zeitraum erfolgen. Es kann dadurch vorkommen, dass der Optimierungsalgorithmus zu langsam ist, um großen Änderungen des Prozesses zu folgen. Besonders groß ist diese Gefahr, falls durch eine Anlagenrevision eine wesentliche Änderung des Prozesses auftritt.

Falls es viele Regler gibt, die alle mit der gleichen Quality-Funktion optimiert werden, so steigt weiterhin die Wahrscheinlichkeit, dass ein einzelner Regler nicht optimiert werden kann, da nicht klar ist, welcher der einzelnen Regler für eine Verschlechterung/Verbesserung der Quality verantwortlich ist. Dieser Umstand wird als strukturelles *credit assignment problem* [Kaelbling et al., 1996, Kaiser et al., 1996] bezeichnet.

In dieser Arbeit werden Ansätze zur Anpassung der Reglerparameter aus diesen Gründen nicht weiter verfolgt.

3.3.2. Anpassung der Reglereingabe

Eine zweite Möglichkeit, einen adaptiven Regler zu konstruieren, besteht in der Modifikation der Reglereingabe. Dem Regler wird dabei nicht die eigentliche Regelgröße $x(t)$ übergeben, sondern eine vorverarbeitete Regelgröße $x'(t)$. Im Sinne eines adaptiven Verfahrens ist die nützlichste Methode die Verwendung eines Prognosemodells für die Regelgröße x wie in Abbildung 3.4 zu sehen. Mithilfe eines solches Prognosemodells soll der zukünftige Wert $x(t+k)$ der Regelgröße $x(t)$ in k Zeitschritten geschätzt werden. Dem Regler wird dann nicht der eigentliche Messwert $x(t)$ sondern die Prognose $x(t+k)$ präsentiert. Ist die Prognose korrekt, so kann der Regler durch den zeitlichen Vorlauf schneller auf eine Abweichung der Regelgröße reagieren, und die entstehende Regelabweichung wird geringer. Diese Kombination eines Reglers und eines Modells wird als Model Predictive Control (MPC) bezeichnet [Dittmar and Pfeiffer, 2004].

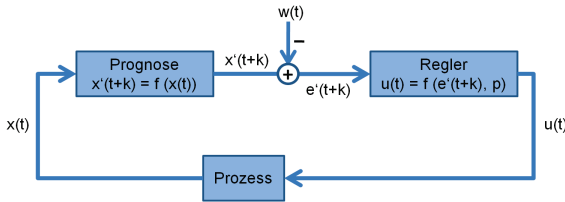


Abbildung 3.4. – Modifizierung einer Reglereingabe $x(t)$ durch ein vorgeschaltetes Prognosemodell. Die Ausgabe des Modells $x'(t)$ und der Sollwert $w(t)$ ergeben dann die Regeldifferenz $e(t)$, die vom Regler genutzt wird um die Stellausgabe $u(t)$ zu berechnen.

Komplexere Varianten der MPC prognostizieren nicht nur einen Wert der Regelgröße sondern den erwarteten zeitlichen Verlauf über mehrere Schritte. Ausgehend von diesem zeitlichen Verlauf kann eine optimale Stellsequenz berechnet werden, die dann ausgeführt wird.

Die Berechnung dieser Stellsequenz ersetzt allerdings den konventionellen Regler, weshalb MPC-Verfahren dieses Typs unter die in Abschnitt 3.4 aufgeführten Verfahren fallen.

Ein Spezialfall des MPC, der im Umfeld der Regelung von Verbrennungsprozessen häufig zum Einsatz kommt, ist die Ersetzung eines Messwertes durch einen Softsensor [Fortuna et al., 2007]. Dabei wird aus anderen Messwerten $x_2(t)$ des Prozesses ein Ersatzsignal $x'_1(t)$ für die Regelgröße $x_1(t)$ berechnet. Solange die Regelgröße $x_1(t)$ verfügbar ist, arbeitet der Regler mit dieser. Ist die Regelgröße nicht mehr verfügbar, wird stattdessen auf das Ersatzsignal $x'_1(t)$ des Softsensors umgeschaltet. In Abbildung 3.5 ist diese Funktionsweise dargestellt.

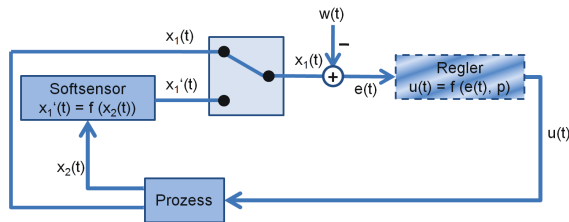


Abbildung 3.5. – Nutzung eines Softsensors mit der Eingabe $x_2(t)$ für eine nicht immer verfügbare Regelgröße $x_1(t)$. Je nach Verfügbarkeit wird zwischen der Ausgabe des Softsensors $x'_1(t)$ und der Regelgröße $x_1(t)$ umgeschaltet. Der Regler ist schraffiert dargestellt, da ein Softsensor prinzipiell auch ohne den Verbund mit einem Regler existieren kann.

Softsensoren kommen vor allem dann zum Einsatz, wenn eine Sensormessung unzuverlässig ist, häufig gewartet werden muss oder mit hohen Kosten verbunden ist. Ein Beispiel dafür sind Messungen der Produktqualität mittels zeitdiskreter Laboranalysen, u.a. Freikalk-Messungen in Zementwerken. Der Freikalk-Messwert wird mittels einer zeitintensiven Laboranalyse bestimmt. Aufgrund der Dauer und der Kosten dieser Analyse wird meist nur alle zwei bis vier Stunden eine Zementprobe aus dem laufenden Betrieb analysiert.

Da die Analyse, je nach angewandtem Verfahren, bis zu einer Stunde dauern kann, vergehen in einem ungünstigen Fall bis zu fünf Stunden, bis eine plötzliche Verschlechterung des Freikalk-Wertes und damit der Zement-Qualität bemerkt wird. Um solche Qualitätseinbußen zu verhindern wird ein Softsensor eingesetzt, der aus anderen, billigeren und kontinuierlichen Messwerten den aktuellen Freikalk-Wert schätzt. Ist diese Schätzung korrekt, kann sie verwendet werden, um schon früh Änderungen der Zementqualität zu erkennen und dementsprechend die Prozessführung zu verändern. Ein Beispiel für einen solchen Softsensor im Bereich der Freikalkschätzung ist in Abbildung 3.6 dargestellt.

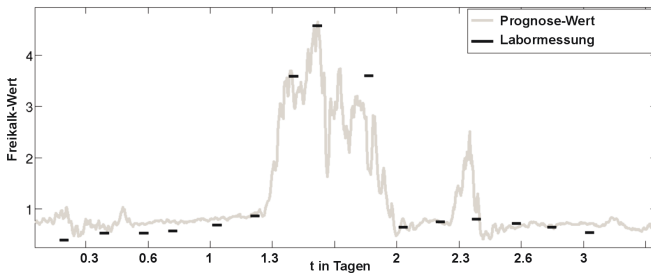


Abbildung 3.6. – Beispiel für Freikalk-Labormessungen und Freikalk-Prognose in einem Zementwerk.

Sowohl ein MPC, welches ein Prognosemodelle verwendet, als auch ein Softsensor mit angebundendem Regler bestehen dabei aus zwei Komponenten. Das sind zum einen der Regler selbst³, der auf Basis der Prognose oder der Softsensor-Ausgabe eine Stellausgabe berechnet, zum anderen das Prognose-/Softsensormodell. Für die technische Realisierung des Modells gibt es viele verschiedene Möglichkeiten.

³Bei Softsensoren im Bereich der Feuerleistungsführung wird manchmal auch der Softsensor allein ohne nachgeschalteten Regler verwendet. Das Softsensor-Signal dient dann als Orientierung oder Warnung für den Anlagenfahrer, der somit als "menschlicher Regler" angesehen werden kann.

Sie kann mithilfe eines linearen Regressionsmodells, mittels verschiedener neuronaler Netze, mit Bayes'schen Modellen oder zahlreichen anderen Möglichkeiten erfolgen. Um dem Anspruch einer adaptiven Regelung zu genügen, ist es nur wichtig, dass sich das Prognosemodell automatisch an aktuelle Daten anpasst. In Abschnitt 3.4 wird erläutert, wie ein solches Modell aufgebaut ist.

Wird ein MPC-Ansatz verwendet, um eine konventionelle Regelung mit adaptiven Komponenten zu erweitern, so kann der bisher benutzte Regler unverändert übernommen werden. Vom Aspekt der Sicherheit betrachtet ist das sehr vorteilhaft. Es ist zwar möglich, dass das vorgeschaltete Prognosemodell eine fehlerhafte Prognose liefert, dies lässt sich durch Vergleich der Modellausgabe mit realen Messwerten aber sowohl offline als auch online relativ einfach feststellen und beheben⁴.

Ebenso wie Verfahren zur Änderung von Reglerparametern kann eine zusätzliche Prognose durch ein Modell nicht die fehlerhafte Reaktion eines konventionellen Reglers auf Nichtlinearität und Nichtmonotonie verhindern. Da besonders die Anwendung als Softsensor im Bereich der Feuerleistungsführung oft benötigt wird, wird diese Art der Anpassung eines Reglers trotzdem in dieser Arbeit intensiv untersucht.

⁴In Abschnitt 6.1 dieser Arbeit wird detailliert auf die unterschiedliche Schwierigkeit der Erkennung einer fehlerhaften Prognose und eines fehlerhaften Reglers eingegangen.

3.4. Lernfähige Verfahren

Im vorherigen Abschnitt wurden Ansätze vorgestellt, die die Parameter eines Reglers graduell verändern oder die Eingabe eines Reglers durch ein Prognosemodell oder einen Softsensor modifizieren. Allen Verfahren ist gemeinsam, dass der ursprüngliche Regler mit seiner charakteristischen Funktionsgleichung und der prinzipiellen Verbindung zwischen Eingabe und Ausgabe erhalten bleibt.

Die größte Veränderung des bestehenden Systems tritt jedoch auf, wenn diese Reglerfunktion geändert wird. Praktisch bedeutet dies ein Austauschen des vorhandenen konventionellen Reglers durch einen anderen Reglertyp.

Dieser Austausch birgt die größten Möglichkeiten, da damit auch die Probleme der Nichtmonotonie, des SISO-Verhaltens und der Nichtlinearität der konventionellen Regler beseitigt werden können. Im Unterschied zu den bisher vorgestellten Verfahren können diese Verfahren auch selbstlernend im Hinblick auf die Stellstrategie sein, das heißt sie können nicht nur bestehende Funktionalität anpassen, sondern eigenständig eine Funktionalität entwickeln.

Andererseits birgt diese Änderung auch die größten Risiken, da eine veränderte Reglerfunktion zu unerwartetem oder unerwünschtem Verhalten führen kann.

Konventionelle Regler wie ein PID enthalten meistens kein eigenes Prozessmodell. Im vorherigen Abschnitt wurden Möglichkeiten genannt, wie durch ein zusätzliches adaptives Prozessmodell ein MPC-System entsteht. Viele Ansätze, die sich für einen selbstlernenden Regler benutzen lassen, sind Erweiterungen dieser Prognosemodelle des MPC. Um ein solches Modell als Regler zu nutzen, ist es notwendig, dass eine der Eingangsgrößen des Modells der Wert der Stellgröße ist. Ist dies der Fall, dann lässt sich für einen Systemzustand voraussagen, wie die Zielgröße sich bei unterschiedlichen Stellgrößenwerten verändern wird. Es wird anschließend der Stellwert an das Leitsystem gegeben, bei dem sich die Zielgröße dem Sollwert am meisten nähert.

Ebenso wie für die Erstellung eines MPC-Prognosemodells existieren auch für die Erstellung eines Reglermodells viele unterschiedliche Methoden. Im Folgenden wird ein Überblick über Methoden gegeben, mit denen ein solches Modell erstellt werden kann.

Im letzten Abschnitt wurden mit MPC und Softsensor zwei Ansätze präsentiert, die ein Modell des Prozesses benötigen, um den Verlauf einer Zielgröße vorherzusagen. Um ein solches Modell zu entwickeln sind verschiedene Methoden anwendbar. Zum einen kann ein physikalisches oder strömungstechnisches Modell des Prozesses aufgrund von Expertenwissen aufgestellt werden. Ein solches Modell wird auch *white-box model* [Imboden and Koch, 2003] oder *first-principle model* genannt [Rodrigueza and Pereza, 2005]. Für Verbrennungsprozesse allgemein finden sich ausführliche Beispiele für solche Modelle in [Griebe, 2007, Epple, 2009] und für den Prozess der Müllverbrennung in [Wolf, 2005, Görner et al., 1999]. Diese Modelle haben allerdings den Nachteil, dass sie sehr präzises Wissen über die Anlage und den verwendeten Brennstoff benötigen, sehr rechenaufwendig sind und meist nicht adaptiv auf Veränderungen der Anlage reagieren. In dieser Arbeit wird dieser physikalische Ansatz deshalb nicht verfolgt.

Stattdessen sollen Modelle verwendet werden, die selbständig ein Abbild des Prozesses aufgrund der vorhandenen Daten in Form von Messwerten lernen. Ein solches Modell wird auch *black-box model* genannt, da die genauen Zusammenhänge zwischen Eingabe und Ausgabe innerhalb des Verfahrens gelernt werden und nicht unbedingt nach aussen sichtbar sind⁵. Lernfähige Verfahren werden dabei meist in drei Klassen eingeteilt [Bishop, 2006, Sutton and Barto, 1998]:

⁵Im Unterschied dazu sind beim *white-box model* alle Zusammenhänge per Definition bekannt, da sie von einem Experten definiert werden.

1. **Supervised Learning:** Überwachtes Lernen. Dem Lernverfahren wird zu jedem Datenbeispiel auch der Ausgabewert oder die zugehörige Klasse präsentiert (im folgenden auch Teacher genannt). Die Aufgabe des Lernverfahrens ist diese Abbildung zu erlernen.
2. **Unsupervised Learning:** Unüberwachtes Lernen. Die Ausgabewerte oder Klassen sind nicht bekannt oder werden dem Lernverfahren nicht präsentiert.
3. **Reinforcement Learning:** Verstärkungslernen. Dem Lernverfahren wird nicht der "korrekte" Ausgabewert präsentiert, sondern eine Bewertung, ob die Ausgabe des Lernverfahrens gut oder schlecht ist.

Für diese Arbeit werden Methoden des Supervised Learning eingesetzt, wenn die Approximation einer Messstelle für ein Prognosemodell oder einen Softsensor erfolgen soll. Der Teacher ist hierbei die tatsächliche Sensor- oder Labormessung. Soll hingegen ein Reglermodell gelernt werden, werden Methoden des Reinforcement Learning angewandt. Der Grund dafür ist, dass eine optimale Regelstrategie nicht bekannt ist, so dass nur das Ergebnis der Regeleingriffe als gut oder schlecht bewertet werden kann.

Sowohl für Supervised als auch für Reinforcement Learning existieren verschiedene Ansätze ein Modell aus Daten zu lernen. Das Modell kann beispielsweise ein künstliches Neuronales Netz, ein Bayes'sches Modell oder ein Regressionsmodell sein. In dieser Arbeit wurden künstliche Neuronale Netze als Basismodell gewählt. Zu Vergleichszwecken werden viele Untersuchungen ebenfalls mittels linearer Regressionsmodelle vorgenommen.

Künstliche Neuronale Netze sind der Oberbegriff für eine große Anzahl an informationsverarbeitenden Systemen. Gemeinsam ist allen neuronalen Netzen, dass sie aus mehreren Elementen bestehen, die durch Verbindungen miteinander kommunizieren [Zell, 1997]. Für jedes der oben genannten Lernparadigmen existieren Neuronale Netze. Bekannte Vertreter für Supervised Learning sind das Multi-Layer Perceptron (MLP) oder Learning Vector Quantization (LVQ), für Reinforcement Learning existiert beispielsweise das Neural Fitted Q-Iteration (NFQ) und für Unsupervised Learning die Self-Organizing Map (SOM). Eine umfassende Darstellung der Arten von künstlichen Neuronalen Netzen und ihrer Trainingsverfahren finden sich in [Bishop, 1995, Zell, 1997].

Im Rahmen dieser Arbeit werden zwei Arten von künstlichen Neuronalen Netzen verwendet. Für die Approximation eines Zielwertes, wie sie für MPC und Softsensoren benötigt wird, wird das MLP genutzt. Für die Regelung einer Stellgröße wird hingegen das NFQ verwendet. Beide Verfahren sowie das zu Vergleichszwecken genutzte Regressionsmodell werden im Folgenden kurz erläutert.

3.4.1. Multi-Layer Perceptron

Das Multi-Layer Perceptron stellt eine Erweiterung des Perceptron-Modells [Rosenblatt, 1958] dar. Ein Perceptron ist ein neuronales Netz, in dem die einzelnen Neuronen in Schichten angeordnet sind. Beim ursprünglichen Perceptron existieren drei Schichten: eine Eingabeschicht, die die Eingaben des Modells repräsentiert. Darauf folgt eine Hidden-schicht, dessen Neuronen jeweils mit allen Eingaben der Eingabeschicht verbunden sind. Die letzte Schicht ist die Ausgabeschicht, welche die Ausgaben des Modells präsentiert und ebenfalls mit allen Neuronen der Hiddenschicht verbunden ist. Bei einem Multi-Layer Perceptron wird dieses Modell erweitert, indem statt einer Hiddenschicht mehrere Hiddenschichten vorkommen können. Das MLP gehört damit zur Klasse der mehrschichtigen, vollvermaschten, feedforward Netze. In Abbildung 3.7 ist ein solches MLP abgebildet.

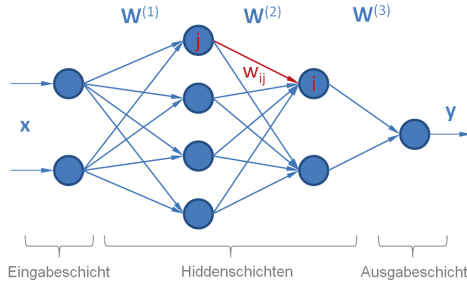


Abbildung 3.7. – Beispiel für den Aufbau eines Multi-Layer Perceptrons mit zwei Hiddenschichten. Die Eingaben \mathbf{x} werden von der Eingangsbeschicht an die Neuronen der ersten Hiddenschicht verteilt. In der Hiddenschicht werden die Signale verarbeitet und weitergeleitet, bis schließlich aus der Ausgabeschicht das Ergebnis \mathbf{y} des MLP für die derzeitige Eingabe ausgegeben wird. Neuronen sind untereinander durch Gewichte verbunden, die bestimmen, wie stark ein Signal weitergegeben wird. Im Beispiel ist ein solches Gewicht w_{ij} von Neuron j zu Neuron i markiert.

Die Informationsverarbeitung geschieht innerhalb eines Perceptrons ausgehend von der Eingangsbeschicht in Richtung der Ausgabeschicht. Die Ausgabe der Neuronen der vorhergehenden Schicht wird multipliziert mit den Gewichten der Verbindungen zur nächsten Schicht und bildet dort die Eingabe für die jeweiligen Neuronen. Diese Eingabe wird mithilfe einer Aktivierungsfunktion in eine Aktivierung umgerechnet, aus der mit einer Ausgabefunktion die Ausgabe des jeweiligen Neurons berechnet wird. Aktivierungs- und Ausgabefunktion unterscheiden sich, je nachdem welcher Typ Neuron verwendet wird. In dieser Arbeit werden Neuronen mit Skalarprodukt-Aktivierung und Sigmoid-Ausgabefunktion verwendet. Die Aktivierung z_i^l und Ausgabe y_i^l des Neurons i der Schicht l berechnet sich dann wie folgt:

$$z_i^{(l)} = \sum_{j=1}^n o_j^{(l-1)} w_{ij} \quad y_i^{(l)} = \frac{1}{1 + e^{-z_i^{(l)}}} \quad (3.2)$$

Wird durch das MLP die Prognose eines Labor- oder Messwertes berechnet, so werden an der Eingabeschicht die benötigten Messwerte wie Sauerstoffgehalt, Feuerraumtemperatur o.ä. angelegt. Die Ausgabe des Netzes repräsentiert dann den approximierten Messwert, beispielsweise die Dampfleistung in einer Minute.

Diese Approximation stimmt natürlich nur, wenn das MLP alle benötigten Informationen als Eingaben erhält und die Gewichte des MLP Werte haben, die diese Eingaben so verrechnen, dass das MLP eine Abbildung der beobachteten Zustände auf die Zielgröße darstellt. Damit diese Gewichte die passenden Werte annehmen, muss das MLP mit Beispieldaten trainiert werden. Zum Training eines MLP stehen verschiedene Verfahren zur Verfügung.

Gemeinsam ist allen Verfahren, dass sie iterativ die Gewichte des Netzwerkes anpassen, um die Ausgabe des Netzwerkes den Teachern der Trainingsbeispiele anzupassen. In dieser Arbeit wird als Standard der Levenberg-Marquardt Algorithmus [Marquardt, 1963, Hagan and Menhaj, 1994] benutzt.

Eine entscheidende Frage beim Training eines MLP ist, nach wie vielen Trainingsschritten der Lernvorgang abgebrochen werden soll. Aufgrund des Bias-Varianz-Dilemmas [Bishop, 2006] kann ein zu kurzes Training ebenso zu einem schlechten Resultat führen wie ein zu langes Training. Im ersten Fall wurde die zu approximierende Funktion noch nicht ausreichend gelernt, im zweiten Fall hat das Netz die Trainingsdaten "auswendig" gelernt und kann nicht auf andere Daten generalisieren. In dieser Arbeit wird der verfügbare Datensatz deshalb stets in einen Trainings-, einen Validierungs- und einen Testdatensatz aufgeteilt. Mit dem Trainingsdatensatz wird das Neuronale Modell trainiert. Der Validierungsdatensatz wird nach jedem Trainingsschritt genutzt um festzustellen, ob das neuronale Modell noch generalisieren kann. Der Testdatensatz wird schließlich zur Bestimmung der Testergebnisse benutzt.

Ein solches MLP kann trainiert werden um als Prognosemodell in einem MPC oder als Softsensor eingesetzt zu werden. Prinzipiell ist es auch möglich, mithilfe eines MLP-Prognosemodells einen Regler zu konstruieren. Wird als einer der Eingaben eine Stellgröße verwendet, so kann eine Prognose des MLPs für einen aktuellen Zustand nicht nur mit dem aktuellen Wert der Stellgröße sondern auch mit anderen möglichen Stellgrößenwerten berechnet werden. Wird anschließend der Stellgrößenwert ausgegeben, bei dem die Zielgrößenprognose am nächsten am gewünschten Sollwert war, so kann das MLP als Regler eingesetzt werden. Eine Erweiterung dieses Vorgehens zur Nutzung eines MLP als Regler stellt das NFQ dar, das im nächsten Abschnitt vorgestellt wird.

3.4.2. Neural Fitted Q-Iteration

Neural Fitted Q-Iteration (NFQ) [Riedmiller, 2005a] ist ein Verfahren, mit dem ein MLP nicht nur zur Prognose eines Wertes sondern auch als Regler genutzt werden kann. Eine einfache Möglichkeit, ein MLP zur Regelung zu benutzen, besteht darin, mehrere Prognosen der Zielgröße mit demselben Systemzustand und verschiedenen Stellgrößenwerten zu erstellen und anschließend den Stellgrößenwert, der die beste Zielgrößenprognose erhielt, zur Regelung zu verwenden.

Das NFQ verwendet ein ähnliches Prinzip, allerdings wird hierbei nicht der Wert der Zielgröße approximiert, sondern ein Q-Wert. Der Q-Wert ist ein Begriff des Reinforcement Learnings [Watkins and Dayan, 1992, Sutton and Barto, 1998]. In der Systematik des Reinforcement Learnings wird der Regler als Agent betrachtet, der das zu regelnde System beobachtet und von dort eine Information über den aktuellen Zustand s_t erhält. Der Agent kann eine Aktion a_t ausführen, um das System zu beeinflussen und in einen neuen Zustand s_{t+1} zu bringen. Der Agent erhält dann vom System eine Belohnung, den Reward r_t , der ausdrückt, ob der neue Zustand s_{t+1} "gut" oder "schlecht" ist. Die Systematik, nach der der Agent die Aktion auswählt, wird als Policy π bezeichnet. In Tabelle 3.1 ist an einem Beispiel dargestellt, wie sich

diese Systematik auf die Feuerleistungsführung übertragen lässt.

Reinforcement Learning	Feuerleistungsregelung	Beispiel
Agent	Regler einer Stellgröße	Regler der Rostgeschwindigkeit
Umwelt	Kessel und Leitsystem	
Policy π	Reglerfunktion und -parameter	$u = 1, 1 \left[e(t) + \frac{1}{200} \int_0^t e(\tau) d\tau + 2, 3 \frac{d}{dt} e(t) \right]$
Zustand s	Aktuelle Messwerte	$\left(\begin{array}{l} O_2 = 7, 5\% \\ T_{Decke} = 781 \text{ }^\circ\text{C} \end{array} \right)$
Aktion a	neuer Wert der Stellgröße	Rostgeschwindigkeit $v_{Rost} = 35\%$
Reward r	negative absolute Regelabweichung der Zielgröße	Abweichung Dampfleistung vom Sollwert $0 - Soll_{Dampf} - Ist_{Dampf} = -1, 2 \frac{t}{h}$

Tabelle 3.1. – Beispiel für die Anwendung von Begriffen des Reinforcement Learning auf den Prozess der Verbrennung in einer Müllverbrennungsanlage.

Der Q-Wert $Q^\pi(s, a)$ gibt an, wie hoch der zukünftige erwartete Gesamtreward für den Zustand s bei Ausführung der Aktion a mit der Policy π ist. Er betrachtet im Zeitpunkt t nicht nur den unmittelbar auf die ausgeführte Aktion folgenden Reward r_t , sondern auch die später zu erwartenden Rewards r_{t+1}, r_{t+2}, \dots . Da die Unsicherheit über den zu erwartenden Reward steigt, je weiter er in der Zukunft liegt, werden die zukünftigen Rewards mit dem Faktor γ diskontiert. Der Q-Wert ist also:

$$Q^\pi(s_t, a_t) = E^\pi(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots) \quad (3.3)$$

Laut Bellman [Bellman, 1957] lässt sich diese Berechnungsvorschrift auch rekursiv ausdrücken, indem sich der Q-Wert einer Zustands-Aktions-Kombination aus dem erhaltenen Reward und dem diskontierten Q-Wert des Folgezustands zusammensetzt:

$$Q^\pi(s_t, a_t) = E^\pi(r_t + \gamma Q^\pi(s_{t+1}, a_{t+1})) \quad (3.4)$$

Am Beispiel der Feuerleistungsführung ist der Reward für eine Aktion beispielsweise die negierte absolute Regelabweichung der Dampfleistung direkt nach Ausführung der Aktion. Je geringer die Abweichung der Dampfleistung vom Sollwert ist, desto höher ist der Reward. Der Q-Wert enthält in diesem Beispiel nicht nur die Abweichung der Dampfleistung nach Ausführung der Aktion, sondern auch (diskontiert) die Abweichung in den folgenden Zeitschritten. Er dient also zur Bewertung der längerfristigen Auswirkungen einer Aktion.

Genau dieser Q-Wert soll durch das NFQ approximiert werden und so zur Entscheidungsgröße für die Auswahl der Aktion und damit für die Veränderung der Stellgröße werden. Zur Approximation des Q-Wertes wird dabei ein MLP verwendet, dessen Ausgabewert der Q-Wert ist und dessen Eingabewerte der aktuelle Prozesszustand s_t sowie der Wert der zu stellenden Stellgröße a_t sind. Der Lernvorgang eines NFQ zur Approximation der Q-Werte läuft in zwei alternierenden Phasen ab:

- **Interaktionsphase:** In dieser Phase werden durch das NFQ Aktionen ausgeführt. Die Auswahl der Aktion geschieht, indem für den aktuellen Prozesszustand s_t eine endliche Zahl unterschiedlicher Aktionen $a_1, a_2, \dots, a_i, \dots, a_n$ als Eingangswerte des NFQ angelegt werden. Für jede der Zustands-Aktions-Kombinationen (s_t, a_i) approximiert das NFQ einen Q-Wert Q_i . Die Aktion, welche den höchsten Q-Wert liefert, wird anschließend ausgeführt. Der Prozess gerät dann in einen neuen Zustand s_{t+1} , und es wird ein Reward r_t berechnet. Das Datentupel $S_t = (s_t, a_t, s_{t+1}, r_t)$ wird gespeichert.
 - **Lernphase:** Das NFQ wird mit den bisher gesammelten Datentupeln S trainiert. Als Trainingswert wird allerdings nicht der Reward r sondern der Q-Wert benötigt. Dieser berechnet sich durch $Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} (Q(s_{t+1}, a_{t+1}))$. Der benötigte Q-Wert des Folgezustandes $Q(s_{t+1}, a_{t+1})$ wird erzeugt, indem die höchste Ausgabe des NFQ für alle möglichen Aktionen a im Zustand s_{t+1} generiert wird.
-

Beide Phasen wechseln sich wiederholt ab. Zum Training des Netzes in der Lernphase wird in dieser Arbeit der Levenberg-Marquardt Algorithmus [Marquardt, 1963] verwendet. Mit steigender Zahl der Iterationen aus Lernphase und Interaktionsphase lernt das NFQ die Approximation der Q-Werte für alle Zustands-Aktions-Kombinationen. Ein Beispiel für die Funktionsweise eines NFQ ist in Abbildung 3.8 dargestellt.

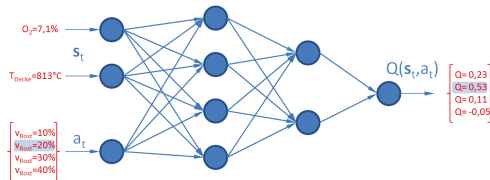


Abbildung 3.8. – Funktionsweise eines NFQ in der Feuerleistungsführung. Als Basis dient ein MLP. Eingangswerte sind zum einen Messungen, die den Zustand des Prozesses beschreiben (im Beispiel O_2 -Wert und Deckentemperatur) sowie eine Stellgröße (im Beispiel die Rostgeschwindigkeit). Für den aktuellen Zustand werden dann die Q-Werte $Q(s_t, a_t)$ für unterschiedliche Aktionen berechnet. Anschließend wird die Aktion ausgeführt, die den höchsten Q-Wert erzielt (im Beispiel $v_{Rost} = 20\%$ mit einem Q-Wert von 0,53).

3.4.3. Lineare Regression

Lineare Regression ist ein Spezialfall der Regressionsanalyse [Backhaus et al., 2006]. Ziel von Regressionsanalysen ist die Bestimmung des Zusammenhangs zwischen einer abhängigen Zielvariable und mehreren unabhängigen Eingabevariablen⁶. Bei der linearen Regression wird angenommen, dass zwischen den Inputwerten $x_1 \dots x_n$ und der Zielvariable y ein linearer Zusammenhang besteht:

$$y = f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (3.5)$$

Die unbekanntenen Parameter $\beta_0 \dots \beta_n$ werden durch ein Schätzverfahren, in dieser Arbeit die Methode der kleinsten Quadrate [Legendre, 1805], aus den vorhandenen Datenbeispielen bestimmt. Zu beachten ist dabei, dass sich das "linear" auf die Parameter $\beta_0 \dots \beta_n$ des Modells bezieht. Lineare Regressionsmodelle können auch einen nichtlinearen Zusammenhang wie $f(x) = x^2$ approximieren, falls x^2 dem Verfahren als Input zur Verfügung gestellt wird.

In dieser Arbeit werden allerdings nur die einfachen Inputwerte benutzt, weshalb das lineare Regressionsverfahren damit deutlich weniger mächtig hinsichtlich seiner potentiellen Approximationsfähigkeit ist als ein neuronales Netz.

Soll ein lineares Regressionsmodell nicht nur zur Approximation einer Messgröße sondern auch als Regler eingesetzt werden, so kann dies genauso geschehen wie beim MLP und NFQ. Einer der Inputwerte $x_1 \dots x_n$ muss die zu regelnde Stellgröße sein. Mithilfe des Regressionsmodells werden dann im aktuellen Zustand für unterschiedliche Werte der Stellgröße die zu erwartenden Werte der Zielgröße oder Q-Werte berechnet und der Stellwert ausgeführt, der am nächsten am gewünschten Zielwert liegt.

⁶Die Trainingsverfahren neuronaler Netze sind ebenfalls Regressionsverfahren, nur dass die Regression auf einem wesentlich komplexeren Modell erfolgt.

3.5. Probleme beim Training selbstlernender Modelle

Im vorherigen Abschnitt wurde gezeigt, welche lernfähigen Verfahren prinzipiell genutzt werden können, um eine Feuerleistungsführung um lernfähige Komponenten zu ergänzen. Sichtet man die Literatur zu den vorgestellten Verfahren, so finden sich auch zahlreiche Anwendungen, bei denen neuronale Netze für die Regelung von Prozessen genutzt werden:

- In [Riedmiller et al., 2007a] wird ein NFQ und in [Pomerleau, 1990] ein MLP benutzt, um innerhalb kurzer Zeit zu lernen, ein Auto in der Fahrbahn-Spur zu halten.
- Neuronale Netze werden in [Martinetz et al., 1995] eingesetzt, um die Walzenkräfte in einer industriellen Walzstraße für Stahl- und Aluminiumprodukte zu optimieren.
- In [Bozoki and Friedman, 1994] wird ein Neuronales Netz benutzt, um die Steuerung von Elektronenbeschleunigern zu optimieren und an veränderliche Eigenschaften anzupassen.

Wenn neuronale Netze und NFQs so erfolgreich bei diesen Vorgängen eingesetzt werden können, warum finden sich dann bisher keine Neuronalen Netze bei der Regelung industrieller Verbrennungsprozesse? In [Jain and De Silva, 1999] wird als Beispiel eine Erhebung unter den Raffinerien in Japan aufgeführt: 91,3% der Stellgrößen werden durch einfache oder erweiterte PID-Regler gesteuert, 8,1% per Hand oder andere Regelverfahren und nur 0,6% durch Regler mit lernfähigen Komponenten.

Im Folgenden werden drei Schwachpunkte aufgeführt, welche sich beim Einsatz neuronaler Modelle in Verbrennungsprozessen als wesentliche Behinderungen herausstellten. Für jeden dieser Schwachpunkte folgt anschließend ein eigenes Kapitel, in dem Maßnahmen zur Behebung dieses Schwachpunktes diskutiert werden.

3.5.1. Verrauschte Messdaten

Ein großes Problem bei der Anwendung selbstlernender Verfahren auf den Daten realer Verbrennungssysteme besteht in dem großen Rauschen, das in den Daten enthalten ist. Der Begriff Rauschen bezeichnet dabei eine Verfälschung des Messwertes, so dass der gemessene Wert nicht den tatsächlichen Zustand des Prozesses abbildet.

In Abbildung 3.9 sind die Approximationen einer einfachen Funktion durch zwei Netzwerke dargestellt. Beide Netze wurden mit demselben Trainingsalgorithmus und derselben Architektur trainiert. Der einzige Unterschied bestand darin, dass im zweiten Fall die Trainingsdaten mit einem Rauschen von 20% der Zielwertvarianz belegt waren, während sie im ersten Fall unverrauscht vorlagen.

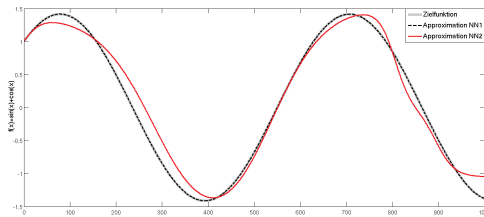


Abbildung 3.9. – Auswirkungen von Rauschen auf den Trainingsdaten auf die Approximation einer einfachen Funktion durch ein Neuronales Netzwerk. Ein Netzwerk, das auf unverrauschten Daten trainiert wurde (NN1, blau gestrichelt), kann die Funktion so gut approximieren, dass die Linien der Ursprungsfunktion und Approximation übereinander liegen. Ein Netz, das auf verrauschten Daten trainiert wurde (NN2, rot), liefert hingegen eine wesentlich schlechtere Approximation.

Führt man diesen einfachen Versuch 100 Mal durch, so stellt sich heraus, dass im unverrauschten Fall 7% der Netze einen Approximationsfehler größer 0,01 haben. Im verrauschten Fall hingegen sind es 74%.

Rauschen kann also die Approximationfähigkeit lernfähiger Verfahren stark verschlechtern. In Kapitel 4 wird gezeigt, wie groß die negativen Auswirkungen sind, die Rauschen auf die Qualität eines selbstlernenden Verfahrens hat, wie sich die Stärke des Rauschens abschätzen lässt und mit welchen Methoden negative Auswirkungen verringert werden können.

3.5.2. Veränderliche Prozessbedingungen

Veränderliche Prozessbedingungen wurden in dieser Arbeit als eine der wesentlichen Motivationen genannt, ein selbstlernendes Verfahren in der Feuerleistungsführung einzusetzen. Aufgrund des Verschleißes von Anlagenbauteilen, der zunehmenden Verschlackung des Kessels und wechselnder Brennstoffe kann ein Regler mit einem festen Parametersatz nicht über den gesamten Zeitraum des Betriebs einer Anlage optimal sein.

Ein lernfähiges Verfahren kann sich prinzipiell an neue Daten und veränderte Systembedingungen anpassen und so langfristig optimal bleiben. Um dies zu gewährleisten, muss das lernfähige Verfahren allerdings in regelmäßigen Abständen auf neuen Datensätzen nachtrainiert werden. In Abbildung 3.10 ist an einem Beispiel dargestellt, dass Neuronale Modelle prinzipiell die Fähigkeit haben sich an veränderliche Bedingungen anzupassen. Gleichzeitig ist aber auch zu sehen, dass durch den Prozess der Anpassung kritische Modellfehler auftreten können.

Soll ein Modell adaptiv an einen Prozess angepasst werden, treten deshalb die folgenden Fragen auf:

- Wie viel bringt die Adaption eines Modells langfristig gegenüber einem unveränderten Modell?
- Sollen alte Datensätze weiter zum Training verwendet werden oder nur neue Daten?

- Soll der Lernprozess des neuen Modells komplett neu gestartet werden, oder kann altes Wissen von den bisherigen Modellen übernommen werden?
- Können alte Modelle weiter genutzt werden?
- Wie kann sichergestellt werden, dass ein neues Modell aufgrund eines Fehlers im Lernvorgang nicht schlechter ist als das bisherige Modell?

Diese Fragen und die damit einhergehenden Varianten zur kontinuierlichen Adaption eines Modells an einen veränderlichen Prozess werden in Kapitel 5 untersucht.

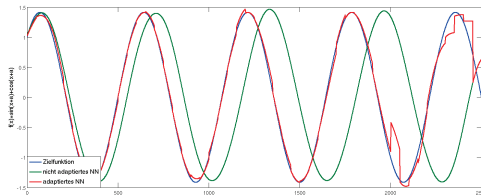


Abbildung 3.10. – Beispiel für die Auswirkung einer kontinuierlichen Veränderung der Basisfunktion. Blau dargestellt ist die Funktion $f(x) = \sin(x + a) + \cos(x + a)$. Der Parameter a steigt mit jedem Zeitschritt ein wenig an. Die Approximation eines neuronalen Modells, welches nur einmal mit den Anfangsdaten trainiert wurde und dann nicht mehr geändert wurde (grün), stimmt am Anfang mit der Zielfunktion überein, entfernt sich dann aber immer mehr von ihr. Wird das Modell hingegen kontinuierlich mit neuen Daten nachtrainiert, kann es der Veränderung der Zielfunktion folgen (rot). Gleichzeitig ist aber auch zu sehen, dass es Abschnitte gibt, in denen die Approximation des roten Modells durch einen Fehler beim Nachtraining stark verschlechtert wird.

3.5.3. Sicherheitskriterien und Daten geregelter Prozesse

Das wichtigste Kriterium beim Einsatz eines selbstlernenden Verfahrens in der Feuerleistungsführung ist die funktionale Sicherheit des neuen Reglers oder Modells. "Funktionale Sicherheit" bedeutet dabei im Kontext dieser Arbeit die Wahrscheinlichkeit, mit der ein Regler keine Folge von Aktionen ausführt, die schädlich für die Anlage oder für die vereinbarten Ziele sind. Ein konventioneller Regler verfügt über (vergleichsweise) wenige Parameter und hat ein relativ einfach verständliches Verhalten. Wie in Abschnitt 2.1.3 gezeigt wurde, kann dieses Verhalten in einigen Situationen sicherheitskritisch sein und ein Eingreifen des Anlagenfahrers erfordern. Da konventionelle Systeme ihre Funktionalität aber nicht automatisch ändern, lässt sich nach einer langen Betriebszeit des Reglers durch einen Anlagenfahrer oft einschätzen, wann eine kritische Reaktion auftreten kann.

Ein selbstlernendes System hingegen ist wesentlich komplexer als ein konventioneller Regler und verändert sich durch automatische Adaptivität. Es fällt deswegen schwerer zu beurteilen in welchen Situationen ein solches System einen ungünstigen Stelleingriff durchführen würde. Nach jedem Nachtraining des Systems muss also erneut eine Überprüfung stattfinden, die sicherstellt, dass die Funktionsweise des lernfähigen Systems den Anforderungen genügt.

Falls es trotz solcher Maßnahmen vorkommt, dass ein Regler mit schädlicher Funktionsweise aktiviert wird, ist es notwendig, diesen Regler zu deaktivieren und stattdessen mit einer Fall-Back-Strategie zu arbeiten.

In Kapitel 6 wird vorgestellt, welche Risiken im Detail beim Training eines neuronalen Reglers bestehen, wie mit Daten eines bereits geregelten Prozesses umgegangen werden kann und welche Vorkehrungen getroffen werden können um das Risiko einer sicherheitskritischen Fehlfunktion zu minimieren.

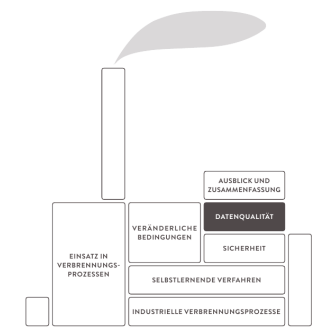
3.6. Fazit

In diesem Kapitel wurden verschiedene Ansätze vorgestellt, mit denen es möglich ist, adaptive oder selbstlernende Verfahren in eine komplexe Feuerleistungsführung zu integrieren. Dazu wurde zunächst definiert, wie in dieser Arbeit die Begriffe “adaptiv” und “selbstlernend” verwendet werden. Selbstlernende Systeme können aus Datenbeispielen einen nicht explizit definierten Zusammenhang lernen. Adaptive Systeme hingegen passen die Parameter eines bereits vorhandenen Systems an.

Anschließend wurde untersucht, wie lernfähige Systeme in Feuerleistungsführungen eingebunden werden können. Die komplette Ersetzung des vorhandenen Systems durch ein monolithisches lernfähiges System wurde als zu riskant eingestuft. Stattdessen sollen die vorhandenen Regler mit lernfähigen oder adaptiven Komponenten ergänzt werden. Dazu wurden vier mögliche Ansatzpunkte ausgemacht, von denen in dieser Arbeit vor allem die beiden Ansätze der Änderung der Reglereingabe und der Änderung der Reglerfunktion verfolgt werden. Bei beiden Ansätzen wird ein Prozessmodell benötigt bzw. gelernt, das Zielgrößen aufgrund von Messgrößen approximiert und die Auswirkungen von Stellgrößenänderungen auf die Zielgröße vorhersagt. Aus der Auswahl möglicher Ansätze um ein solches Prozessmodell zu erzeugen wurden das Multi-Layer-Percepton, das Neural Fitted Q-Iteration und das Lineare Regressionsmodell vorgestellt.

Aus vielen Literaturbeispielen geht hervor, dass diese Ansätze prinzipiell zur Regelung eines Prozesses geeignet sind und mit den Problemen der veränderlichen Prozessbedingungen und nichtlinearer/nichtmonotoner Zusammenhänge umgehen können. Gleichzeitig wurde aber festgestellt, dass lernfähige Systeme bisher kaum in der Feuerleistungsführung industrieller Prozesse eingesetzt werden. Im letzten Abschnitt wurden drei wesentliche Probleme beim Training neuronaler Modelle, die für diesen mangelnden Einsatz sorgen, angesprochen. In den folgenden drei Kapiteln wird jedes dieser Probleme genauer untersucht, und es werden Lösungsmöglichkeiten vorgeschlagen.

Umgang 4. mit schlechter Datenqualität



In Abschnitt 3.5.1 wurde erläutert, dass ein industrieller Verbrennungsprozess oft Messdaten in sehr geringer Qualität oder in sehr ungünstiger Abtastung liefert. Dies führt oft dazu, dass ein datengetriebenes Lernverfahren keine Approximation der gewünschten Zusammenhänge in der erforderlichen Qualität liefern kann.

Ein wesentlicher Aspekt für die Qualität der Daten ist neben der Anzahl der Datenpunkte und der Auswahl der besten Informationsquelle¹ das Rauschen auf den Daten. In Abschnitt 4.1 wird erläutert, wie der Begriff des Rauschens in dieser Arbeit definiert wird. Handelt es sich um industrielle Prozessdaten, so ist die Stärke des Rauschens im Vorhinein meist nicht bekannt. In Abschnitt 4.2 wird deshalb erklärt, mit welchen Methoden dieses Rauschen abgeschätzt werden kann.

Stellt sich heraus, dass die vorhandenen Daten einem starken Rauschen unterliegen, dann ergibt sich die Frage, welche Auswirkungen

¹Die Auswahl der richtigen Messstellen ist sogar von entscheidender Bedeutung. Eine Merkmalsauswahl wurde für jedes in dieser Arbeit vorkommende Modell durchgeführt. Der Prozess der Merkmalsauswahl selbst steht aber nicht im Fokus dieser Arbeit. Die verwendeten Verfahren sind deshalb in Anhang A dargestellt.

dieses Rauschen auf ein datengetriebenes Verfahren hat und wie sich negative Auswirkungen vermeiden lassen. In Abschnitt 4.3 werden entsprechende Ansätze vorgestellt und die Ergebnisse bei Anwendung der Verfahren auf Daten der Feuerleistungsführungen ausgewertet. Das Kapitel endet mit einer Zusammenfassung in Abschnitt 4.4.

4.1. Verrauschte Daten

Verrauschte Messdaten sind ein wesentliches Problem bei der Erstellung von datengetriebenen Modellen für Verbrennungsprozesse. Bevor auf die Probleme bei der Erstellung eines Modells mit verrauschten Daten eingegangen werden kann, muss zunächst geklärt werden, wie im Kontext dieser Arbeit der Begriff “Rauschen” definiert werden soll.

Im Bereich der Lernfähigen Systeme wird Rauschen häufig als eine Ungenauigkeit angesehen, die für eine (mit einer bestimmten Verteilung wirkende) Schwankung der Beobachtung sorgt [Bishop, 2006]. Im Bereich der Regelungstechnik ist Rauschen eine Unterart der an einem System wirkenden Störgrößen [Lutz and Wendt, 2002, Mann et al., 2005] die entweder gefiltert oder ausgeregelt werden muss.

Im Kontext der genannten Arbeiten ist Rauschen also eine Ungenauigkeit bei der Beobachtung eines Systemzustandes durch die der beobachtete Wert nicht dem tatsächlichen Wert entspricht. In Abbildung 4.1 ist der Effekt von Rauschen abgebildet. Eine konventionelle Form der Definition von Rauschen lässt sich wie folgt aufstellen:

Definition 4.1.1 (***messbedingtes Rauschen***)

Messbedingtes Rauschen bezeichnet Unterschiede eines Messwertes bei Wiederholung einer Messung unter gleichen Bedingungen, die durch eine Ungenauigkeit des Messgerätes oder Messverfahrens entstehen.

Handelt es sich bei dem zu messenden Wert oder Objekt um eine statische Einheit (beispielsweise die Größe eines Objektes), so ist diese Definition zutreffend und ausreichend. In der Praxis eines Verbrennungsprozesses kommt allerdings der Umstand hinzu, dass sich eine Messung nicht unter vollkommen gleichen Bedingungen wiederholen lässt. Wird beim Verbrennen eines Stückes Kohle die entstehende Wärme gemessen, so lässt sich diese Messung nicht wiederholen, weil das Referenzexemplar verbrannt ist und nicht ein zweites Mal verbrennen kann. Die Messung lässt sich nur wiederholen, indem man ein zweites **ähnliches** Stück Kohle verbrennt. Entsteht bei dieser zweiten Messung ein anderer Messwert, so kann nicht genau gesagt werden, ob die Veränderung aufgrund einer Ungenauigkeit des Messgerätes (messbedingtes Rauschen) zustande kam, oder ob die Veränderung dadurch kam, dass das zweite Messobjekt dem ersten nicht so ähnlich war wie man annahm. Möglicherweise hatte das zweite Stück Kohle einen geringfügig höheren Wasseranteil als das erste. Kann man eine Messung also nicht mit genau demselben Messobjekt wiederholen, dann entsteht zusätzlich systembedingtes Rauschen:

Definition 4.1.2 (*systembedingtes Rauschen*)

Systembedingtes Rauschen bezeichnet Unterschiede eines Messwertes bei Wiederholung einer Messung die durch eine nicht beobachtete Veränderung der Messprobe zwischen den einzelnen Messungen entstehen.

Diese Definition von Rauschen ist wesentlich seltener. Sie findet sich vor allem dann, wenn zeitlich ablaufende Prozesse untersucht werden, so z.B. bei der Definition des Kalman-Filters [Kalman, 1960, Welch and Bishop, 1995] bei der zwischen *process noise* und *measurement noise* unterschieden wird. Das Gesamttrauschen ξ entsteht dann durch Summation von messbedingtem Rauschen ξ_{mess} und systembedingtem Rauschen ξ_{sys} .

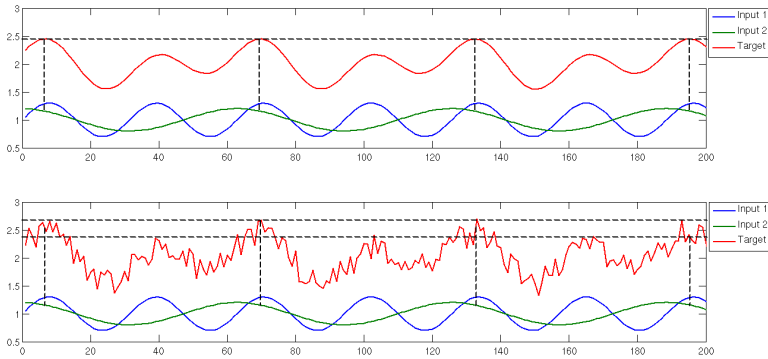


Abbildung 4.1. – Beispiel für ein unverschmutztes (oben) und verschmutztes (unten) Signal. Das Ziel (rot) ergibt sich aus der Summation der beiden Eingabesignale (blau und grün). Im unverschmutzten Fall entsteht bei gleichen Eingaben auch der gleiche Wert des Ziels. Im verschmutzten Fall entstehen unterschiedliche Zielwerte bei gleichen Eingaben.

Für diese Arbeit hat der Umgang mit verschmutzten Daten aus zwei Gründen eine hohe Bedeutung:

1. Messbedingtes Rauschen ist meist zufällig und kann deshalb nicht vorhergesagt werden. Die Stärke des Rauschens bestimmt also, wie genau ein Messwert durch ein Modell approximiert werden kann.
2. Rauschen führt zu widersprüchlichen Daten, da ursprünglich gleiche Zustände durch Rauschen mit einem unterschiedlichen Messwert wahrgenommen werden. Widersprüchliche Daten wiederum beeinträchtigen die Approximationsfähigkeit von datengetriebenen Modellen, wie sie in dieser Arbeit verwendet werden.

Ein gutes Beispiel für die Stärke des Messrauschens findet sich bei der Analyse von Freikalkwerten in der Zementproduktion. Der Freikalkwert dient dabei als ein Gütekriterium für die Qualität des produzierten Zement-Klinkers und wird durch die Laboranalyse einer

Stichprobe bestimmt [Alsop, 2005]. Um festzustellen wie groß das Rauschen dieser Messung ist, wurde im Rahmen dieser Arbeit folgendes Experiment bei vier unterschiedlichen Zementwerken durchgeführt: eine Probe des Klinkers wurde entnommen, homogenisiert und aufgeteilt in mehrere Einzelproben. Jede dieser Proben wurde separat analysiert und der Freikalkwert bestimmt. Ist das Messverfahren exakt und die Stichprobe homogen, so müsste jede Einzelmessung den gleichen Wert liefern und die Standardabweichung der Messungen gleich Null sein. Die Ergebnisse dieses Tests sind in Tabelle 4.1 dargestellt.

	Zem A	Zem B	Zem C	Zem D
\emptyset	1,71	2,80	2,33	0,76
std	0,18	0,16	0,14	0,12
min	1,38	2,67	1,91	0,51
max	2,1	3,21	2,58	0,96
$\frac{\text{var}(\xi)}{\text{var}(y)} \%$	17,10%	3,45%	3,28%	10,13%

Tabelle 4.1. – Mittelwert, Standardabweichung sowie minimaler und maximaler Wert der mehrfachen Freikalkanalyse einer einzigen Stichprobe in vier Zementwerken. Der Wert $\frac{\text{var}(\xi)}{\text{var}(y)}$ in der letzten Zeile gibt an, wie groß die Varianz der Analyse der einzelnen Stichprobe gegenüber der Varianz der Freikalkwertmessungen aus zwei Wochen ist. Er dient als Wert für das prozentuale Rauschen der Messung.

Wie zu sehen ist, ist die Standardabweichung deutlich höher als null. Setzt man die Varianz dieser Mehrfachmessung $\text{var}(\xi)$ in ein Verhältnis zur Varianz der Messwerte innerhalb von drei Wochen $\text{var}(y)$, so ergeben sich Werte von 3%-17% für den Anteil des Rauschens. Ein ähnlich hohes Rauschen liegt auch bei anderen Messwerten vor.

Soll ein Prognosemodell für den Freikalkwert erstellt werden, kommt zu diesem experimentell bestimmten Messrauschen noch das erläuterte Systemrauschen hinzu. Dieses lässt sich wesentlich schwerer abschätzen, da es davon abhängt, wie viel Informationen die verwendeten Messungen über den Freikalkwert liefern. Soll nun ein lernfähiges Verfahren für die Approximation solcher Prozesswerte eingesetzt werden, so ist es

wichtig, die Stärke des Gesamtrauschens zu kennen, um die Güte des Modells bewerten zu können. Ein Ansatz zur Abschätzung des Gesamtrauschens wird im nächsten Abschnitt dargestellt.

4.2. Abschätzung der möglichen Approximationsgüte

Wie in Abbildung 4.1 zu sehen ist, ist eine Auswirkung von Rauschen, dass zu zwei Samples mit gleichen Eingabewerten $\mathbf{x}_1 = \mathbf{x}_2$ unterschiedliche Zielwerte $y_1 \neq y_2$ existieren können. Dies hat unmittelbare Auswirkungen auf die Approximationsleistung eines Modells. Kann keines der beiden Samples ausgeschlossen werden und stehen zur Approximation des Zielwertes nur die vorhandenen Eingabewerte zur Verfügung, so kann ein deterministisches Modell bei gleichen Eingabewerten nicht zwei unterschiedliche Zielwerte approximieren. Im besten Fall wird der Mittelwert beider Zielwerte approximiert. Dies führt unvermeidlich zu einem Approximationsfehler von $mse = (\frac{1}{2}(y_1 - y_2))^2$.

Da unterschiedliche Zielwerte bei gleichen Eingabewerten nur durch ein Rauschen des Zielkanals² zu erklären sind, ergibt sich ein Zusammenhang zwischen dem Rauschen des Zielkanals und dem minimal möglichen Fehler bei der Approximation der Daten:

²Genauso gut kann natürlich auch der Eingabekanal verrauscht sein. In beiden Fällen stimmt die Zuordnung zwischen Eingabe- und Zielwert nicht, und beides behindert eine Approximation des Ziels. Für einen Schätzer, der eine Approximation $y = f(x)$ aus Datenbeispielen lernen soll, ist es gleichermaßen hinderlich, wenn die y -Werte oder die x -Werte verrauscht sind. Die hier vorgestellten Verfahren untersuchen die Stärke des Rauschens, unabhängig davon ob das Rauschen von einem Eingabe- oder Zielkanal ausgeht. Es wird der Effekt des kombinierten Rauschens geschätzt und in ein Rauschen des Zielkanals umgerechnet.

Definition 4.2.1 (*Rauschen als Untergrenze Fehler*)

Auf einem genügend großen Datensatz ist der durch ein deterministisches Modell erreichbare Approximationsfehler größer oder gleich der Varianz des Rauschens des Zielkanals.

Aus dieser Definition ergeben sich wichtige Schlussfolgerungen bei der Entwicklung eines Modells für einen Prozess. Kann man das Rauschen des Zielkanals bei den vorhandenen Eingabekanälen bestimmen, so lässt sich daraus der minimal erreichbare Fehler abschätzen. Ist dieser Fehler größer als eine bestimmte Qualitätsschranke, so lässt sich mit den gegebenen Eingaben kein Modell entwickeln, das den Qualitätsanforderungen genügt. In diesem Fall bleiben dem Modellentwickler die Möglichkeiten zu untersuchen, ob weitere Eingabekanäle verfügbar sind, so dass sich das *systembedingte Rauschen* verringern lässt, oder durch andere Messverfahren oder Messgeräte das *messbedingte Rauschen* zu verkleinern.

Um eine solche Entscheidung treffen zu können, muss zunächst die Größe des Rauschens bekannt sein. Eine einfache Variante die Varianz der Rauschverteilung zu bestimmen, ist in Abbildung 4.2 gezeigt. Existieren im Datensatz genügend Samples, die die gleichen Eingabewerte haben, so nähert sich die Varianz der Stichprobe der Zielwerte mit gleichem Eingabewert der Varianz der Rauschverteilung an.

Dieses Verfahren zur Abschätzung des Rauschens ist allerdings stark abhängig von der Abtastung der Messwerte, der Anzahl der Eingaben und der Anzahl der verfügbaren Samples. Im Beispiel aus Abbildung 4.2 mit zwei Eingabekanälen tritt im Abstand von 57 Samples die gleiche³ Konstellation der beiden Eingaben auf. Dadurch ergeben sich bei 400 Samples sieben Datenpunkte mit denselben Eingabewerten.

³Im Fall von reelwertigen Messwerten muss zuerst festgelegt werden, wann zwei Werte als "gleich" angesehen werden. Im Beispiel wurden zwei Werte als gleich angesehen, wenn ihre absolute Differenz weniger als 0,05 betrug.

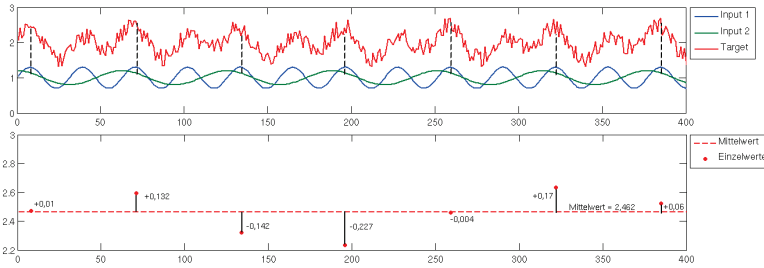


Abbildung 4.2. – Das verrauschte Ziel (rot) ergibt sich aus der Summation der beiden Eingabesignale (blau und grün). Sieben Messpunkte, an denen die beiden Eingaben den gleichen Wert haben, wurden markiert. Die Unterschiede der Zielwerte an diesen Stellen zum Mittelwert der Zielwerte wurden im unteren Bild aufgetragen. Die Stichprobenvarianz dieser Unterschiede entspricht mit 0,0203 fast der Varianz der Verteilung des Rauschens (0,02), das auf den Zielkanal addiert wurde.

Würde das Ziel noch von einem weiteren (von den anderen Eingaben unabhängigen) Eingabekanal abhängen, käme es entsprechend seltener zu gleichen Eingabekombinationen aller drei Eingaben, und entsprechend mehr Beispielsamples wären nötig, um sieben Datenpunkte mit gleichen Eingabewerten zu finden.

Bei realen Verbrennungsprozessen ist es durchaus möglich, dass ein zu bestimmender Zielwert von 5-20 untereinander unabhängigen Eingabewerten abhängt. Dementsprechend selten kommt es vor, dass eine Konstellation aller Eingaben mehrfach vorkommt und sich das oben erwähnte Verfahren anwenden lässt. Um trotzdem eine Abschätzung des Rauschens vornehmen zu können, wurde der Gammatest [Stefánsson et al., 1997, Evans et al., 2002] entwickelt, der im nächsten Abschnitt vorgestellt wird.

4.2.1. Der Gammatest zur Abschätzung des Rauschens

Durch den Gammatest [Stefánsson et al., 1997, Evans et al., 2002] soll der minimal mögliche Approximationsfehler auf einer Datenbasis ermittelt werden. Der Test versucht dabei das Problem, dass nicht mehrere vollkommen gleiche Eingabekombinationen vorliegen, zu umgehen, indem aus den vorliegenden Abständen des Zielwertes bei Punkten mit ähnlichen Werten im Eingaberaum ein verbleibendes Rauschen für identische Punkte interpoliert wird.

Dazu werden zu jedem Eingabewert s_i zunächst die k ähnlichsten Eingaben gesucht. Anschließend wird der mittlere Euklidische Abstand dieser ähnlichen Samples im Eingaberaum bestimmt. Ebenso wird der mittlere Abstand der Zielwerte dieser nächsten Nachbarn bestimmt. Aus beiden Merkmalen lässt sich eine Gerade schätzen, die abbildet, wie sich der durchschnittliche Abstand zweier Punkte im Eingaberaum auf den Abstand der Zielwerte auswirkt. Der Schnittpunkt dieser Gerade mit der y -Achse, und damit der interpolierte Abstand zweier Punkte im Eingaberaum von Null, gibt dann Auskunft über das Rauschen. Existiert kein Rauschen, so liegt der Schnittpunkt mit der y -Achse im Koordinatenursprung. Liegt er bei einem höheren y -Wert, so bedeutet dies, dass auch bei einem Abstand zweier Punkte im Eingaberaum von 0 ein Unterschied der Zielwerte >0 und damit ein Rauschen in dieser Stärke zu erwarten ist.

In Abbildung 4.3 ist dieser Vorgang dargestellt, während in Algorithmus 9 in Anhang B.3.1 noch einmal der Ablauf des Algorithmus als Pseudocode gezeigt wird.

4.2.2. Versuche mit künstlichen Daten

Um zu testen, inwiefern der Gammatest geeignet ist die Stärke des Rauschens abzuschätzen, wurden in dieser Arbeit verschiedene Versuche durchgeführt. Als erstes wurden künstliche Eingabesignale erzeugt, aus denen ein Zielsignal errechnet wurde. Auf dieses Zielsignal wurde ein zufälliges Rauschen addiert, das vom Gammatest geschätzt werden soll. Besonders interessant ist dabei, ob die Stärke des Rauschens und

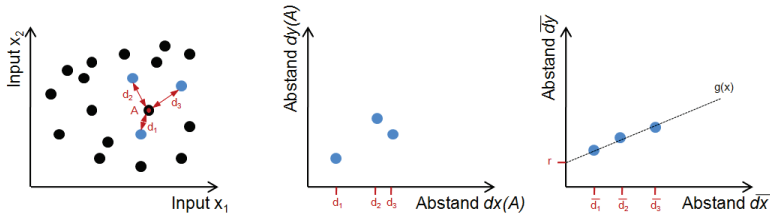


Abbildung 4.3. – Funktionsweise des Gammatests. Zu jedem Datenpunkt werden die k nächsten Nachbarn im Eingaberaum bestimmt (*links*). Ihr Abstand im Eingaberaum dx und die absolute Differenz dy ihrer Zielwerte zum aktuellen Punkt werden berechnet (*Mitte*). Aus den Mittelwerten $d\bar{x}$, $d\bar{y}$ dieser Abstände für alle Punkte lässt sich eine Gerade interpolieren, deren Schnittpunkt mit der y -Achse dem Rauschen entspricht (*rechts*).

die Art der Zielfunktion einen Einfluss auf die Schätzung des Gammatestes haben. Es wurden jeweils zwei Eingabekanäle x_1 und x_2 mit je 1000 Samples erzeugt. Als Funktionen für die Berechnung des Ziele wurden gewählt:

- $y = x_1 + x_2 + \xi$
- $y = x_1 \cdot x_2 + \xi$
- $y = (x_1 + x_2)^2 + \xi$
- $y = \sin(x_1) + \cos(x_2) + \xi$

Die Ergebnisse der Abschätzung der Stärke des Rauschens ξ sind in Tabelle 4.2 aufgeführt.

Wie aus den Ergebnissen des Tests zu sehen ist, funktioniert die Abschätzung der Stärke des Rauschens $\frac{\text{var}(\xi)}{\text{var}(y)}$ für die Funktionen $y = x_1 + x_2 + \xi$, $y = x_1 \cdot x_2 + \xi$ und $y = (x_1 + x_2)^2 + \xi$ sehr gut. Für die Funktion $y = \sin(x_1) + \cos(x_2) + \xi$ hingegen ist die Abschätzung sehr ungenau.

$\frac{\text{var}(\xi)}{\text{var}(y)}$	$x_1 + x_2$	$x_1 \cdot x_2$	$(x_1 + x_2)^2$	$\sin(x_1) + \cos(x_2)$
1%	0,97%	1,13%	1,14%	17,6%
5%	4,90%	5,07%	5,08%	21,5%
10%	9,8%	10,0%	10,0%	26,3%
20%	19,7%	19,8%	19,8%	36,2%
50%	49,0%	49,3%	49,3%	65,8%
100%	98,4%	98,4%	98,2%	114%

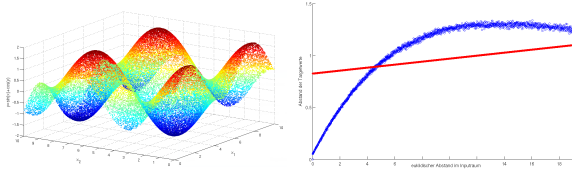
Tabelle 4.2. – Ergebnisse des Gammatests zur Abschätzung des Rauschens auf unterschiedlichen Funktionen. Die Varianz des künstlich zugefügten Rauschens $\text{var}(\xi)$ in Relation zur Varianz des Zielsignals $\text{var}(y)$ ist in der linken Spalte dargestellt. Das vom Gammatest geschätzte Rauschen ist in den anderen Spalten für die jeweilige Funktion aufgeführt.

Die Ursache für dieses Verhalten liegt in der Nichtmonotonie und Periodizität der Sinus- bzw. Cosinus-Funktion. Im Abstand von 2π treten für zwei Eingabewerte $x_a = x_b + \pi$ die gleichen Ausgabewerte $y_a = y_b$ auf. Für die Funktionsweise des Gammatestes ist ein solches Verhalten sehr schädlich. Der Abstand der Punkte im Eingaberaum ist zwar sehr groß, der Abstand der Zielwerte hingegen sehr klein. Dies führt dazu, dass die berechnete Gerade einen geringeren Anstieg hat und ein größeres Rauschen geschätzt wird. In Abbildung 4.4 ist dieses Verhalten dargestellt.

Dieser Fehler bei der Schätzung des Rauschens tritt nicht nur bei periodischen Funktionen auf, sondern auch bei stark nichtlinearen Funktionen. Er kann verkleinert werden, wenn eine größere Anzahl von Datenpunkten zur Verfügung steht und die Anzahl der nächsten Nachbarn entsprechend gewählt wird.

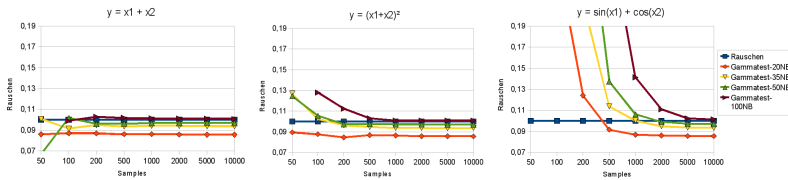
Laut [Stefánsson et al., 1997, Evans et al., 2002] konvergiert das Ergebnis des Gammatests für eine genügend große Anzahl an Punkten zum Wert des Rauschens. In Abbildung 4.5 sind die Ergebnisse des Gammatests für verschiedene Funktionen und verschiedenen große Mengen an Datensamples dargestellt⁴.

⁴Die numerischen Ergebnisse aller Versuche sind in Tabelle B.2 bis B.4 in Anhang B.3 dargestellt.



(a) $y = \sin(x_1) + \cos(x_2)$. (b) Abstände und Regressionsgrade.

Abbildung 4.4. – Periodische Funktion $y = \sin(x_1) + \cos(x_2)$ (links) und euklidische Abstände der Datenpunkte im Eingaberaum in Relation zum Abstand der Funktionswerte. Rot ist die Regressionsgerade über den Daten dargestellt.



(a) $y = x_1 + x_2$ (b) $y = (x_1 + x_2)^2$ (c) $y = \sin(x_1) + \cos(x_2)$

Abbildung 4.5. – Schätzung des Rauschens durch den Gammatest für verschiedene Funktionen in Abhängigkeit von der Anzahl der verwendeten Samples und der Anzahl der betrachteten nächsten Nachbarn. Blau dargestellt ist die tatsächliche Stärke des Rauschens. Je mehr Samples verwendet werden, desto genauer schätzt der Gammatest das Rauschen.

Wie zu sehen ist, wird die Schätzung des Gammatestes genauer, je mehr Samples zur Verfügung stehen. Für die Anzahl der benutzten nächsten Nachbarn sind Werte zwischen 35 und 50 Samples zielführend, was auch den Schätzungen in [Stefánsson et al., 1997, Evans et al., 2002] entspricht.

Für die Anwendung in der Feuerleistungsführung kann die Anzahl der Datenpunkte allerdings nicht unbegrenzt erhöht werden, da nur eine endliche (und im Fall von Labormessungen sehr kleine) Menge an Datenbeispielen zur Verfügung steht.

Kritisch ist noch ein weiterer wichtiger Punkt: Eine Annahme bei der Anwendung des Gamma-Testes ist stets, dass die verwendeten Eingabekanäle einen direkten Zusammenhang mit dem Zielkanal haben. Werden hingegen zusätzliche Kanäle als Eingabe verwendet, die in keinem Zusammenhang mit dem Ziel stehen, so wird die Abschätzung des Rauschens beeinflusst. In Abbildung 4.6 ist die Veränderung der Schätzung des Gammatestes bei Hinzufügung unnützer Kanäle dargestellt⁵.

Interessant ist dabei die Auswirkung der zusätzlichen Kanäle auf die Abschätzung des Rauschens. Da im erweiterten Eingaberaum die Euklidischen Distanzen zwischen den Samples steigen, vermutet man zunächst ein Ansteigen des geschätzten Rauschens, da die Gamma-Regressionsgrade einen geringeren Anstieg haben müssten. Gleichzeitig wird durch die zusätzlichen Eingaben aber auch die Bestimmung des "korrekten" nächsten Nachbarn verzerrt. Immer öfter werden Punkte im erweiterten Eingaberaum als nächste Nachbarn klassifiziert, die im ursprünglichen Eingaberaum nicht nebeneinander lagen. Dadurch steigt auch die Differenz der Funktionswerte und der Anstieg der Regressionsgeraden. Im vorliegenden Beispiel mit gleich verteilten zusätzlichen Eingaben dominiert der zweite Effekt den ersten, und das geschätzte Rauschen sinkt unter den tatsächlichen Wert.

⁵Die numerischen Ergebnisse aller Versuche sind in Tabelle B.5 in Anhang B.3 dargestellt.

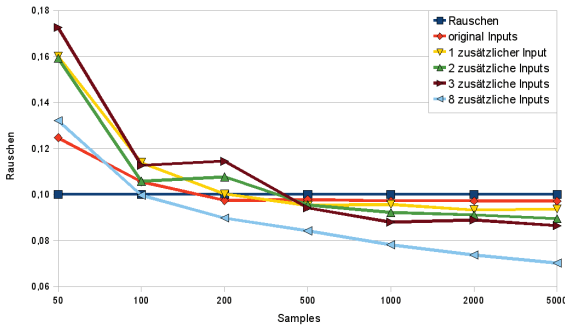


Abbildung 4.6. – Veränderung der Schätzung des Rauschens durch den Gammatest bei Hinzufügen zusätzlicher Eingabekanäle, die keinen Bezug zur Zielfunktion $y = (x_1 + x_2)^2$ haben. Die Varianz des zum Zielkanal hinzugefügten Rauschens betrug stets 10% der Zielvarianz.

Unter Beachtung dieser Kritikpunkte soll nun untersucht werden, wie sich der Gammatest bei der Anwendung auf Daten von Verbrennungsprozessen verhält.

4.2.3. Versuche mit realen Daten

Im letzten Abschnitt wurden die Funktionsweise sowie die Eigenschaften des Gammatests zur Bestimmung des Rauschens und damit des minimal möglichen Fehlers vorgestellt. Der Gammatest kann prinzipiell das Rauschen der Daten abschätzen, ist aber abhängig von der Anzahl der verfügbaren Datensamples und der möglichst optimalen Auswahl⁶ der Eingabekanäle.

In diesem Abschnitt soll das Ergebnis des Gammtestes mit dem Ergebnis neuronaler Netze und linearer Regressionsmodelle auf den Daten verschiedener Verbrennungsprozesse verglichen werden.

⁶Die in dieser Arbeit verwendeten Verfahren zur Auswahl der Eingabekanäle werden in Anhang A erläutert.

Dazu wurden im Rahmen dieser Arbeit Prognosemodelle für die Dampfleistung mit den Daten von zwei verschiedenen Müllverbrennungsanlagen sowie Softsensor-Modelle für den Freikalkwert an fünf verschiedenen Zementwerken erstellt. Die Daten wurden aufgeteilt in Trainings- und Testdaten. Die Trainingsdaten wurden benutzt, um die entsprechenden Modelle zu trainieren. Anschließend wurde auf den Testdaten zum einen der Approximationsfehler der Modelle bestimmt und zum anderen der Gammatest zur Abschätzung des Rauschens angewandt. Dieser Test wurde neben den Verbrennungsdaten auch auf den schon bekannten künstlichen Datensätzen der letzten Abschnitte durchgeführt. Die Ergebnisse sind in Tabelle 4.3 und Abbildung 4.7 dargestellt.

	$(x_1 + x_2)^2$	$\sin(x_1) + \cos(x_2)$	Zem1	Zem2	Zem3
var(r)	0,09	0,09			
NN- mse_{Q05}	0,09	0,09	0,84	0,52	0,39
LRG- mse_{Q05}	0,14	0,96	0,89	0,69	0,50
Gamma $_{k=35}$	0,08	0,10	0,56	0,27	0,24
Gamma $_{k=50}$	0,08	0,12	0,62	0,32	0,27
	Zem4	Zem5	MVA1	MVA2	
var(r)					
NN- mse_{Q05}	0,87	0,82	0,37	0,36	
LRG- mse_{Q05}	0,97	0,85	0,45	0,38	
Gamma $_{k=35}$	0,63	0,52	0,40	0,34	
Gamma $_{k=50}$	0,71	0,60	0,43	0,36	

Tabelle 4.3. – Abschätzung des Rauschens durch den Gammatest sowie Ergebnisse des Trainings eines neuronalen Netzes (NN- mse_{Q05}) sowie eines linearen Regressionsmodells (LRG- mse_{Q05}). Der Gammatest wurde mit 35 und 50 Nachbarn durchgeführt. Bei den künstlichen Datensätzen hatte das hinzugefügte Rauschen die Varianz 0,09.

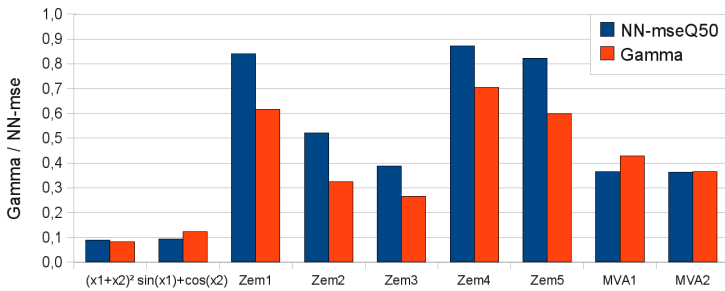


Abbildung 4.7. – Gegenüberstellung der Abschätzung des Rauschens durch den Gammatest mit 50 Nachbarn und dem besten Ergebnis des Trainings eines neuronalen Netzes.

Für jeden Test wurden insgesamt 50 Neuronale Modelle mit unterschiedlicher Startinitialisierung trainiert. Betrachtet man dann das 5%-Quantil der Fehler des neuronalen Modells ($NN\text{-}mse_{Q05}$), so entspricht dieser Fehler genau dem hinzugefügten Rauschen auf den künstlichen Datensätzen. Zum Vergleich wurde auch ein lineares Regressionsmodell trainiert, welches die nichtlinearen Datensätze natürlich deutlich schlechter approximiert.

Bei den realen Datensätzen ist das tatsächliche Rauschen nicht bekannt. Es wird hier davon ausgegangen, dass das 5%-Quantil der Fehler des neuronalen Modells zumindest in der Nähe dieses Rauschens ist. Die Abschätzung des Rauschens durch den Gammatest gelingt auf den künstlichen Daten und den Daten der Müllverbrennungsanlagen mit geringen Abweichungen. Auf den Daten der Zementwerke hingegen schätzt der Gammatest stets ein geringeres Rauschen als der Fehler des besten Approximationsmodells. Der Unterschied zwischen den Daten der MVAn und der Zementwerke besteht vor allem in der Menge der vorhandenen Daten und der Stärke des Rauschens. In Abschnitt 4.2.1 wurde gezeigt, dass bei zu wenigen Datenpunkten der Gammatest eher ein zu hohes als ein zu niedriges Rauschen schätzt. Da datengetriebene Lernverfahren anfällig auf geringe Datenmengen reagieren, ist es also durchaus möglich, dass das Approximationsergebnis der neuronalen Modelle auf den Zementdaten schlechter ist, als es aufgrund des geschätzten Rauschens möglich wäre. Stimmt das Ergebnis des Gammatests, dann ist auf diesen Anlagen ein besseres Approximationsergebnis möglich.

Außerdem fällt auf, dass das geschätzte Rauschen auf den Zementdaten höher ist als das Rauschen auf MVA-Daten. Dieses hohe Rauschen in Verbindung mit wenigen Datenpunkten kommt als Ursache für das schlechte Abschneiden der lernfähigen Modelle in Betracht. Im nächsten Abschnitt werden deshalb Verfahren vorgestellt, mit denen unter Beachtung des ermittelten Rauschens das Approximationsergebnis der neuronalen Modelle verbessert werden kann.

4.3. Auswirkungen von Rauschen auf den Lernvorgang

Im vorherigen Abschnitt ging es darum abzuschätzen, wie groß das Rauschen auf einem Datensatz ist. Dies wurde untersucht, damit der minimal mögliche Fehler eines Modells auf diesem Datensatz bestimmt werden kann. Eine zweite wichtige Frage, die sich in Verbindung mit Rauschen stellt, ist, inwiefern Rauschen auf einem Trainingsdatensatz Einfluss auf die Approximationsfähigkeit eines datengetriebenen Lernverfahrens hat.

Um dies zu untersuchen, wurde in dieser Arbeit folgender Versuch durchgeführt: Wie bereits in Abschnitt 4.2 wurde ein künstliches Zielsignal y erzeugt, das sich aus mehreren Eingabesignalen x_1, \dots, x_4 zusammensetzt. Als Funktion zur Berechnung des Zieles wurde $y = x_1 + x_2 + x_3x_4 + x_4 + x_1x_3 + \xi$ gewählt. Anschließend wurden mehrere Datensätze erzeugt: ein Testdatensatz, in dem kein Rauschen vorkommt, der Rauschterm ξ also gleich 0 ist, und Trainingsdatensätze, in denen die Varianz des Rauschens eine Größe zwischen 5%-100% der Varianz des Zieles hat. Anschließend wurden jeweils ein Neuronales Netz und ein lineares Regressionsmodell mit den Daten der verschiedenen (verrauschten) Trainingsdatensätze trainiert und auf dem unverrauschten Testdatensatz getestet. Die Approximationsgüte der Verfahren abhängig vom Rauschen des Trainingsdatensatzes ist in Abbildung 4.8 dargestellt⁷.

Es ist klar zu erkennen, dass der Approximationsfehler steigt, je stärker das Rauschen auf den Trainingsdaten ist. Neuronale Netze werden von diesem Effekt wesentlich stärker betroffen als lineare Regressionsmodelle. Liegt kein Rauschen vor, erzielt das Neuronale Netz einen sehr viel geringeren Fehler als das lineare Regressionsmodell.

⁷Die numerischen Ergebnisse aller Versuche sind in Tabelle B.6 in Anhang B.3 dargestellt.

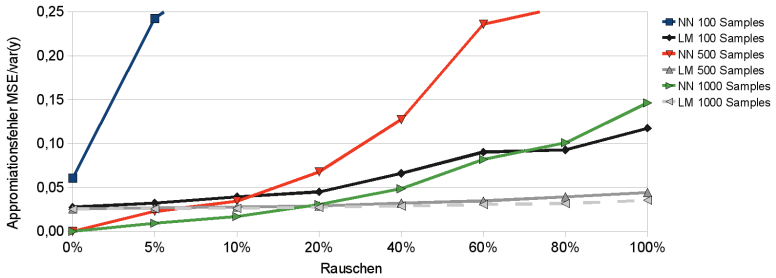


Abbildung 4.8. – Auswirkungen von verrauschten Trainingsdaten auf die Approximationsfähigkeit von Modellen auf unverrauschten Testdaten. Abgebildet ist der Median des mittleren quadratischen Approximationsfehlers auf unverrauschten Testdaten (Y-Achse) im Verhältniss zur Stärke des Rauschens (X-Achse). Zur Approximation des Zielwertes wurden ein lineares Modells (LM) und ein neuronales Netzwerk (NN) angewendet. Der Approximationsfehler ist in Relation zur Varianz des Zieles angegeben. Ein Fehler von 0,5 bedeutet also: der mittlere quadratische Fehler ist gleich der Hälfte der Varianz des Zieles. Die Versuche wurden jeweils mit 100, 500 und 1000 Trainingsamples durchgeführt.

Dies war zu erwarten, da die zu approximierende Funktion nichtlinear ist und deshalb von einem linearen Modell nicht korrekt approximiert werden kann. Steigt das Rauschen jedoch an, schwindet der Vorteil des Neuronalen Netzes. Ab einer Varianz des Rauschens von 5%-20% (abhängig von der Größe des Beispieldatensatzes) wird mit einem einfachen linearen Modell schließlich ein geringerer Fehler erzeugt als mit einem Neuronalen Netz.

Woher kommt diese starke Anfälligkeit der Neuronalen Netze gegenüber Rauschen? Um dies zu untersuchen wurde der obige Test mit verschiedenen Netzarchitekturen wiederholt. Die Ergebnisse dieses Tests sind in Abbildung 4.9 dargestellt⁸.

⁸Die numerischen Ergebnisse aller Versuche sind in Tabelle B.7 in Anhang B.3 dargestellt.

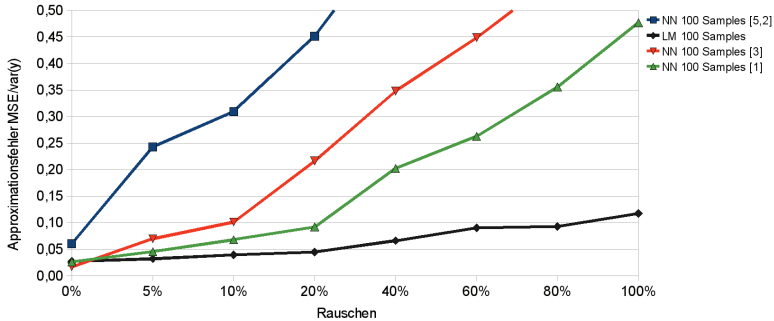


Abbildung 4.9. – Einfluss der Größe des Neuronalen Netzwerkes auf die Approximationsfähigkeit bei verrauschten Daten. *Blau:* NN mit zwei Hiddenschichten mit fünf und zwei Neuronen. *Orange:* NN mit einer Hiddenschicht mit 3 Neuronen. *Grün:* NN mit einem einzigen Hiddenneuron.

Es wird deutlich, dass die Verschlechterung des Ergebnisses durch Rauschen größer ist, wenn das Netzwerk mehr Neuronen beinhaltet. Wird kein *early stopping*⁹ verwendet, sind die Fehler noch wesentlich größer. Das Netz beginnt also zu *overfitten*, es lernt die verrauschten Datenpunkte "auswendig" und approximiert nicht mehr die zugrundeliegende Funktion. Dieser Effekt tritt selbst dann auf, wenn der zum Training verwendete Datensatz aufgeteilt wird in einen Trainings- und einen Validierungsdatensatz. Regularisierung [Dan Foresee and Hagan, 1997] als weitere Maßnahme gegen Overfitting verringert den Anstieg des Fehler etwas, die Problematik der Verschlechterung des Ergebnisses bleibt aber bestehen¹⁰.

⁹*Early stopping* ist sehr einfache Maßnahme gegen *Overfitting*. Der Datensatz wird aufgeteilt in einen Trainings- und einen Validierungsteil. Steigt der Approximationsfehler auf dem nicht zum Training genutzten Validierungsdatensatz, wird das Training beendet, das das Netz die Fähigkeit zu generalisieren verliert und die Trainingsdaten "auswendig" lernt.

¹⁰Der Einfluss von Regularisierung auf neuronale Modelle wird in Abschnitt 4.3.3 ausführlich untersucht.

These 4 (*Rauschen und Approximationsleistung*)

Rauschen auf einem Trainingsdatensatz wirkt sich negativ auf die Approximationsleistung eines neuronalen Modells aus. Je größer das Rauschen ist, desto größer wird der Approximationsfehler auf einem Testdatensatz.

Diese Ergebnisse stehen auf den ersten Blick im Widerspruch zu Verfahren wie *Weight Noise Injection* [Murray and Edwards, 1993] oder *Input Noise Injection/Jitter* [An, 1996]. Bei *Weight Noise Injection* werden die Gewichte des Netzes während des Trainings mit einem leichten Rauschen versehen, während bei *Jitter* bewusst Eingabe- oder Zieldaten mit zusätzlichem Rauschen versehen werden. Beides führt sowohl in der Theorie [Holmstrom and Koistinen, 1992] als auch in der Praxis [Sietsma and Dow, 1991] zu besseren Klassifikation-/Approximationsergebnissen. Bei beiden Verfahren gibt es allerdings einen signifikanten Unterschied zu der hier aufgeführten Problematik. Sowohl bei *Weight Noise Injection* als auch bei *Jitter* wird das künstlich hinzugefügte Rauschen in jedem Trainingsschritt des Netzes verändert. Das Netz kann sich so nicht fälschlich an einen verrauschten Punkt anpassen. In den oben aufgeführten Versuchen ist das Rauschen jedoch Bestandteil der Ursprungsdaten, und der Rauschwert jedes Datenpunktes bleibt während des gesamten Trainings konstant.

4.3.1. Datenauswahl unter Beachtung von Rauschen

Im vorherigen Abschnitt 4.3 wurde gezeigt, dass Rauschen negative Effekte auf die Approximationsleistung eines Modells haben kann. Im Umkehrschluss bedeutet das, dass die Approximationsleistung verbessert werden kann, wenn das Rauschen auf den Trainingsdaten verringert wird. Dies könnte z.B. geschehen, indem das Messrauschen durch Verwendung anderer Messgeräte oder das Systemrauschen durch die Verwendung anderer Informationskanäle reduziert wird. In der Praxis der Feuerleistungsführung stehen allerdings oft beide Wege nicht zur Verfügung oder sind mit hohen Kosten verbunden.

In diesem Abschnitt wird deshalb eine weitere Möglichkeit untersucht. Unter der Annahme, dass der Rauschterm ξ einer Verteilung unterliegt, die überwiegend¹¹ unabhängig von den Eingabe- und Zielwerten ist, gibt es Samples des Datensatzes, die einem größeren Rauschen unterliegen und solche, die nur einen geringen Rauschanteil haben. Das Gesamt rauschen auf den Daten könnte demzufolge reduziert werden, indem die Samples mit dem größten Rauschen aus dem Datensatz entfernt werden. Gleichzeitig wird allerdings auch der Datensatz kleiner.

Um zu klären wie sich das Entfernen von Samples mit großem Rauschen auf das Approximationsergebnis auswirkt, wurde folgender Versuch durchgeführt: Für drei verschiedene Funktionen wurde je ein künstlicher Datensatz aus 2000 Samples erstellt. Diese Datensätze wurden jeweils in einen Trainings- und einen Testdatensatz aufgeteilt. Wie bereits in Abschnitt 4.3 bleibt der Testdatensatz ohne Rauschen. Bei den Trainingsdatensätzen hingegen wurde dem Ziel y ein $N(\mu, \sigma^2)$ -verteiltes Rauschen hinzugefügt, wobei der Erwartungswert $\mu = 0$ und die Varianz $\sigma^2 = \frac{1}{2}var(y)$ gewählt wurde.

Mit dem Trainingsdatensatz wurden Approximationsmodelle trainiert, wobei schrittweise die am meisten verrauschten Samples aus dem Trainingsdatensatz entfernt wurden. Anschließend wurde der Fehler der trainierten Modelle auf dem (unverrauschten) Testdatensatz bestimmt. Die Ergebnisse sind in Tabelle 4.4 zu sehen.

Wie zu sehen ist, verbessern sich die Ergebnisse bei einer Entfernung der am meisten verrauschten Samples. Allerdings wird eine Mindestanzahl an Punkten benötigt, um die zugrunde liegende Funktion

¹¹Bei Verbrennungsprozessen verändert sich die Verteilung des systembedingten Rauschens je nach Brennstoff und Last. Der konkrete Wert des Rauschens eines Samples bleibt aber unabhängig, da ein vollständig abhängiger Rauschwert sonst durch ein neuronales Netz aus den anderen Größen approximiert werden könnte, was bei keinem der vorliegenden Datensätze der Fall war.

Verbleibende Samples	Rauschen $\frac{\text{var}(\xi)}{\text{var}(y)}$	Funktionen			
		$x_1 + x_2$	$x_1 \cdot x_2$	$(x_1 + x_2)^2$	$\sin(x_1) + \cos(x_2)$
1000	50,00%	0,62%	1,29%	0,72%	4,40%
900	30,00%	0,51%	0,94%	0,50%	2,68%
800	20,00%	0,42%	0,82%	0,39%	2,85%
700	15,00%	0,36%	0,70%	0,33%	2,27%
600	11,00%	0,30%	0,54%	0,29%	1,64%
500	7,00%	0,21%	0,48%	0,25%	1,36%
400	5,00%	0,19%	0,41%	0,16%	1,54%
300	2,50%	0,14%	0,36%	0,14%	1,31%
200	1,00%	0,10%	0,32%	0,11%	1,68%
100	0,30%	0,06%	0,31%	0,07%	5,86%
90	0,22%	0,05%	0,27%	0,06%	8,16%
80	0,17%	0,05%	0,32%	0,07%	14,13%
70	0,13%	0,05%	0,35%	0,06%	22,37%
60	0,10%	0,05%	0,38%	0,05%	42,75%
50	0,07%	0,05%	0,51%	0,05%	44,59%
40	0,04%	0,04%	1,09%	0,06%	75,89%
30	0,03%	0,06%	2,10%	0,06%	137,79%
20	0,01%	0,20%	7,75%	0,20%	185,44%
10	0,00%	25,74%	42,60%	22,36%	236,63%

Tabelle 4.4. – Dargestellt ist der Testfehler eines neuronalen Netzes auf unverrauschten Testdaten in Abhängigkeit von der Anzahl der Trainingsdaten und des Rauschens auf den Trainingsdaten. Blau hinterlegt sind die vier jeweils besten Testfehler bei der Approximation jeder Funktion.

approximieren zu können. Wird diese Anzahl unterschritten, steigt der Testfehler erneut an.

Die Mindestanzahl der Punkte, die benötigt wird, hängt ab von der Komplexität der Funktion und der Struktur des eingesetzten Netzwerkes.

Es ist allerdings nicht immer notwendig, verrauschte Punkte komplett aus dem Datensatz zu entfernen. Wie in Kapitel 3.4.1 gezeigt, wird der Trainingsdatensatz eines neuronalen Netzes in Trainings-, Validierungs- und Testdaten aufgespalten. In dieser Arbeit wurde untersucht, ob es einen Unterschied macht, ob die verrauschten Samples eher für die Validierung oder das Training des Netzes genutzt werden. Die Ergebnisse sind in Tabelle 4.5 dargestellt.

Trainingsverfahren	$x_1 + x_2$	$x_1 \cdot x_2$	$(x_1 + x_2)^2$	$\sin(x_1) + \cos(x_2)$
Alle Punkte	0,57%	1,56%	0,92%	2,67%
ohne verrauschte Samples	0,28%	1,00%	0,55%	1,84%
Rauschen Validierung	0,24%	0,97%	0,42%	1,22%
Rauschen Training	0,61%	1,67%	1,25%	3,30%

Tabelle 4.5. – Auswirkungen der Nutzung verrauschter Samples in Validierung und Test: das künstliche Rauschen auf dem Datensatz betrug 50% der Zielvarianz. Im Versuch *Alle Punkte* wurden aus allen 1000 Samples 30% zufällig zur Validierung genutzt. Im Versuch *ohne verrauschte Samples* wurden die 300 am meisten verrauschten Punkte aus dem Datensatz entfernt. Aus den restlichen 700 Samples wurden 30% zufällig zur Validierung genutzt. Im Versuch *Rauschen Validierung* wurden die 300 am meisten verrauschten Punkte zur Validierung und die restlichen 700 zum Training benutzt. Im Versuch *Rauschen Training* wurden die 300 am wenigsten verrauschten Punkte zur Validierung und die restlichen 700 zum Training benutzt.

Es wird deutlich, dass verrauschte Daten im Trainingsdatensatz wesentlich schädlicher für das Approximationsresultat sind als verrauschte Daten im Validierungsdatensatz. Dieses Ergebnis lässt sich logisch nachvollziehen. Der Validierungsdatensatz wird vor allem benutzt um festzustellen, ob ein **Anstieg** des Fehlers während des Trainings und somit ein Auswendiglernen des Fehlers erfolgt. Ist das Rauschen zufällig verteilt, so ist der absolute Gesamtfehler auf verrauschten Daten zwar höher als auf unverrauschten, das Umkehren des Fehlergradienten durch Auswendiglernen erfolgt aber zu einem ähnlichen Zeitpunkt.

Werden verrauschte Daten deshalb vornehmlich zur Validierung genutzt, kann das Approximationsergebnis noch einmal verbessert werden, da mehr unverrauschte Punkte für den Trainingsdatensatz zur Verfügung stehen.

These 5 (*Rauschen für Validierungsdaten*)

Wird ein Approximationsmodell erstellt und werden zum Training des Modells

Validierungs- sowie Trainingsdaten benötigt, so ist es vorteilhaft, verrauschte Punkte im Validierungsdatensatz zu verwenden und unverrauschte im Trainingsdatensatz.

Um trotz verrauschter Daten bessere Approximationsresultate zu erhalten, gibt es mehrere Möglichkeiten. Zum einen kann der Datensatz untersucht werden. Gelingt es besonders verrauschte Punkte zu identifizieren, können diese aus dem Trainingsdatensatz des Modells ausgeschlossen werden. Das Trainingsverfahren bleibt bei diesem Ansatz unverändert, es wird nur der zum Training verwendete Datensatz modifiziert. Zum anderen kann das Trainingsverfahren so modifiziert werden, dass Trainingsmethoden angewandt werden, die weniger stark durch Rauschen beeinflusst werden als die Standardverfahren. In diesem Fall bleibt der Datensatz unverändert. Drittens ist es möglich, beide Verfahren zu kombinieren und sowohl das Trainingsverfahren als auch den verwendeten Datensatz anzupassen. Im Folgenden werden Verfahren für jede dieser drei¹² Methoden erläutert und an realen Datensätzen aus verschiedenen Verbrennungsprozessen getestet.

4.3.2. Verfahren zur Ausreißerkennung

Im vorangegangenen Abschnitt wurde gezeigt, dass das Approximationsergebnis eines Modells verbessert werden kann, wenn die am meisten verrauschten Punkte aus dem Trainingsdatensatz entfernt werden oder hauptsächlich zur Validierung des Modells genutzt werden. Dazu wurden in den vergangenen Abschnitten künstliche Datensätze genutzt, bei denen der Rauschanteil jedes einzelnen Datenpunktes bekannt war.

¹²Es ist noch ein weiteres Verfahren denkbar: die Verwendung mehrerer Modelle statt eines einzelnen. Dieser Ansatz wird unter dem Stichwort "Ensembles" in Kapitel 5 und 6 ausführlich getestet.

Bei einem realen Datensatz ist dieser Rauschanteil unbekannt und muss zuerst geschätzt werden um anschließend die am meisten verrauschten Punkte aus dem Datensatz auszuschließen. Stark verrauschte Messwerte werden in der Literatur meist als "Ausreißer" oder *Outlier* bezeichnet. Der Umstand, dass Ausreißer die Approximationsfähigkeit von Modellen stören, ist hinreichend bekannt¹³, weshalb zahlreiche Verfahren existieren, um Ausreißer zu erkennen. Viele dieser Verfahren haben allerdings Voraussetzungen oder Einschränkungen, die einen Einsatz auf Daten von Verbrennungsprozessen nicht möglich machen.

In [Papadimitriou et al., 2003] werden Verfahren zum Entdecken von Ausreißern in folgende Kategorien eingeteilt:

1. **verteilungsbasierte Ansätze:** Die Werte der Samples werden mit einer definierten Verteilung verglichen. Samples, deren Wert außerhalb einer bestimmten Grenze der definierten Verteilung liegen, gelten als Ausreißer [Barnett and Lewis, 1994].
2. **tiefenbasierte Ansätze:** Die Lage der Beispieldaten im Raum wird durch mehrschichtige konvexe Hüllen (ähnlich Hüllen der Computergrafik) approximiert. Datenpunkte in den äußeren Hüllen werden als Ausreißer klassifiziert.
3. **clusterbasierte Ansätze:** Die Daten werden mit einem Clusterverfahren verarbeitet. Punkte, die keinen Cluster zugeordnet werden können oder Cluster, die nur aus wenigen Punkten bestehen, gelten als Ausreißer.
4. **distanzbasierte Ansätze:** Ein Datenpunkt wird als Ausreißer klassifiziert, wenn alle anderen Datenpunkte im Datenraum weiter als einen bestimmten Radius vom untersuchten Punkt entfernt sind [Knorr et al., 2000].

¹³Beispielsweise [Bishop, 1995] unter dem Schlagwort "Outlier".

- dichtebasierte Ansätze:** Die Entfernung eines Punktes zu seinen Nachbarn wird ins Verhältnis gesetzt zur Dichte der Datenpunkte in der Nachbarschaft. Punkte, die im Verhältnis zur Dichte ihrer Nachbarn weit von diesen entfernt sind, gelten als Ausreißer [Breunig et al., 2000].

Die Verfahren der Kategorie 1-3 kommen für Daten aus Verbrennungsprozessen aufgrund der Verteilung der Prozessdaten nicht in Frage. In Abbildung 4.10 ist die Verteilung zweier Signale aus unterschiedlichen Verbrennungsprozessen dargestellt.

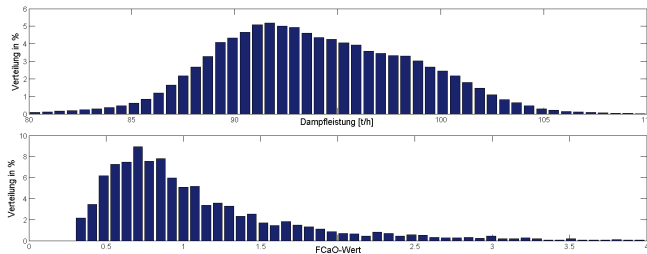


Abbildung 4.10. – Beispiele für die Verteilung der wichtigen Prozesswerte Dampfleistung in einer Müllverbrennungsanlagen (oben) und Freikalkwert in einem Zementwerk (unten).

Die Verteilungen dieser Signale entspricht keiner der gebräuchlichen statistischen Verteilungen. Das macht die Anwendung von Verfahren der Kategorie 1. sehr schwierig, da zuerst eine Verteilung geschätzt werden muss. Bei den Verfahren der Kategorie 1.-3. wird weiterhin angenommen, dass Punkte, die weit von den Schwerpunkten einer Verteilung entfernt sind, Ausreißer sind. Dies ist auch richtig: Datenpunkte mit sehr seltenen Werten entstehen vor allem dann, wenn die Anlage in einem sehr ungünstigen Zustand läuft. Das ist bei den meisten Anlagen sehr selten der Fall, weshalb solche Datenpunkte, zeitlich gesehen, Ausreißer sind. Gerade solche gefährlichen Zustände sollen durch die zu trainierenden Approximationsmodelle aber abgebildet und keinesfalls ignoriert werden, weshalb es kontraproduktiv wäre, sie

aus den Trainingsdaten auszuschließen.

Wesentlich interessanter sind deshalb distanz- und dichtebasierte Ansätze. In [Papadimitriou et al., 2003, Breunig et al., 2000, Kriegel et al., 2011] wird allerdings aufgeführt, dass distanzbasierte Ansätze nur dann zu guten Ergebnissen führen, wenn in allen Bereichen des Datenraumes eine relativ gleichförmige Dichte an Datenpunkten vorliegt. Wie in Abbildung 4.10 zu sehen ist, trifft diese Annahme für geregelte Prozesse im Bereich der Feuerungsführung nicht zu, da eine starke Häufung von Daten um den Sollwert oder den Normalzustand des Prozesses auftritt.

Von den in [Papadimitriou et al., 2003] zusammengefassten Methoden verbleiben also die dichtebasierten Ansätze, welche bei der Schätzung der Ausreißer die lokale Umgebung der Datenpunkte statt globaler Parameter des Datensatzes berücksichtigen. In [Breunig et al., 2000] wird der Local Outlier Factor (LOF) als dichtebasierter Ansatz vorgestellt. Er kann auch auf realen Datensätzen Ausreißer erkennen und wird deshalb in dieser Arbeit als Verfahren zur Ausreißererkenntnis getestet.

Local-Outlier Factor (LOF)

Das LOF-Verfahren, wie es in [Breunig et al., 2000] beschrieben wird, basiert auf der Bestimmung der *k-distance* und der *reachability-distance* rd jedes Punktes des Datensatzes. Innerhalb des Datensatzes wird dabei nicht unterschieden zwischen Eingabe- und Zielkanälen. Die *k-distance* eines Punktes s_i ist dann der euklidische Abstand des k 'ten Nachbarn von s_i zu diesem. Die $rd_k(s_i, s_j)$ zweier Punkte s_i und s_j wird dann definiert als:

$$rd_k(s_i, s_j) = \max(k - \text{distance}(s_i), d(s_i, s_j)) \quad (4.1)$$

Man beachte, dass diese Definition nicht symmetrisch ist. $rd_k(s_i, s_j)$ ist also nicht zwingend gleich $rd_k(s_j, s_i)$. Dieser Umstand ist in Abbildung 4.11 noch einmal erklärt.

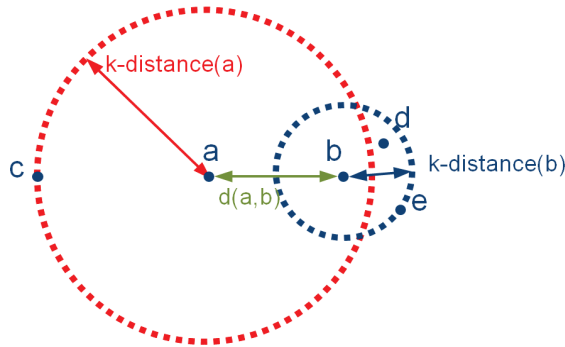


Abbildung 4.11. – k -distance und rd_k zweier Punkte bei der Anwendung des LOF-Verfahrens in einem Datenraum. Wenn $k = 2$ ist, sind die Punkte b und c die Nachbarn von a . Punkt b hingegen hat die Nachbarn d und e , da sie näher zu ihm liegen als a . Demzufolge ist die $rd_2(a, b)$ gleich der k -distance(a), die $rd_2(b, a)$ hingegen ist gleich $d(a, b)$.

Die *local reachability distance* (lrd) eines Punktes berechnet sich dann wie folgt:

$$lrd_k(s_i) = \frac{k}{\sum_{j=1}^k rd_k(s_i, s_j)} \quad (4.2)$$

Mithilfe dieser *local reachability distance* lässt sich dann der *Local Outlier Factor* (LOF) wie folgt berechnen:

$$lof_k(s_i) = \frac{\sum_{j=1}^k \frac{lrd_k(s_j)}{lrd_k(s_i)}}{k} \quad (4.3)$$

Der LOF eines Punktes ist damit ein Maß dafür, wie stark sich die Nachbarschaft dieses Punktes von der Nachbarschaft seiner Nachbarn

unterscheidet. LOF hat den Vorteil, dass es ein lokales Maß ist, also auch bei Datensätzen funktioniert, deren Datenpunkte nicht an allen Stellen des Datenraumes in der gleichen Konzentration vorliegen.

Der LOF-Ansatz hat allerdings auch zwei Nachteile, die bei der Verarbeitung von Prozessdaten zu Problemen führen können. Zum einen werden einzelne Punkte, die weit von anderen Datenhäufungen entfernt sind, mit einer hohen Wahrscheinlichkeit als Outlier klassifiziert. Im Sinne einer reinen Outlierdetektion ist das auch vollkommen korrekt. Für den Anwendungszweck, der in dieser Arbeit untersucht wird, ist aber wichtiger, ob der Zusammenhang zwischen Eingaben und Ziel schlüssig mit dem Rest des Datensatzes ist.

Der zweite Nachteil ist, dass bei LOF alle Kanäle des Eingabe-Ziel-Raums gleichberechtigt sind. Für den Zweck, den besten Datensatz für die Approximation eines Ziels zu finden, ist es aber sehr wahrscheinlich, dass verschiedene Eingaben unterschiedlich wichtig für die Approximation sind. Aus diesen Gründen wurde im Rahmen dieser Arbeit ein weiteres Verfahren entwickelt um Ausreißer zu erkennen. Dieses Verfahren wurde Noise Estimation with Local Linear Models (NELLM) [Steege et al., 2012] genannt.

Noise Estimation with Local Linear Models (NELLM)

Die Basiskomponente des NELLM Verfahrens ist das Local Linear Model (LLM) [Walter et al., 1990]. LLMs bestehen aus linearen Approximationsmodellen, die jeweils nur für einen beschränkten Teil des Eingaberaumes zuständig sind. Durch die Kombination vieler linearer Modelle kann dann ein nichtlinearer Zusammenhang über dem gesamten Eingaberaum abgebildet werden.

In dieser Arbeit werden LLMs nicht zur Approximation des Ziels genutzt, sondern zur Abschätzung des Rauschens. Dazu werden auf einem Datensatz S zunächst die nächsten Nachbarn N_i eines jeden Datenpunktes $s_i \in S$ im Eingaberaum bestimmt. Auf diesen Nachbarn

wird ein Lokales Lineares Modell bestimmt, welches aus den Eingaben den Zielwert bestimmt. Anschließend wird überprüft, welchen Fehler der aktuelle Punkt s_i gegenüber der Approximation durch das lineare Modell hat.

Ist dieser Fehler klein, so wird ein geringes Rauschen angenommen, ist er hingegen groß, so wird ein großes Rauschen angenommen. Diese Abschätzung wird für alle Punkte des Datensatzes wiederholt, wobei für jeden Punkt ein neues Lineares Modell auf Basis seiner Nachbarpunkte aufgebaut wird. Die Funktionsweise des Algorithmus ist in Algorithmus 1 und Abbildung 4.12 zusammengefasst.

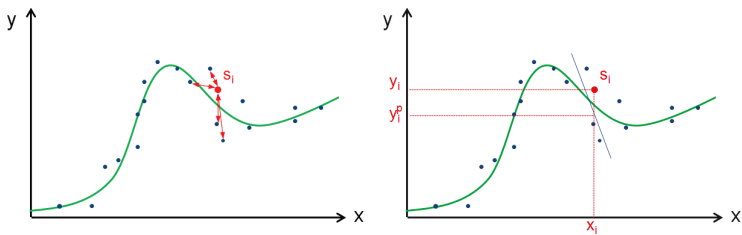


Abbildung 4.12. – Funktionsweise des NELLM-Algorithmus mit eindimensionalem Eingaberaum. Dargestellt sind die (unbekannte) Zielfunktion (grün) und die verrauschten Samples des Datensatzes (blau). Zu jedem Punkt s_i werden die nächsten Nachbarn im Eingaberaum x detektiert (links). Über diesen Nachbarn wird eine Regressionsgerade des Ziels y geschätzt (rechts). Anschließend wird der Zielwert y_i des Punktes s_i mit dem geschätzten Zielwert y_i^p verglichen. Die Differenz beider Werte ist das geschätzte Rauschen des Punktes s_i .

Um die Leistung der verschiedenen Verfahren vergleichen zu können, wurden in dieser Arbeit Tests mit den Daten verschiedener realer Verbrennungsprozesse durchgeführt. Für diese Tests wurden jeweils 1-2 Jahre Daten aus fünf unterschiedlichen Zementwerken und zwei Müllverbrennungsanlagen benutzt. Aus den vorhandenen Daten

Algorithmus 1 Outlier-Erkennung NELLM

Eingabe: Datensatz $S = \{s_i | i = 1, \dots, n; s_i = (\mathbf{x}_i, y_i); \mathbf{x}_i \in \mathbb{R}^m; y_i \in \mathbb{R}\}$ bestehend aus Eingabewerten \mathbf{x} und Zielwerten y

Anzahl nächster Nachbarn k

Ausgabe: Wichtungen $\mathbf{w} = \{w_i | i = 1, \dots, n; w_i \in \mathbb{R}; 0 \leq w_i \leq 1; \}$

$dx(s_i, s_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 \forall s_i, s_j \in S, i \neq j$ // bestimme euklidischen Abstand aller Punkte im Eingaberaum

for all $s_i \in S$ **do**

$N_i \leftarrow \{s_j | j = 1, \dots, k; s_j \in S, i \neq j, s_j \text{ ist } k\text{-nächster Nachbar von } s_i\}$

$y_j^p = p_1 x_{1j} + p_2 x_{2j} + \dots + p_m x_{mj} + p_{m+1}$ // stelle lineares Regressionsmodell für alle $\{y_j\}$ mit $s_j \in N_i$ auf

$e_i = |y_i - y_i^p|$ // bestimme Regressionsfehler

end for

$e_{max} = \max(e_i); e_{min} = \min(e_i);$

$w_i = \frac{e_i - e_{min}}{e_{max} - e_{min}}; \forall i = 1, \dots, n$

return $\mathbf{w} = \{w_i | i = 1, \dots, n\}$

wurden zufällige Datenausschnitte extrahiert, die in einen Trainings- und einen Testbereich aufgespalten wurden. Auf den Trainingsdaten wurden die beiden vorgestellten Verfahren LOF und NELLM zur Detektion von Ausreißern angewandt. Anschließend wurden die 20% der Datenpunkte, die die höchste Ausreißerwertung erhielten, aus dem Trainingsdatensatz entfernt. Der verbleibende Datensatz wurde zum Training eines Neuronalen Approximationsmodells genutzt, wobei die am meisten verrauschten 20% der Daten zum Bestimmen des Validierungsfehlers genutzt wurden. Die so erhaltenen Modelle wurden anschließend auf dem Testdatensatz getestet, und der durchschnittliche Mean Squared Error (MSE) wurde bestimmt. Dieser Vorgang wurde 500 Mal mit unterschiedlichen Datenausschnitten wiederholt. Um einen Vergleich für die Ergebnisse zu haben, wurde weiterhin ein neuronales Modell ohne Ausreißervorverarbeitung und Ausschluss von Datenpunkten trainiert. Die Ergebnisse sind in Tabelle 4.6 dargestellt¹⁴.

$\overline{mse}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5
Normal	0%	0%	0%	0%	0%
NELLM	-8,5%	-6,6%	-10,8%	-2,8%	-7,3%
LOF	15,2%	-9,5%	6,9%	-9,2%	0,5%

Tabelle 4.6. – Median $\overline{mse}_{Q50\%}$ des Approximationsfehlers von MLP-Modellen die mit Daten von jeweils einem der beiden Vorverarbeitungsverfahren trainiert wurden. Der Median wurde jeweils über je 500 Versuchen pro Verfahren bestimmt. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben. Die absoluten Fehler sind in Anhang B.3 aufgeführt.

Wie zu sehen ist, ist das NELLM-Verfahren am besten geeignet um die Approximationsleistung des Modells zu steigern. Im Schnitt kann eine Verbesserung der Approximation um 5,7% gegenüber dem Standard-Training erreicht werden. Mit dem LOF-Verfahren sind bei einigen Anlagen größere Verbesserungen möglich, gleichzeitig gibt

¹⁴Die absoluten Ergebniszahlen und die Parameter des Verfahrens sind in Tabelle B.8 in Anhang B.3 aufgeführt.

$\overline{mse}_{Q50\%}$	MVA1	MVA2	\emptyset
Normal	0%	0%	0%
NELLM	-2,0%	-1,7%	-5,7%
LOF	1,3%	2,2%	1,0%

Tabelle 4.7. – Median $\overline{mse}_{Q50\%}$ des Approximationsfehlers von MLP-Modellen die mit Daten von jeweils einem der beiden Vorverarbeitungsverfahren trainiert wurden. Der Median wurde jeweils über je 500 Versuchen pro Verfahren bestimmt. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben. Die absoluten Fehler sind in Anhang B.3 aufgeführt.

es aber Anlagen, bei denen eine Verschlechterung gegenüber dem Standard-Training zu beobachten ist. Zurückzuführen ist dieser Unterschied auf die eingangs erwähnte unterschiedliche Behandlung des Ziels in den beiden Verfahren. NELLM ist targetorientiert, während LOF alle Kanäle gleich behandelt.

Da bei dem NELLM-Verfahren LLMs eingesetzt werden, um das Rauschen eines Samples zu schätzen, wurde außerdem überprüft, welche Approximationsleistung ein LLM gegenüber dem sonst verwendeten Neuronalen Modell hat. Die Ergebnisse dieses Versuches sind in Anhang B.3 aufgeführt. Es zeigt sich, dass das LLM im Durchschnitt eine deutlich schlechtere Approximationsqualität hat als das neuronale Modell. Ein LLM kann also aufgrund der geringeren Anfälligkeit linearer Modelle gegenüber Rauschen (siehe Abschnitt 4.3) genutzt werden, um die Abweichung eines Samples vom Normverhalten innerhalb eines beschränkten Eingabebereiches zu schätzen. Soll aber ein möglichst generalisierendes Modell für die Approximation eines Ziels auf neuen Daten gebildet werden, ist das LLM den neuronalen Modellen unterlegen.

4.3.3. Robuste Trainingsverfahren

Im letzten Abschnitt wurden Methoden vorgestellt, um den zum Training eines Modells verwendeten Datensatz von verrauschten Daten zu bereinigen. Eine andere Methode um die negativen Auswirkungen von Rauschen auf die Approximationsleistung zu verringern besteht darin, das verwendete Trainingsverfahren zu modifizieren um das Training weniger anfällig für die Auswirkungen von Rauschen zu machen, als dies beim Standardverfahren der Fall ist.

Aus den in Kapitel 3 genannten Gründen werden in dieser Arbeit vor allem neuronale Modelle zur Approximation von Prozessgrößen verwendet. Eines der Probleme neuronaler Modelle ist das *Bias-Varianz-Dilemma* [Geman et al., 1992], also die Problematik, dass neuronale Netze die Trainingsdaten auswendig lernen und nicht mehr generalisieren können. Wie in dieser Arbeit gezeigt wurde, verstärkt Rauschen auf den Trainingsdaten diesen Effekt. Umso wichtiger ist es, Trainingsverfahren zur verwenden, die nicht anfällig für Overfitting sind.

In [Liano, 1996] stellt der Autor eine einfache Modifikation des Verfahrens zur Bestimmung des Approximationsfehlers vor, welche sich auf die meisten Trainingsverfahren neuronaler Modelle anwenden lässt und die Anfälligkeit von neuronalen Netzen gegenüber Ausreißern im Trainingsdatensatz verringern soll. Normalerweise¹⁵ wird beim Training neuronaler MLPs der MSE als Fehlermaß für die aktuelle Approximationsgüte des Netzes verwendet. Der MSE wird berechnet als:

$$mse = \frac{1}{2} \sum_{i=1}^n (y_i^{daten} - y_i^{netz})^2 \quad (4.4)$$

wobei y^{daten} die Zielwerte eines Datensatzes bezeichnet und y^{netz} für die dazugehörige Approximation des neuronalen Modells steht.

¹⁵In den Lehrbüchern [Zell, 1997, Bishop, 1995] wird der MSE als einziges Fehlermaß vorgestellt. In Matlab ist der MSE als Fehlermaß für neuronale Netze der voreingestellte Standard.

In [Liano, 1996] schlägt der Autor vor, statt des MSE ein anderes Fehlermaß namens Least Mean Log Squares (LMLS) zu verwenden. Der LMLS wird wie folgt berechnet:

$$lmls = \sum_{i=1}^n \log\left(1 + \frac{1}{2}(y_i^{daten} - y_i^{netz})^2\right) \quad (4.5)$$

In Abbildung 4.13 sind beide Fehlerfunktionen abgebildet.

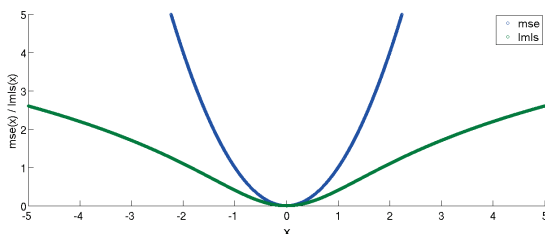


Abbildung 4.13. – Vergleich der Fehlerfunktionen des MSE (blau) und des LMLS (grün).

Wie zu sehen ist, werden große Abweichungen zwischen dem Zielwert und der Approximation vom MSE verstärkt, während sie vom LMLS abgeschwächt werden. Theoretisch verringert dies die Reaktion eines Netzwerkes auf starke Ausreißer. Liegt ein Punkt weit ab der "normalen" Approximation des Modells, so erzeugt er einen großen Fehler. Mit dem MSE-Fehlermaß würde dieser Fehler noch verstärkt, und das Lernverfahren würde deshalb während des Trainings eine Tendenz haben, die Approximationsfunktion zugunsten solcher einzelner Punkte zu verändern, so lange sie einen hohen Fehler haben.

Mit dem LMLS werden einzelne hohe Fehler hingegen abgeschwächt, weshalb das Netz mehr generalisieren soll. Ein Nachteil des Verfahrens ist die implizite Annahme, dass nur Ausreißer, also stark verrauschte Punkte, einen hohen Approximationsfehler erzeugen.

Wesentlich bekanntere Methoden ein Auswendiglernen bei neuronalen Modellen zu verhindern sind *weight decay* [Werbos, 1988] und Regularisierung für Neuronale Netze [Bishop, 1995]. Das Grundprinzip beider Verfahren besteht darin, die Gewichte innerhalb eines Neuronalen Modells nicht zu groß werden zu lassen. Wie in [Bartlett, 1999] gezeigt wird, sind große Gewichte ein wesentliches Indiz für *overfitting*. Wird die Gesamtsumme der Gewichte in die Berechnung des Fehlermaßes mit einbezogen, werden große Gewichte während des Lernvorganges verhindert. Problematisch ist dabei allerdings, ein gutes Maß zwischen der Wichtung des Approximationsfehlers und den Netzwerkgewichten zu finden. In [Irmer, 2011] wurden Regularisierungsmethoden für das Training von Neuronalen Netzen auf den Daten von Verbrennungsprozessen untersucht und Bayes'sche Regularisierung (BRG) [Dan Foresee and Hagan, 1997, MacKay, 1991] als die beste Methode identifiziert.

Eine dritte Methode gegen Overfitting, die in dieser Arbeit untersucht wurde, ist Input Noise Injection (INI) [An, 1996, Holmstrom and Koistinen, 1992] bzw. Weight Noise Injection (WNI) [Murray and Edwards, 1993, Ho et al., 2009]. Das Prinzip beider Ansätze ist künstliches Rauschen in jedem Trainingsschritt auf die Eingabedaten (INI) oder die Gewichte des Netzwerkes (WNI) zu addieren. Das dabei verwendete Rauschen ist in seiner Wirkung grundsätzlich verschieden vom in den Daten enthaltenen Messrauschen, welches das Approximationsergebnis verschlechtert. Bei WNI/INI wird das künstliche Rauschen in jedem Trainingsschritt verändert. Das Modell kann sich deshalb nicht an den Rauschwert anpassen, wie es bei Messrauschen der Fall sein kann. In [Zur et al., 2009] wird INI auf realen medizinischen Datensätzen mit anderen Methoden wie *weight decay* und *early stopping* verglichen. Die damit trainierten Modelle erzielten in allen Fällen einen besseren Approximationsfehler als die anderen verglichenen Methoden.

Um die Leistung der verschiedenen Verfahren vergleichen zu können, wurden Tests mit dem LMLS-Fehler, Bayes'scher Regularisierung und INI durchgeführt. Es wurden dafür dieselben Daten und Netzwerkkonfigurationen wie in Abschnitt 4.3.2 verwendet, um eine Vergleichbarkeit der unterschiedlichen Verfahren zu sichern. Die Ergebnisse sind in Tabelle 4.8 dargestellt¹⁶.

$\bar{e}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5
Normal	0%	0%	0%	0%	0%
BRG	-11,9%	-3,7%	-6,6%	2,5%	-11,8%
LMLS	-2,3%	1,6%	-1,4%	0,5%	-0,5%
INI	-4,2%	-0,8%	-6,9%	-0,8%	-4,0%

Tabelle 4.8. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 500 Versuchen pro Verfahren. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben.

$\bar{e}_{Q50\%}$	MVA1	MVA2	\emptyset
Normal	0%	0%	0%
BRG	-0,8%	0,1%	-4,6%
LMLS	0,1%	0,7%	-0,2%
INI	-2,8%	-1,7%	-3,0%

Tabelle 4.9. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 500 Versuchen pro Verfahren. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben.

Durch das LMLS-Verfahren wird keine merkliche Veränderung des Approximationsfehlers erzeugt. Sowohl durch Bayes'sche Regularisierung als auch durch Input Noise Injection lassen sich allerdings bessere Approximationsergebnisse erzielen. Bei Anwendung von Bayes'scher Regularisierung sind die Verbesserungen teilweise deutlich höher, allerdings tritt auf den Daten der Anlage Zem4 und MVA2 eine Verschlechterung des Ergebnisses ein. INI hingegen sorgt für eine kleinere Verbesserung des Fehlers, diese konnte aber bei sämtlichen

¹⁶Die absoluten Ergebniszahlen und die Parameter des Verfahrens sind in Tabelle B.8 in Anhang B.3 aufgeführt.

untersuchten Anlagen erzielt werden. Zur Untersuchung der Ursachen dieser Unterschiede wurde ein weiterer Versuch durchgeführt: auf den Daten der Anlage MVA1 wurden zusätzliche künstliche Ausreißer im Trainingsdatensatz eingefügt und anschließend die MLPs mit den unterschiedlichen Verfahren auf diesem Datensatz trainiert. Je besser die Modelle die Funktionswerte dieser künstlichen Punkte, die in keinem Bezug zu den realen Werten stehen, approximieren, desto anfälliger sind sie gegenüber Rauschen und Ausreißern in realen Daten. In Abbildung 4.14 ist das Simulationsresultat auf diesen Trainingsdaten dargestellt.

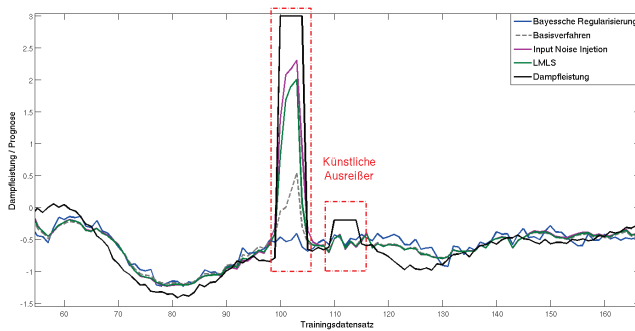


Abbildung 4.14. – Darstellung der Approximation der Dampfleistung durch verschiedene Verfahren. In den Datensatz wurden künstlich erzeugte Ausreißer eingefügt (rot markiert). Die Ergebnisse wurden 100 Mal wiederholt und gemittelt. Es ist erkennbar, wie unterschiedlich sich die Verfahren an diese Ausreißer anpassen.

Wie zu sehen ist, reagiert das Verfahren der Bayesschen Regularisierung am geringsten auf die künstlichen Ausreißer. Vergleicht man die Summe der Netzwerkgewichte, dann sieht man die Ursache für diese geringe Beeinflussung (siehe Tabelle 4.10). Das mit Bayes'scher Regularisierung trainierte MLP hat durchschnittlich sehr viel geringere Gewichte als die mit anderen Verfahren trainierten MLPs. Es sind

Fälle denkbar, in denen dieses Verhalten zu schlechteren Resultaten führt, bei stark verrauschten Trainingsdaten hat es allerdings meist positive Auswirkungen.

	Normal	BRG	LMLS	INI
Mittelwert Gewichte w_{ij}	2,582	0,135	2,681	2,629

Tabelle 4.10. – Mittlerer Wert der Gewichte der MLP die mit den aufgeführten Verfahren auf dem Datensatz von MVA1 trainiert wurden. Alle Verfahren starteten mit denselben Initial-Gewichten bei identischen Netzwerkarchitekturen. Die Versuche wurden 100 Mal wiederholt. Wie zu sehen ist, entstehen bei Bayesscher Regularisierung deutlich geringere Gewichte.

4.3.4. Kombination von robusten Trainingsverfahren und Datenvorverarbeitung

In Abschnitt 4.3.2 und Abschnitt 4.3.3 wurde gezeigt, dass sowohl durch Modifikation des Trainingsdatensatzes als auch durch Nutzung eines robusten Trainingsverfahrens eine Verbesserung des Approximationsergebnisses auf verrauschten Daten erzielt werden kann.

Da beide Ansätze an unterschiedlichen Stellen des Modelltrainings ansetzen, lassen sie sich problemlos kombinieren. In diesem Abschnitt wird untersucht, ob sich dadurch eine weitere Verbesserung des Ergebnisses erzielen lässt oder ob der Fehler konstant bleibt.

In Abschnitt 4.3.2 zeigte sich, dass NELLM am besten geeignet ist, um die Datenpunkte eines Trainingsdatensatzes auf ihren Rauschanteil zu untersuchen und dementsprechend zu filtern. In Abschnitt 4.3.3 zeigten BRG und INI von den robusten Trainingsverfahren die besten Ergebnisse. Um zu untersuchen, ob sich durch eine Kombination dieser Verfahren eine weitere Verbesserung ergibt, wurde deshalb eine Vorverarbeitung der Trainingsdaten durch NELLM mit jeweils dem robusten Trainingsverfahren BRG und INI kombiniert. Es wurden

erneut dieselben Ausgangsdaten und Netzwerkkonfigurationen wie in Abschnitt 4.3.2 verwendet. Die Ergebnisse der Tests sind in Tabelle 4.11 aufgeführt¹⁷. Um den Vergleich mit den Einzelverfahren zu erleichtern, sind ebenfalls die Ergebnisse aus Abschnitt 4.3.2 und 4.3.3 noch einmal dargestellt.

$\bar{e}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5
Normal	0%	0%	0%	0%	0%
NELLM BRG	-15,4%	-5,2%	-13,7%	-1,3%	-12,3%
NELLM INI	-9,2%	-7,8%	-8,5%	-3,2%	-6,7%
NELLM	-8,5%	-6,6%	-10,8%	-2,8%	-7,3%
BRG	-11,9%	-3,7%	-6,6%	2,5%	-11,8%
INI	-4,2%	-0,8%	-6,9%	-0,8%	-4,0%

Tabelle 4.11. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 500 Versuchen pro Verfahren. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben. Die Kombination der Verfahren NELLM, BRG und INI bringt eine Verbesserung gegenüber den Resultaten der Einzelverfahren.

$\bar{e}_{Q50\%}$	MVA1	MVA2	\emptyset
Normal	0%	0%	0%
NELLM BRG	-3,7%	-1,3%	-7,6%
NELLM INI	-3,1%	-2,7%	-5,9%
NELLM	-2,0%	-1,7%	-5,7%
BRG	-0,8%	0,1%	-4,6%
INI	-2,8%	-1,7%	-3,0%

Tabelle 4.12. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 500 Versuchen pro Verfahren. Es ist jeweils die prozentuale Verbesserung gegenüber dem Basisverfahren angegeben. Die Kombination der Verfahren NELLM, BRG und INI bringt eine Verbesserung gegenüber den Resultaten der Einzelverfahren.

¹⁷Die absoluten Ergebniszahlen und die Parameter des Verfahrens sind in Tabelle B.8 in Anhang B.3 aufgeführt.

Aus den Ergebnissen geht hervor, dass es durchaus sinnvoll ist, beide Verfahren zu kombinieren. Wird ein robustes Trainingsverfahren wie BRG oder INI zusätzlich mit dem Vorverarbeitungsansatz NELLM kombiniert, so ergibt sich daraus bei allen Testanlagen eine Verbesserung der Approximationsgüte.

Etwas differenziert ist das Ergebnis wenn man vergleicht, welchen Zusatznutzen die robusten Trainingsverfahren gegenüber der reinen Vorverarbeitung mit NELLM haben. In vielen Fällen gibt es auch hier eine zusätzliche Steigerung des Ergebnisses, in manchen Fällen wird die Approximationsleistung allerdings auch schlechter. Im Durchschnitt aller Testdaten erzielt die Kombination aus NELLM und BRG allerdings die besten Resultate. Der durchschnittliche Approximationsfehler des Standardmodells konnte durch Anwendung dieser beiden Methoden um 7,6% gesenkt werden.

These 6 (Robuste Trainingsverfahren)

Robuste Erweiterungen der Trainingsverfahren Neuronaler Modelle wie Bayes'sche Regularisierung und die Auswahl der Trainingsdaten nach dem NELLM-Verfahren verbessern die Approximationsleistung neuronaler Modelle auf verrauschten Daten.

4.4. Fazit

In diesem Kapitel wurden die Auswirkungen schlechter Datenqualität, insbesondere verrauschter Daten, auf lernfähige Verfahren untersucht. Am Beispiel der mehrfachen Untersuchung einer Stichprobe in Zementwerken wurde gezeigt, dass Rauschen in den Daten eines Verbrennungsprozesses vorkommt und eine starke Ausprägung haben kann. Rauschen wurde dabei unterschieden in *messbedingtes Rauschen*, das durch ungenaue Messgeräte entsteht, und *systembedingtes Rauschen*, das durch fehlende Informationen zustande kommt.

Es wurde gezeigt, dass die Varianz des Rauschens auf einem Datensatz eine Untergrenze für den möglichen Fehler eines Approximationsmodells bildet. Um die Stärke des Rauschens zu ermitteln, wurde der Gammatest vorgestellt. Anhand künstlicher Datensätze wurde die Zuverlässigkeit des Gammatestes in Abhängigkeit von der Charakteristik der Datensätze, der Anzahl der Samples und der Parameter des Gamatests analysiert. Anschließend wurde der Gammatest auf realen Daten angewandt und mit dem Approximationsergebnis neuronaler Modelle verglichen. Es stellte sich heraus, dass besonders bei den Daten von Zementwerken das Approximationsergebnis deutlich schlechter ist als das geschätzte Rauschen.

Als mögliche Ursache dafür wurde das hohe Rauschen auf den Zementdaten ausgemacht. Es wurde gezeigt, dass Rauschen auf den Trainingsdaten den Lernvorgang eines neuronalen Modells behindert. Dieser negative Effekt kann verringert werden, wenn besonders stark verrauschte Punkte (Ausreißer) aus dem Trainingsdatensatz entfernt werden oder nur zur Validierung des Modells genutzt werden.

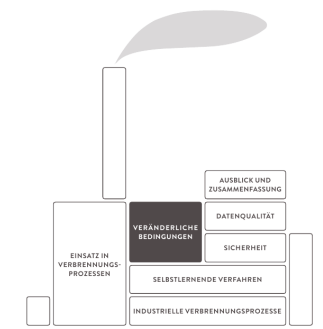
Daraufhin wurden zwei Ansätze vorgestellt, mit denen die negativen Auswirkungen der Ausreißer begrenzt werden können. Zum einen wurden Verfahren vorgestellt, mit denen der Beitrag jedes Datenpunktes zum Rauschen abgeschätzt werden kann. Ist dieser Wert bekannt, können die Punkte mit dem höchsten Rauschen entfernt werden, was

zu einer Verbesserung der Approximationsleistung führt. Zum anderen können robustere Trainingsverfahren verwendet werden, die nicht so stark wie das Standardverfahren auf Ausreißer in den Daten reagieren. Bei den Verfahren der Ausreißererkenkung schnitt das *Noise Estimation with Local Linear Models*-Verfahren (NELLM) am besten ab. In der Klasse der robusten Trainingsverfahren konnten die besten Resultate mit Bayes'sche Regularisierung (BRG) erzielt werden. Beide Verfahren lassen sich kombinieren und erzielen so im Schnitt eine Verbesserung des Approximationsergebnisses um 7,6%. Betrachtet man nur die Ergebnisse auf den Daten der Zementwerke, die laut Gammatest das höchste Rauschen hatten, so beträgt die Verbesserung sogar 9,6%.

Durch die Anwendung robuster Trainingsverfahren und einer guten Auswahl der Trainingssamples lässt sich der Nutzen neuronaler Modelle für die Feuerleistungsführung also deutlich erhöhen. Eine Zusammenfassung dieser Ergebnisse und das neue Verfahren NELLM wurden in [Steege et al., 2012] publiziert.

Im folgenden Kapitel soll untersucht werden, wie dieser Nutzen nicht nur für einen kleinen Zeitraum, sondern über mehrere Jahre und bei sich ändernden Prozessbedingungen erhalten werden kann.

5. Einfluss veränderlicher System- bedingungen



In Abschnitt 3.5.2 wurde erläutert, dass ein industrieller Verbrennungsprozess nicht stets unter den gleichen Bedingungen abläuft. Die verwendeten Brennstoffe unterscheiden sich, und durch Verschlackung der Kesselwände sowie Abnutzung der mechanischen Bauteile verändern sich auch die Bedingungen, unter denen die Verbrennung abläuft. Diese Veränderung lässt sich durch konventionelle Regelungen nur schlecht ausgleichen (siehe Abschnitt 2.1.3) und ist eine wesentliche Motivation für die Verwendung selbstlernender Verfahren.

In diesem Kapitel werden verschiedene Verfahren vorgestellt die dazu dienen, datengetriebene Modelle an die aktuellen Prozessbedingungen anzupassen. In Abschnitt 5.1 wird zunächst ein Überblick und eine Einordnung bekannter Verfahren gegeben. Es ergeben sich dabei zwei wesentliche Gruppen: solche Verfahren, die nur ein Modell verwenden und Verfahren, die mehrere Modelle gleichzeitig verwenden.

Beide Gruppen werden in Abschnitt 5.3 und Abschnitt 5.4 näher erläutert und ihre Ergebnisse verglichen. Dazu werden zum einen Daten von verschiedenen Zementwerken und zum anderen in Abschnitt 5.2 erläuterte Testdaten genutzt.

Ein Vergleich der erzielten Ergebnisse mit den Daten von Müllverbrennungsanlagen wird in Abschnitt 5.5 gezogen. Wie gewohnt endet das Kapitel mit einer Zusammenfassung der Ergebnisse in Abschnitt 5.6.

5.1. Adaptivität - Überblick

Datengetriebene, lernfähige Modelle können sich prinzipiell an neue Zusammenhänge anpassen, wenn dem Modell entsprechende neue Daten präsentiert werden und die Parameter des Lernverfahrens eine weitere Anpassung zulassen. Schon die einfache Delta-Lernregel¹ für einschichtige neuronale Feedforward Netze erlaubt prinzipiell Adaptivität, vorausgesetzt der Lernparameter η ist noch groß genug. Aufgrund dieser prinzipiellen Anpassbarkeit liegt die Vermutung nahe, dass bestehende künstliche neuronale Modelle recht einfach an neue Zusammenhänge angepasst werden können, indem sie mit neuen Daten nachtrainiert werden. In der Praxis treten bei diesem Nachtraining allerdings Probleme auf. In [McCloskey and Cohen, 1989] analysierte McCloskey ein Problem, dass er *catastrophic interference* in neuronalen Netzen benannte. *Catastrophic interference* bedeutet, dass das Lernen von neuen Zusammenhängen die Leistung des Modells auf vorher gelernten Zusammenhängen beeinträchtigt.

Das Auftreten und die Auswirkungen von *catastrophic interference* lassen sich an einem einfachen Beispiel nachweisen: Es soll mittels eines Feedforward Multi-Layer Perceptron eine Funktion der Form $f(x) = \sin(x) + \xi$ interpoliert werden, wobei ξ ein gleichverteiltes zufälliges Rauschen im Bereich $[0, 0.15]$ ist und die x -Werte im Bereich $[-5, 5]$ liegen. Das Netz wird mit Beispieldaten der Form $s = (x, f(x))$ trainiert und anschließend auf neuen Testdaten der Fehler der

¹Delta-Lernregel: $\Delta_p w_{ij} = \eta o_{pi}(t_{pj} - o_{pj})$ [Zell, 1997] Dabei ist $\Delta_p w_{ij}$ die Änderung des Netzwerkgewichtes w_{ij} für ein Sample p mit Trainingswert t_{pj} und Netz-Ausgabe o_{pj} .

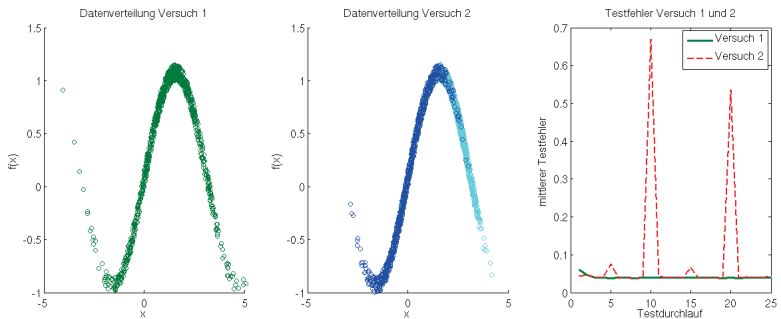


Abbildung 5.1. – *Catastrophic interference* am Beispiel der Approximation einer Sinusfunktion durch ein neuronales Netz. Werden Daten der gesamten Sinusfunktion (grün) präsentiert, kann das Netz diese gut approximieren. Werden jedoch eine Zeit lang nur Daten eines Abschnittes der Sinusfunktion (dunkelblau/hellblau) präsentiert, “vergisst” das Netz die Approximation der anderen Hälfte und erzeugt auf diesen Daten erneut einen hohen Fehler. Ein Umschalten zwischen den beiden Datenhälften fand bei den Zeitpunkten 5, 10, 15 und 20 statt wobei jeweils bei 0-4, 10-14 und 20-24 Datenverteilung 1 und ansonsten Datenverteilung 2 präsentiert wurde.

Netzwerkapproximation bestimmt. In Abbildung 5.1 sind die Beispielfunktion und der Fehler der Netzwerkapproximation dargestellt. Dieses Trainieren und Testen wird mehrmals mit demselben Netz durchgeführt, wobei sich der durchschnittliche Fehler kaum ändert. Um die Auswirkungen von catastrophic interference darzustellen, wird die Datenmenge nun in zwei Bereiche aufgeteilt. Im ersten Bereich liegen hauptsächlich Samples mit einem x -Wert zwischen -5 und 1.5 (der ansteigende Ast der Sinusfunktion) im zweiten Bereich die Werte zwischen 1.5 und 5 (der abfallende Ast der Sinusfunktion).

Wird ein Netz auf dem ersten Beispieldatensatz trainiert, so fällt der Approximationsfehler auf diesem Datensatz schnell auf ein ähnliches Niveau wie im ersten Versuch. Präsentiert man dem Netz hingegen

Daten des zweiten Datensatzes, so hat es auf diesen zunächst einen hohen Approximationsfehler. Lässt man das Netz nun einige Schritte auf dem zweiten Datensatz lernen, so fällt der Fehler auch hier sehr schnell ab. Das Netz hat sich an den zweiten Datensatz angepasst. Werden nun wieder Daten aus dem ersten Datensatz präsentiert, so ist der Fehler aber erneut sehr hoch. Durch das Lernen des zweiten Datensatzes hat das Netz die Approximationsfähigkeit auf dem ersten Datensatz verloren, obwohl beide Datensätze nicht widersprüchlich zueinander sind und sich bei gemeinsamer Präsentation durch ein einzelnes Netzwerk erlernen lassen. *Catastrophic interference* ist damit ein Spezialfall des allgemeinen Plastizitäts-Stabilitäts-Dilemmas [Grossberg, 1987], also des Problems neue Zusammenhänge zu lernen, ohne alte Zusammenhänge zu vergessen.

These 7 (Catastrophic Interference)

Ein neuronales Modell kann sich an neue Daten anpassen, wenn ihm diese präsentiert werden. Gleichzeitig können dadurch aber bereits gelernte Zusammenhänge verloren werden. Sind alte und neue Daten widersprüchlich, kann die Approximation sogar komplett fehlschlagen.

Das Vorkommen von neuen Zusammenhängen oder Datenverteilungen kann also das Ergebnis neuronaler Approximationsmodelle verschlechtern. Eine solche Veränderung der Zusammenhänge wird im allgemeinen als *concept drift* bezeichnet [Schlimmer and Granger, 1986]. In [Tsymbal, 2004] findet sich eine gute Unterscheidung mehrerer Arten von *concept drift*. Tsymbal erläutert dabei verschiedene Möglichkeiten *concept drift* zu unterscheiden: zum einen nach der Schnelligkeit, in der sich die Daten verändern: bei *sudden/abrupt concept drift* tritt diese Veränderung sehr schnell und mit einer großen Veränderung des zugrunde liegenden Zusammenhanges auf. Bei *gradual concept drift* hingegen ist die Veränderung langsam und erfolgt in kleinen Schritten.

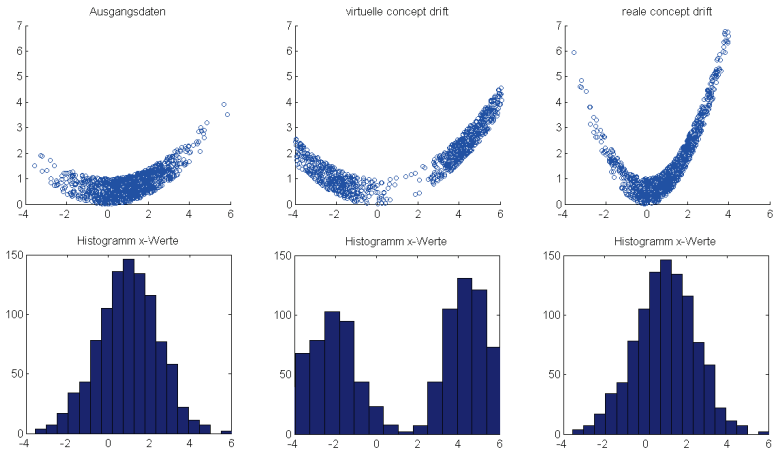


Abbildung 5.2. – *Virtual und real concept drift.* Bei virtueller *concept drift* verändert sich nur die Verteilung der Ausgangsdaten. Die Funktion selbst bleibt gleich. Bei realer *concept drift* hingegen verändert sich die zugrunde liegende Funktion.

Ein zweite Möglichkeit der Einteilung ist nach der grundlegenden Natur der *concept drift*: *virtual concept drift* ist dabei nur eine Veränderung der Verteilung der Samples im Wertebereich. Der eigentliche Zusammenhang zwischen den Messgrößen ändert sich hingegen nicht, es erscheinen aber mehr Samples aus Bereichen, die vorher selten waren. Bei *real concept drift* ändert sich dagegen tatsächlich der Zusammenhang zwischen den einzelnen Größen. In Abbildung 5.2 ist der Unterschied zwischen *real* und *virtual concept drift* an einem Beispiel dargestellt.

Neben der Einteilung von *concept drift* in verschiedene Klassen definiert Tsymbal [Tsymbal, 2004] auch Anforderungen, die ein Modell erfüllen muss, das in einer Umgebung mit *concept drift* bestehen soll.

Es muss sich demnach:

1. schnell an an eine auftretende *concept drift* anpassen,
2. robust gegenüber Rauschen sein und
3. wiederkehrende alte *concepts* erkennen.

Verfahren, die mit *concept drift* umgehen, teilt Tsymbal in zwei Klassen ein: *instance selection* beinhaltet die Auswahl eines möglichst passendes Trainingsbereiches aus den Daten (*training window*) und die Anwendung des Lernverfahrens auf den Daten dieses Bereiches. *Instance weighting* hingegen ist die Nutzung aller vorhandenen Daten, wobei allerdings jedes Sample ein Gewicht hat, das seinen Einfluss bestimmt. Diese Gewichte verschieben sich dann abhängig von den aktuellen Daten.

He wählt in [He, 2011] eine ähnliche Einteilung. Zusätzlich unterscheidet er aber zwischen Verfahren, die nur ein einzelnes Netz benutzen und Verfahren, die mit mehreren Netzen arbeiten (Ensemble-Verfahren), die auf unterschiedlichen Zeitbereichen trainiert wurden. Eine solche Definition findet sich auch in [Tsymbal et al., 2008] wobei der Autor bei den Ensemble-Verfahren noch einmal zwischen *dynamic selection* (Auswahl eines einzelnen Modells aus einem Pool) und *dynamic voting* (Wichtung und Verrechnung aller Modelle im Pool) unterscheidet.

Zusammengefasst lassen sich vier wesentliche Ansätze definieren, wie mit *concept drift* umgegangen werden kann. Ihre Eigenschaften sind in Tabelle 5.1 und Tabelle 5.2 dargestellt.

Bei den aufgeführten Publikationen handelt es sich allerdings fast ausschließlich um Methoden zur Klassifikation von Daten, in einigen Fällen sogar um den Sonderfall der Binären Klassifikation, wobei nur zwischen zwei Klassen unterschieden werden muss. Vielfach ist der Klassifikations-Aspekt ein wichtiger Bestandteil der angewandten Algorithmen.

einzelnes Modell	instance selection	Zum Anpassen des Modells wird eine bestimmte Menge von Daten ausgewählt und das Modell mit diesen Daten trainiert. Jedes Sample der ausgewählten Datenmenge hat den gleichen Einfluss auf das Training. Vertreter: <i>FLORA</i> [Widmer and Kubat, 1996], <i>example selection</i> [Klinkenberg, 2004], <i>data accumulation</i> [He, 2011]
	instance weighting	Zum Anpassen des Modells werden alle vorhandenen Daten genutzt, aber jedes Sample bekommt ein Gewicht das bestimmt, wie groß sein Einfluss auf das Training ist. Vertreter: <i>STAGGER</i> [Schlimmer and Granger, 1986], <i>example weighting</i> [Klinkenberg, 2004]

Tabelle 5.1. – Verfahren zum Umgang mit concept drift

Pool von Modellen	ensemble selection	Es werden Modelle auf unterschiedlichen Datenbereichen trainiert. Von den verfügbaren Modellen im Modellpool wird eines ausgewählt, das für die aktuellen Daten aktiviert wird. Vertreter: <i>dynamic selection</i> [Tsymbol et al., 2008]
	ensemble weighting	Es werden Modelle auf unterschiedlichen Datenbereichen trainiert. Jedes Modell aus dem Modellpool erhält ein Gewicht. Die Ausgaben aller Modelle werden unter Beachtung ihres Gewichtes zu einer Gesamtausgabe verrechnet. Vertreter: <i>dynamic voting</i> [Tsymbol et al., 2008], <i>CVM-selection</i> [Schaffer, 1993], <i>Learn++</i> [Polikar et al., 2001], <i>Learn++.NSE</i> [Elwell and Polikar, 2011], <i>Dynamic Weighted Majority</i> [Kolter and Maloof, 2003, Kolter and Maloof, 2007], <i>Incremental Cluster Evolving</i> [Protzel et al., 2000]

Tabelle 5.2. – Verfahren zum Umgang mit concept drift

Im Algorithmus *Learn++* [Polikar et al., 2001] existiert beispielsweise zum einen die Bedingung $\epsilon_t > \frac{1}{2}$ welche festlegt, dass für die Hälfte der Samples das Klassifikationsergebnis richtig sein muss, zum anderen wird auch das Gesamtergebnis durch *weighted majority* über die einzelnen Klassen bestimmt. Ein ähnlicher Bezug zur Klassifikation findet sich auch in [Widmer and Kubat, 1996, Schlimmer and Granger, 1986, Tsymbal et al., 2008, Elwell and Polikar, 2011, Kolter and Maloof, 2003, Kolter and Maloof, 2007]. Für die in dieser Arbeit vorgestellte Problemstellung sind hingegen kontinuierliche Funktionsapproximations-Aufgaben von weit größerem Interesse.

In den Abschnitten 5.3 bis 5.4 werden deshalb Modifikationen dieser Verfahren vorgestellt, die es ermöglichen, auch auf Problemstellungen der Funktionsapproximation zu arbeiten. Zunächst wird allerdings die Anwendung aus dem Bereich der Feuerleistungsführung vorgestellt, mit der anschließend die Leistungsfähigkeit der einzelnen Verfahren getestet wird.

5.2. TestszENARIO

Wie in Kapitel 2 erläutert ist eines der wesentlichen Probleme bei der Regelung industrieller Verbrennungsprozesse mit konventionellen Systemen die kontinuierliche Veränderung des Prozesses unter anderem durch Verschlackungen und Brennstoffwechsel. Diese Veränderung ist die Motivation für die Einführung adaptiver Elemente. Bevor in den folgenden Abschnitten konkrete Verfahren aufgeführt werden, mit denen ein Modell an eine veränderliche Umgebung angepasst werden kann, wird an dieser Stelle die Testumgebung erläutert, mit der die verschiedenen Modelle anschließend auf ihre Verwendbarkeit in der Feuerleistungsführung getestet werden.

Durch die Tests soll vor allem die Frage geklärt werden, welches Adaptionsverfahren sich am besten eignet um langfristig in einer Feuerleistungsführung eingesetzt zu werden. Aus diesem Grund werden

die Verfahren auf realen Daten verschiedener Verbrennungsprozesse getestet. Da besonders Langzeiteffekte und die Auswirkungen von Revisionen² von Interesse sind, wird die Güte von Freikalkprognosen in Zementwerken als Vergleichskriterium benutzt. Wie in Abschnitt 2.2 erläutert, dient der Freikalk-Wert als ein wesentliches Kriterium zur Bewertung der Zementgüte. Die Messung dieses Wertes erfolgt allerdings meist durch kostenintensive Laborversuche, welche nur in bestimmten Zeitabständen durchgeführt werden. Bei den meisten untersuchten Zementwerken lag ein solcher Freikalk-Wert alle zwei bis sechs Stunden vor. Wird nun ein Softsensor-Modell aufgestellt, welches den Freikalk-Wert aus anderen (kontinuierlichen) Sensormessungen schätzen kann, ist dies von großem Vorteil. Der vom Softsensor gelieferte Wert kann entweder in die automatische Regelung eingebunden werden oder vom Anlagenfahrer genutzt werden, um Prozessänderungen vorzunehmen. In Abbildung 5.3 sind eine solche Softsensor-Prognose und die vom Labor bestimmten Freikalkwerte zu sehen.

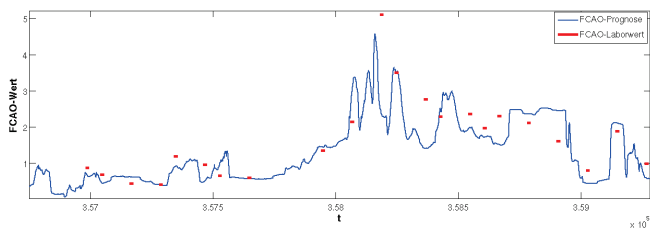


Abbildung 5.3. – Ergebnisse eines Softsensors zur Freikalkprognose in einem Zementwerk. Rot dargestellt sind vom Labor ermittelte Freikalkwerte, blau dargestellt ist der vom Softsensor berechnete Freikalkwert. Der dargestellte Zeitraum erstreckt sich über 24 Stunden.

²Eine Revision (Rev.) im Kontext technischer Großanlagen ist die Überprüfung und ggf. Ersetzung aller wesentlichen Bauteile. Bei Verbrennungskesseln werden dabei meist die Verschlackung entfernt, verschlissene Bauteile (besonders Roste) ersetzt und Verschleiß- sowie Temperaturschutzbeläge erneuert.

Ausschlaggebend für den Nutzen des Softsensors ist die Qualität der Prognose, also die Übereinstimmung des vom Modell berechneten Freikalkwertes mit den vom Labor bestimmten Freikalkwerten. Der Prognosefehler $e = |y^{model} - y^{labor}|$, die Abweichung der Modell-Prognose y^{model} vom gemessenen Laborwert y^{labor} wird deshalb als Kriterium für die Qualität des verwendeten Modells genutzt.

Verschiedene Adaptionsverfahren lassen sich vergleichen, indem man die Prognose mit unterschiedlichen Adaptionsverfahren auf den gleichen Daten mit dem gleichen Ausgangsmodell simuliert und anschließend die sich ergebenden Approximationsfehler vergleicht. In dieser Arbeit wird diese Simulation mit den Daten von fünf verschiedenen Zementwerken auf einem Zeitbereich von jeweils zwei Jahren durchgeführt. Als zu adaptierendes Modell dient dabei jeweils ein neuronales Netz in der Form eines Multi-Layer-Perceptron. Der Zielwert für das neuronale Netz ist der Freikalkwert. Als Eingangsdaten dienen andere Sensormesswerte wie Temperaturmessstellen an verschiedenen Stellen des Ofens, Materialflussmessungen, gemessene Drücke und aus Kamerabildern extrahierte Merkmale³.

Um die Übertragbarkeit der Resultate auf andere Anwendungen zu zeigen wird ebenfalls eine Simulation mit Daten von zwei Müllverbrennungsanlagen durchgeführt. Eine ausführliche Beschreibung dieser Versuche findet sich in Abschnitt 5.5.

Mithilfe dieser Simulation auf realen Anlagendaten lässt sich gut abschätzen, welches Verfahren am besten für den Einsatz in der Feuerleistungsregelung geeignet ist. Allerdings kann auf realen Daten nur schwer analysiert werden, aus welchen Gründen die Verfahren unterschiedlich abschneiden oder wie sich eine verfahrenstechnische Verbesserung erzielen ließe. Auch die Nachvollziehbarkeit und Wiederholbarkeit der Experimente ist, ohne die Daten öffentlich zur Verfügung zu stellen, nicht möglich.

³Genauere Anmerkungen zur Auswahl und Nutzung von Messungen finden sich in Anhang A

Aus diesen Gründen wurden weitere Experimente mit künstlich erstellten Datensätzen durchgeführt. Diese Datensätze bestehen aus jeweils fünf Inputkanälen x_1, \dots, x_5 und einem Targetkanal y , der sich wie folgt berechnet:

$$y = \alpha x_1 x_2 + x_3 + \alpha x_4 + x_1 x_5 + \xi \quad (5.1)$$

ξ ist ein normal verteiltes Rauschen $\mathcal{N}(0, 0.3)$ welches auf das Targetsignal aufaddiert wird und dem Modell nicht als Input präsentiert wird. Die Inputdaten x_1, \dots, x_5 werden zufällig aus einer $\mathcal{N}(0, 1)$ -Verteilung bestimmt. Besondere Bedeutung kommt dem Faktor α zu. Dieser Faktor simuliert eine langsame Veränderung der Prozessdynamik. Mit Prozessdynamik ist dabei in dieser Arbeit das Zusammenwirken einzelner Faktoren zum gemessenen Zielwert gemeint. Dazu wurden in dieser Arbeit drei Datensätze mit jeweils unterschiedlichem Verlauf des Parameters α erzeugt, welche in Abbildung 5.4 zu sehen sind.

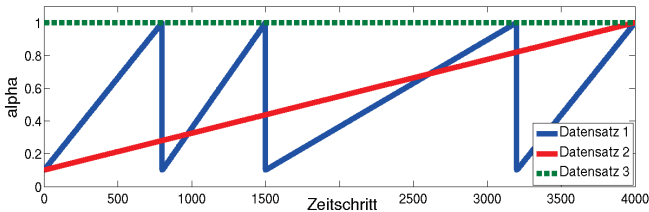


Abbildung 5.4. – Zeitlicher Verlauf des Parameters α in den drei künstlich erzeugten Datensätzen zum Testen verschiedener Adaptionverfahren.

Der Zweck dieser drei künstlichen Datensätze besteht darin drei mögliche Dynamikänderungen eines Prozesses zu untersuchen:

1. Die Dynamik ändert sich, kehrt aber regelmäßig auf ihren alten Wert zurück. Bei Verbrennungsprozessen taucht dieser Fall oft auf, da im Laufe der Zeit Verschlackungen und Verschleiß zunehmen, diese aber in regelmäßigen Abständen durch Wartungen und Revisionen der Anlage wieder entfernt werden.

2. Die Dynamik ändert sich und kehrt nicht auf ihren Ursprungswert zurück. Dieser Fall ist in der Praxis relativ selten. Er tritt dann auf, wenn sich Prozessmerkmale nur sehr langsam ändern und die betroffenen Bauteile sehr selten gewartet werden.
3. Die Dynamik ändert sich nicht. In diesem Fall ist eigentlich keine Adaption vorhandener Modelle nötig. Dieser Datensatz wurde trotzdem als Kontrollgruppe eingeführt um zu untersuchen, ob die Adaptionmethoden hier zu einer Verschlechterung führen.

Als zu adaptierendes Modell dient, genau wie bei den Freikalkmodellen, ein neuronales Netz in der Form eines Multi-Layer-Perceptron.

Besondere Bedeutung kommt dem Anfangsmodell zur Approximation der Zieldaten N_{init} zu. Bei einer realen Anlage wird dieses Modell häufig besonders stark optimiert und kontrolliert, da es das erste Modell ist, das auf der Anlage eingesetzt wird und sehr stark sowohl vom Anlagenpersonal als auch vom Ersteller der Konfiguration des Modells überwacht wird. Um diesen Umstand bei den Tests zu berücksichtigen, wurde auf den Anfangsdaten S_{init} zehn Mal ein Modell trainiert und nur das Modell mit dem geringsten Approximationsfehler als Anfangsmodell N_{init} genutzt.

Im Folgenden werden die im Rahmen dieser Arbeit entwickelten und getesteten Adaptionverfahren sowie ihre Ergebnisse auf den vorgestellten Datensätzen erläutert. Es wird dabei unterschieden, ob nur ein Modell adaptiert werden soll oder ob ein Pool von Modellen genutzt werden kann.

5.3. Adaptivität bei Verwendung eines einzelnen Modells

In diesem Abschnitt werden Methoden vorgestellt, mit denen ein Modell an einen sich ändernden Prozess angepasst werden kann. Im Unterschied zu Abschnitt 5.4 werden dabei nur Verfahren betrachtet, die mit einem einzelnen aktiven Modell arbeiten und nicht einen Pool an Modellen verwenden. Folgende Voraussetzungen gelten für die Anwendung der vorgestellten Verfahren:

1. Es gibt einen initialen Datensatz S_{init} der aus n Datenpunkten $s_i \in S_{init}, i = 1 \dots n$ besteht.
2. Jedes Sample $s = (x_1, \dots, x_m, y)$ setzt sich zusammen aus m Eingabe-/Messwerten x_1, \dots, x_m und einem Zielwert y . Am Beispiel von Zementwerken ist der Zielwert y gleich dem gemessenen Freikalkwert.
3. Es existiert ein initiales Modell N_{init} welches trainiert wurde aus den Eingangswerten x_1, \dots, x_m den Zielwert y zu approximieren.
4. In regelmäßigen Zeitabständen kommen neue Datenpunkte der Form $s = (x_1, \dots, x_m, y)$ zum bestehenden Datensatz hinzu.

Basierend auf diesen Voraussetzungen wurden in dieser Arbeit die drei folgenden Verfahren entworfen und getestet, mit denen ein einzelnes Modell adaptiert werden kann.

5.3.1. Datenakkumulation

Das Verfahren der Datenakkumulation (AKK) ist prinzipiell sehr einfach. Der zu Anfang bestehende Datensatz S_{init} wird um jedes neu auftauchende Sample $s = (x_1, \dots, x_m, y)$ erweitert. Es entsteht so ein ständig wachsender Datensatz $S_{accu}^t = S^{t-1} \cup s_t$ in dem sowohl der initiale Datensatz S_{init} als auch alle folgenden Datenpunkte bis zum aktuellen Zeitpunkt t enthalten sind.

Das bereits existierende Modell N_{init} wird nun mit dem erweiterten Datensatz trainiert und damit an die aktuellen Daten angepasst. Ein solches Nachtraining kann theoretisch jedes Mal durchgeführt werden, wenn ein neues Sample dem Datensatz hinzugefügt wurde. Aus praktischen Gründen wird aber eine Zeitdauer t_{min} gewartet, bis eine bestimmte Menge S_{min} an neuen Datenpunkten zusammengekommen ist.

In Algorithmus 2 und Abbildung 5.5 ist der Ablauf zur Anpassung eines Modells nach dem Prinzip der Daten-Akkumulation dargestellt.

Algorithmus 2 Adaptionsmethode der Datenakkumulation

Eingabe: Datensatz S_{init} , Modell N_{init} , Wartezeit für Nachtraining t_{min}

Ausgabe: adaptiertes Modell N_{adapt}^t zum Zeitpunkt t

```

 $S_{accu} \leftarrow S_{init}$ 
 $p \leftarrow 0$ 
while 1 do
   $t \leftarrow time()$ 
   $s_t \leftarrow observation(t)$ 
   $S_{accu} \leftarrow S_{accu} \cup s_t$ 
   $p \leftarrow p + 1$ 
  if  $p > t_{min}$  then
     $p \leftarrow 0$ 
     $N_{adapt}^t \leftarrow train(N_{adapt}^t, S_{accu})$ 
  end if
end while

```

Besondere Beachtung verdient der Schritt des Trainings des Modells mit dem akkumulierten Datensatz $train(N_{adapt}^t, S_{accu})$. Von Bedeutung ist dabei die Frage, ob beim Trainieren das alte Modell als Anfangszustand genutzt wird, oder ob ein zufällig neu initialisiertes Modell genutzt werden soll. Diese Frage ist dann entscheidend, wenn

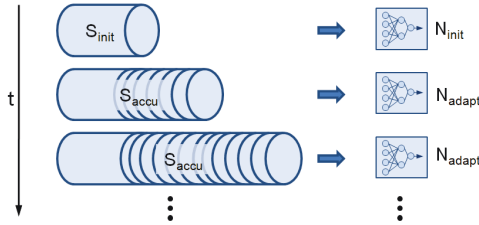


Abbildung 5.5. – Datenakkumulation: Der Ursprungsdatensatz S_{init} wird mit jedem Zeitschritt um das aktuelle Sample s_t erweitert. Sind genügend neue Daten vorhanden wird das Modell auf dem gesamten Datensatz S_{accu} trainiert.

viele lokale Extrempunkte bei der Abbildung der Modellparameter auf die Approximationsgüte existieren⁴. Wird das vorhandene Modell zur Initialisierung genutzt, so ist die Wahrscheinlichkeit groß, dass das adaptierte Modell in einem ähnlichen (lokalen) Extrempunkt konvergiert. Wird hingegen ein neu initialisiertes Modell genutzt, kann ein vollkommen anderer Extrempunkt gefunden werden.

In dieser Arbeit werden beide Konzepte untersucht um festzustellen, ob sich bei beiden Ansätzen ein signifikanter Unterschied ergibt und welcher Ansatz erfolgversprechend ist.

Dadurch, dass das Netz kontinuierlich mit neuen Daten aus dem Datensatz trainiert wird, kann es sich an neue Daten anpassen. Da bei jedem Training auch alle alten Daten enthalten sind, wird außerdem die Gefahr des Vergessens von alten Zusammenhängen gemindert.

⁴Der Zusammenhang zwischen den Werten der Parameter eines Modells und der Approximationsgüte kann als Gebirge im Parameterraum abgebildet werden. Ein Parametereinstellung mit hoher Approximationsgüte ist ein Berggipfel. Probleme beim Training ergeben sich, wenn bei der Problemstellung mehrere unterschiedliche hohe Gipfel existieren (lokale Extrempunkte), die durch Täler mit geringerer Approximationsgüte getrennt sind. Viele Trainingsverfahren finden nur den nächsten Gipfel und haben Probleme den global höchsten Gipfel zu finden.

Vollständig ausgeschaltet ist diese Gefahr allerdings nicht, da der Anteil eines Datenabschnittes an den Gesamtdaten, und damit auch ihr Einfluss auf das Trainingsergebnis, mit wachsender Datenmenge sinkt.

Durch die stetig steigende Datenmenge ergibt sich der Nachteil, dass eine immer größer werdende Zahl an Ressourcen für das Speichern der Trainingsdaten und für das Trainieren des Netzes aufgewendet werden muss. Ist die Abtastrate der Daten sehr hoch ergeben sich so schnell große Datenmengen. Bei der Anwendung als Softsensor für Freikalkwerte tritt beispielsweise nur alle zwei Stunden ein Sample auf, was eine Menge von 4.380 Samples pro Jahr ergibt. Soll hingegen eine Dampfleistungsprognose für eine Müllverbrennungsanlage erstellt werden, so liegt sekundlich ein Sample vor, und es ergibt sich eine Datenmenge von 31.536.000 Datenpunkten pro Jahr.

Die Anwendung dieses Verfahrens ist außerdem vor allem bei virtueller *concept drift* (siehe Abbildung 5.2) sinnvoll. Liegt hingegen reale *concept drift* vor, ändert sich also nicht nur die Verteilung der Eingangsdaten, sondern auch der Zusammenhang zwischen den Eingangsdaten und der Ausgabe, dann ist eine Anwendung von Datenakkumulation nicht sinnvoll. Die akkumulierte Datenmenge enthält dann Datenpunkte, in denen sowohl der alte, als auch der neue Zusammenhang enthalten ist, was zu Widersprüchen führt.

In einem solchen Fall ist die Anwendung einer *sliding window* Technik, wie im nächsten Verfahren, sinnvoller.

5.3.2. Bereichsauswahl

Um die Nachteile der Datenakkumulation (hoher Speicherbedarf, Widersprüche in Daten) zu vermeiden wird in [Widmer and Kubat, 1996, Klinkenberg, 2004, He, 2011] ein *sliding window* oder *instance selection* Verfahren vorgeschlagen. Das Grundprinzip dieser Ansätze besteht darin, nicht alle bisher gesammelten Daten für ein Training zu verwenden, sondern nur Daten aus einem bestimmten Zeitfenster. Der

zu Anfang bestehende Datensatz S_{init} wird also ebenfalls um jedes neu auftauchende Sample $s = (x_1, \dots, x_m, y)$ erweitert, allerdings wird gleichzeitig ein maximaler Zeitversatz t_{max} definiert.

Alle Samples, deren Zeitdifferenz zur aktuellen Zeit größer ist als t_{max} werden wieder aus dem Datensatz S_{window} entfernt. Dadurch hat der Datensatz S_{window} eine definierte Maximalgröße, und kann nicht wie S_{accu} immer weiter wachsen.

In Algorithmus 3 und Abbildung 5.6 ist der Ablauf zur Anpassung eines Netzes nach dem Prinzip der Bereichsauswahl (BAW) dargestellt.

Algorithmus 3 Adaptionmethode der Bereichsauswahl

Eingabe: Datensatz S_{init} , Modell N_{init} , Wartezeit für Nachtraining t_{min} , maximaler Zeitversatz t_{max}

Ausgabe: adaptiertes Modell N_{adapt}^t zum Zeitpunkt t

```

 $S_{window} \leftarrow S_{init}$ 
 $p \leftarrow 0$ 
while 1 do
   $t \leftarrow time()$ 
   $s_t \leftarrow observation(t)$ 
   $S_{window} \leftarrow S_{window} \cup s_t$ 
   $p \leftarrow p + 1$ 
  if  $p > t_{min}$  then
     $p \leftarrow 0$ 
     $S_{window} \leftarrow \{s_j | s_j \subseteq S_{window}, (t - t(s_j)) < t_{max}\}$ 
     $N_{adapt}^t \leftarrow train(N_{adapt}^t, S_{window})$ 
  end if
end while

```

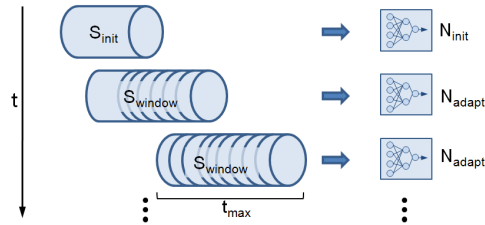


Abbildung 5.6. – Verfahren der Bereichsauswahl. Eine durch t_{max} bestimmte, feste Anzahl an Datenpunkten wird zur Adaption des Modells genutzt. Kommen neue Datenpunkte zum Datensatz hinzu werden dafür alte Datenpunkte aus dem Trainingsdatensatz entfernt.

Ebenso wie bei dem Verfahren der Datenakkumulation können auch bei diesem Verfahren zur Initialisierung des zu adaptierenden Modells verschiedene Methoden verwendet werden. Wie im vorhergehenden Abschnitt auch wird in dieser Arbeit die Initialisierung sowohl mit dem alten Modell als auch mit zufälligen Parametern untersucht.

5.3.3. Bereichsauswahl mit Kontrolle

Bei den bisherigen Verfahren der Datenakkumulation und der Bereichsauswahl wurde stets ein bestehendes Modell durch ein neues oder adaptiertes Modell ausgetauscht, sobald genug Daten für eine Adaption vorlagen. Beide Verfahren basieren damit implizit auf zwei wesentlichen Annahmen:

1. Aktuellere Trainingsdaten ermöglichen eine bessere Approximation des Zielwertes für die nächsten Testwerte.
2. Ein Modell, das mit aktuellen Daten trainiert wird, ist immer besser als ein Modell, das mit länger zurückliegenden Daten trainiert wurde.

Es stellt sich aber die Frage, ob beide Annahmen wirklich zutreffen. Insbesondere beim Neu-Training eines neuronalen Netzes kann es

durch die zufällige Initialisierung der Gewichte zu Beginn des Trainings durchaus zu unterschiedlichen Resultaten beim Training zweier Netze auf den gleichen Daten kommen. Es ist also keinesfalls sicher, dass ein neu trainiertes Netz auch wirklich bessere Ergebnisse liefert, als das bereits existierende Netz.

Aus diesem Grund wurde ein weiteres Verfahren entwickelt, das das Adaptieren an neue Daten mit einer zusätzlichen Kontrolle der Approximationsleistung des adaptierten Netzwerkes verbindet und deshalb Bereichsauswahl mit Kontrolle (BAWk) genannt wird. Das (Nach-)Training des neuen Netzwerkes erfolgt dabei nach dem gleichen Schema wie im vorherigen Abschnitt. Allerdings werden die verfügbaren neuen Daten S_{window} jetzt in zwei Datensätze $S_{teil,train}$ und $S_{teil,test}$ aufgeteilt, wobei im Datensatz $S_{teil,test}$ die neueren Daten enthalten sind.

Nachdem mit den Daten in $S_{teil,train}$ ein neues Modell N_{tmp}^t trainiert und validiert wurde, wird überprüft, ob der Approximationsfehler $e(N_{tmp}^t, S_{teil,test})$ des neuen Modells kleiner ist als der Fehler $e(N_{adapt}^t, S_{teil,test})$ des alten Modells. Nur wenn diese Bedingung tatsächlich erfüllt ist, wird auf das neue Modell umgestellt, andernfalls wird weiter das alte Modell genutzt.

In Algorithmus 4 und Abbildung 5.7 ist der Ablauf zur Anpassung eines Netzes nach diesem Prinzip dargestellt.

Das Verfahren der Bereichsauswahl mit Kontrolle bietet eine größere Sicherheit gegenüber Fehlern im Netztraining. Der Nachteil des Verfahrens besteht darin, dass ein Teil der neuen Daten $S_{teil,test}$ für den zusätzlichen Test des alten gegen das neue Netz reserviert werden muss und nicht für das Training verwendet werden kann.

Im folgenden Abschnitt werden die Ergebnisse der Tests der bisher vorgestellten Adaptionsverfahren auf künstlichen und realen Daten vorgestellt.

Algorithmus 4 Adaptionsmethode der Bereichsauswahl mit Kontrolle

Eingabe: Datensatz S_{init} , Modell N_{init} , Wartezeit für Nachtraining t_{min} , maximaler Zeitversatz für Trainingsdaten $t_{max,train}$, maximaler Zeitversatz für Testdaten $t_{max,test}$

Ausgabe: adaptiertes Modell N_{adapt}^t zum Zeitpunkt t

```

 $S_{window} \leftarrow S_{init}$ 
 $p \leftarrow 0$ 
while 1 do
   $t \leftarrow time()$ 
   $s_t \leftarrow observation(t)$ 
   $S_{window} \leftarrow S_{window} \cup s_t$ 
   $p \leftarrow p + 1$ 
  if  $p > t_{min}$  then
     $p \leftarrow 0$ 
     $S_{teil,train} \leftarrow \{s_j | s_j \subseteq S_{window}, t_{max,test} < (t - t(s_j)) < t_{max,train}\}$ 
     $S_{teil,test} \leftarrow \{s_j | s_j \subseteq S_{window}, (t - t(s_j)) < t_{max,test}\}$ 
     $N_{tmp}^t \leftarrow train(N_{adapt}^t, S_{teil,train})$ 
    if  $e(N_{tmp}^t, S_{teil,test}) < e(N_{adapt}^t, S_{teil,test})$  then
       $N_{adapt}^t \leftarrow N_{tmp}^t$ 
    end if
     $S_{window} \leftarrow S_{teil,train} \cup S_{teil,test}$ 
  end if
end while

```

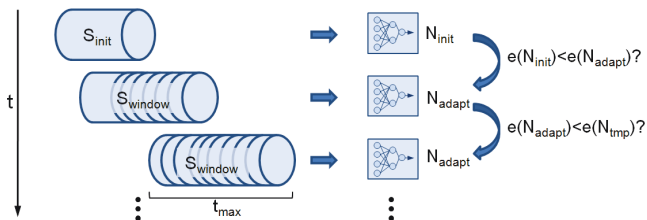


Abbildung 5.7. – Verfahren der Bereichsauswahl mit Kontrolle zur Anpassung eines Modells an neue Daten. Es entspricht dem Verfahren der Bereichsauswahl, allerdings wird nach dem Training eines neuen Modells überprüft, ob dessen Approximationsleistung besser ist als die des alten Modells, bevor ein Austausch vorgenommen wird.

5.3.4. Ergebnisse auf künstlichen Daten

Die bisher vorgestellten Verfahren wurden mit den in Kapitel 5.2 erläuterten Datensätzen getestet. In Tabelle 5.3 sind die Ergebnisse der einzelnen Adaptions-Verfahren dargestellt. Alle Versuche mit Zementdaten wurden je 50 Mal mit zufälliger Initialisierung der Startmodelle wiederholt. Bei allen Testverfahren kam dasselbe Startmodell zum Einsatz, um die Vergleichbarkeit noch einmal zu erhöhen. Als Vergleichskriterium wird der Median, also das 50%-Quantil, aller Versuche über dem mittleren Fehler auf dem gesamten Datensatz gewählt. Der dargestellte Fehler $\bar{e}_{Q50\%}$ berechnet sich bei einem Datensatz von n Samples und m Wiederholungen also wie folgt:

$$\bar{e}_{Q50\%} = \text{median}\left\{\sum_{i=1}^n \frac{|y_i^{modell} - y_i^{labor}|}{n}\right\} \quad (5.2)$$

Statt des Mittelwertes wurde bewusst der Median als Vergleichskriterium gewählt. Es kann vorkommen, dass bei einer schlechten Startinitialisierung ein Prognosefehler erzeugt wird, der sehr stark vom normalen Prognoseergebnis abweicht und den Mittelwert verzerrt. Der Median wird durch solche Ausreißer hingegen nicht beeinflusst.

$\bar{e}_{Q50\%}$	kD1	kD2	kD3	\bar{kD}
ohne Adaption (OA)	0%	0%	0%	0%
Datenakkumulati- on (AKK) 1	-27,8%	-35,4%	-6,7%	-23,3%
Datenakkumulati- on (AKK) 2	-5,9%	-15,5%	53,3%	10,7%
Bereichsauswahl (BAW) 1	-33,3%	-53,3%	-3,7%	-30,1%
Bereichsauswahl (BAW) 2	1,0%	-17,6%	88,9%	24,1%
Bereichsauswahl mit Kon- trolle (BAWk) 1	-36,3%	-53,6%	-4,4%	-31,4%
Bereichsauswahl mit Kon- trolle (BAWk) 2	-19,3%	-50,6%	-0,7%	-23,5%

Tabelle 5.3. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 50 Versuchen pro Verfahren und künstlichem Datensatz. Das Verfahren ohne Adaption (OA) des Modells dient als Referenz. Es wird die prozentuale Verbesserung des Fehlers gegenüber diesem Verfahren dargestellt. Das Verfahren mit dem besten Resultat pro Datensatz wurde jeweils blau markiert. Verfahren die schlechter abschnitten als das Referenzverfahren, wurden rot markiert. Die Nummer hinter dem Namen des Verfahrens bezieht sich auf die Initialisierung des Modells bei dem jeweiligen Verfahren. Die Zahl 1 bezeichnet Verfahren, bei denen die Gewichte der alten Netze als Startwert genutzt werden, während die Zahl 2 für Verfahren mit neu initialisierten Modellen steht.

Wie aus der Tabelle zu erkennen ist, lassen sich durch die Adaptionsverfahren besonders auf dem Datensatz 1 und 2 deutlich bessere Approximationsergebnisse erzielen als ohne eine Adaption der Modelle. Verfahren, bei denen die alten Gewichte der Modelle weiter genutzt werden, schneiden besser ab als solche, bei denen vor jedem Training neu initialisiert wurde. Die Ergebnisse der einzelnen Verfahren auf den drei Datensätzen werden im Folgenden genauer analysiert. In Abbildung 5.8 sind dazu die mittleren Approximationsfehler aller Verfahren (bei Nutzung der alten Gewichte zur Initialisierung) auf Datensatz kD1 (mit Sägezahn-förmigem α) dargestellt.

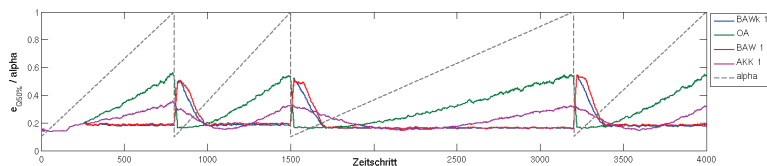


Abbildung 5.8. – Mittlerer Fehler der Adaptionverfahren auf dem künstlichen Datensatz 1. Dargestellt sind nur Verfahren, die Gewichte der alten Netze als Startwert nutzen. Ebenfalls dargestellt ist der Verlauf des Parameters α , welcher maßgeblich für die Veränderung der Systemdynamik ist.

Es ist erkennbar, wie durch die Veränderung der Parameters α das Approximationsergebnis der nicht adaptierten Modelle (OA) schlechter wird. Das Verfahren der Datenakkumulation liefert ebenfalls schlechtere Ergebnisse, sobald sich der Parameter α ändert, der Fehler steigt allerdings langsamer an als beim Verfahren ohne Adaption (OA). Dies war zu erwarten, da bei der Datenakkumulation alle bisherigen Daten berücksichtigt werden. Das führt dazu, dass die Datenbasis des Modells Daten mit unterschiedlichen Zusammenhängen zwischen den Eingaben und dem Zielwert enthält (reale *concept drift*) und damit inkonsistent ist. Das Modell approximiert dann Werte für einen Durchschnittswert von α . Liegt α tatsächlich in diesem durchschnittlichen Bereich zwischen 0,4 und 0,6, so erzielt das AKK-Verfahren aufgrund seiner großen Datenbasis bessere Resultate als die anderen Verfahren. Bei allen anderen Werten von α sind die Resultate aber schlechter.

Die Verfahren zur Bereichsauswahl (BAW und BAWk) erzielen auf Datensatz 1 die besten Ergebnisse. Aufgrund der Nutzung eines kleinen aktuellen Datenfensters können sie sich schnell an die Veränderung der Zusammenhänge zwischen Eingaben und Zielwert anpassen. Probleme gibt es allerdings an den “Revisionspunkten“, an denen α auf den Startwert zurückgesetzt wird. Der Datenbereich, den die Verfahren der

Bereichsauswahl nutzen, liegt zu diesem Zeitpunkt größtenteils noch auf alten Daten mit hohem α . Deswegen approximieren die Verfahren weiterhin die alte Dynamik und erzeugen so kurzzeitig den höchsten Fehler. Erst wenn genügend neue Daten in das Bereichsfenster aufgenommen wurden, wird das Netz angepasst, und der Fehler sinkt wieder.

Auf Datensatz 2 sind die Ergebnisse ähnlich. Die Verfahren der Bereichsauswahl sind erneut dem Verfahren der Datenakkumulation überlegen, welches durch einen widersprüchlichen Basisdatensatz nur mittelmäßige Ergebnisse erzielen kann.

Im Gegensatz dazu stehen die Ergebnisse von Datensatz 3. Bei diesem Datensatz tritt keine Änderung des Parameters α auf, wodurch sich der Vorteil der Datenakkumulation zeigt. Mit einem absoluten Fehler⁵ von $\bar{e}_{Q50\%} = 0,126$ erzielt das Datenakkumulationsverfahren einen Fehler, der nahe am Optimum ist. Bei dem verwendeten künstlichen Rauschen wäre theoretisch ein Fehler von 0,123 möglich gewesen.

Bei der Vorstellung der Verfahren wurde bei allen Verfahren unterschieden zwischen einer Neuinitialisierung mit zufälligen Gewichten vor dem Training und der Nutzung der alten Gewichte als Startinitialisierung. Bei allen drei künstlichen Datensätzen zeigt sich die Nutzung der alten Gewichte als vorteilhaft gegenüber einer Neuinitialisierung. In Abbildung 5.9 ist der Verlauf des durchschnittlichen Fehlers beim Verfahren der Datenakkumulation einmal mit und einmal ohne Neuinitialisierung dargestellt.

Wie zu sehen ist, ist der Mittelwert des Approximationsfehlers bei Neuinitialisierung größer als bei Nutzung der alten Netzwerkgewichte. Zudem fällt auf, dass der Approximationsfehler bei Neuinitialisierung stärker schwankt als bei Nutzung der alten Gewichte. Dieses Verhalten kommt daher, dass bei jedem Nachtraining mit Neuinitialisierung die Chance besteht, zufällig eine sehr schlechte Initialisierung zu wählen, die beim Training zu einem schlechten lokalen Extrempunkt und

⁵Die absoluten Fehlerwerte sind in Anhang B.4 aufgeführt.

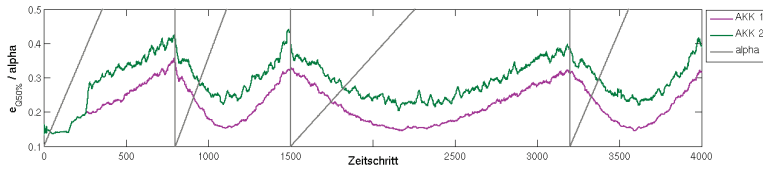


Abbildung 5.9. – Mittlerer Fehler des Adaptionverfahrens der Datenakkumulation auf dem künstlichen Datensatz 1. Deutlich ist der Unterschied zwischen Nutzung der alten Gewichte (AKK 1) und zufälliger Neuinitialisierung (AKK 2) im mittleren Approximationsfehler erkennbar.

einer schlechten Prognose führt. Beim nächsten Nachtraining besteht dann wieder die Chance, ein besseres Optimum zu finden. Da ein Nachtraining alle 20 Schritte durchgeführt wird, erklärt sich so die stärkere Schwankung des Approximationsfehlers.

Der Unterschied in der Höhe des Approximationsfehlers hingegen hat als Ursache das bei allen Versuchen verwendete Startmodell. Wie in Abschnitt 5.2 erläutert, wurde das Anfangsmodell N_{init} mehrfach auf den Startdaten trainiert und ein besonders gutes Modell ausgewählt. Dies soll dem Umstand Rechnung tragen, dass auch in der Realität dieses erste Modell meist per Hand mit besonderer Sorgfalt erstellt wird. Die Wahrscheinlichkeit, mit einem guten Modell zu starten, ist also relativ hoch. Während bei den Verfahren mit Nutzung der alten Gewichte dieses Startmodell sich auch in den weiteren Adaptionsschritten zu einem gewissen Teil wiederfindet, wird bei einer Neuinitialisierung das Startmodell nicht weiter genutzt, was bei allen drei künstlichen Datensätzen zu einem insgesamt schlechteren Ergebnis führt.

Das Problem, dass eine Neuinitialisierung zu einem schlechteren Modell führen kann, erklärt auch, warum der Ansatz BAWk bei den Datensätzen 1 und 2 am besten abschneidet. Dadurch, dass das neu trainierte Netz auf einem Teil der Testdaten noch einmal mit dem alten

Netz verglichen wird, kann zumindest zum Teil verhindert werden, dass ein Netz mit schlechterem Approximationsergebnis an die Stelle eines Netzes mit gutem Ergebnis tritt.

Zusammengefasst lässt sich feststellen, dass auf künstlichen Daten eine Nutzung der alten Gewichte stets vorteilhafter ist als eine Neuinitialisierung. Ändert sich die Prozessdynamik, dann ist ein Adaptionsverfahren mit Bereichsauswahl sinnvoll, andernfalls werden mit Datenakkumulation die besten Resultate erzielt.

5.3.5. Ergebnisse auf realen Daten aus Zementwerken

Die Tests mit Freikalkmodellen auf den Daten realer Zementwerke wurden nach demselben Schema wie die Tests auf künstlichen Daten durchgeführt. Die Ergebnisse sind in Tabelle 5.4 dargestellt. Statt 50 mal, wie bei den künstlichen Datensätzen, wurde jeder Versuch je 100 Mal mit zufälliger Initialisierung der Startmodelle wiederholt, da die Varianz der Ergebnisse auf den Daten der Zementwerke höher war als auf den künstlichen Daten. Bei jedem dieser 100 Versuche wurde das zufällig initialisierte Startmodell für alle Adaptionsmethoden eingesetzt um die Vergleichbarkeit der Ergebnisse zu erhöhen. Als Vergleichskriterium wurde erneut der Median gewählt.

Werden die Ergebnisse der Adaptionsverfahren auf realen Zementdaten mit den Ergebnissen der künstlichen Daten verglichen, ergeben sich deutliche Unterschiede. In 73% der Fälle werden mit Neuinitialisierung bessere Ergebnisse erzielt als durch Nutzung der alten Gewichte. Das Verfahren zur Adaption durch Datenakkumulation und Neuinitialisierung schneidet auf allen fünf untersuchten Zementwerken am besten ab. Verfahren der Bereichsauswahl schneiden hingegen schlecht ab und erzielen teilweise Ergebnisse die schlechter sind, als ein Verfahren ohne jegliche Adaption an neue Daten.

$\bar{\epsilon}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5	\bar{Zem}
ohne Adaption (OA)	0%	0%	0%	0%	0%	0%
Datenakkumulati- on (AKK) 1	-12,9%	-8,7%	-39,9%	-9,7%	-18,7%	-18,0%
Datenakkumulati- on (AKK) 2	-16,8%	-16,4%	-46,3%	-9,8%	-22,1%	-22,3%
Bereichsaus- wahl (BAW) 1	40,8%	-4,9%	-37,1%	0,9%	-14,8%	-3,0%
Bereichsaus- wahl (BAW) 2	9,0%	-9,3%	-35,7%	0,6%	-13,3%	-9,7%
Bereichsauswahl mit Kontrolle (BAWk) 1	72,1%	-8,3%	-37,7%	1,7%	-18,5%	1,8%
Bereichsauswahl mit Kontrolle (BAWk) 2	4,3%	-5,7%	-39,0%	-3,7%	-12,2%	-11,3%
Revision (Rev.)	3	4	0	2	3	

Tabelle 5.4. – Median $\bar{\epsilon}_{Q50\%}$ des Approximationsfehlers über je 100 Versuchen pro Verfahren und Zement-Datensatz. Das Verfahren ohne Adaption (OA) des Modells dient als Referenz. Es wird die prozentuale Verbesserung des Fehlers gegenüber diesem Verfahren dargestellt. Das Verfahren mit dem besten Resultat pro Datensatz wurde jeweils blau markiert. Verfahren die schlechter abschnitten als das Referenzverfahren, wurden rot markiert. Die Nummer hinter dem Namen des Verfahrens bezieht sich auf die Initialisierung des Modells bei dem jeweiligen Verfahren. Die Zahl 1 bezeichnet Verfahren, bei denen die Gewichte der alten Netze als Startwert genutzt werden, während die Zahl 2 für Verfahren mit neu initialisierten Modellen steht. Zusätzlich aufgeführt ist auch die Anzahl der Revisionen im Testzeitraum bei Zementwerken.

Wie ist dieser Unterschied zwischen künstlichen Daten und Zementdaten zu erklären? Ein Hinweis darauf ergibt sich, wenn man die Menge der verfügbaren Daten und die Komplexität der zu approximierenden Funktion vergleicht. Die künstliche Funktion $y = \alpha x_1 x_2 + x_3 + \alpha x_4 + x_1 x_5 + r$ enthält zwar Nichtlinearität und ein unbekanntes Rauschen, trotzdem ist sie vermutlich bedeutend einfacher aus den vorhandenen Inputkanälen x_1, \dots, x_5 abzuschätzen als die Approximation des Freikalkwertes aus Temperatur- und Kamerasignalen in einem Zementofen.

Bei den bisher durchgeführten Versuchen wurden jeweils 300 Samples für das Training eines Approximationsmodells verwendet. Möglicherweise sind für die komplexe Freikalkprognose aber wesentlich mehr Samples notwendig. Um dies zu überprüfen, wurden die Tests an Zementwerk 1 und 4, bei denen bisher die schlechtesten Ergebnisse bei Bereichsauswahl erzielt wurden, mit einer Anzahl von 600 Samples je Training wiederholt. Zum Vergleich wurde auch der Versuch auf künstlichen Daten mit Datensatz 1 mit 30 statt 300 Samples wiederholt. Die Ergebnisse sind in Tabelle 5.5 dargestellt.

$\bar{e}_{Q50\%}$	Zem1 (300)	Zem1 (600)	Zem4 (300)	Zem4 (600)	kD1 (30)	kD1 (300)
OA	0%	0%	0%	0%	0%	0%
OA	/ 0,51	/ 0,50	/ 0,70	/ 0,65	/ 0,57	/ 0,33
AKK 1	-12,9%	-11,7%	-9,7%	-2,3%	-55,7%	-35,4%
AKK 1	/ 0,45	/ 0,45	/ 0,63	/ 0,62	/ 0,25	/ 0,22
AKK 2	-16,8%	-15,0%	-9,8%	-3,5%	-38,7%	-15,5%
AKK 2	/ 0,43	/ 0,43	/ 0,64	/ 0,63	/ 0,35	/ 0,28
BAW 1	40,8%	12,6%	0,9%	-1,5%	-42,7%	-53,3%
BAW 1	/ 0,72	/ 0,57	/ 0,71	/ 0,64	/ 0,34	/ 0,16
BAW 2	9,0%	-4,5%	0,6%	-0,5%	22,2%	-17,6%
BAW 2	/ 0,56	/ 0,48	/ 0,71	/ 0,65	/ 0,70	/ 0,28
BAWk 1	72,1%	33,0%	1,7%	-1,7%	-43,8%	-53,6%
BAWk 1	/ 0,88	/ 0,67	/ 0,72	/ 0,64	/ 0,32	/ 0,16
BAWk 2	4,3%	3,8%	-3,7%	-1,7%	-16,9%	-50,6%
BAWk 2	/ 0,54	/ 0,52	/ 0,68	/ 0,64	/ 0,48	/ 0,17

Tabelle 5.5. – Prognosefehler Median $\bar{e}_{Q50\%}$ (absolut / relativ zu OA) auf den Daten zweier Zementwerke und eines künstlichen Datensatzes mit jeweils unterschiedlicher Anzahl an Datenpunkten pro Training.

Durch die Erhöhung der Anzahl der genutzten Samples pro Modell verringern sich der absolute Fehler des Ursprungsmodells und die Fehler aller Bereichsadaptionsverfahren. Der absolute Fehler der Akkumulationsverfahren hingegen ändert sich kaum. Betrachtet man die relativen Fehler der Modelle im Vergleich zum nicht adaptierten Ursprungsmodell, so verbessern sich deshalb alle Verfahren (bis auf die Akkumulationsverfahren) durch eine Hinzunahme von mehr Samples.

Dieser Effekt ist auch zu beobachten, wenn das Training auf künstlichen Daten mit 30 statt 300 Samples pro Modell durchgeführt wird. Der absolute Fehler der Akkumulationsverfahren bleibt ähnlich, der Fehler aller anderen Verfahren ist mit nur 30 Samples wesentlich schlechter.

Es zeigt sich also, dass für eine Auswahl des besten Adaptionsverfahrens zwei Kriterien von Bedeutung sind:

1. die Stärke der *concept drift* des Prozesses,
2. die Anzahl der verfügbaren Samples pro Zeiteinheit in Relation zur Komplexität der Funktion.

These 8 (Adaption einzelner Modelle und Datenmenge)

Je stärker die concept drift ist und je häufiger Änderungen im Prozess vorkommen, desto mehr Nutzen bringt ein Bereichsauswahl-Verfahren. Ist die concept drift hingegen schwach, die zu approximierende Funktion sehr komplex oder stehen pro Modell nur wenige Datenbeispiele zur Verfügung, dann werden bessere Resultate erzielt, wenn ein Verfahren der Datenakkumulation benutzt wird.

Aus praktischer Sicht ist dieses Resultat sehr unbefriedigend. Zu dem Zeitpunkt, an dem die Entscheidung über ein Adaptionsmodell getroffen werden muss, liegen im Allgemeinen nicht genügend Daten vor, um die Stärke der *concept drift* oder die Schwierigkeit der Funktionsapproximation abschätzen zu können⁶. Die bisherigen Ergebnisse legen nahe, dann das Verfahren der Datenakkumulation zu verwenden, da dieses bei den meisten Anlagen den geringsten Fehler erzielte. Datenakkumulationsverfahren haben aber den großen Nachteil, dass dafür alle bisherigen

⁶Bei allen in dieser Arbeit untersuchten Projekten lagen zum Zeitpunkt der Aufsetzung des Adaptionsverfahrens Daten im Bereich von ein bis zwei Monaten vor. Da Freikalkmessungen nur vier bis zehn Mal pro Tag durchgeführt werden, entspricht das den oben aufgeführten 300 Samples.

Daten für das Training genutzt werden müssen. Mit der Zeit sammeln sich so große Datenmengen an, dass sie ein Training immer zeit- und speicheraufwendiger machen⁷.

Verstärkt sich die concept drift, würde das Ergebnis des Akkumulationsverfahrens außerdem schnell schlechter und in extremen Fällen sogar unbrauchbar werden.

Im Folgenden soll deshalb untersucht werden, ob durch die Verwendung von mehreren Modellen ein Resultat erzielt werden kann, welches eine ähnliche Güte hat wie das Akkumulationsverfahren, ohne ein Aufheben sämtlicher Datensamples zum Nachtraining zu erfordern.

5.4. Adaptivität bei Verwendung eines Ensembles von Netzen

Bei den bisher vorgestellten Verfahren wurde stets nur ein einzelnes Modell genutzt. Eine Ausnahme bildet nur der Ansatz *Bereichsauswahl mit Kontrolle (BAWK)*, bei dem ein adaptiertes Modell auf einem Abschnitt der Testdaten noch einmal mit der Leistung des Vorgängermodells verglichen wird. Nachdem dieser Vergleich abgeschlossen war, wird im Weiteren aber nur das bessere Modell genutzt.

Es sind viele Situationen denkbar, in denen dieses Vorgehen nicht optimal ist. Beispielsweise unterscheidet sich die Situation im Kessel vor und nach einer Revision erheblich. Ein Modell wird auf einem Zustand *A* "nach einer Revision" trainiert und adaptiert sich dann im Laufe der Zeit an den Zustand *B* "normaler Betriebsmodus". Wird nun eine weitere Revision durchgeführt, dann befindet sich der Kessel wieder im Zustand *A*. Zu diesem Zeitpunkt wäre es sehr nützlich, wenn sofort das alte Modell für diesen Zustand aktiviert wird, statt erst langsam das aktuelle Modell an Zustand *A* anzupassen.

⁷Die Erfahrung aus der Praxis bei der Aufsetzung von Softsensor-Modellen in Zementwerken zeigen, dass im Allgemeinen nicht mehr als sechs Monate bzw. 500-1000 Samples in der Datenbank zur Verfügung stehen und zur Adaption genutzt werden können.

Mit Hilfe von Ensembles ist ein solches Vorgehen möglich. In [Protzel et al., 2000] stellt der Autor beispielsweise den Ansatz *Incremental Cluster Evolving* vor, mit dem durch mehrere neuronale RBF-Netze die *concept drift* eines künstlichen Datensatzes ausgeglichen wird. *Learn++.NSE* [Elwell and Polikar, 2011] erzielt ein ähnliches Resultat, wobei MLPs statt RBF-Netzen als Teile des Ensembles genutzt werden.

In dieser Arbeit wird unterschieden zwischen zwei Arten, Ensembles zu nutzen:

1. **Ensemble Selection:** Es existiert ein Pool von Modellen. Auf Testdaten der Vergangenheit (offline) können mehrere Modelle dieses Pools getestet werden, für die aktuelle Datensituation (online) wird aber nur ein einzelnes Modell genutzt.
2. **Ensemble Weighting:** Es existiert ein Pool von Modellen. Sowohl offline als auch online können mehrere Modelle des Pools gleichzeitig genutzt werden.

Im Folgenden werden beide Konzepte erläutert und ihre Ergebnisse auf den Testdaten vorgestellt.

5.4.1. Ensemble Selection

Wie bei allen Ensemble-Verfahren wird auch bei *Ensemble Selection* ein Pool von Modellen statt eines einzelnen Modells verwendet. Dieser Pool P besteht aus allen Modellen, die bis zum aktuellen Zeitpunkt trainiert wurden. Sollte der Pool mit der Zeit zu groß für die verfügbaren Ressourcen (Speicherplatz, Rechenkapazität) werden, so müssen Modelle aus dem Pool entfernt werden.

Bei *Ensemble Selection* wird aus diesem Modellpool P in regelmäßigen Zeitabständen ein Modell N_{adapt}^t ausgewählt, welches dann bis zur nächsten Überprüfung aktiv ist. Um festzustellen, welches Modell am besten geeignet ist, werden sämtliche Modelle des Modellpools

auf den letzten verfügbaren Daten $S_{teil,test}$ getestet. Das Modell, das auf diesen Daten den geringsten Fehler $e(N_i, S_{teil,test})$ hat, wird als aktives Modell ausgewählt. In Algorithmus 5 und Abbildung 5.10 ist die Funktionsweise dieses Algorithmus dargestellt.

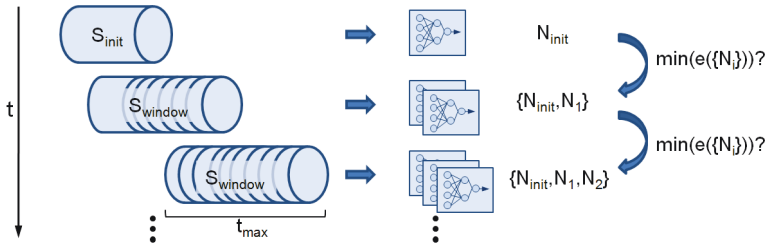


Abbildung 5.10. – Verfahren der *Ensemble Selection* zur Anpassung eines Modells an neue Daten unter Nutzung eines Pools von vorhandenen Modellen.

Wie bei allen bisherigen Verfahren auch, kann beim Training des Modells entweder ein altes Modell als Basis verwendet werden oder ein komplett neu initialisiertes Modell trainiert werden. Für eine Initialisierung mit Hilfe eines alten Modells stehen nun allerdings alle Modelle des Pools zu Auswahl. Mögliche Strategien sind:

- das bisher aktive Modell nutzen
- das Modell mit dem besten Testfehler auf den neuen Daten nutzen
- das erste Modell nutzen.

Für die rechenintensiven Tests in dieser Arbeit wurde die erste Strategie gewählt.

Algorithmus 5 Ensemble Selection

Eingabe: Datensatz S_{init} , Modell N_{init} , Modellpool P , Wartezeit für Nachtraining t_{min} , maximaler Zeitversatz für Trainingsdaten $t_{max,train}$, maximaler Zeitversatz für Testdaten $t_{max,test}$

Ausgabe: adaptiertes Modell N_{adapt}^t zum Zeitpunkt t

```

 $S_{window} \leftarrow S_{init}$ 
 $p \leftarrow 0$ 
 $P \leftarrow \{N_{init}\}$ 
while 1 do
   $t \leftarrow time()$ 
   $s_t \leftarrow observation(t)$ 
   $S_{window} \leftarrow S_{window} \cup s_t$ 
   $p \leftarrow p + 1$ 
  if  $p > t_{min}$  then
     $p \leftarrow 0$ 
     $S_{teil,train} \leftarrow \{s_j | s_j \subseteq S_{window}, t_{max,test} < (t - t(s_j)) < t_{max,train}\}$ 
     $S_{teil,test} \leftarrow \{s_j | s_j \subseteq S_{window}, (t - t(s_j)) < t_{max,test}\}$ 
     $N_{tmp}^t \leftarrow train(N_{adapt}^t, S_{teil,train})$ 
     $P \leftarrow P \cup \{N_{tmp}^t\}$  // füge Modell dem Pool hinzu
    for all  $N_i \in P$  do
       $e_i = e(N_i, S_{teil,test})$  // bestimme Testfehler aller Pool-Modelle
    end for
     $N_{adapt}^t \leftarrow \{N_j | N_j \in P, e_j < e_i \forall i, i \neq j\}$  // wähle bestes Modell
     $S_{window} \leftarrow S_{teil,train} \cup S_{teil,test}$ 
  end if
end while

```

Der Vorteil des *Ensemble Selection* Verfahrens besteht darin, dass in einem Fall in dem sich verschiedene Systemzustände abwechseln, schnell auf ein passendes Modell umgeschaltet werden kann. Trotzdem wird mit diesem Ansatz nicht das volle Potential von Ensembles genutzt. Es wird zwar das beste Modell aus einem Pool genutzt, aber es werden nicht die Fähigkeiten mehrerer Modelle kombiniert, um insgesamt eine bessere Leistung zu erzielen als mit jedem Einzelmodell möglich wäre. Das folgende Verfahren soll diesen Nachteil ausgleichen.

5.4.2. Ensemble Weighting

Bei *Ensemble Selection* wurde ein Pool an Modellen angelegt, von denen stets nur ein einzelnes Modell aktiv war. Ein wesentlicher Vorteil von Ensembles besteht aber darin, die Ausgaben mehrerer Modelle gleichzeitig zu nutzen und dadurch ein Ergebnis zu erhalten, das sicherer und besser ist, als die Ausgabe eines einzelnen Modells.

Aufbauend auf dem im vorangegangenen Abschnitt vorgestellten Verfahren *Ensemble Selection* lässt sich sehr schnell ein Verfahren konstruieren, das nicht nur ein Modell nutzt. Wie bei *Ensemble Selection* wird ein Pool P von Modellen aufgebaut, in dem in regelmäßigen Abständen neue Modelle trainiert und dem Pool hinzugefügt werden. Ebenso wird wie bei *Ensemble Selection* in regelmäßigen Abständen der Fehler $e(N_i, S_{teil, test})$ aller Modelle N_i des Pools auf den aktuellen Daten $S_{teil, test}$ bestimmt. Dieser Fehler wird anschließend genutzt, um ein Gewicht $\zeta(N_i)$ für jedes Modell zu berechnen. Die Gewichte sind dabei umgekehrt proportional zum Fehler des Modells $\zeta(N_i) \sim \frac{1}{e(N_i, S_{teil, test})}$ und summieren sich zu eins auf: $\sum_{i=1}^n \zeta(N_i) = 1$.

Die Ausgabe des Verfahrens *Ensemble Weighting* entspricht dann der gewichteten Summe aller Einzelmodelle, die bis zum bisherigen Zeitpunkt im Modellpool vorhanden sind:

$$y^{Pool} = \sum_{i=1}^n \zeta(N_i) \cdot y^{N_i}, N_i \in P \quad (5.3)$$

Auf diese Weise werden alle Modelle in die Berechnung der Ausgabe mit einbezogen. Der Algorithmus des Trainings und der Verwaltung der Modelle ist identisch zu dem in Algorithmus 5 vorgestellten Verfahren, einzig die Ausgabe des Gesamtmodells ändert sich.

Ein Nachteil dieses Verfahrens wird schnell ersichtlich. Während bei den bisherigen Verfahren nur ein Modell online lief, sind nun zu jedem Zeitpunkt alle bisher trainierten Modelle aktiv. Werden keine Modelle aus dem Pool entfernt, und regelmäßig neue Modelle trainiert, ist es nur eine Frage der Zeit, bis die Rechenleistung nicht mehr ausreicht, um für jeden Onlineschritt eine Ausgabe aller Modelle zu berechnen. Ein weiterer Nachteil besteht darin, dass auch schlechte Modelle in die Ausgabe einbezogen werden. Diese erhalten zwar nur ein sehr kleines Gewicht, können aber trotzdem das Ergebnis verschlechtern.

Ein einfacher Ansatz um beide Probleme zu umgehen, besteht darin, nicht alle Modelle in die Ausgabe einzubeziehen, sondern nur die m besten Modelle. In bestimmten Abständen wird weiterhin für jedes Modell des Gesamtpools P der Fehler auf den Testdaten $S_{teil,test}$ bestimmt. Anschließend werden die m besten Modelle herausgesucht, die dann den Arbeitspool $P^m \subset P$ bilden. Die Gewichte $\zeta(N_i)$ der Modelle im Arbeitspool sind proportional zum Fehler des jeweiligen Modells auf den Testdaten. Die Gesamtausgabe des Verfahrens berechnet sich wie folgt:

$$y^{Pool_m} = \sum_{i=1}^m \zeta(N_i) \cdot y^{N_i}, N_i \in P^m \quad (5.4)$$

In der Praxis lässt sich dieses Verfahren, das im Folgenden als *Ensemble Weighting* ^{m} bezeichnet wird, weit besser einsetzen, als das normale *Ensemble Weighting*, da nur einige Modelle gleichzeitig im Einsatz sein müssen. In Abbildung 5.11 ist der Unterschied der Gewichte beider *Ensemble Weighting* Verfahren an einem Beispiel mit Daten aus einem Zementwerk dargestellt.

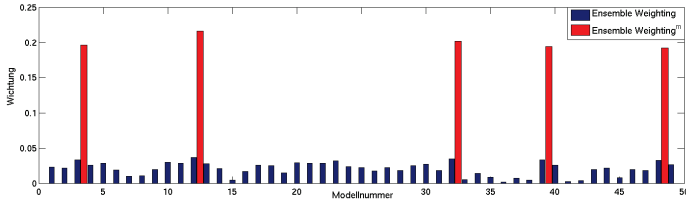


Abbildung 5.11. – Vergleich der Gewichte für die beiden vorgestellten *Ensemble Weighting* Verfahren auf den Daten von Zementwerk 1. Während bei *Ensemble Weighting* alle Modelle mit einem Gewicht in die Berechnung der Ausgabe eingehen, erhalten bei *Ensemble Weighting^m* nur die besten m Modelle ein Gewicht $\zeta(N_i) > 0$. Da die Summe der Gewichte gleich 1 sein muss, entstehen bei *Ensemble Weighting^m* größere Gewichte.

5.4.3. Ergebnisse auf künstlichen Daten

Die bisher vorgestellten Verfahren wurden mit den in Kapitel 5.2 erläuterten Datensätzen mit dem gleichen Verfahren und den gleichen Parametern wie bereits in Abschnitt 5.3.4 getestet. In Tabelle 5.6 sind die Ergebnisse der einzelnen Adaptions-Verfahren dargestellt. Zum Vergleich ist ebenfalls das Ergebnis des Verfahrens Datenakkumulation 2 aus dem letzten Abschnitt dargestellt, welches bisher das beste Ergebnis auf den Daten der Zementwerke erzielen konnte.

Mit Ausnahme von Ensemble Weighting erzielen alle Ensemble-Verfahren eine Verbesserung des Ergebnisses gegenüber dem Ansatz ohne Adaption. Ebenso wie die Verfahren zur Bereichsauswahl ist auch bei den Ensemble-Verfahren der Ansatz des Weitertrainings mit alten Gewichten auf künstlichen Daten besser als der Ansatz der Neuintialisierung. Der Unterschied zwischen den beiden Verfahren ist bei Ensembles (mit Ausnahme des Verfahrens EW 2) allerdings deutlich geringer als bei Bereichsverfahren.

$\bar{\epsilon}_{Q50\%}$	kD1	kD2	kD3	$k\bar{D}$
ohne Adaption (OA)	0%	0%	0%	0%
Datenakkumulation (AKK) 2	-5,9%	-15,5%	53,3%	10,7%
Ensemble Selection (ES) 1	-37,9%	-50,3%	-2,2%	-30,1%
Ensemble Selection (ES) 2	-37,3%	-49,7%	-1,5%	-29,5%
Ensemble Weighting (EW) 1	-25,2%	-31,0%	0,7%	-18,5%
Ensemble Weighting (EW) 2	57,2%	57,7%	234,8%	116,6%
Ensemble Weighting mit m Modellen (EW _m) 1	-38,2%	-50,0%	-4,4%	-30,9%
Ensemble Weighting mit m Modellen (EW _m) 2	-37,3%	-49,1%	-3,7%	-30,0%

Tabelle 5.6. – Median $\bar{\epsilon}_{Q50\%}$ des Approximationsfehlers über je 50 Versuchen pro Verfahren und künstlichem Datensatz. Das Verfahren ohne Adaption (OA) des Modells dient als Referenz. Es wird die prozentuale Verbesserung des Fehlers gegenüber diesem Verfahren dargestellt. Das Verfahren mit dem besten Resultat pro Datensatz wurde jeweils blau markiert. Verfahren, die schlechter abschnitten als das Referenzverfahren, wurden rot markiert. Die Nummer hinter dem Namen des Verfahrens bezieht sich auf die Initialisierung des Modells bei dem jeweiligen Verfahren. Die Zahl 1 bezeichnet Verfahren, bei denen die Gewichte der alten Netze als Startwert genutzt werden, während die Zahl 2 für Verfahren mit neu initialisierten Modellen steht.

Da sich bei den Bereichsadaptionsverfahren bereits zeigte, dass die Ergebnisse auf realen Daten teilweise von den künstlichen Daten abweichen, müssen vor einer endgültigen Aussage zur Qualität der Verfahren die Ergebnisse auf realen Daten berücksichtigt werden.

5.4.4. Ergebnisse auf realen Daten aus Zementwerken

Die Tests auf den Daten realer Zementwerke für das Ziel der Freikalk-Approximation wurden mit den gleichen Daten und Parametern wie bereits in Abschnitt 5.3.5 getestet. In Tabelle 5.7 sind die Ergebnisse der einzelnen Adaptions-Verfahren und des Verfahrens Datenakkumulation 2 aus dem letzten Abschnitt dargestellt.

$\bar{e}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5	\bar{Zem}
ohne Adaption (OA)	0%	0%	0%	0%	0%	0%
Datenakkumulation (AKK) 2	-16,8%	-16,4%	-46,3%	-9,8%	-22,1%	-22,3%
Ensemble Selection (ES) 1	-7,3%	-7,9%	-42,5%	-3,1%	-12,6%	-14,7%
Ensemble Selection (ES) 2	-7,4%	-8,9%	-44,7%	-3,3%	-14,0%	-15,7%
Ensemble Weighting (EW) 1	-12,1%	-13,4%	-38,0%	-7,8%	-18,4%	-18,0%
Ensemble Weighting (EW) 2	3,9%	-0,3%	-36,9%	-4,6%	-21,0%	-11,8%
Ensemble Weighting mit m Modellen (EWm) 1	-18,6%	-13,4%	-44,8%	-10,2%	-18,8%	-21,2%
Ensemble Weighting mit m Modellen (EWm) 2	-18,0%	-14,8%	-45,2%	-10,5%	-19,6%	-21,6%
Revision (Rev.)	3	4	0	2	3	

Tabelle 5.7. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 100 Versuchen pro Verfahren und Zement-Datensatz. Das Verfahren ohne Adaption (OA) des Modells dient als Referenz. Es wird die prozentuale Verbesserung des Fehlers gegenüber diesem Verfahren dargestellt. Das Verfahren mit dem besten Resultat pro Datensatz wurde jeweils blau markiert. Verfahren die schlechter abschnitten als das Referenzverfahren, wurden rot markiert. Die Nummer hinter dem Namen des Verfahrens bezieht sich auf die Initialisierung des Modells bei dem jeweiligen Verfahren. Die Zahl 1 bezeichnet Verfahren, bei denen die Gewichte der alten Netze als Startwert genutzt werden, während die Zahl 2 für Verfahren mit neu initialisierten Modellen steht. Zusätzlich aufgeführt sind auch die Anzahl der Revisionen im Testzeitraum bei Zementwerken.

Mit Ausnahme von Ensemble Weighting 2 erzielen alle Ensemble-Verfahren eine Verbesserung des Ergebnisses gegenüber dem Ansatz ohne Adaption. Das ist insofern bemerkenswert, da auch bei Zementwerk 1 und 4 eine Verbesserung erzielt werden kann. Bei diesen Zementwerken und einer Größe des Trainings-Datensatzes von 300 Samples konnten die Bereichsauswahl-Verfahren des vorherigen Abschnittes nur sehr schlechte Resultate erzielen und verloren deutlich gegenüber dem Verfahren der Datenakkumulation. Obwohl die Model-

le der Ensemble Verfahren ebenfalls jedes für sich nur 300 Samples zum Training verwenden, wird durch die Kombination der einzelnen Modelle in einem Ensemble ein deutlich besseres Resultat erzielt. Im Fall von Zementwerk 1 und 4 ist dieses Resultat sogar besser als das Verfahren der Datenakkumulation, im Fall der anderen Zementwerke zumindest ähnlich gut.

Ensemble Weighting^m erzielt dabei deutlich bessere Resultate als *Ensemble Weighting* oder *Ensemble Selection*. Der Grund hierfür liegt in der Verbindung der Ansätze der Selektion von Modellen und der Kombination von Modellen. Bei *Ensemble Selection* werden schlechte Modelle des Modellpools nicht genutzt, sondern es wird nur die Ausgabe des besten Modells weiterverarbeitet. Bei *Ensemble Weighting* werden die Ergebnisse aller Modelle kombiniert, allerdings werden schlechte Modelle nicht davon ausgenommen. *Ensemble Weighting^m* sortiert die schlechten Modelle aus und kombiniert gleichzeitig die Ausgaben guter Modelle. Vergleicht man nicht nur den Mittelwert, sondern auch die Standardabweichung⁸ des Prognosefehlers, so ist zu sehen, dass sich dadurch Freikalkprognosen mit einem geringen Fehler ergeben, welcher auch nur einer kleinen Schwankung unterliegt.

These 9 (*Adaption von Modell-Ensembles*)

Kann statt eines einzelnen Modells ein Pool von Modellen verwendet werden, so ist es vorteilhaft, nur die besten Modelle des Pools zu kombinieren. Die dadurch erstellte Approximation ist besser als die Approximation des besten Modells und besser als die linear gewichtete Kombination aller Modelle.

Ensemble Weighting^m ist außerdem das Verfahren, bei dem der Unterschied zwischen einer Initialisierung des Modells mit alten Daten und einer Neuinitialisierung am wenigsten Auswirkungen

⁸Die Standardabweichungen der einzelnen Verfahren sind in Anhang B.4 aufgeführt.

auf das Ergebnis hat. Im Durchschnitt aller Zementwerke unterschieden sich die Ergebnisse beider Varianten nur um 0,7%. Beim Verfahren der Datenakkumulation sind es hingegen 4,27%. Dadurch ist es weniger gravierend, falls bei der Aufsetzung des Adaptionsverfahrens der falsche Ansatz zur Initialisierung der Modelle gewählt wird.

Durch diese Vorteile wird *Ensemble Weighting*^m zu einer guten Alternative zum Verfahren der Datenakkumulation. Es ist besonders dann von Vorteil, wenn nicht genügend Ressourcen zur Verfügung stehen, um alle bisher beobachteten Daten zum Training zu nutzen oder falls eine starke concept drift erwartet wird, welche zu Inkonsistenzen auf großen Datenmengen führen würde.

Im vorherigen Kapitel wurde ein zusätzlicher Versuch durchgeführt um zu bestimmen, wie sich größere Trainings-Datensätze auf die Resultate auswirken. Dieser Versuch wurde auch bei den Ensemble-Verfahren durchgeführt. Die Ergebnisse sind in Tabelle 5.8 dargestellt.

$\bar{e}_{Q50\%}$	Zem1 (300)	Zem1 (600)	Zem4 (300)	Zem4 (600)	kD1 (30)	kD1 (300)
OA	0% / 0,51	0% / 0,50	0% / 0,70	0% / 0,65	0% / 0,57	0% / 0,33
AKK 2	-16,8% / / 0,43	-15,0% / / 0,43	-9,8% / 0,64	-3,5% / 0,63	-38,7% / / 0,35	-15,5% / / 0,28
ES 1	-7,2% / 0,48	-14,0% / / 0,44	-3,1% / 0,68	-0,8% / 0,64	-23,9% / / 0,44	-50,3% / / 0,17
ES 2	-7,4% / 0,47	-12,5% / / 0,44	-3,3% / 0,68	-0,6% / 0,64	-18,8% / / 0,47	-49,7% / / 0,17
EW 1	-12,1% / / 0,45	-14,2% / / 0,43	-7,8% / 0,65	-3,7% / 0,62	-24,8% / / 0,43	-31,0% / / 0,23
EW 2	3,9% / 0,53	1,8% / 0,52	-4,6% / 0,67	2,8% / 0,67	4,2% / 0,60	57,7% / 0,53
EW _m 1	-18,6% / / 0,42	-19,0% / / 0,41	-10,2% / / 0,63	-4,2% / 0,62	-36,0% / / 0,37	-50,0% / / 0,17
EW _m 2	-18,0% / / 0,42	-19,8% / / 0,41	-10,5% / / 0,63	-4,8% / 0,62	-31,1% / / 0,40	-49,1% / / 0,17

Tabelle 5.8. – Prognosefehler Median $\bar{e}_{Q50\%}$ (absolut / relativ zu OA) auf den Daten zweier Zementwerke und eines künstlichen Datensatzes mit jeweils unterschiedlicher Anzahl an Datenpunkten pro Training.

Ebenso wie die Verfahren der Bereichsauswahl profitieren die Ensemble-Verfahren von einem größeren Trainingsdatensatz. Der absolute Fehler aller Verfahren wird kleiner. Gleichzeitig sinkt allerdings auch der Fehler des Basisverfahrens ohne Adaption, weshalb die relativen Fehler im Vergleich zum Basisverfahren gleich bleiben. Die Unterschiede zwischen den Ergebnissen mit dem kleinen und dem großen Trainingsdatensatz sind aber wesentlich geringer als beim Ansatz der Bereichsauswahl. *Ensemble Weighting^m* bleibt, unabhängig von der Größe des Trainingsdatensatzes, das beste Verfahren.

5.5. Übertragbarkeit auf Müllverbrennungsanlagen

Die in den letzten Kapiteln vorgestellten Ergebnisse wurden auf den Daten von Zementwerken mit dem Ziel des Freikalkwertes erzielt. Es stellt sich die Frage, ob die dargestellten Ergebnisse sich auch auf Müllverbrennungsanlagen übertragen lassen. Um diese Frage zu klären wurden die bisher vorgestellten Verfahren ebenfalls auf den Daten von zwei Müllverbrennungsanlagen angewandt. Zielwert war in beiden Fällen die Dampfleistung, bei der eine Prognose mit einem Zeithorizont von zwei Minuten erstellt wurde. Als Qualitätskriterium gilt die Abweichung des prognostizierten Dampfleistungswertes vom zwei Minuten später auftretenden tatsächlichen Dampfleistungswert.

Neben dem Unterschied zwischen den beiden Prozessen der Müllverbrennung und der Zementherstellung und den damit einhergehenden unterschiedlichen Messstellen unterscheiden sich die Datensätze auch durch die Anzahl der Messwerte: Freikalkwerte im Zement werden nur 4-12 Mal pro Tag bestimmt. Dampfleistungswerte der Müllverbrennung hingegen sind sekundlich verfügbar, so dass sich 86400 Datenpunkte pro Tag ergeben. Um trotz dieser Menge an Daten Versuche zur Adaption durchführen zu können, wurde in dieser Arbeit nur jeder hundertste Datenpunkt zum Datensatz hinzugefügt.

Weiterhin wurden nur 40 Wiederholungen der Versuche statt der bei Zementdaten üblichen 200 Wiederholungen durchgeführt. Die Ergebnisse der unterschiedlichen Adaptionverfahren mit den Daten der beiden untersuchten Müllverbrennungsanlagen sind in Tabelle 5.9 dargestellt.

$\bar{e}_{Q50\%}$	MVA1	MVA2	\overline{MVA}	\overline{Zem}
ohne Adaption (OA)	0%	0%	0%	0%
Datenakkumulation (AKK) 1	-11,0%	-17,4%	-15,4%	-18,0%
Datenakkumulation (AKK) 2	-11,9%	-18,4%	-16,3%	-22,3%
Bereichsauswahl (BAW) 1	-1,3%	-12,0%	-8,6%	-3,0%
Bereichsauswahl (BAW) 2	-1,7%	-12,2%	-8,8%	-9,7%
Bereichsauswahl mit Kontrolle (BAWk) 1	-3,0%	-16,4%	-12,1%	1,8%
Bereichsauswahl mit Kontrolle (BAWk) 2	0,0%	-2,4%	-1,6%	-11,3%
Ensemble Selection (ES) 1	-3,4%	-9,4%	-7,5%	-14,7%
Ensemble Selection (ES) 2	0,0%	-8,2%	-5,6%	-15,7%
Ensemble Weighting (EW) 1	-8,1%	-13,0%	-11,4%	-18,0%
Ensemble Weighting (EW) 2	64,0%	21,4%	35,1%	-11,8%
Ensemble Weighting mit m Modellen (EWm) 1	-9,3%	-13,2%	-12,0%	-21,2%
Ensemble Weighting mit m Modellen (EWm) 2	-8,5%	-12,6%	-11,3%	-21,6%
Revision (Rev.)	2	1		

Tabelle 5.9. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 40 Versuchen pro Verfahren und MVA-Datensatz. Das Verfahren ohne Adaption des Modells OA dient als Referenz.

Auch bei Müllverbrennungsanlagen liefern die Verfahren der Datenakkumulation und *Ensemble Weighting*^m die besten Resultate. Im Gegensatz zu Zementanlagen, wo die Anwendung des Akkumulationsverfahrens tatsächlich eine Option ist⁹, kann das Akkumulationsverfahren auf Müllverbrennungsdaten nicht in der Praxis angewandt werden. Bei sekundlicher Abtastrate kommen innerhalb von zwei Jahren so viele Daten zusammen, dass eine Adaption des Netzes mit allen vorhandenen Daten mit der verfügbaren Rechentechnik nicht möglich ist.

⁹Vorausgesetzt es ist eine Datenbank mit entsprechend vielen Daten vorhanden.

Von den anderen Verfahren erzielt das Verfahren der Bereichsauswahl mit Kontrolle mit 12,1% Verbesserung ein geringfügig besseres Resultat als *Ensemble Weighting^m* 1. Da bei *Ensemble Weighting^m* die Verbesserung auf beiden Anlagen auftrat, bei BAWk nur bei einer Anlage, und *Ensemble Weighting^m* unabhängig von der verwendeten Initialisierungsmethode ist, ist es trotzdem vorteilhaft *Ensemble Weighting^m* als Adaptionsverfahren zu benutzen.

5.6. Fazit

In diesem Kapitel wurden Verfahren vorgestellt, mit denen ein lernfähiges System an sich verändernde Prozesse angepasst wird. Zuerst wurde erklärt, wie solche Veränderungen in industriellen Verbrennungsprozessen zustande kommen. In der Literatur sind Veränderungen dieser Art als *concept drift* bekannt. Anschließend wurde erläutert, dass ein einfaches Nachtraining eines lernfähiges Modells auf neuen Daten nicht ausreicht um das Modell zu adaptieren, da aufgrund von *catastrophic interference* bereits gelernte Zusammenhänge durch neue Daten verdrängt werden können und so sogar ein schlechteres Verhalten des adaptierten Modells möglich ist.

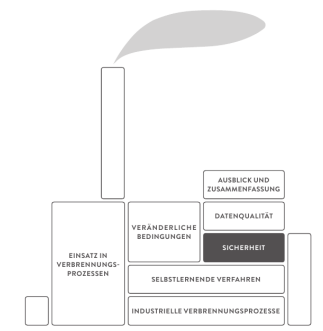
Anschließend wurden verschiedene Verfahren vorgestellt, mit denen ein Modell an sich verändernde Prozesse angepasst werden kann. Diese Verfahren wurden, je nach Anzahl der aktiven Modelle, der Gruppe Einzel-Adaptions-Verfahren oder der Gruppe der Ensemble-Verfahren zugeordnet. Alle Verfahren wurden auf den Datensätzen von fünf Zementwerken, zwei Müllverbrennungsanlagen und drei künstlichen Datensätzen getestet. Aus dem Bereich der Einzel-Adaptions-Verfahren erzielte das Verfahren der Datenakkumulation die besten Ergebnisse. Alle anderen Einzel-Adaptions-Verfahren schnitten deutlich schlechter ab und erzielten teilweise sogar schlechtere Resultate als ein nicht adaptiertes Modell. Es konnte allerdings gezeigt werden, dass diese Verschlechterung sehr stark von der Menge der Trainingsdaten pro Einzelmodell abhängt und verringert werden kann, wenn mehr Daten

zur Verfügung stehen.

In der Gruppe der Ensemble-Verfahren konnten mit *Ensemble Weighting^m* die besten Ergebnisse erzielt werden. Durch die Kombination mehrerer Modelle zu einer Prognose und den gleichzeitigen Ausschluß von schlechten Modellen ist dieses Verfahren den anderen Ensemble-Verfahren überlegen. Die Ergebnisse von *Ensemble Weighting^m* sind insgesamt zwar etwas schlechter als die Ergebnisse des Verfahrens der Datenakkumulation, da Datenakkumulation aber bei großen Datenmengen praktisch oft nicht anwendbar ist, stellt *Ensemble Weighting^m* eine gute Alternative zur Adaption von Modellen an Prozessveränderungen dar. Eine Auswahl dieser Ergebnisse und der verwendeten Verfahren wurde in [Steege and Groß, 2012] publiziert.

In diesem Kapitel wurden vor allem die theoretischen Grundlagen der kontinuierlichen Adaption von Prozessmodellen an veränderliche Prozesse erläutert. Anhand von künstlichen Datensätzen wurden die Merkmale verschiedener Adaptionmethoden analysiert und anschließend *offline* auf den Daten realer Zementwerke und Müllverbrennungsanlagen das beste Verfahren für die Anwendung dieser Arbeit bestimmt. Die beiden Verfahren *Ensemble Selection* und *Ensemble Weighting^m* wurden nach Abschluß der Analyse in die Software der Firma STEAG Powitec GmbH integriert und im Betrieb in Zementwerken und Müllverbrennungsanlagen getestet. Die Ergebnisse dieser *Online*-Tests finden sich zusammen mit den anderen *Online*-Ergebnissen in Kapitel 7.

6. Funktionale Sicherheit des lernenden Systems



Soll ein lernfähiges Verfahren in der Feuerleistungsführung eines industriellen Verbrennungsprozesses eingesetzt werden, spielt der Aspekt der funktionalen Sicherheit des Verfahrens eine enorm wichtige Rolle.

Ein konventioneller Regler wird von einem Experten konfiguriert und mit festen Parametern versehen. Im Unterschied dazu lernt ein adaptives System seine Parameter aufgrund der vorhandenen Daten und verändert diese Parameter. Während die Funktionalität des konventionellen Systems also stets gleich bleibt, kann sich die Funktionalität des lernfähigen Systems ändern. Diese Änderung kann positive Auswirkungen haben, wenn das System sich dem Prozess anpasst und stets optimal arbeitet, sie kann aber auch negative Auswirkungen bis hin zum Ausfall der Anlage haben. In diesem Abschnitt wird dargestellt, welche Möglichkeiten es gibt, die Funktionalität des lernfähigen Systems zu prüfen und welche Maßnahmen während des Lernvorgangs ergriffen werden können, um sicher zu stellen, dass keine gravierenden Fehler im Lernprozess auftauchen.

Zuerst erfolgt dazu eine Definition von Sicherheit im Kontext der Feuerleistungsführung durch lernfähige Systeme in Abschnitt 6.1. Sicherheitsanforderungen an ein lernfähiges System unterscheiden sich je nach der Art des Einsatzes des Systems. Kritisch sind dabei vor allem lernfähige Systeme, die direkt Stellgrößen beeinflussen. Warum gerade diese Systeme auf Daten geregelter Prozesse häufig ein fehlerhaftes Lernverhalten haben und welche Maßnahmen dagegen ergriffen werden können, wird in Abschnitt 6.2 erläutert.

Eine weitere Besonderheit sind selten auftretende aber wichtige Zustände. Die Behandlung dieser Zustände wird in Abschnitt 6.3 erläutert. Anschließend wird in Abschnitt 6.4 erläutert, warum Informationen über die Stellgröße beim Lernvorgang beachtet werden müssen und wie nach Abschluss des Lernvorgangs eine Überprüfung stattfinden kann.

Da nicht vollkommen ausgeschlossen werden kann, dass trotz dieser Maßnahmen ein Lernvorgang fehlschlägt, wird in Abschnitt 6.5 erläutert, wie durch die Kombination mehrerer Regler in Ensembles die Auswirkungen eines fehlerhaften Lernvorganges auf das Gesamtsystem minimiert werden kann.

Die Ergebnisse des Kapitels werden schließlich in Abschnitt 6.6 zusammengefasst.

6.1. Prozessregelung und funktionale Sicherheit

Sicherheit ist selbst im Kontext informationsverarbeitender Systeme ein relativ weit gefasster Begriff. Im Englischen wird unterschieden zwischen *safety*, welche den Schutz der Umgebung vor einem System beinhaltet, und *security*, welche den Schutz des Systems vor äußeren Einflüssen (z.B. Viren oder Hackerangriffe) darstellt. In dieser Arbeit wird ausschließlich Sicherheit im Sinne von *safety* behandelt. Die Norm IEC 61508 nennt eine solche Sicherheit “funktionale Sicherheit“ und definiert sie als “Freiheit von unvertretbaren Risiken“ eines technischen Systems.

Im Fokus dieser Arbeit steht die Veränderung einzelner Komponenten eines Regelungssystems innerhalb einer digitalen Feuerleistungsführung. Eine solche Feuerleistungsführung wurde konzipiert, um die Regelgrößen des Verbrennungsprozesses so zu regeln, dass der Prozess der Verbrennung innerhalb definierter Grenzen abläuft und der Zweck der Anlage erfüllt werden kann. Daraus leitet sich für diese Arbeit folgende Definition von Sicherheit ab:

Definition 6.1.1 (*Sicherheit eines Feuerleistungssystems*)

Ein Feuerleistungssystem ist sicher, wenn es die angesteuerten Stellgrößen so stellt, dass der Prozess der Verbrennung stets innerhalb vorgegebener Grenzen abläuft.

Eine vollkommene Sicherheit kann dabei nie garantiert werden. Technische Defekte¹, Bedienfehler oder außergewöhnliche Belastungen²

¹Fällt beispielsweise der Motor des Zuteilers in einer MVA aus, kann kein Brennstoff auf den Rost gelangen, und die Verbrennung wird stoppen.

²Enthält der verbrannte Hausmüll z.B. große Mengen heizwertreichen Materials (z.B. Fett oder Öl) und gelangt dieses Material unvermischt auf den Rost, so lässt sich ein starker Anstieg der Kesseltemperatur oft durch keinen Stelleingriff verhindern.

können ein Verbleiben des Prozesses innerhalb definierter Grenzen unmöglich machen. Die Feuerleistungsführung sollte die zur Verfügung stehenden Stellgrößen allerdings stets so stellen, dass solche Grenzverletzungen möglichst selten vorkommen und Dauer und Ausprägung der Übertretung der Grenze möglichst klein sind.

Im Bereich der Regelungstechnik ist im Zusammenhang mit Sicherheit der Begriff der Stabilität von großer Bedeutung. In [Lutz and Wendt, 2002] wird explizit definiert: “Voraussetzung für die technische Anwendbarkeit einer Regelung ist die Stabilität des Regelkreises”³.

In Abbildung 6.1 ist am Beispiel von drei Prozessverläufen dargestellt, was unter Stabilität zu verstehen ist.

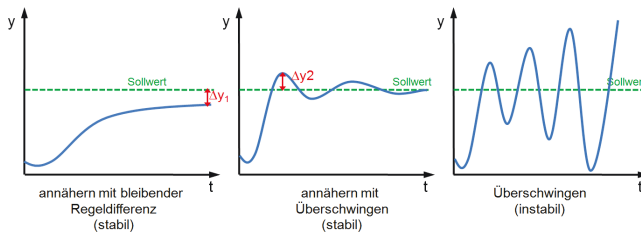


Abbildung 6.1. – Drei mögliche Prozessverläufe einer Zielgröße nach einer initialen Abweichung vom Sollwert durch Änderung des Sollwertes oder eine Störung. Im ersten Fall nähert sich die Zielgröße asymptotisch dem Sollwert an, es verbleibt aber eine Regelabweichung Δy_1 . Im zweiten Fall erfolgt ein Überschwingen mit der Regelabweichung Δy_2 bevor sich die Zielgröße dem Sollwert nähert. Im dritten Fall wird das Überschwingen immer größer: das System ist instabil.

³ [Lutz and Wendt, 2002], S.196, Abschnitt “Definition der Stabilität”. Eine ähnliche Definition findet sich auch in [Mann et al., 2005] S.91 und anderen Büchern zu Grundlagen der Regelungstechnik.

Die Untersuchung, ob eine Regelung stabil ist oder nicht, ist nicht trivial. Die Regelung kann natürlich am Prozess getestet werden. Ist sie allerdings nicht stabil, wird der Prozess die Sicherheitsgrenzen verlassen und dadurch ein Schaden (technisch oder wirtschaftlich) verursacht. Wesentlich vorteilhafter ist es den Regler bereits vor der Aktivierung zu testen. Dazu existieren in der Regelungstechnik verschiedene Berechnungen, wie das *Routh-Hurwitz-Kriterium* [Hurwitz, 1964], das *Nyquist-Kriterium* [Nyquist, 1932] oder die Methoden nach Lyapunov [Lyapunov, 1992].

Alle genannten Methoden haben allerdings den Nachteil, das sie eine genaue mathematische Beschreibung des unregulierten Systems und des verwendeten Reglers benötigen⁴. Aus beiden Gleichungen kann die charakteristische Funktion des geregelten Systems bestimmt werden, aus der sich wiederum Aussagen zur Stabilität ableiten lassen. Eine solche charakteristische Funktion für ein so komplexes System wie eine MVA oder einen Zementofen aufzustellen, ist allerdings nahezu unmöglich. Theoretisch wäre es möglich, über Versuche mit sprunghaften Änderungen von einzelnen Größen zumindest eine Abschätzung dieser Funktion zu erhalten. Da sich der verwendete Brennstoff und die Charakteristik der Anlage aber ständig ändern⁵, ist auch eine Bestimmung der notwendigen Parameter über Versuche kaum möglich.

Die Aufstellung der charakteristischen Funktionen des Systems ist also für die Anwendungen dieser Arbeit nicht durchführbar. Trotzdem ist es möglich, in begrenztem Umfang Aussagen über die Stabilität eines Reglers zu treffen. Dabei helfen folgende Besonderheiten der Anwendung die sich in den Lernvorgang eines adaptiven Systems einbauen lassen:

⁴Handelt es sich beim Regler um einen prädiktiven Regler wie ein NFQ, wird der theoretische Nachweis der Stabilität noch schwerer. In [Rau, 2003] werden verschiedene Methoden vorgestellt, um einen solchen Nachweis zu führen. Die Grundbedingung, dass Prozess- und Reglermodell bekannt sein müssen, bleibt erhalten und wird ergänzt um die Bedingung, dass das Prädiktionsmodell hinreichend viele Zeitschritte exakt den Prozess abbilden muss.

⁵Siehe *concept drift* in Kapitel 5.

1. Die Feuerleistungsführung soll in den meisten Fällen nicht neu auf ein komplett unbekanntes System aufgesetzt werden. Es existieren bereits Regler, welche (überwiegend) auch das Kriterium der Stabilität erfüllen.
2. Die Systemgleichung des Systems ist nicht in mathematischer Form bekannt, aber es existieren Erfahrungen und technisches Wissen, aus denen sich einige grundlegende Zusammenhänge ableiten lassen.

Neben der Sicht der Regelungstechnik auf die Stabilität eines Systems existiert auch eine Betrachtung von Sicherheit im Kontext von lernfähigen Systemen und Reinforcement Learning. In [Heger, 1994] wird beispielsweise das Minimax-Lernverfahren vorgestellt. Im Unterschied zum normalen Q-Learning-Verfahren soll damit eine Strategie gelernt werden, die große Risiken auf jeden Fall vermeidet, auch wenn diese selten vorkommen.

[Hans et al., 2008] stellt dar, wie am Beispiel einer Gasturbinen-Simulation sicheres Reinforcement-Learning mit einer Sicherheitsbewertung und einer Fallback-Strategie durchgeführt werden kann. In [Geibel, 2001, Geibel and Wysotzki, 2005] stellt der Autor fest, dass manche Zustände eines Systems “fatal” sind. Übergänge in fatale Zustände müssen auf jeden Fall vermieden und fatale Zustände demzufolge sehr negativ bewertet werden.

Den vorgestellten Ansätzen ist gemein, dass stets präzises Wissen über das System vorausgesetzt wird. Beim Minimax-Lernverfahren und dem System von [Geibel, 2001] müssen alle fatalen Zustände bekannt sein. [Hans et al., 2008] setzt voraus, dass eine sichere Fallback-Strategie und eine Funktion zur Bewertung der Sicherheit eines Zustandes vorhanden sind. Bei vielen Anwendungen in der Verbrennungsführung ist eine dieser Annahmen nicht gegeben.

Ein Überblick über Verfahren zur Kontrolle der Sicherheit von künstlichen neuronalen Netzen findet sich in [Taylor, 2006]. Der Autor

analysiert verschiedene Verfahren zur Sicherstellung der funktionalen Sicherheit, die jeweils spezifisch für unterschiedliche Netzwerktypen und Anwendungen sind. Für diese Arbeit von Bedeutung ist dabei vor allem der *Safety Lifecycle* aus [Kurd and Kelly, 2003], in dem vorgestellt wird, wie durch Analyse des Netzwerkverhaltens und der anschließenden Generierung neuer Trainingssamples funktionale Sicherheit verbessert werden kann. Ebenso interessant sind *Rule Extraction* Methoden, mit denen versucht wird das Verhalten neuronaler Modelle in Regeln zu übersetzen. Von den in [Andrews et al., 1995] behandelten Methoden zur *Rule Extraction* ist für diese Arbeit vor allem die *pedagogical Rule Extraction* von Bedeutung, bei der nicht aus der Netzwerstruktur sondern aus den Ausgaben auf Regeln geschlossen wird.

Methoden des *Run Time Monitoring*, also der Überwachung online laufender Netze, werden in [Taylor, 2006] ebenfalls aufgeführt. Für diese Arbeit relevant ist dabei vor allem der Ansatz von [Liu et al., 2004, Liu et al., 2005] der darauf basiert die aktuellen Daten mit den Trainingsdaten des Netzwerkes zu vergleichen und so ein Maß zu generieren das aussagt, ob das Netzwerk beim Training aus ähnlichen Daten lernen konnte oder ob diese Daten neu für das Netzwerk und die Netzwerkantworten deshalb unsicher sind.

Ein weiterer sehr vielversprechender Ansatz wird in [Abbeel and Ng, 2005] mit dem *Apprenticeship Learning* vorgestellt. Das Lernverfahren soll nicht ein Modell lernen und darauf aufbauend eine Regelstrategie entwickeln, sondern eine ihm präsentierte Regelstrategie imitieren und verbessern. Im Laufe dieses Kapitels wird ein Lernvorgang entwickelt, der ähnlich dem *'Apprenticeship Learning* auf einer vorhandenen Strategie aufbaut, "fatale" Zustände wie in [Geibel, 2001] vermeidet und eine Fallbackstrategie wie [Hans et al., 2008] verwendet. Für die Nutzung in Verbindung mit einem NFQ in der Feuerleistungsführung ist allerdings nur die Idee gemeinsam; die Implementierung unterscheidet sich deutlich von den angesprochenen Verfahren.

In den folgenden Abschnitten werden Methoden vorgestellt, die Wissen über den Prozess sowie die Idee der Vermeidung fataler Zustände und der Nachahmung einer Regelstrategie nutzen um:

1. bereits beim Training eines selbstlernenden Verfahrens das Lernen instabiler Regelfunktionen zu verhindern,
2. einen trainierten Regler vor dem Einsatz auf instabiles Verhalten zu untersuchen oder
3. einen Regler im Einsatz zu überwachen und angemessen auf falsche Regelstrategien zu reagieren.

Die Notwendigkeit dieser Verfahren zur Überwachung des Reglers und ihre Anwendbarkeit sind sehr stark abhängig von der Art des Reglers bzw. des lernfähigen Verfahrens, das mit dem Regler verknüpft wird.

In Kapitel 3.1 dieser Arbeit wurden verschiedene Möglichkeiten vorgestellt, um einen Regler mit einem lernfähigen System zu kombinieren. Die vorgestellten Varianten sind: 1. Änderung der Parameter des Reglers, 2. Änderung der Eingabe des Reglers und 3. Änderung der Reglerfunktion.

Von diesen vier Varianten werden in dieser Arbeit vor allem die Möglichkeiten 2.: Änderung der Eingabe des Reglers und 3.: Änderung der Reglerfunktion untersucht und verwendet. Die Änderung der Eingabe des Regler erfolgt durch ein lernfähiges Prognosemodell, das entweder einem konventionellem Regler vorgeschaltet ist oder von einem menschlichen Operator beobachtet wird. Eine Änderung der Reglerfunktion erfolgt, indem der bisherige Regler zum Stellen einer Stellgröße durch einen lernfähigen Regler wie ein NFQ ersetzt wird. Beide Möglichkeiten unterscheiden sich wesentlich in dem damit verbundenen Risiko und werden im Folgenden einzeln dargestellt. In Abbildung 6.2 ist dargestellt, welche Maßnahmen zur Gewährleistung der funktionalen Sicherheit im Folgenden vorgestellt werden.

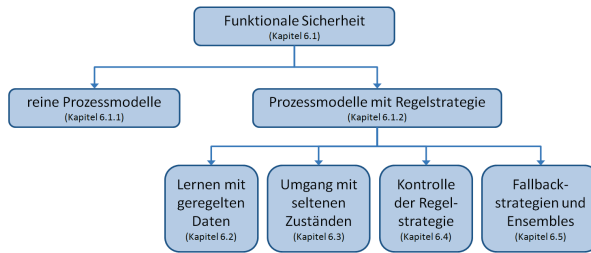


Abbildung 6.2. – Übersicht zu den Maßnahmen zur Gewährleistung der funktionalen Sicherheit, die in diesem Kapitel vorgestellt werden, und des jeweiligen Abschnittes in der Arbeit, in dem sie erläutert werden.

6.1.1. Prozessmodell als Eingabe des Reglers

Bei der in Abschnitt 3.3.2 beschriebenen Methode wird ein konventioneller Regler⁶ mit einem selbstlernenden Prozessmodell kombiniert. Statt von der Messstelle der Anlage erhält der Regler seine Eingabe aus dem Prozessmodell. Dabei werden vor allem zwei Typen von Modellen verwendet:

1. **Softsensor:** Ein Softsensor-Modell wird anstelle eines normalen Messwertes verwendet, wenn dieser häufiger ausfällt oder die Messung nur selten verfügbar ist (Laborwerte).
2. **Prognosemodell:** Ein Prognosemodell soll einen zeitlichen Vorlauf vor dem tatsächlichen Messwert generieren. Es soll beispielsweise der Dampfleistungswert in zwei Minuten prognostiziert werden, um schneller auf Abweichungen reagieren zu können.

In beiden Fällen muss durch das Prozessmodell keine Regelstrategie gelernt werden. Das Modell muss keine Entscheidung treffen, wie

⁶Bei Freikalk-Softsensoren tritt häufig auch der Fall auf, dass die Modellausgabe nicht direkt verwendet, sondern auf der Warte dem Anlagenfahrer dargestellt wird. In diesem Fall stellt der Anlagenfahrer in gewisser Weise den Regler dar, der durch Steuereingriffe auf die Modellausgabe reagiert.

die aktuelle Situation verbessert werden kann. Diese Strategie ist weiterhin im konventionellen Regler enthalten, welcher durch einen menschlichen Experten parametrisiert wurde.

Ein solches Prognose- oder Softsensormodell kann folglich nur die Sicherheit der Feuerleistung gefährden, falls es eine falsche Approximation des Zielwertes ausgibt und der nachgeschaltete Regler aufgrund dieser Eingabe eine falsche Aktion ausführt. Bei der Nutzung eines Prozessmodells als Eingabe für einen Regler muss also sichergestellt werden, dass die Approximation des Zielwertes möglichst gut ist.

In den Kapiteln 4 und 5 wurden bereits Verfahren vorgestellt, mit denen eine möglichst gute Approximationsqualität für Anwendungen in der Feuerleistungsführung sichergestellt werden soll.

Trotzdem ist es möglich, dass ein schlechtes Approximationsmodell aktiviert wird. Häufig lässt sich ein solcher Fall sehr schnell feststellen, indem ständig die Qualität der Approximation durch Beobachtung des Approximationsfehlers beurteilt wird. Lässt diese Qualität nach, kann ein anderes Modell aktiviert werden oder der ursprüngliche Messwert (sofern vorhanden) genutzt werden. Der Aufbau eines solchen Systems zur Sicherstellung einer guten Approximationsqualität ist in Abbildung 6.3 dargestellt.

Zusammengefasst kann das Sicherheitsrisiko durch Prozessmodelle, die als Softsensor oder Prognosemodell verwendet werden, gut behandelt werden. Eine unerwartete Fehlfunktion kann nur auftreten, wenn die Approximation des Modells falsch ist⁷. Diese Approximation lässt sich zum einen bereits vor der Aktivierung des Modells offline prüfen, zum anderen kann sie auch online während der Laufzeit des Modells einfach überprüft werden. Sollten dabei Fehler festgestellt werden, kann statt des Prognosemodells der tatsächliche Messwert verwendet werden.

⁷Ein falsches Verhalten kann natürlich auch auftreten, wenn der nachfolgende konventionelle Regler trotz korrektem Input falsch reagiert. In diesem Fall ist der konventionelle Regler aber falsch entworfen und es liegt kein Fehler des vorgelagerten Lernsystems vor.

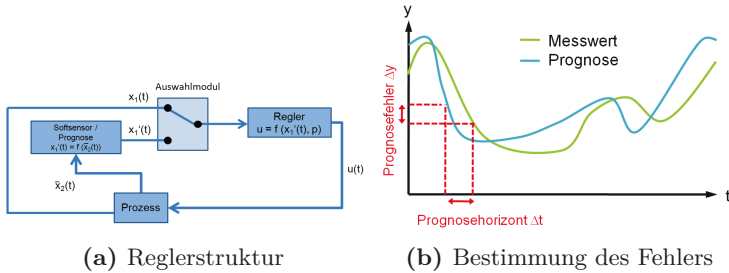


Abbildung 6.3. – Reglerstruktur mit Prognosemodell und Auswahlmodul. Nimmt die Qualität der Prognose $\hat{x}_1(t)$ ab, wird stattdessen der normale Messwert $x_1(t)$ verwendet. Die Qualität der Prognose wird über den Prognosefehler $|\Delta y|$ im Prognosehorizont Δt bestimmt.

6.1.2. Lernfähige Regler

In Abschnitt 3.4 wurde beschrieben, wie ein konventioneller Regler durch einen lernfähigen Regler ersetzt werden kann. Eine Gemeinsamkeit mit den Verfahren des letzten Abschnittes ist, dass auch in diesem Fall ein Prozessmodell gelernt wird. Dieses Prozessmodell wird aber nicht nur genutzt, um die Approximation eines Zielwertes zu berechnen, sondern auch um zu entscheiden wie dieser Zielwert durch einen Stelleingriff optimiert werden kann. Diese Regelstrategie wurde im vorherigen Abschnitt noch durch den konventionellen Regler entschieden. Während im letzten Abschnitt die Qualität und Sicherheit des Modells und des Lernverfahrens allein an der Qualität der Approximation des Zielwertes festgemacht werden konnte, muss bei Verwendung eines lernfähigen Reglers auch die Qualität der Regelstrategie bewertet und auf Sicherheitsrisiken überprüft werden.

Diese Überprüfung ist wesentlich schwieriger als die Bewertung einer Zielwertprognose. Das Regelverhalten eines Reglers kann zwar offline auf den vorhandenen Daten simuliert werden. Tritt dabei ein Unterschied zur bisherigen Regelstrategie auf, so kann aber oftmals

nicht bestimmt werden, ob dieser Unterschied sich positiv oder negativ auf den Gesamtprozess auswirken würde. Die Qualität eines Reglers kann letztlich also nur im closed loop Betrieb online festgestellt werden.

Selbst im Online Fall ist die Bestimmung der Qualität aufwendig. Eine Stellgröße kann zu einem Zeitpunkt nur mit einem Wert gefahren werden. Kommt es zu einer Abweichung der Zielgröße von ihrem Sollwert, so sind folgende Varianten möglich:

1. Der Regler hat die Stellgröße mit der bestmöglichen Strategie gefahren. Eine andere Strategie hätte zu noch größeren Abweichungen geführt.
2. Die Strategie des aktuellen Reglers ist nicht optimal, aber besser als die Strategie des bisher verwendeten Reglers.
3. Die Strategie des aktuellen Reglers ist schlechter als die des bisherigen Reglers.

Die Schwierigkeit besteht darin herauszufinden, welcher der drei Fälle in der aktuellen Situation vorliegt. In den folgenden Abschnitten dieses Kapitels werden verschiedene Möglichkeiten untersucht um sicherzustellen, dass die Regelstrategie eines lernfähigen Reglers so gut wie möglich ist. Die vorgestellten Verfahren setzen dabei an drei unterschiedlichen Stellen an:

1. Verfahren, die bereits während des Trainings des Reglers durch Zusatzwissen und Bedingungen das Lernen einer "falschen" Regelstrategie verhindern (Abschnitt 6.2).
 2. Verfahren, mit denen die erlernte Regelstrategie vor der ersten Zuschaltung des Reglers überprüft werden kann (Abschnitt 6.4).
 3. Verfahren, mit denen die Regelstrategie während der Laufzeit des Regler überprüft und korrigiert wird (Abschnitt 6.5).
-

6.2. Daten geregelter Prozesse und Expertenwissen

In dieser Arbeit werden lernfähige Regler wie das NFQ angewandt. Diese Regler erhalten Daten eines Prozesses und einer Stellgröße, schätzen daraus die Zusammenhänge zwischen den Datenkanälen und leiten dann eine Stellstrategie für die Stellgröße ab.

Alle datengetriebenen Lernverfahren funktionieren allerdings nur, wenn die Beispieldaten, die zum Lernen verwendet werden, eine entsprechende Qualität haben. In Kapitel 4 und 5 wurde bereits dargestellt, welchen Einfluss Rauschen und veränderliche Prozesswerte auf den Lernvorgang haben. Soll nicht nur ein Prognosemodell, sondern eine Regelstrategie gelernt werden, so ist außerdem der Verlauf der bisher ausgeführten Regelstrategie für den Lernvorgang von enormer Bedeutung.

Eine wesentliche Bedingung für das Lernen einer Stellstrategie ist zunächst, dass in den verfügbaren Daten eine Veränderung des Wertes der Stellgröße vorliegt. Bei einem rein datengetriebenen Lernverfahren lässt sich nur aus einer Veränderung der Stellgröße und Beobachtung der Reaktion des Prozesses ableiten, welchen Einfluss die Stellgröße auf andere Messgrößen hat. Hat die Stellgröße immer denselben Wert, können diese Daten nicht verwendet werden um eine Stellstrategie zu lernen, sondern es muss erst ein Versuchsprogramm mit der Stellgröße durchgeführt werden. Bei den Stellgrößen, die in dieser Arbeit untersucht und beeinflusst wurden, lag allerdings in allen Fällen bereits eine Führung der Stellgröße durch konventionelle Regler oder menschliche Operatoren vor.

Aber auch wenn die Stellgröße in den genutzten Daten oft genug verändert wird, können sich Probleme beim Lernen einer optimalen Stellstrategie ergeben. Die beiden häufigsten Probleme mit den größten Auswirkungen bei der Regelung eines Verbrennungsprozesses sind:

- eine konventionelle Regelstrategie ohne Exploration
- selten vorkommende aber wichtige Ausnahmesituationen

Beide Umstände führen häufig dazu, dass das Lernen einer Stellstrategie auf den Daten eines Verbrennungsprozesses fehlschlägt und falsche oder unvollständige Stellstrategien gelernt werden. Auf beide Umstände eines Verbrennungsprozesses wird deshalb im Folgenden eingegangen, und es wird erläutert mit welchen Erweiterungen ein Lernverfahren versehen werden muss, damit eine gute Stellstrategie gelernt wird.

6.2.1. Lernen auf Daten eines geregelten Prozesses

Viele Lernverfahren haben implizit oder explizit die Voraussetzung, dass die Daten, auf denen gelernt wird, einen großen Raum des Merkmalsraums abdecken, die Daten also sehr explorativ sind. In [Sutton and Barto, 1998] wird diese Exploration als eine Voraussetzung für das Finden einer guten Strategie genannt.

Ist dies nicht der Fall, so wird ein in Phasen ablaufender Lernvorgang vorgeschlagen (wie beispielsweise beim NFQ [Riedmiller, 2005a]) in dem abwechselnd auf den vorhandenen Daten gelernt und anschließend die neu gelernte Strategie angewandt wird, um neue Daten über die Effektivität dieser Strategie zu erhalten.

Die Daten, die zum Training von Stellgrößen in Verbrennungsprozessen zur Verfügung stehen, sind allerdings weder explorativ, noch lässt sich ein mehrstufiger Lernvorgang durchführen, da dieser mit hohen Risiken verbunden ist. In Abbildung 6.4a wurde das Problem der einseitigen Daten an einem Beispiel erläutert. Dargestellt ist der Zusammenhang zwischen einer Stellgröße und einer Zielgröße in einer Anlage. Im Beispiel existiert ein einfacher P-Regler, der die Stellgröße je nach aktuellem Wert der Zielgröße verändert. Da die Parameter des Reglers nicht geändert werden, reagiert er auf jede Veränderung der Zielgröße mit derselben Änderung der Stellgröße. Werden die erhaltenen Daten im Stellgröße-Zielgröße-Merkmalraum abgebildet, so entsteht eine Gerade. Abseits dieser Gerade existieren keine Datenpunkte im Datensatz.

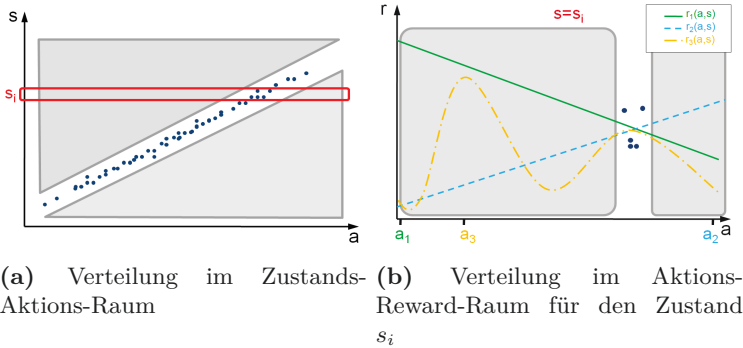


Abbildung 6.4. – Verteilung der Datenpunkte im Zustands-Aktions-Raum und im Aktions-Reward-Raum für einen geregelten Prozess mit einem einfachen P-Regler. Da der Regler für denselben Zustand immer dieselbe Aktion ausführt, gibt es große Bereiche in beiden Räumen, in denen keine Datenbeispiele vorliegen (grau markierte Bereiche). Soll ein Approximationsmodell den Reward für alle Aktionen approximieren, sind bei solchen spärlichen Verteilungen viele unterschiedliche Approximationen ($r_1(s, a)$ bis $r_3(s, a)$) möglich. Die “beste“ Aktion (a_1 bis a_3) der jeweiligen Reward-Approximation ist diejenige Aktion welche laut Modell den höchsten Reward erzeugt. Wie zu sehen ist, ist diese ”beste Aktion“ bei allen drei Approximationen unterschiedlich.

Diese Unterbesetzung des Datenraumes setzt sich fort, wenn man für einen bestimmten Zustand s_i die Verteilung des erhaltenen Rewards über den möglichen Aktionen betrachtet. Im Beispiel (siehe Abbildung 6.4b) ist dies die Abweichung der Zielgröße vom Sollwert eine Minute nach Ausführung der Aktion a . Da im Zustand s_i vom P-Regler immer dieselbe Aktion a_i ausgeführt wurde, wurden auch nur für diese Aktion Rewards beobachtet. Ein NFQ oder andere lernfähige Verfahren versuchen den möglichen Reward über allen Aktionen eines Zustandes zu approximieren. Da im vorliegenden Datensatz aber nur Datenbeispiele für eine einzige Aktion enthalten sind, ist die Abschätzung des Rewards für die anderen Aktionen willkürlich. In Abbildung 6.4b sind drei mögliche Approximationen des Rewards

($r_1(s, a)$ bis $r_3(s, a)$) abgebildet.

Jede dieser drei Approximationen hat auf dem vorhandenen Datensatz den gleichen Approximationsfehler. Würde aber die jeweils "beste" Aktion (also die mit dem geschätzten höchsten Reward) ausgeführt, würde dies bei allen drei Approximationen zu unterschiedlichen Ergebnissen führen. Jedes der drei Approximationsmodelle würde also eine vollkommen unterschiedliche Stellstrategie fahren, obwohl alle denselben Approximationsfehler auf den vorhandenen Daten haben.

Die Ursache dieses Problems liegt in der nicht-explorativen Stellstrategie des P-Reglers⁸.

Da die meisten Stellgrößen in Verbrennungsprozessen über solche konventionellen PID-Regler gestellt werden, ist die nicht-explorative Verteilung der Daten im Stellgröße-Zielgröße-Merkmalraum eines der größten Probleme beim Lernen einer Stellstrategie. Wird dieses Problem nicht beachtet, führt es in den meisten Fällen dazu, dass eine scheinbar willkürliche Stellstrategie gelernt wird, welche im Einsatz auf der Anlage schnell zu Sicherheitsproblemen und einer Abschaltung des lernfähigen Reglers führt.

These 10 (Lernen auf Nicht-explorativen Daten)

Datengetriebenes Lernen einer optimalen Stellstrategie in einem Zustands-Aktionsraum ist nicht möglich, falls keine Exploration der Stellstrategie erfolgte, für jeden Zustand s_i also stets dieselbe (oder eine sehr ähnliche) Aktion a_j ausgeführt wurde.

⁸Sie tritt ebenfalls bei komplexeren PID-Reglern auf, falls der integrale und der differentiale Fehler dem Approximationsmodell ebenfalls als Input präsentiert werden.

Um eine korrekte und überall gültige Approximation des Rewards und damit eine optimale Stellstrategie des Reglers zu ermöglichen, wäre es theoretisch nötig, zuerst eine Exploration des Zustands-Aktionsraumes durchzuführen. Praktisch ist dies aus mehreren Gründen nicht möglich:

- Viele der möglichen Zustands-Aktions-Kombinationen, insbesondere solche, die stark von der bisherigen Strategie abweichen, führen zu einem schlechten oder sicherheitskritischen Verhalten und dürfen nicht ausgeführt werden.
- Bei vielen Stellgrößen reagiert der Prozess sehr langsam. Eine Exploration des Zustandsraumes würde deshalb sehr viel Zeit in Anspruch nehmen. Lange Explorationsfahrten mit einer nicht optimalen Strategie gefährden die Wirtschaftlichkeit der Anlage.
- Wird ein Modell nachtrainiert, so müsste vor jedem Nachtraining erneut eine Explorationsfahrt durchgeführt werden, oder die Daten alter Explorationsfahrten müßten in den Trainingsdatensatz integriert werden.

Wie kann trotz der Unmöglichkeit einer kompletten Exploration des Datenraumes und der in These 10 genannten Nichterlernbarkeit einer optimalen Stellstrategie ein Lernverfahren auf bereits geregelten Prozessdaten angewandt werden? In dieser Arbeit wird im Folgenden ein Lösungsansatz gezeigt, der verfolgt werden kann, wenn die Bedingung gilt, dass nicht die "optimale" Stellstrategie gefunden werden soll, sondern (insbesondere im ersten Lernschritt) nur eine Stellstrategie, die ähnlich gut ist wie die aktuelle Stellstrategie.

Dieser Ansatz ähnelt dem in [Abbeel and Ng, 2005] vorgeschlagenen Ansatz des *Apprenticeship Learning*, da eine vorhandene Strategie nachgebildet und verbessert werden soll. [Abbeel and Ng, 2005] benutzt dabei allerdings Reinforcement-Learning Verfahren, die nur bekannte Aktionen bewerten, und Modelle, die (verglichen mit der Anwendung dieser Arbeit) geringe Zustands-Aktionsräume haben. Die in dieser

Arbeit verwendeten NFQs bewerten allerdings nicht nur bekannte Strategien sondern extrapolieren auch auf unbekannte Situationen und andere Stellstrategien. Dies ist in vielen Fällen von Vorteil, wird aber bei bereits geregelten Daten zum Nachteil.

Das NFQ-Lernverfahren soll also so beeinflusst werden, dass eine Strategie, die stark von der bisher verfolgten Strategie abweicht, als unvorteilhaft bewertet und vermieden wird. Das Werkzeug dafür sind sogenannte "künstliche Datenpunkte", die bereits in der Konzeption des NFQ-Lernverfahrens vorgesehen sind. In [Riedmiller, 2005a] wird beschrieben, wie künstliche Datenpunkte mit dem Zielzustand und einem hohen Reward als *Hint-To-Goal-Mechanismus* genutzt werden können. Obwohl das Verfahren den Zielzustand nicht durch eigene Exploration gesehen hat, weiß es, in welche Richtung dieses Ziel liegt und dass für diesen Zustand ein hoher Reward zu erwarten ist. Voraussetzung für den Einsatz künstlicher Datenpunkte ist, dass es einen Experten gibt, der weiß, dass für diesen Zustand ein hoher Reward zu erwarten ist.

Ein ähnlicher Mechanismus wird in [Abu-Mostafa, 1994] als *learning from hints* beschrieben und in [Lampinen et al., 1999] erweitert, wobei ebenfalls künstliche Datenpunkte (kDP) zum Lernprozess hinzugefügt werden um diesen zu beschleunigen. Auch in [Niyogi et al., 1998] wird dargestellt, wie kleine Datensätze durch zusätzliche virtuelle Datenpunkte ergänzt werden können und dadurch das Lernverhalten von RBF-Netzwerken verbessert wird⁹.

Für die in dieser Arbeit untersuchten Prozesse kann dieser Mechanismus genutzt werden. Zwar ist nicht bekannt, wie die optimale Stellstrategie aussieht, es ist aber bekannt, dass eine große Abweichung von der aktuellen Strategie unvorteilhaft ist. Eine solche Informa-

⁹Eine ausführliche Darstellung der verschiedenen Methoden zur Erzeugung und Einbindung virtueller Punkte in Lernverfahren wurde in [Goldhan, 2012] erarbeitet.

tion kann durch künstliche Datenpunkte ebenso vermittelt werden wie Informationen über das Ziel im Hint-To-Goal-Mechanismus. Deshalb wurde in dieser Arbeit eine Modifikation des Hint-To-Goal-Mechanismus entwickelt und in [Steege et al., 2010] publiziert. In Algorithmus 6 und Abbildung 6.5 ist dargestellt, welche Schritte unternommen werden müssen, um künstliche Datenpunkte dieser Art zu berechnen und in den Lernprozess zu integrieren.

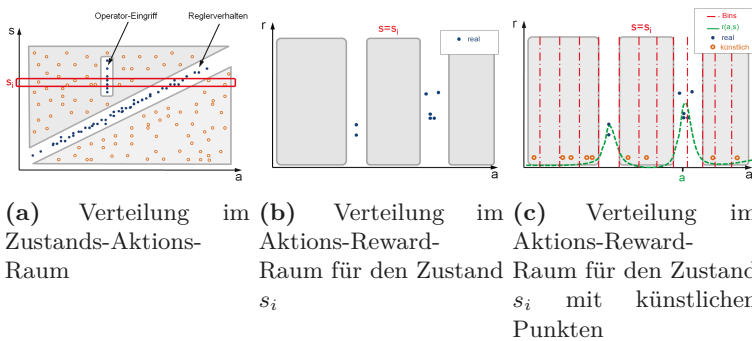


Abbildung 6.5. – Verteilung der Datenpunkte eines geregelten Prozesses. Zusätzlich ist der Eingriff eines Operators dargestellt, der nicht der üblichen Regelstrategie entspricht. Durch das Einfügen künstlicher Datenpunkte mit einem niedrigeren Reward für nicht beobachtete Aktionen nimmt die Approximation des Rewards $r(s, a)$ über allen Aktionen eine Form an, bei der nicht beobachtete Aktionen schlechter bewertet werden. Die Regelstrategie wird dadurch sicherer, und wechselnde Strategien, wie in Abbildung 6.4 gezeigt, werden vermieden. Die Entscheidung, an welchen Stellen im Zustands-Aktionsraum künstliche Datenpunkte eingefügt werden, wird durch Clustern der Daten im Zustandsraum und anschließende Einteilung des Aktionsraumes in beobachtete und nicht beobachtete Aktions-Bins getroffen.

Zuerst müssen dazu die vorhandenen Datenpunkte in Cluster eingeteilt werden, wobei das Clustern allein nach dem Zustand erfolgt und nicht die ausgeführte Aktion oder den erhaltenen Reward berücksichtigt. In dieser Arbeit wurde zu Clustering der Daten ein k-means-Algorithmus verwendet.

Anschließend wird über jedem Cluster C_i ein Histogramm der in diesem Zustands-Cluster vorhandenen Aktionen aufgebaut¹⁰. Befinden sich in jedem der Histogramm-Bins Datenpunkte mit Aktionen, so ist der Aktionsraum für diesen Zustand vollständig exploriert, und es müssen keine künstlichen Datenpunkte hinzugefügt werden. Bei Daten bereits geregelter Anlagen ist dieser Fall allerdings sehr selten.

Gibt es Histogramm-Bins, in denen keine realen Datenpunkte vorliegen, werden für diese Bins künstliche Datenpunkte erzeugt. Die Anzahl der erzeugten künstlichen Datenpunkte kann mittels eines Parameters n_{kp} beeinflusst werden und orientiert sich an der Anzahl der realen Datenpunkte im aktuellen Zustandscluster.

Wird als Modell/Lernverfahren ein NFQ verwendet, so benötigt das NFQ Datenpunkte der Form $\eta = (s, a, s', r)$ mit dem Zustand s , der Aktion a , dem Folgezustand s' und dem erhaltenen Reward r . Bei künstlichen Datenpunkten $\eta_{art} = (s_{art}, a_{art}, s'_{art}, r_{art})$ müssen die einzelnen Komponenten berechnet werden:

- s_{art} : Der künstliche Zustand ist gleich dem Mittelwert des Zustandes aller Punkte im aktuellen Cluster.
- a_{art} : Die künstliche Aktion ist der Aktionswert des aktuellen Histogramm-Bins, in dem Punkte eingefügt werden sollen.

¹⁰In Abbildung 6.5a ist ein solcher Cluster durch den rot umrandeten Zustand s_i dargestellt. Die anschließend aufgebauten Histogramm-Bins sind durch rote Striche in Abbildung 6.5c gekennzeichnet.

- r_{art} : Der künstliche Reward ist gleich dem minimalen, bei den Datenpunkten des Clusters beobachteten Reward. Von diesem minimalen Reward wird zusätzlich ein Betrag r_{kp} abgezogen¹¹. Der Reward der künstlichen Datenpunkte eines Clusters ist also immer kleiner als der Reward aller realen Datenpunkte eines Clusters. Das ist beabsichtigt, da bisher nicht beobachtete Aktionen nicht ausgeführt werden sollen und folglich für diese Aktionen ein geringer Reward approximiert werden soll.
- s'_{art} : Der Folgezustand stellt bei künstlichen Datenpunkten das größte Problem da. Es ist nicht möglich abzuschätzen in welchem Zustand der Prozess eingetreten wäre, falls die künstliche Zustands-Aktionskombination ausgeführt würde. Auch im Ursprungsansatz des Hint-to-Goal in [Riedmiller, 2005a] tritt dieses Problem auf. Als Lösung wird der Folgezustand mit einem Flag versehen und im anschließenden NFQ-Lernprozess nicht berücksichtigt. Für künstliche Datenpunkte ist der Q-Wert also gleich dem künstlichen Reward, während bei normalen Datenpunkten der Q-Wert des Folgezustandes mit eingeht. Dieser Umstand muss bei der Wahl von r_{kp} berücksichtigt werden.

Durch den vorgestellten Algorithmus wird der ursprüngliche Datensatz um künstliche Datenpunkte erweitert. Diese künstlichen Datenpunkte sollen verhindern, dass aufgrund der sehr einseitigen Datenverteilung bei geregelten Prozessen eine Stellstrategie gelernt wird welche zu wesentlich schlechteren Ergebnissen führt als die bisherige Strategie.

¹¹Für die Berechnung des künstlichen Rewards sind auch andere Verfahren denkbar. Beispielsweise kann auch der mittlere Reward des nächstliegenden Histogrammbin betrachtet werden. Die Gefahr bei allen Ansätzen besteht darin, eine Funktion zu wählen, welche den künstliche Reward atypisch zur Verteilung des Rewards der realen Datenpunkte wählt. Der auf beiden Datenmengen trainierte Lernprozess erhält dann widersprüchliche Daten und kann keine Approximation des Rewards mehr liefern. In der bisherigen Praxis hat sich die vorgestellte Funktion hingegen bewährt.

Algorithmus 6 Künstliche Datenpunkte für Lernen auf geregelten Daten

Eingabe: Datensatz S_{init} mit $\eta_i = (s, a, s', r), \eta_i \in S_{init}, i = 1 \dots n$,
 Rate künstlicher Punkte pro Cluster n_{kp} , Rewardabschlag für künstliche Punkte r_{kp} ; Anzahl Aktions-Bins n_{action}

Ausgabe: Datensatz zum Trainieren eines Modells S_{train}

```

 $S_{train} \leftarrow S_{init}$ 
 $C = \{C_i \mid C_i = (S_{C_i}, A_{C_i}, S'_{C_i}, R_{C_i}), i = 1 \dots m\} \leftarrow$ 
 $cluster(S_{init}, s, m)$  // clustere den Datensatz  $S_{init}$  nach dem Zu-
 $stand s$  in  $m$  Cluster
for all  $C_i \in C$  do
   $A_{hist} \leftarrow hist(A_{C_i}, n_{action})$  // Histogramm der Aktionen in  $A_{C_i}$ 
  mit  $n_{action}$  Bins
   $h_1 \leftarrow n_{kp} \mid C_i \mid$  // künstl. Punkte pro Cluster
   $h_2 \leftarrow round(\frac{h_1}{|\{b_j \mid b_j = 0, j = 1, \dots, n_{action}\} \in A_{hist}|})$  // künstl. Punkte pro
  leerem Histo.Bin
  for all  $\{b_j \mid b_j = 0, j = 1, \dots, n_{action}\} \in A_{hist}$  do
     $r_{art} \leftarrow min(R_{C_i}) - r_{kp}$ 
     $s_{art} \leftarrow mean(S_{C_i})$ 
     $a_{art} \leftarrow A_{hist}(b_j)$ 
     $s'_{art} \leftarrow f$  // Folgezustand = Flag
    for  $k = 1$  to  $h_2$  do
       $S_{train} \leftarrow S_{train} \cup \eta_k = (s_{art}, a_{art}, s'_{art}, r_{art})$ 
    end for
  end for
end for

```

Um die Auswirkungen des Algorithmus zu demonstrieren werden im folgenden Ergebnisse beim Training von lernfähigen Modellen auf geregelten Prozessen mit und ohne künstliche Daten dargestellt.

6.2.2. Lernen einer sicheren Regelstrategie am Cart-Pole-Simulator

Um die Unterschiede zwischen dem Training eines Modells mit geregelten und ungeregelten Daten an einem einfachen nachvollziehbaren Beispiel zu verdeutlichen, wurde die Cart-Pole-Simulationsumgebung gewählt.

Die Cart-Pole-Simulation, die manchmal auch invertiertes Pendel genannt wird, ist ein im Bereich des Reinforcement Learning und der Regelungstechnik sehr gebräuchlicher Benchmark, um die Güte eines Lern- oder Regelungsansatzes zu überprüfen. Es wird das Verhalten eines Stabes simuliert, der auf einem Wagen an einem Gelenk montiert ist (siehe Abbildung 6.6). Die Aufgabe des Regelungssystems besteht darin, durch Bewegung des Wagens den Stab stets aufrecht zu halten¹².

In der Literatur existieren viele Ansätze, um automatisch eine Regelungsstrategie für das Cart-Pole-Problem zu finden. In keinem der Ansätze¹³, die im Rahmen dieser Arbeit untersucht wurden, erfolgte der Lernvorgang allerdings auf bereits geregelten Daten. Stattdessen erfolgt immer ein Lernen auf den Daten einer zufälligen Regelstrategie oder ein sequentieller Lernvorgang, in dem eine Strategie gelernt, anschließend erprobt und dann erneut gelernt wird.

Für diese Arbeit wurden unterschiedliche Ansätze erprobt, um die Auswirkungen auf die Regelstrategie zu untersuchen. Dazu wurden folgende Konfigurationen getestet:

¹²Eine ausführliche Beschreibung der Simulationsumgebung, der Parameter des Systems und der Literaturquellen, in denen der Simulator als Benchmark genutzt wird, findet sich in Anhang B.2.

¹³Siehe Anhang B.2.

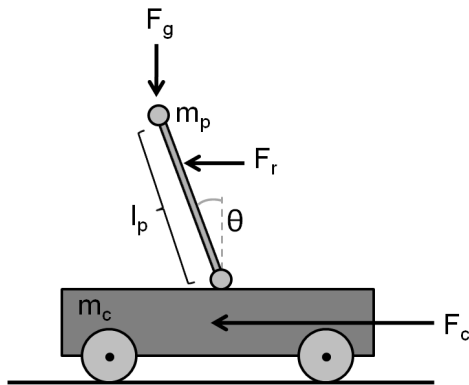


Abbildung 6.6. – Modell der Cartpole-Simulationsumgebung. Ein Stab mit der Länge l_p und der Masse m_p ist auf einem Wagen mit der Masse m_c montiert. Auf den Stab wirkt die Gravitationskraft F_g und eine Kraft mit zufälliger Stärke F_r , die Seitenwind entsprechen soll. Ein Regelungssystem kann mit einer Kraft F_c den Wagen bewegen und dadurch den Stab beeinflussen.

1. PID-Regler: Ein einfacher PID-Regler. Eingabe des Reglers ist die aktuelle Abweichung des Winkels des Stabes vom Sollwert von 0° . Die Parameter K_P , T_N und T_V des PID-Reglers sind so gewählt, dass der Regler den Stab gut balanciert, das Ergebnis aber nicht optimal¹⁴ ist.
2. NFQ, Zufalls-Daten: Ein NFQ-Regler, der mit zuvor generierten explorativen Daten des Cartpole-Simulators trainiert wurde. Um die Trainingsdaten zu generieren, wurden der Winkel, die Geschwindigkeit und die Beschleunigung des Stabes sowie die auf den Wagen wirkende Kraft auf zufällige Werte gesetzt und ein Schritt simuliert um ein Datentupel (s, a, s', r) zu erhalten. Dieser Vorgang wurde 500 Mal wiederholt und erzeugte so einen sehr explorativen Datensatz zum Training des NFQ. Bei einem realen Verbrennungsprozess liegen solche Daten allerdings so gut wie nie vor.
3. NFQ, PID-Daten: Der in Versuch 1. verwendete PID-Regler wurde benutzt um 500 Datentupel (s, a, s', r) zu generieren. Der PID-Regler bestimmt die ausgeführte Aktion a in den Trainingsdaten. Die Startpositionen werden nicht zufällig bestimmt, sondern sind abhängig vom jeweils letzten Zustand. Dieses Trainingsszenario entspricht den von einer Verbrennungsanlage erhaltenen Daten. Anschließend wird mit den Trainingsdaten ein NFQ trainiert, das den Prozess regelt.
4. NFQ, PID-Daten + kDP: Es werden ebenso wie in Versuch 3. 500 Datentupel (s, a, s', r) unter Verwendung eines PID-Reglers erzeugt. Zusätzlich werden 1000 künstliche Datenpunkte (kDP) mit dem in Abschnitt 6.2.1 vorgestellten Algorithmus 6 erzeugt und dem Trainingsdatensatz hinzugefügt. Mit dem kombinierten

¹⁴Der Regler wurde bewußt nicht optimal parametrisiert da untersucht werden soll ob das Lernfähige Verfahren eine bessere Strategie findet. Außerdem entspricht dies der Situation der meisten Verbrennungsanlagen, deren Regler ebenfalls gut, aber nicht optimal eingestellt sind.

Datensatz wird ein NFQ trainiert, das anschließend den Prozess regelt.

5. idealer Regler: In diesem zusätzlichen Versuch wird in jedem Simulationsschritt die Aktion ausgeführt, die den Stab die nächsten zwei Schritte am besten balanciert. Dies ist nur durchführbar, da in dieser Simulation das Prozess vollständig bekannt ist. Der Versuch dient nur zu Vergleichszwecken und ist an einer realen Anlage nicht durchführbar.

In Tabelle 6.1 sind die Ergebnisse dargestellt, die sich mit den unterschiedlichen Verfahren erzielen lassen. Jeder Versuch wurde 50 Mal mit allen fünf Methoden wiederholt. Ein Simulationslauf enthielt jeweils 50 Simulationsschritte. Wenn der Stab während eines Simulationschrittes umfiel (also der Winkel größer $1,5\text{rad}$ wurde), wurde der Stab für den nächsten Simulationsschritt erneut in die aufrechte Ausgangsposition gesetzt. Der Fehler \bar{e} entspricht der mittleren Abweichung des Stabwinkels von der aufrechten Position, gemittelt über den 500 Simulationsschritten. Um die Qualität der Regelung zu bestimmen wurden drei Kennzahlen bestimmt: $Q_{0,1}(\bar{e})$, $Q_{0,5}(\bar{e})$ und $Q_{0,9}(\bar{e})$ stehen für das 10%-, 50%- und 90%-Quantil¹⁵ von \bar{e} über allen 50 Wiederholungen der Simulation.

Das beste Ergebnis erzielt, wie erwartet, der ideale Regler. Bei diesem Regler ist die ideale Aktion aufgrund der vollkommenen Kenntnis des Prozesses bekannt. Der verbleibende Fehler von $Q_{0,5}(\bar{e}) = 0,032\text{rad}$ entsteht durch die Auswirkungen der Kraft des Windes F_r , die nicht bekannt ist.

Das zweitbeste Resultat wird von dem NFQ-Regler erreicht, welcher mit explorativen Daten trainiert wurde. In der Realität der Verbrennungsführung stehen solche Daten zum Training eines NFQ aber nicht zur Verfügung. Trainiert man einen NFQ-Regler hingegen nur mit Datenpunkten, die von einem geregelten Prozess stammen (Versuch

¹⁵Es wurde bewusst das Quantil statt des Mittelwerts gewählt, da sonst ein einziger besonders schlechter Versuch das Gesamtergebnis stark beeinflussen kann.

	$Q_{0,5}(\bar{e})$
PID-Regler	0,094
NFQ, Zufalls-Daten	0,068
NFQ, PID-Daten	0,767
NFQ, PID-Daten + kDP	0,109
idealer Regler	0,033

Tabelle 6.1. – Winkelabweichungen der Cartpole-Simulation mit unterschiedlichen Reglern. Jede Simulation über je 50 Simulationschritte wurde 50 Mal durchgeführt. Der aufgeführte Fehler entspricht dem Median über 50 Simulationen. Das 10%- und 90%-Quantil sind zusätzlich in Tabelle B.14 in Anhang B.5 abgebildet.

NFQ mit PID-Daten), so ist das Resultat bedeutend schlechter. Die resultierende Regelstrategie ist so schlecht, dass der Stab selten länger als 10 Schritte balanciert werden kann. Dies bestätigt These 10. Aus Daten, in denen keine Exploration der Stellstrategie vorliegt, kann keine optimale Strategie abgeleitet werden. Es kann auch eine Erklärung sein, warum datengetriebene Lernverfahren bisher kaum auf bereits geregelten Prozessdaten eingesetzt werden.

Das Lernen einer sehr schlechten Stellstrategie kann allerdings durch die in Algorithmus 6 gezeigte Ergänzung des Datensatzes um künstliche Datenpunkte verhindert werden. Die Ergebnisse zeigen, dass ein NFQ, das mit einem solchen erweiterten Datensatz trainiert wurde, eine wesentlich bessere Strategie lernt als ein NFQ, das allein mit den geregelten Beispieldaten trainiert wurde. Durch die künstlichen Daten nach Algorithmus 6 kann also auch auf geregelten Datensätzen eine sinnvolle Stellstrategie gelernt werden. Die Ergebnisse zeigen aber auch, dass diese Stellstrategie oft etwas schlechter ist als die Stellstrategie des PID-Reglers, der einen Teil der Beispieldaten lieferte. An diesem Punkt stellt sich die Frage, warum mit viel Aufwand ein neuronales Modell trainiert werden soll, wenn das Ergebnis etwas schlechter oder gleich dem bisher schon laufenden Regler ist?

Es darf dabei aber nicht übersehen werden, dass ein lernfähiger Regler sich immer wieder an neue Daten anpassen kann. Der oben gezeigte Versuch demonstrierte, dass mit Hilfe künstlicher Daten ein erster NFQ-Regler trainiert werden kann, welcher das Problem ähnlich gut wie oder geringfügig schlechter beherrscht als der bisherige Regler. Der Regler kann also aktiviert werden und aktiv regeln. Da seine Regelstrategie sich geringfügig von der des bisher laufenden PID-Reglers unterscheidet, entstehen neue Datenpunkte, in denen eine Regelstrategie enthalten ist, die bisher nicht auf der Anlage gefahren wurde. In der Kombination der alten und neuen Daten ist nun also eine kleine Exploration enthalten. Trainiert man auf diesen Daten erneut ein Modell, kann daraus eine bessere Strategie abgeleitet werden. In Tabelle 6.2 sind die Ergebnisse eines solchen iterativen Trainings aufgeführt.

	$Q_{0,5}(\bar{e})_{0-50}$	$Q_{0,5}(\bar{e})_{0-500}$	$Q_{0,5}(\bar{e})_{450-500}$
PID-Regler	0,094	0,097	0,097
NFQ, Zufalls-Daten	0,068	0,276	0,5014
NFQ, PID-Daten	0,767	1,012	1,102
NFQ, PID-Daten + kDP	0,109	0,102	0,099
idealer Regler	0,033	0,035	0,035

Tabelle 6.2. – Ergebnisse eines mehrfachen iterativen Trainings der unterschiedlichen Regler. Nach jeweils 50 Simulationsschritten wurden die NFQ-Regler nachtrainiert. Bei diesem Nachtraining wurden die bisherigen Simulationsschritte dem Datensatz als Datenpunkte hinzugefügt. Alle Versuche wurden 50 Mal wiederholt. $Q_{0,5}(\bar{e})_{0-500}$ bezeichnet den Median des Fehlers über dem gesamten Simulationslauf von je 500 Schritten. $Q_{0,5}(\bar{e})_{0-50}$ bezeichnet den Median des Fehlers über den ersten 50 und $Q_{0,5}(\bar{e})_{450-500}$ den Median des Fehlers über den letzten 50 Simulationsschritten.

Wie zu sehen ist, kann das NFQ+kDP-Verfahren im Laufe der Simulation ein etwas besseres Ergebnis erzielen. Die Resultate des NFQ, das mit reinen PID-Daten trainiert wurde, und auch die Resultate des NFQ, das mit Zufalls-Daten trainiert wurde, werden hingegen schlechter. Dieses Ergebnis erstaunt auf den ersten Blick, ist aber erklärbar: zur Validierung¹⁶ des Trainings eines NFQ werden im Allgemeinen 25%-40% der Gesamtdaten genutzt. Werden die Validierungsdaten zufällig aus dem Gesamtdatensatz ausgewählt oder nur die jeweils neuesten Daten zur Validierung genutzt, so nimmt der ursprüngliche (zufällige und explorierte) Datensatz im Laufe des Trainings einen immer kleineren Stellenwert gegenüber den neuen geregelten Daten ein. Die neuen geregelten Daten enthalten sehr viel weniger Zustands-Aktionskombinationen, weshalb der Lern- und Validierungsvorgang auf diesen Daten schwieriger ist und mehr Fehler bei der Ableitung der korrekten Regelstrategie möglich sind. Führt man die Validierung weiterhin nur auf den ersten (explorativen) Datenpunkten durch, dann kann ein Ansteigen des Fehlers durch eine falsche Regelstrategie vermieden werden¹⁷. In Abbildung 6.7 sind die durchschnittlichen Fehler der verschiedenen Verfahren über dem Simulationszeitraum dargestellt.

Das Ergebnis des NFQ+kDP-Verfahren kann weiter verbessert werden, wenn die in Kapitel 5 benutzten Verfahren zur Kombination von Modellen in Ensembles genutzt werden. Alle bisherigen Versuche wurden auf einem Cartpole-Simulator durchgeführt, bei dem die wesentlichen Parameter der Simulation (Reibung, Stablänge, Gravitation, etc.) über allen Versuchen konstant waren. Wie in Kapitel 5 geschildert wurde, ist das bei realen Anlagen aufgrund von Verschleiß, Verschlackung und Revisionen aber nicht der Fall.

¹⁶Validierungsdaten dienen dazu festzustellen, wann der optimale Punkt zum Abbruch eines Trainingsvorgangs erreicht ist, bevor das Modell beginnt "auswendig" zu lernen und seine Fähigkeit zur Generalisierung verliert.

¹⁷Die Validierung nur auf zurückliegenden ursprünglichen Daten steht allerdings im Widerspruch zur Motivation dieser Arbeit einen Regler zu trainieren der sich an veränderliche Umgebungen anpasst.

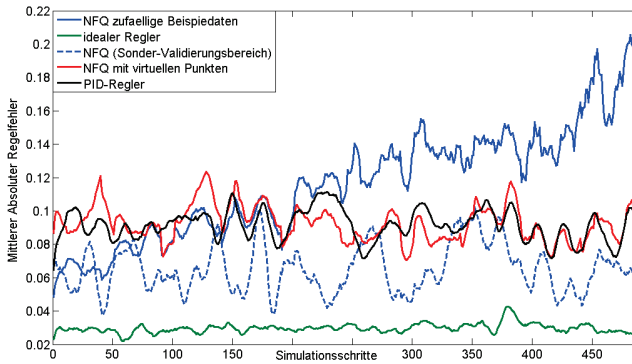


Abbildung 6.7. – Durchschnittliche Ergebnisse der verschiedenen Verfahren mit dem Cartpole-Simulator. Das beste Ergebnis wird vom idealen Regler (grün) erzeugt. Ein mit explorativen Daten trainiertes NFQ (blau) kann anfangs ebenfalls gute Resultate liefern, wird dann aber schlechter. Die Verschlechterung kann verhindert werden (blau gestrichelt) wenn zur Validierung des Trainings nur die explorativen Daten genutzt werden. Ein PID (schwarz) und das NFQ+kDP-Verfahren (rot) erzielen ähnliche Resultate. Der Fehler eines NFQ, das nur auf PID-Daten trainiert wurde, liegt im Schnitt bei 1 und wurde hier nicht abgebildet.

Verändert man die Parameter der Simulation langsam, um diesem Umstand Rechnung zu tragen, so werden auch die Resultate des ursprünglichen PID-Reglers schlechter, während der lernfähige Regler sich anpasst.

6.2.3. Lernen einer sicheren Regelstrategie für eine Müllverbrennungsanlage

Die im letzten Abschnitt gezeigten Experimente beruhten auf künstlichen Simulationen in einer vollständig bekannten Simulationsumgebung. In diesem Abschnitt werden die Ergebnisse von Experimenten mit Reglern in realen Müllverbrennungsanlagen aufgeführt.

Diese Experimente wurden wie folgt durchgeführt: verschiedene NFQ-Regler wurden nach dem in Abschnitt 6.2.2 beschriebenen Schema auf Daten der Anlage für verschiedene Stellgrößen trainiert. Anschließend wurde offline, also ohne den Regler tatsächlich aktiv eingreifen zu lassen, simuliert, was der Regler in einer bestimmten Situation getan hätte. In diesen Experimenten können auch schlechtere Regler über längere und damit aussagekräftigere Zeiträume simuliert werden. Der Nachteil dieses Verfahrens besteht allerdings darin, dass eine offline-Simulation des Reglers keine wirkliche Aussage über die Qualität des Reglers erlaubt. Es kann nur festgestellt werden, inwiefern die Strategie des neuen Reglers dem alten Regler gleicht. In manchen Fällen ist dieser Vergleich aber bereits ausreichend. Sollte der neu trainierte Regler eine vollkommen andere Stellstrategie aufweisen als der bisherige Regler, so kann im Allgemeinen davon ausgegangen werden, dass die neue Strategie nicht korrekt ist und zu Schäden führen würde.

Die Ergebnisse dieser Experimente für das Stellen der Rostgeschwindigkeit in einer MVA sind in Tabelle 6.3 dargestellt. Die Stellgröße Rostgeschwindigkeit wird in der betreffenden Anlage durch ein System von zwei PIDs gefahren. Die lernfähigen NFQs erhielten als Inputgrößen alle Messgrößen, welche auch die PIDs erhalten, sowie zwei weitere Größen, die laut dem verwendeten Merkmalsselektionsverfahren Einfluss auf die Zielgröße haben. Anschließend wurden NFQs zum einen mit dem Datensatz der Anlage und zum anderen mit dem Datensatz sowie künstlichen Datenpunkten gemäß Algorithmus 6 trainiert¹⁸.

Anschließend wurde auf einem Testdatensatz der (open-loop) Betrieb beider NFQ-Regler simuliert. Für die Auswertung wurden folgende Kennzahlen ermittelt: $\bar{\epsilon}$ ist der durchschnittliche Approximationsfehler des NFQ für die Zielgröße (in diesem Fall die Regelabweichung des Dampfes). Würde es sich um ein reines Prognosemodell handeln, wäre diese Größe entscheidend für die Qualität des Modells. Da es sich

¹⁸Die detaillierten Parameter der Versuche finden sich in Anhang B.5.

um einen Regler handelt, ist aber die Stellstrategie entscheidend. Die Größe $a_{diff} = |a_{NFQ} - a_{PID}|$ ist deswegen als Maß für die Abweichung der Stellstrategie des NFQ vom tatsächlichen laufenden PID. Eine geringe Abweichung beider Regler ist allerdings schwer hinsichtlich ihrer Auswirkungen auf den Prozess zu bewerten. Möglicherweise verbessert eine geringfügig andere Strategie die Prozessführung, möglicherweise wird sie verschlechtert. Ohne einen Closed-Loop-Test lässt sich darüber keine Aussage treffen. Im Unterschied dazu sind große Abweichungen der Stellstrategie mit sehr großer Wahrscheinlichkeit schädlich. Die Größe a_{krit} gibt deshalb an, wie viele Stelleingriffe des NFQ sich um mehr als 50% des Stellbereiches von den Aktionen des PID-Systems unterscheiden.

Approx. Zielgröße	$Q_{0,5}(\bar{e})$
NFQ, PID-Daten	0,167
NFQ, PID-Daten + kDP	0,226
kritische Aktionen	$Q_{0,5}(a_{krit})$
NFQ, PID-Daten	35,27%
NFQ, PID-Daten + kDP	3,10%
Abweichung $a_{diff} = a_{NFQ} - a_{PID} $	$Q_{0,5}(\bar{a}_{diff})$
NFQ, PID-Daten	0,358
NFQ, PID-Daten + kDP	0,128

Tabelle 6.3. – Ergebnisse der Simulation des Rostgeschwindigkeit-Reglers auf Daten von MVA 1. Aufgeführt ist der mittlere Fehler bei der Approximation der Dampf-Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen realen PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 100 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 100 Simulationen. Das 10%- und 90%-Quantil sind zusätzlich in Tabelle B.15 in Anhang B.5 abgebildet.

Wie zu sehen ist, ist der durchschnittliche Approximationsfehler $\bar{\epsilon}$ des NFQ, das ohne virtuelle Punkte trainiert¹⁹ wurde, geringer. Dies war auch zu erwarten. Die künstlichen Datenpunkte laut Algorithmus 6 haben jeweils einen eigenen, abgeschätzten Reward. Das NFQ+kDP-Verfahren muss den künstlichen und den tatsächlichen Reward approximieren, weshalb die Approximation auf den realen Testdaten schlechter gelingt als beim normalen NFQ.

Für einen Regler ist dieser Wert aber nicht so bedeutend wie die Qualität der Stellstrategie. Die Abweichung a_{diff} der NFQ-Strategie von der Strategie des realen PID ist beim NFQ+kDP-Verfahren nur halb so groß wie beim normalen NFQ-Training. Die Bedeutung dieser Abweichung lässt sich allerdings schwer bewerten. Wesentlich leichter fällt diese Bewertung bei der Anzahl der kritischen Aktionen. Im Median aller Versuche erzielt das NFQ+kDP-Verfahren hier einen Wert von 3,10%. Das normale NFQ-Training ohne virtuelle Punkte hingegen erzeugt einen Wert von 35,27%. Das bedeutet ca. 35% der Stellengriffe eines NFQ das ohne virtuelle Punkte trainiert wurde, würden um mehr als 50% des Stellbereiches von den Aktionen des bisherigen Reglers abweichen. In Abbildung 6.8 ist dargestellt, wie beide Stellstrategien aussehen würden.

Für das Beispiel der Steuerung der Rostgeschwindigkeit in einer MVA kommt der Einsatz eines NFQ, trainiert ohne virtuelle Punkte, folglich nicht in Frage. Die Stellstrategie eines solchen NFQ weicht viel zu stark von der bisherigen Strategie ab. Mit dem NFQ+kDP-Verfahren hingegen kann ein NFQ trainiert werden, welches im closed-loop-Betrieb getestet werden kann. Die Ergebnisse dieses closed-loop-Betriebs werden in Kapitel 7 vorgestellt.

Neben der Rostgeschwindigkeit wurden obige Experimente auch für die Stellgröße der Zuteilergeschwindigkeit durchgeführt. Die Geschwindigkeit des Zuteilers bestimmt, wie viel Müll pro Zeiteinheit

¹⁹Die genauen Parameter der verwendeten Trainingsverfahren und Datensätze sind in Anhang B.5.1 aufgeführt.

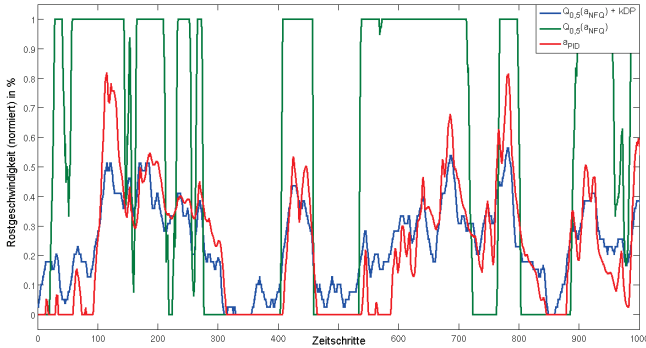


Abbildung 6.8. – Vergleich der Stellstrategie des realen PID (rot) und zweier simulierter NFQ für die Rostgeschwindigkeit einer MVA. Ohne künstliche Datenpunkte (grün) weicht die Stellstrategie des NFQ sehr deutlich von der des PID ab. Eine solche extreme Stellstrategie würde zu großen Problemen führen. Ein NFQ+kDP-Verfahren (blau) hingegen verfolgt eine ähnliche Strategie wie der PID und könnte prinzipiell für den closed-loop-Betrieb aktiviert werden. Die dargestellten Stellstrategien der NFQ-Regler sind der Median aus jeweils 100 Simulationen.

auf den Verbrennungsrost gelangt, und ist eine der maßgeblichen Stellgrößen für den Betrieb des Verbrennungskessels. Eine falsche Stellstrategie hätte ebenso weitreichende Auswirkungen wie bei der Rostgeschwindigkeit. Die Experimente wurden nach demselben Schema wie die Experimente zur Rostgeschwindigkeit durchgeführt. Die Ergebnisse sind in Tabelle 6.4 dargestellt. Ein visueller Vergleich der Stellstrategien des vorhandenen PID-Systems und der beiden NFQ-Arten ist in Abbildung 6.9 zu sehen.

Die Ergebnisse der Stellgröße Rostgeschwindigkeit ähneln denen der Stellgröße Zuteilergeschwindigkeit. Der durchschnittliche Approximationsfehler $\bar{\epsilon}$ des NFQ ohne virtuelle Punkte ist geringer als der des NFQ+kDP-Verfahrens. Die Abweichung a_{diff} der NFQ-Strategie von der Strategie des realen PID ist beim NFQ+kDP-Verfahren wesentlich

Approx. Zielgröße	$Q_{0,5}(\bar{e})$
NFQ, PID-Daten	0,133
NFQ, PID-Daten + kDP	0,145
kritische Aktionen	$Q_{0,5}(a_{krit})$
NFQ, PID-Daten	63,67%
NFQ, PID-Daten + kDP	0%
Abweichung $a_{diff} = a_{NFQ} - a_{PID} $	$Q_{0,5}(\overline{a_{diff}})$
NFQ, PID-Daten	0,530
NFQ, PID-Daten + kDP	0,068

Tabelle 6.4. – Ergebnisse der Simulation des Zuteilergeschwindigkeit-Reglers auf Daten von MVA 1. Aufgeführt ist der mittlere Fehler bei der Approximation des Rewards \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen des aktiven PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 50 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 50 Simulationen. Das 10%- und 90%-Quantil sind zusätzlich in Tabelle B.16 in Anhang B.5 abgebildet.

geringer als beim normalen NFQ-Training²⁰. Auch bei der Anzahl der kritischen Aktionen ist der Unterschied beachtlich. Während das NFQ+kDP-Verfahren im Median und 10%-Quantil aller Versuche keine kritische Abweichung gegenüber der PID-Strategie hat, erzeugt das normale NFQ-Training Strategien, die zu 27,3% beim 10%-Quantil und zu 63,67% im Median kritische Aktionen ausführen.

Für die Zuteilergeschwindigkeit gilt also ebenso wie für die Rostgeschwindigkeit, dass der Einsatz eines NFQ, trainiert ohne virtuelle Punkte, nicht in Frage kommt. Das NFQ+kDP-Verfahren hingegen kann prinzipiell eingesetzt werden. Die Ergebnisse dieses closed-loop-Betriebs werden in Kapitel 7 vorgestellt.

²⁰Die genauen Parameter der verwendeten Trainingsverfahren und Datensätze sind in Anhang B.5.1 aufgeführt.

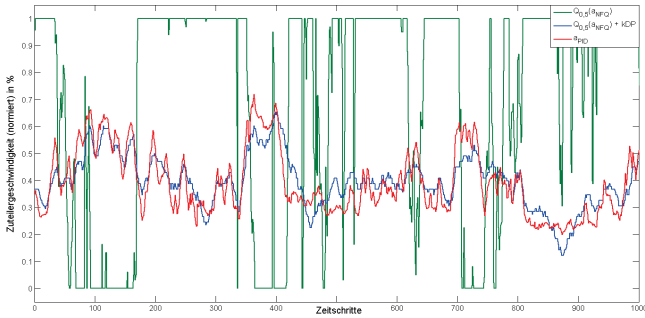


Abbildung 6.9. – Vergleich der Stellstrategie des realen PID (rot) und zweier simulierter NFQ für die Zuteilergeschwindigkeit einer MVA. Die dargestellten Stellstrategien der NFQ-Regler sind der Median aus jeweils 50 Simulationen.

6.2.4. Lernen einer sicheren Regelstrategie für ein Zementwerk

Im vorherigen Abschnitt wurden Experimente zur Sicherheit von Stellstrategien für bereits geregelte Stellgrößen in Müllverbrennungsanlagen vorgestellt. In diesem Abschnitt werden diese Experimente für Stellgrößen in Zementwerken wiederholt. Die Durchführung der Experimente ist identisch zu den in Abschnitt 6.2.3 erläuterten Versuchen. Für eine bereits geregelte Stellgröße werden NFQ-Regler trainiert. Dabei wird zum einen das normale NFQ-Trainingsverfahren auf den ursprünglichen Anlagendaten durchgeführt, zum anderen wird das NFQ+kDP-Verfahren verwendet, bei dem der Anlagendatensatz um zusätzliche künstliche Datenpunkte ergänzt wird, um eine große Abweichung der NFQ-Stellstrategie von der bisherigen Stellstrategie zu verhindern.

Ein erstes Experiment wurde auf der Stellgröße der Ammoniakwassermenge (NH_4OH -Menge) der Selective Non-Catalytic Reduction (SNCR) eines Zementwerks durchgeführt. Die SNCR-

Anlage befindet sich häufig im Steigschacht eines Zementwerks, in der Verbindung zwischen Drehrohrofen und Vorwärmer. Die Aufgabe der SNCR besteht darin durch Eindüsung von Ammoniakwasser die im Sekundärluftstrom enthaltenen Stickoxide (NO_x) in N_2 und H_2O umzuwandeln. Die Menge des eingedüsten Ammoniakwassers muss dabei ständig an die aktuelle Stickoxid-Konzentration angepasst werden. Wird zu viel eingedüst, so bildet sich überschüssiges Ammoniak im Rauchgas, wird zu wenig eingedüst, werden Stickoxid-Grenzwerte überschritten.

Die Experimente wurden nach demselben Schema wie die Experimente in Abschnitt 6.2.3 durchgeführt. Die Ergebnisse sind in Tabelle 6.5 dargestellt.

Approx. Zielgröße	$Q_{0,5}(\bar{e})$
NFQ, PID-Daten	0,368
NFQ, PID-Daten + kDP	0,387
kritische Aktionen	$Q_{0,5}(a_{krit})$
NFQ, PID-Daten	5,72%
NFQ, PID-Daten + kDP	0,52%
Abweichung $a_{diff} = a_{NFQ} - a_{PID} $	$Q_{0,5}(\bar{a}_{diff})$
NFQ, PID-Daten	0,209
NFQ, PID-Daten + kDP	0,068

Tabelle 6.5. – Ergebnisse der Simulation des Ammoniakwasser-Reglers auf Daten von Zementwerk 6. Aufgeführt sind der mittlere Fehler bei der Approximation der NO_x -Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen realen PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 100 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 100 Simulationen. Das 10%- und 90%-Quantil sind zusätzlich in Tabelle B.17 in Anhang B.5 abgebildet.

Die Unterschiede zwischen den beiden NFQ-Trainings²¹ sind bei dieser Stellgröße geringer als bei den Stellgrößen in Müllverbrennungsanlagen. Es bestätigt sich die Vermutung, dass das NFQ+kDP-Verfahren zwar zu einem höheren Approximationsfehler, aber gleichzeitig zu einer geringeren Anzahl an kritischen Aktionen führt. Die Anzahl der kritischen Aktionen beim normalen NFQ-Training ist bei dieser Stellgröße zwar wesentlich geringer als im MVA-Bereich, vergleicht man allerdings die Stellstrategie beider Regler (siehe Abbildung 6.10) so fällt auf, dass die Stellstrategie des NFQ-Trainings ohne künstliche Punkte wesentlich sprunghafter und unregelmäßiger ist als die Strategie des NFQ+kDP-Verfahrens. Die NFQ-Strategie verletzt an diesen Stellen zwar noch nicht die a_{krit} -Bedingung 50% des Stellbereiches von der PID-Strategie abzuweichen, die bestehende Abweichung ist aber wesentlich größer als beim NFQ+kDP-Verfahren und würde wahrscheinlich zu schlechten Ergebnissen führen. Closed-Loop Ergebnisse zum Test dieses Verfahrens finden sich in Kapitel 7.

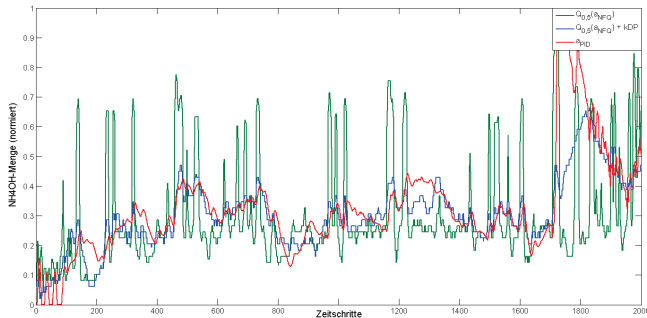


Abbildung 6.10. – Vergleich der Stellstrategie des realen PID (rot) und zweier simulierter NFQ für die NH_4OH -Menge der SNCR eines Zementwerkes. Die dargestellten Stellstrategien der NFQ-Regler sind der Median aus jeweils 100 Simulationen.

²¹Die genauen Parameter der verwendeten Trainingsverfahren und Datensätze sind in Anhang B.5.1 aufgeführt.

Ein weiterer Test des NFQ+kDP-Verfahrens wurde für die Stellgröße des Energieeintrags durchgeführt. Der Energieeintrag beschreibt die Menge an Energie, die durch verschiedene Quellen²² in den Ofen eingebracht wird. Im Unterschied zu den bisher untersuchten Stellgrößen wird diese Stellgröße oftmals nicht durch einen PID-Regler sondern per Hand durch einen Operator eingestellt. Durch den Energieeintrag soll die Temperatur in der Sinterzone möglichst optimal für die aktuelle Rohmehlmenge und Temperatur eingestellt werden. Als Zielgröße der NFQ-Regler wurde deshalb die Abweichung der Temperatur am Ofeneinlaß von einem definierten Sollwert gewählt. In Tabelle 6.6 sind die Ergebnisse der Simulation aufgeführt.

Approx. Zielgröße	$Q_{0,5}(\bar{e})$
NFQ, Anlagen-Daten	0,043
NFQ, Anlagen-Daten + kDP	0,088
kritische Aktionen	$Q_{0,5}(a_{krit})$
NFQ, Anlagen-Daten	40,11%
NFQ, Anlagen-Daten + kDP	0,22%
Abweichung $a_{diff} = a_{NFQ} - a_{real} $	$Q_{0,5}(\bar{a}_{diff})$
NFQ, Anlagen-Daten	0,438
NFQ, Anlagen-Daten + kDP	0,110

Tabelle 6.6. – Ergebnisse der Simulation des Energieeintrags auf Daten von Zementwerk 2. Aufgeführt sind der mittlere Fehler bei der Approximation der Ofeneinlaß-Temperatur-Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den durchgeführten Aktionen $a_{diff} = |a_{NFQ} - a_{real}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 50 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 50 Simulationen. Das 10%- und 90%-Quantil sind zusätzlich in Tabelle B.18 in Anhang B.5 abgebildet.

²²Unter anderem: Kohle- und Ölmenge am Hauptbrenner, Reifenzugabe und Menge der Sekundärbrennstoffe.

Auch in diesem Versuch bestätigen sich die bisherigen Ergebnisse. Das NFQ+kDP-Verfahren führt zu einer schlechteren Approximation des Zielwertes, aber zu wesentlich weniger als "kritisch" eingestuften Aktionen. In Abbildung 6.11 ist ein Vergleich der Regelstrategien dargestellt. Es ist deutlich zu erkennen, dass ein NFQ-Regler, der ohne zusätzliche künstliche Punkte trainiert²³ wurde, in sehr vielen Situationen den maximalen Energieeintrag wählen würde. Eine solche Strategie hätte Schäden am Drehrohren durch Überhitzung zur Folge und würde sehr schnell zu einer Deaktivierung des NFQ durch einen Operator führen. Die Strategie des NFQ+kDP-Verfahrens ist wesentlich ausgeglichener.

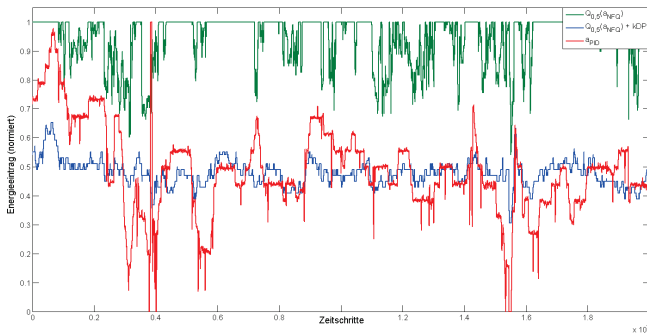


Abbildung 6.11. – Vergleich der realen Stelleingriffe (rot) und zweier simulierter NFQ für den Energieeintrag eines Zementwerkes. Die dargestellten Stellstrategien der NFQ-Regler sind der Median aus jeweils 50 Simulationen.

Allerdings sind auch beim NFQ+kDP-Verfahren Abweichungen zur Operator-Strategie zu erkennen. Diese Abweichungen sind größer als bei den bisherigen Tests und lassen sich auf folgende Faktoren zurückführen:

²³Die genauen Parameter der verwendeten Trainingsverfahren und Datensätze sind in Anhang B.5.1 aufgeführt.

- Ein menschlicher Operator benutzt zur Entscheidung mehr Faktoren als ein PID-Regler. Nicht alle Entscheidungsaspekte finden sich in Informationskanälen, die im Datensatz des NFQ enthalten sind.
- Ein menschlicher Operator stellt nicht kontinuierlich, sondern nur in gewissen Zeitabschnitten (beispielsweise einmal pro Stunde). Dadurch ergeben sich Mehrdeutigkeiten im Datensatz, und der Grund für einen Stelleingriff lässt sich allein aus den Daten schwieriger identifizieren.
- Die konkrete Stellstrategie unterscheidet sich mitunter deutlich von Operator zu Operator. Auch dies führt zu Mehrdeutigkeiten.

Für den Fall, dass eine Stellgröße durch menschliche Operatoren gestellt wird, lässt sich das NFQ+kDP-Verfahren zwar anwenden, die Resultate sind aber nicht so überzeugend wie im Fall der Stellstrategie durch einen PID-Regler.

Falls eine Stellgröße durch einen Operator gestellt wird, ist demzufolge eine Modifikation oder Erweiterung von Algorithmus 6 sinnvoll. Eine solche Erweiterung ist auch vor dem Hintergrund von selten auftretenden Zuständen, die aber bestimmte Aktionen der Stellgröße erfordern, sinnvoll. Diese Erweiterung wird im nächsten Abschnitt vorgestellt.

6.3. Seltene Zustände und Expertenwissen

Im letzten Abschnitt wurde erläutert, wie durch Hinzufügen virtueller Punkte ein zu großes Abweichen des Reglers von der bisherigen Stellstrategie verhindert werden kann.

Neben dieser Anwendung können virtuelle Datenpunkte noch bei einem weiteren Schwachpunkt datengetriebener Lernverfahren eingesetzt werden. Diesen Schwachpunkt stellen sehr selten auftretende Zustände dar, die ein bestimmtes, von der normalen Strategie abweichendes Regelverhalten benötigen. Ein Beispiel für einen solchen Zustand ist in Abbildung 6.12 auf den Daten von MVA 1 dargestellt.

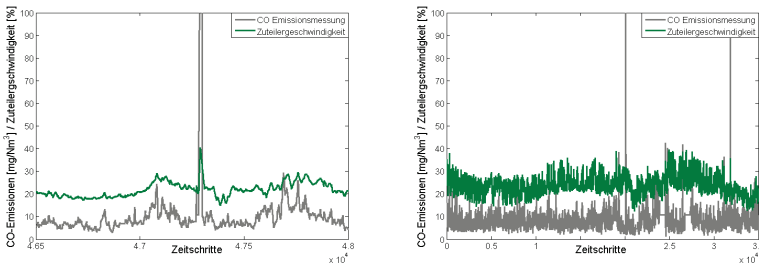


Abbildung 6.12. – Beispiel für das Auftreten von seltenen Zuständen und die entsprechende Reaktion der Stellgröße: grau dargestellt sind die CO -Emissionen in MVA 1. Tritt eine CO -Spitze auf, wie im linken Bild, muss die Stellgröße Zuteiler (grün) sofort stark erhöht werden um eine Grenzwertüberschreitung zu verhindern. Im rechten Bild ist zu sehen, dass solche CO -Spitzen nur äußerst selten vorkommen. Der Zeitbereich des linken Bildes umfasst 12 Stunden, im rechten Bild ist der Verlauf von 12 Tagen zu sehen.

In diesem Beispiel tritt eine hohe CO -Emission auf, die durch eine sofortige Erhöhung des Zuteilers wieder reduziert werden kann. Wird der Zuteiler bei Auftreten einer solchen CO -Spitze nicht erhöht, droht eine Überschreitung des Halbstunden-Grenzwertes für CO -Emissionen.

Eine entsprechende Regelstrategie kann durch ein lernfähiges Verfahren durchaus gelernt werden, wenn hohe *CO*-Emissionen mit einem entsprechenden niedrigen Reward versehen werden. In Abbildung 6.12 ist allerdings auch zu sehen, dass der entsprechende Zustand nur sehr selten auftaucht. Am Beispiel von *CO*-Emissionen ist es durchaus möglich, dass ein bis zwei Wochen vergehen, bevor eine solche *CO*-Spitze auftritt.

Wird ein NFQ auf einem solchen Datenbereich trainiert, so kann es die richtige Reaktion auf eine *CO*-Spitze prinzipiell nicht lernen, da diese in den Trainingsdaten nicht vorhanden ist. Selbst wenn eine *CO*-Spitze im Datensatz vorhanden ist, dann ist deren Anteil mit einer Dauer von 5-15 Minuten bei einer durchschnittlichen Größe des Trainingsdatensatzes von ca. 15000 Minuten so gering, dass ein Lernen der richtigen Strategie unwahrscheinlich ist.

Um trotzdem ein lernfähiges Verfahren zur Regelung des Prozesses zu nutzen und korrekt auf solche seltenen Zustände zu reagieren sind drei Varianten möglich:

1. Es wird detektiert, ob ein seltener und kritischer Zustand auftritt. Ist dies der Fall, wird das lernfähige Verfahren deaktiviert, die Sonderstrategie für den kritischen Zustand angewandt und anschließend das lernfähige Verfahren wieder aktiviert.
2. Bei einer genügenden Menge an prinzipiell vorhandenen Daten kann der Trainingsbereich des Modells so geändert werden, dass entsprechende Zustände nicht mehr so selten in den Trainingsdaten vorkommen und entsprechend besser gelernt werden.
3. Die Zustände und die richtige Reaktion werden von Experten definiert und dem Trainingsdatensatz als virtuelle Punkte hinzugefügt.

Sowohl für Variante 1. als auch für Variante 3. wird ein Experte benötigt, der definiert, wie ein entsprechender Zustand aussieht

und welche Stellstrategie als richtige Reaktion angewandt werden muss. Variante 2. hingegen kann prinzipiell auch ohne einen solchen Experten auskommen, indem Zustände geclustert werden und für seltene Zustände entsprechende Vorkommen in älteren Daten gesucht und zum Trainingsdatensatz hinzugefügt werden. In der Praxis sind allerdings meist nicht genügend Daten vorhanden, oder die Verarbeitung entsprechender Datenmengen ist zu komplex²⁴.

Sowohl Variante 1. als auch Variante 3. wurden in dieser Arbeit angewandt. Variante 1. entspricht der Anwendung einer Fallbackstrategie. Sie wird in Abschnitt 6.5 genauer beschrieben. Der Nachteil dieser Variante besteht darin, dass erkannt werden muss, ob ein seltener kritischer Zustand vorliegt und ob das lernfähige Verfahren falsch reagiert. Beides kostet Zeit und kann sich bei großer Verzögerung negativ auf die Anlage auswirken.

Durch Variante 3. hingegen kann die Regelstrategie für seltene kritische Zustände vom lernfähigen Modell mit erlernt werden. Die dabei entstehenden virtuellen Punkte können genau so in den Trainingsdatensatz aufgenommen werden wie die virtuellen Punkte, die mit Algorithmus 6 in Abschnitt 6.2.1 erzeugt wurden. Der in dieser Arbeit verwendete Ablauf zur Erzeugung der virtuellen Punkte für seltene kritische Zustände ist dann wie in Algorithmus 7 dargestellt.

In jedem Schritt des Algorithmus werden zwei Datenpunkte erzeugt: Datenpunkt η_j^{right} mit einer richtigen Stellstrategie für den seltenen Zustand und η_j^{false} mit einer falschen Strategie. Dazu wird ein Datenpunkt η_{rand} zufällig aus der Menge der vorhandenen Datenpunkte gewählt. Im Zustandsvektor s_{rand} dieses Datenpunktes wird dann die Bedingung s_{art} für den seltenen Zustand an der Stelle k eingefügt. Am obigen Beispiel von MVA 1 müsste also für den CO -Messwert ein sehr hoher Wert eingetragen werden²⁵.

²⁴Siehe entsprechende Anmerkungen in Kapitel 5.3.1, wo ein ähnliches Problem mit dem Verfahren der Datenakkumulation auftrat.

²⁵In vielen Fällen kann der seltene Zustand nur über mehrere Messwerte definiert werden. Dann müssen mehrere Stellen im Zustandsvektor geändert werden.

Algorithmus 7 Künstliche Datenpunkte für seltene Zustände

Eingabe: Datensatz S_{init} mit $\eta_i = (s, a, s', r), \eta_i \in S_{init}, i = 1 \dots n$,
Rate künstlicher Punkte pro Cluster n_{kp} , Rewardabschlag für falsche
Strategie r_{kp} ;

Ausgabe: Datensatz zum Trainieren eines Modells S_{train}

```

 $S_{train} \leftarrow S_{init}$ 
for  $j = 1$  to  $\frac{1}{2}n_{kp} \cdot |C_i|$  do
   $\eta_{rand} = (s_{rand}, a_{rand}, s'_{rand}, r_{rand}), \eta_{rand} \in S_{init}$  // zufällig ge-
  wählter Punkt aus dem Datensatz
   $s_{art} = (s_{rand}^1, \dots, s_{rand}^{k-1}, s_{art}^k, s_{rand}^{k+1}, \dots, s_{rand}^l)$  // setze im Zustand
  die Bedingung für den seltenen Zustand ein
   $s'_{art} \leftarrow f$  // Folgezustand = Flag
   $\eta_j^{right} = (s_{art}, a_{art}^{right}, s'_{art}, r_{rand})$  // virt. Punkt mit richtiger Stra-
  tegie
   $\eta_j^{false} = (s_{art}, a_{art}^{false}, s'_{art}, r_{rand} - r_{kp})$  // virt. Punkt mit falscher
  Strategie
   $S_{train} \leftarrow S_{train} \cup \{\eta_j^{right}, \eta_j^{false}\}$ 
end for

```

Für den Folgezustand der virtuellen Punkte wird wie in Algorithmus 6 ein Flag f gesetzt. Anschließend müssen durch Expertenwissen die richtige Stellstrategie a_{art}^{right} und die falsche Stellstrategie a_{art}^{false} definiert werden. Im Fall des Zuteilers in MVA 1 wäre die richtige Strategie ein hoher Wert für den Zuteiler und die falsche Strategie ein niedriger Wert. Der Reward für die falsche Strategie wird entsprechend um r_{kp} , verringert und die beiden künstlichen Datenpunkte werden zum Trainingsdatensatz hinzugefügt.

Der Ansatz, auf Basis von Expertenwissen künstliche Datenpunkte zu generieren und zum Training eines neuronalen Netzes zu verwenden, um mehr funktionale Sicherheit zu erhalten, ist nicht neu. Das *Safety Lifecycle* Modell aus [Kurd and Kelly, 2003] arbeitet mit einem ähnlichen Ansatz, allerdings wird dort sämtliches Wissen symbolisch vorgegeben und muss anschließend in Datenbeispiele für das Netzwerk übersetzt werden. Der Beitrag der vorliegenden Arbeit besteht darin, Expertenwissen mit den bereits vorhandenen Anlagendaten zu kombinieren, so die funktionale Sicherheit zu verbessern und das Verfahren auf realen Daten der industriellen Feuerung zum Einsatz zu bringen.

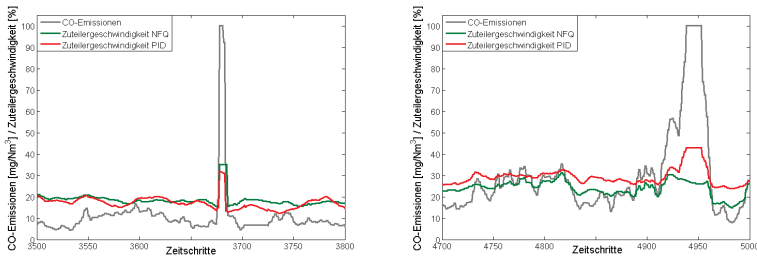


Abbildung 6.13. – Vergleich der Reaktion zweier NFQ-Modelle auf CO -Spitzen (grau) in MVA 1. Im linken Bild ist die Reaktion eines NFQ-Modells (grün) dargestellt, welches mit zusätzlichen virtuellen Punkten trainiert wurde, im rechten Bild hingegen die Reaktion eines Modells ohne virtuelle Punkte in den Trainingsdaten. Zum Vergleich ist jeweils die Reaktion des konventionellen PID-Systems (rot) aufgeführt.

In Abbildung 6.13 sind die simulierten Stellstrategien zweier NFQs zu sehen, von denen eines mit diesen virtuellen Punkten trainiert wurde und das andere nicht. Das mit zusätzlichen virtuellen Punkten trainierte NFQ reagierte deutlich besser auf *CO*-Spitzen als das NFQ, das ohne virtuelle Punkte trainiert wurde.

Durch das Hinzufügen von virtuellen Punkten lassen sich also auch Strategien für seltene kritische Zustände durch lernfähige Verfahren behandeln. Die Nachteile des hier vorgestellten Verfahrens sind:

- die Charakteristik eines seltenen Zustandes und die "richtige" Strategie müssen bisher durch einen Experten in einer sehr strukturierten Form angegeben werden,
- die Rate der virtuellen Punkte n_{kp} hat Einfluss auf die Stärke der Reaktion. Werden zu wenige Punkte hinzugefügt, werden seltene Zustände weiterhin nicht beachtet. Werden zu viele Punkte hinzugefügt, wird das Modell zu einem "Ausnahmeregler", der zwar die seltenen Zustände, aber nicht mehr den Normalzustand regeln kann.
- Der korrekte Rewardabschlag r_{kp} für eine falsche Strategie ist schwierig zu definieren. Wird er falsch gewählt, enthalten die Trainingsdaten dadurch Inkonsistenzen beim Zielkanal, und der Lernvorgang wird erschwert.

Viele dieser Nachteile können durch eine Weiterentwicklung des Systems und eine Vereinfachung der Bedienung verbessert werden. Im Ausblick dieser Arbeit in Kapitel 9 werden konkrete Möglichkeiten dazu vorgestellt.

Trotzdem muss bei dem Verfahren der Erzeugung virtueller Punkte um Modelle an seltene Zustände anzupassen immer eine Abwägung zwischen Aufwand und Nutzen erfolgen. In einigen Fällen kann es schneller und sicherer sein, einen konventionellen Fallbackmechanismus für seltene Zustände zu verwenden, wie er in Abschnitt 6.5 vorgestellt wird.

In diesem Abschnitt sollte vor allem dargestellt werden, welche Probleme seltene Zustände bei der Anwendung selbstlernender Modelle bringen und dass es prinzipiell Möglichkeiten gibt diese Probleme innerhalb der Möglichkeiten des Lernverfahrens durch virtuelle Datenpunkte zu lösen.

6.4. Untersuchung der Regelstrategie eines Modells

In Abschnitt 6.2.1 wurden Verfahren vorgestellt, mit denen der Trainingsdatensatz eines lernfähigen Modells erweitert wird. Diese Erweiterung hat den Zweck, eine starke Abweichung der durch das Modell gestellten Stellgröße von der bisherigen Stellstrategie zu vermeiden. Das vorgestellte Verfahren setzt durch die Erweiterung des Datensatzes bereits vor dem Training des Modells an. In diesem Abschnitt wird erläutert, wie nach dem Training des Modells, aber vor dem ersten Closed-Loop Einsatz, die Stellstrategie des Modells überprüft und sicherheitskritische Modelle abgelehnt werden können.

Zwei Varianten zur Überprüfung der Stellgröße wurden dabei bereits im vorherigen Abschnitt genutzt: die Bestimmung der durchschnittlichen Abweichung der neuen und alten Stellstrategie $\overline{a_{diff}}$ und die Anzahl der kritischen Aktionen a_{krit} .

Beide Werte lassen sich bestimmen, indem die Stellstrategie des neuen Modells auf einem Zeitabschnitt simuliert und mit der Stellstrategie des alten Modells oder Reglers verglichen wird. Die Werte für $\overline{a_{diff}}$ und a_{krit} für eine Anzahl n an Vergleichspunkten berechnen sich dann wie folgt:

$$\overline{a_{diff}} = \frac{\sum_1^n |a_{neu} - a_{alt}|}{n} \quad (6.1)$$

$$a_{krit} = \frac{|\{a^i | a_{diff}^i > a_{max}\}|}{n} \cdot 100\% \quad (6.2)$$

a_{max} ist dabei die maximale Abweichung, die eine Aktion des neuen Modells von der Aktion des alten Modells haben darf, um nicht als sicherheitskritisch zu gelten. In den bisherigen Versuchen wurde a_{max} so gewählt, dass es der Hälfte des üblichen Stellbereiches entsprach. Es lassen sich dann die Werte $\overline{a_{diff}}$ und a_{krit} für ein neu trainiertes Modell gegenüber den bisher ausgeführten Aktionen ermitteln. Wählt man für beide Werte Obergrenzen, so können automatisch Modelle abgelehnt werden, die diese Obergrenzen überschreiten und in ihrer Stellstrategie deutlich von der bisherigen Strategie abweichen.

Die beiden vorgestellten Größen zur Evaluierung haben allerdings zwei wesentliche Nachteile:

1. Die Evaluierung erfolgt auf einem bestimmten Datenbereich. Kommen in diesem Datenbereich bestimmte Zustände nicht (oder sehr selten) vor, so kann über die Stellstrategie in diesen Zuständen keine Aussage getroffen werden.
2. Die Evaluierung erfolgt nur gegen die bereits aktive Stellstrategie, ohne das berücksichtigt wird, wie gut diese Stellstrategie tatsächlich ist. Bei ständigem Nachtraining der Modelle kann sich im Laufe der Zeit die Stellstrategie so kontinuierlich verschlechtern. Am Ende würde sogar die ursprüngliche (gute) Strategie als sicherheitskritisch abgelehnt werden, da sie der mittlerweile aktiven (schlechten) Stellstrategie widerspricht.

Ist Expertenwissen vorhanden, wie die Stellgröße prinzipiell zu stellen ist und wie die Stellgröße in Bezug auf verschiedene Zielgrößen reagieren soll, so lassen sich diese Nachteile durch zusätzliche Analysen vermeiden.

6.4.1. Einfluss der Stellgröße auf Zielgrößen

Ein wichtiger Faktor bei der Analyse eines Modells ist, ob der Einfluss der Stellgröße auf andere Größen des Modells korrekt modelliert ist. Ein NFQ-Modell enthält als Basis ein MLP-Netzwerk, welches mehrere Eingangsgrößen und eine Ausgangsgröße hat²⁶. Die Lernverfahren modellieren die Gewichte des Netzwerkes so, dass durch das Netzwerk eine möglichst gute Approximation der Ausgangsgröße auf dem verwendeten Datensatz entsteht. Dabei müssen aber nicht alle Inputgrößen gleichermaßen genutzt werden. Je nach Verteilung der Zustände im Datensatz und Redundanz der Inputgrößen ist es möglich, dass einige Inputgrößen nur einen sehr geringen Einfluss auf die Approximation der Zielgröße durch das Netzwerk haben. Handelt es sich bei diesen Inputgrößen um wichtige Messgrößen, für die eine bestimmte Reaktion der Stellgröße erwartet wird, so ist aber ein deutlicher Einfluss notwendig.

Im Folgenden wird ein Verfahren vorgestellt, mit dem sich der Einfluss einzelner Inputgrößen auf die Ausgangsgröße eines Modells und auf die Stellstrategie untersuchen lässt. Ist für eine bestimmte Inputgröße bekannt, dass ein hoher Einfluss vorhanden ist, so lassen sich durch diese Analyse Modelle ablehnen, bei denen der entsprechende Einfluss der Inputgröße zu schwach ist.

Der Algorithmus ist nicht beschränkt auf NFQ- oder MLP-Modelle, sondern lässt sich auf alle Modelle anwenden, welche mehrere Eingangsgrößen auf eine Ausgangsgröße abbilden. Eine der Eingangsgrößen sollte eine Stellgröße sein, die gesondert behandelt wird²⁷. Es ist notwendig den Minimal- und Maximalwert der Stellgröße und den üblichen Messbereich aller anderen Größen zu kennen.

²⁶Prinzipiell kann eine MLP auch mehrere Ausgangsgrößen haben, dieser Fall wird allerdings in dieser Arbeit nicht untersucht.

²⁷Es ist problemlos möglich, den Algorithmus für Modelle mit mehreren Ausgangs- und Stellgrößen zu erweitern.

Weiterhin muss eine Menge an Inputzuständen vorhanden sein. Dies kann ein Datensatz der Anlage sein, es können aber auch von einem Experten definierte Zustände sein. Die zweite Option ist empfehlenswert, falls direkt Zustände getestet werden sollen, die im normalen Datensatz nur selten enthalten sind. Die Funktionsweise des Algorithmus ist in Algorithmus 8 dargestellt.

Algorithmus 8 Bestimmung des Einflusses von Inputgrößen auf die Zielgröße

Eingabe: trainierter Regler/Modell N , Datensatz S mit $\eta_i = (s^1, \dots, s^m), \eta \in S, i = 1 \dots n$, Anzahl Bins b , Grenzen $s_{max}^j = (s_{max}^1, \dots, s_{max}^m), s_{min}^j = (s_{min}^1, \dots, s_{min}^m), t_{max}, t_{min}$

Ausgabe: Einfluss der Inputgrößen c^1, \dots, c^m

```

 $c_{max} \leftarrow \sigma(\{t_{max}, t_{min}\})$  // theoretisch maximaler Einfluss
for  $j = 1$  to  $m$  do
  for all  $\eta_i \in S$  do
     $S_{temp} \leftarrow \{\}$ 
    for  $h = s_{min}^j$  to  $s_{max}^j$  in  $\frac{s_{max}^j - s_{min}^j}{b-1}$  stepsize do
       $\eta_{test} \leftarrow (s_i^1, \dots, s_i^{j-1}, h, s_i^{j+1}, \dots, s_i^m)$  // ersetze  $s_i^j$  durch  $h$ 
       $S_{temp} \leftarrow S_{temp} \cup N(\eta_{test})$  // Antwort des Modells auf  $\eta_{test}$ 
    end for
     $S_i^j \leftarrow \sigma(S_{temp})$ 
  end for
   $c^j \leftarrow \frac{100}{c_{max}} \cdot \frac{\sum_{i=1}^n S_i^j}{n}$ 
end for

```

Für jede Inputgröße s^1, \dots, s^m wird der Einfluss c^1, \dots, c^m dieser Größe auf die Zielgröße bestimmt. Es ist dabei unerheblich, ob es sich bei einer Inputgröße s^j um eine normale Meßgröße oder um eine Stellgröße handelt. Für jeden Datenpunkt η_i des vorgegebenen Datensatzes S wird dazu die Veränderung der Modellantwort bei

Veränderung dieser Inputgröße bestimmt. Die Standardabweichung σ über den Antworten dient als Maß für den Einfluss. Ist die Standardabweichung Null, so ist die Antwort des Modells immer gleich, unabhängig von der Veränderung des gewählten Inputs. Das bedeutet, dieser Input hat keinen Einfluss auf die Modellantwort. Dieser Versuch wird für alle Datenpunkte durchgeführt und anschließend das Ergebnis gemittelt. Um eine aussagekräftigere Zahl zu erhalten, wird das Ergebnis mit der theoretisch maximalen Veränderung der Zielgröße normiert.

Dieser Algorithmus wurde auf die Modelle und Daten des Versuches zur Zuteilergeschwindigkeit aus Kapitel 6.2.3 angewendet. Die Ergebnisse sind in Tabelle 6.7 aufgeführt.

Input-/Stellgröße	I_1	I_2	I_3	I_4	I_5
NFQ	1,8%	0,3%	3,1%	3,5%	4,0%
NFQ+kDP	1,5%	0,4%	2,9%	3,0%	3,7%
Input-/Stellgröße	I_6	I_7	I_8	I_9	A_1
NFQ	46,0%	11,5%	15,7%	1,5%	3,8%
NFQ+kDP	49,5%	19,1%	15,8%	4,5%	9,6%

Tabelle 6.7. – Analyse des Einflusses von 10 Inputgrößen auf die Zielgröße eines Modells. Bei der zehnten Inputgröße handelt es sich um eine Stellgröße, welche anhand des Modells gestellt werden soll.

Wie zu sehen ist, ergeben sich sehr unterschiedliche Werte für die einzelnen Inputs. Die Inputgrößen I_1 - I_5 haben verfahrenstechnisch einen schwachen Einfluss auf das Ziel, während Input I_6 - I_7 einen sehr großen Einfluss haben. Vergleicht man hingegen die Werte der beiden Trainings-Verfahren, so sind sie untereinander sehr ähnlich. Einzig bei der Stellgröße, dem Zuteiler, gibt es einen deutlichen Unterschied. Während beim normalen NFQ-Training die Stellgröße Zuteiler nur einen Einfluss von 3,8% auf die Zielgröße hat, ist beim NFQ+kDP-Verfahren der Einfluss mit 9,6% mehr als doppelt so hoch. Dieses Verhalten ist erwünscht, da im Modell die Stellgröße einen großen Einfluss auf das Ziel haben soll.

Durch das vorgestellte Verfahren kann ermittelt werden, wie groß der Einfluss einer Inputgröße auf die Zielgröße des Modells ist. Wird das Modell nicht nur zur Prognose oder als Softsensor, sondern als Regler genutzt, der eine Stellgröße verändert, dann ist allerdings nicht nur der Einfluss einer Größe, sondern auch deren Wirkung auf die Stellstrategie interessant. Dieser Zusammenhang wird im nächsten Abschnitt untersucht.

6.4.2. Wirkrichtung der Stellgröße

Fall es sich bei dem trainierten Modell um einen potentiellen Regler handelt, ist die Abhängigkeit der Stellstrategie von den Inputs von weit größerem Interesse als der Einfluss auf die Zielgröße. Algorithmus 8 lässt sich aber durch eine einfache Modifikation auch dazu nutzen. Statt im Schritt $S_{temp} \leftarrow S_{temp} \cup N(\eta_{test})$ die Ausgabe des Modells auf die Eingabe zu berechnen, wird stattdessen die Menge der optimalen Aktionen A_{temp} bestimmt. Über die Standardabweichung der Aktionen $\sigma(A_{temp})$ lässt sich dann der Einfluss einer Inputgröße auf die Wahl der besten Aktion bestimmen.

Während bei dem bisherigen Verfahren die theoretisch maximale Standardabweichung der Zielgröße $c_{max} \leftarrow \sigma(\{t_{max}, t_{min}\})$ zur Normierung des Ergebnisses diente, kann bei der Ermittlung des Einflusses der Aktionen die reale Standardabweichung der Stellgröße als Bezugsgröße gewählt werden.

In Tabelle 6.8 sind diese Einflüsse für den im letzten Abschnitt erläuterten Versuch dargestellt.

Wie zu sehen ist, hat beim normalen NFQ-Training vor allem I_6 einen sehr starken Einfluss. Mit über 100% der normalen Stellgrößen-Standardabweichung ist davon auszugehen, dass diese Größe zu stark in den Aktionen berücksichtigt wird. Ein großer Einfluss findet sich außerdem bei den Größen I_1 , I_3 und I_9 .

Inputgröße	I_1	I_2	I_3	I_4	I_5
NFQ	29,1%	2,7%	26,2%	21,3%	11,5%
NFQ+kDP	4,9%	4,3%	2,8%	2,3%	1,8%
Inputgröße	I_6	I_7	I_8	I_9	
NFQ	179,6%	28,8%	13,9%	10,9%	
NFQ+kDP	71,1%	73,3%	8,6%	35,3%	

Tabelle 6.8. – Analyse des Einflusses von 9 Inputgrößen auf die Stellgröße eines Modells.

Diese tabellarische Auswertung stellt zwar dar, wie groß der Einfluss einzelner Inputs auf die Stellgröße ist, es ist aber immer noch schwer zu beurteilen ob der Regler eine sinnvolle Strategie verfolgen würde oder nicht.

Wertet man die Matrix der bestmöglichen Aktionen A_{temp} jedoch nicht in Form einer Tabelle, sondern durch eine Grafik aus, so lassen sich wesentlich besser Auswirkungen erkennen. In Abbildung 6.14 ist eine solche grafische Auswertung für die Stellgröße und die vier Inputgrößen I_1 , I_3 , I_6 und I_9 dargestellt.

Es sind deutliche Unterschiede zwischen dem Stellverhalten des normalen NFQ-Verfahrens und des NFQ+kDP-Verfahrens zu erkennen. Für eine Abschätzung der Sicherheit eines Modells ist es zweckdienlich ebenfalls die tatsächlich in der Anlage aufgetretenen Werte der Stellgröße zu vergleichen. In Abbildung 6.14 wurden jeweils das 20%-Quantil und das 80%-Quantil der realen Stellwerte abgebildet. Ein neues Modell sollte sich innerhalb dieser Grenzen bewegen oder zumindest nicht stark von ihnen abweichen. Beim NFQ+kDP-Verfahren werden diese Grenzen größtenteils eingehalten, während die durch normales NFQ-Training erzeugten Modelle teilweise starke Abweichungen aufweisen.

Abbildungen wie in Abbildung 6.14 können nach jedem Training von einem Experten ausgewertet und so die Strategie des Modells

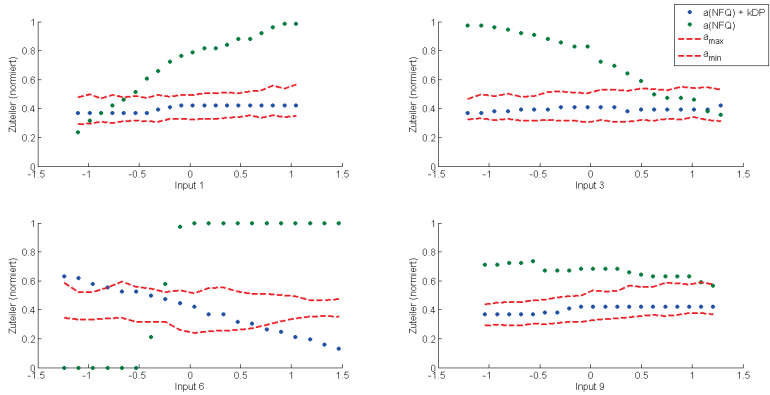


Abbildung 6.14. – Durchschnittlich beste Aktion der Stellgröße Zuteiler in Abhängigkeit der Inputgrößen I_1 , I_3 , I_6 und I_9 . Dargestellt sind jeweils die Aktionen der nach dem NFQ-Verfahren (grün) und dem NFQ+kDP-Verfahren (blau) trainierten Modelle. Zusätzlich dargestellt sind das 20%-Quantil und das 80%-Quantil (rot gestrichelt) der realen Werte der Zuteiler in den entsprechenden Zuständen.

evaluiert werden. Die Kombination aus Wissen über den Einfluss eines Inputs auf die Stellstrategie einerseits und aus dem Wissen über das Stellverhalten in Abhängigkeit vom Wert des Inputs andererseits lassen eine Bewertung des Stellverhaltens des Modells noch vor der Inbetriebnahme zu.

Kann das Expertenwissen zur Evaluierung des Modells in formale Regeln übertragen werden, so kann die Auswertung des Modells auch automatisch geschehen. Ein automatisches Nachtraining ist dadurch möglich, da unsichere Modelle mit bedeutend höherer Wahrscheinlichkeit abgelehnt werden, bevor sie zum Einsatz kommen.

6.5. Ensembles

In den vorherigen Kapiteln wurden Verfahren beschrieben, mit denen die Regelungsstrategie eines lernfähigen Modells bereits vor der Close-Loop Zuschaltung des Modells verbessert werden kann. Alle diese Maßnahmen tragen dazu bei ein Modell zu erhalten, das eine als sicher eingestufte Strategie verfolgt und eine möglichst gute Regelgüte erzeugt.

Da alle vorgestellten Tests offline erfolgen, ist diese Auswertung aber immer unvollständig. Selbst kleine Änderungen der Strategie können im Closed-Loop-Betrieb langfristig zu schlechten Ergebnissen führen. Ebenso ist es schwer zu prüfen wie sich das Modell in allen nur denkbaren, bisher aber nicht aufgetretenen Situationen verhalten würde und ob dieses Verhalten gut oder schädlich ist.

Es bleibt also ein Restrisiko, wenn das Modell aktiviert wird. Um den möglichen Schaden, der durch dieses Restrisiko auftreten könnte, möglichst gering zu halten, ist es notwendig während des Closed-Loop-Betriebes des Modells die Qualität der Prozessführung zu überwachen und im Falle einer falschen Strategie eine Fall-Back-Strategie zu aktivieren.

Beispiele für eine solche Überwachung mit Fall-Back-Strategie finden sich in [Hans et al., 2008, Riedmiller et al., 2007a]. In dieser Arbeit werden zwei Ansätze angewendet, um das Risiko einer falschen Regelstrategie im Onlinefall zu minimieren:

1. Kombination mehrerer Regelstrategien mit Gütemaß,
2. Fallback-Strategie im Fall einer Regelüberschreitung.

Der erste Ansatz entspricht dem Ensemble-Verfahren EWm, das in Abschnitt 5.4.2 dieser Arbeit vorgestellt wurde. Es werden mehrere Prozessmodelle trainiert. Jedes Modell erhält einen Gütewert, der sich auf die bisherige Regelqualität des Modells im Closed-Loop-Betrieb

bezieht. Die Ausgaben aller Modelle werden entsprechend ihres Gütwertes zu einer Gesamt-Stellausgabe verrechnet. Die wesentlichen Unterschiede zu dem in Abschnitt 5.4.2 vorgestellten Verfahren bestehen hierbei darin, dass zum einen Stellgrößenausgaben statt Prognosen zu einer Ausgabe verrechnet werden. Zum anderen wird die Güte eines Modells nicht offline anhand seiner Prognosequalität bestimmt, sondern online anhand der Prozessgüte.

Die Bestimmung dieser Güte ist aufwendig, da der Einfluss eines einzelnen Modells auf die Gesamtleistung²⁸ bestimmt werden muss, und sie kann viel Zeit beanspruchen. Eine Kombination mehrerer lernfähiger Modelle schützt auch nicht vor Fehlern in der Regelstrategie. Da alle Modelle auf den bisher erhaltenen Daten arbeiten, zeigen alle Modelle eine Tendenz häufig aufkommende Fälle gut zu regeln, bei seltenen Situationen aber unvorhersehbar zu reagieren. Aus diesem Grund wird in dieser Arbeit stets eine Fallback-Strategie in Form eines konventionellen Reglers verwendet.

Die Aktivierung der Fallbackstrategie geschieht nach folgendem Schema:

1. Es werden Bedingungen definiert, mit denen erkannt wird, dass möglicherweise eine falsche Regelstrategie gefahren wird. Die Bedingungen können beispielsweise ein hoher Emissionswert bei Schadstoffen oder eine Abweichung der Dampfleistung vom Sollwert sein.

²⁸Die Aufspaltung einer Gesamtleistung auf mehrere Module wird auch als *Reward Dekomposition* bezeichnet. Ausführliche Erläuterungen zu diesem Thema finden sich innerhalb des APD-Forschungsprojektes, das in Anhang A.4 beschrieben wird und in [Eisenbach, 2009].

2. Trifft eine dieser Bedingungen zu, wird das Verhalten der vom lernfähigen Modell gesteuerten Stellgröße verglichen mit dem Verhalten des konventionellen Reglers (falls dieser aktiv gewesen wäre).²⁹
3. Gibt es eine Differenz zwischen den beiden Regelstrategien, die größer ist als ein definierter Schwellwert, so wird das lernfähige Modell deaktiviert und der Fallback-Regler aktiviert.

Durch dieses Verfahren wird sichergestellt, dass im Falle einer falschen Regelstrategie auf eine Fallback-Lösung umgestellt wird.

Wurde das lernfähige Modell deaktiviert, stellt sich die Frage ob und wann es wieder aktiviert werden soll. Drei Möglichkeiten sind denkbar:

1. Sobald alle Bedingungen für eine Deaktivierung des Modells wieder innerhalb der Grenzen sind (ggf. mit einer Hysterese), wird das lernfähige Modell wieder aktiviert.
2. Die Modelle müssen manuell durch einen menschlichen Operator aktiviert werden, indem zum Beispiel eine Störungsmeldung quittiert wird.
3. Falls eine adaptive Anpassung der Modelle implementiert wurde (siehe Kapitel 5), bleiben die Modelle deaktiviert, bis ein neues Modell trainiert wurde. Danach erfolgt automatisch die Aktivierung.

Das erste Verfahren birgt das höchste Risiko. War das lernfähige Modell verantwortlich für die Abweichung, die zur Deaktivierung führte, so ist es wahrscheinlich, dass bei einem erneuten Zuschalten

²⁹Der Vergleich eines aktiven Closed-Loop-Reglers mit eines inaktiven Open-Loop-Regler ist natürlich theoretisch inkorrekt. Insbesondere durch aktive I-Anteile können sich hierbei Differenzen ergeben, die beim Vergleich mit einem aktiven Regler nicht entstehen würden. Da aber effektiv nur ein Regler aktiv die Stellgröße stellen kann, existiert hierzu keine Alternative. Im praktischen Einsatz hat sich dieses Verfahren trotz der theoretischen Bedenken als sehr praktikabel erwiesen.

auch erneut eine falsche Regelstrategie gefahren wird und das Problem erneut auftritt. Dieser Ansatz wird in dieser Arbeit deshalb nicht verwendet.

Stattdessen wird im Rahmen dieser Arbeit so oft wie möglich der dritte Ansatz verwendet. Dieses Schema ist in Abbildung 6.15 noch einmal als Programmablaufplan zu sehen. Bei einigen Anlagen besteht der Betreiber allerdings darauf, dass eine solche Störung ebenfalls durch einen Operator quittiert werden muss. In diesen Fällen wurde eine Kombination der Ansätze 2. und 3. implementiert.

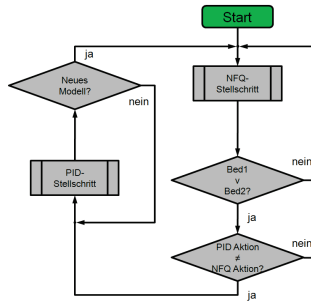


Abbildung 6.15. – Programmablaufplan zur Aktivierung einer Fallbackstrategie. Im dargestellten Beispiel werden zwei Bedingungen überprüft um zu entscheiden, ob das Modell deaktiviert wird, es können aber auch mehr Bedingungen verwendet werden.

In Abbildung 6.16 ist ein Beispiel für die Wirkungsweise dieses Verfahrens bei der Kontrolle eines lernfähigen Reglers in einer Müllverbrennungsanlage dargestellt.

Langfristige empirische Vergleiche mit und ohne diese Sicherheitsvorkehrungen auf realen Anlagen lassen sich nicht durchführen. Gibt es keine Fallbackstrategie wie in diesem Abschnitt erläutert, so führt ein signifikantes Fehlverhalten des Reglers entweder zu einer manuellen Abschaltung des Systems durch einen Operator oder zu einem

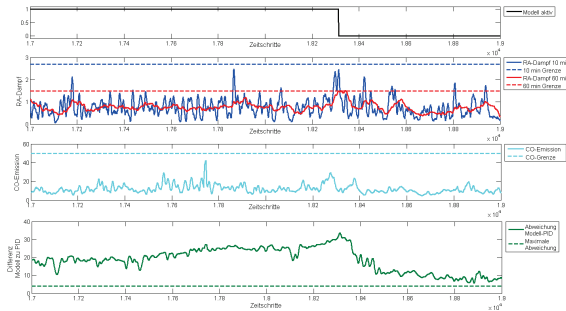


Abbildung 6.16. – Beispiel für die Deaktivierung eines Dampf- und *CO*-Reglers. Das Modell wird deaktiviert, nachdem die mittlere Regelabweichung über 60 Minuten (rot) eine definierte Grenze (rot gestrichelt) überschritten hat. Außerdem wurde geprüft, ob der Unterschied (grün) zwischen NMPC-Regler und PID-Regler-Stellausgabe größer war als eine definierte Grenze (grün gestrichelt). Da das der Fall war, wurde das Modell deaktiviert.

wirtschaftlichen Schaden. Beide Fälle sind vom Anlagenbetreiber nicht gewünscht, weswegen ein solches Sicherheitssystem Grundvoraussetzung für langfristige Versuche mit adaptiven Reglern ist und Versuche ohne ein solches System nicht durchgeführt werden.

6.6. Fazit

Dieses Kapitel behandelte die Gewährleistung der Funktionalität und Sicherheit lernfähiger Regler innerhalb der Prozessführung von Verbrennungsanlagen. Zuerst wurde dazu definiert, was unter Sicherheit zu verstehen ist, nämlich die Vermeidung einer Regelstrategie, die zur Verletzung von Prozesswert-Grenzen führen würde.

Das Risiko, dass eine solche Überschreitung durch ein lernfähiges Modell verursacht wird, hängt von der Einsatzart des Modells ab. Modelle, die als Softsensor oder zur Prognose von Messwerten eingesetzt werden, stellen dabei einen geringere Gefährdung dar als Modelle, die direkt eine Stellgröße beeinflussen. NMPC-Regler lassen sich außerdem schwerer überprüfen als Prognosemodelle, da eine Abschätzung der Korrektheit einer Stellstrategie offline wesentlich schwieriger ist als die Kontrolle einer Prognose.

Es wurden drei Möglichkeiten definiert, das Risiko einer falschen Stellstrategie durch einen lernfähigen Regler zu mindern:

1. Bereits vor dem Training eines Modells kann durch Hinzufügen von künstlichen Datenpunkten die Wahrscheinlichkeit einer falschen Stellstrategie verringert werden [Steege et al., 2010].
 2. Ist das Modell trainiert, kann durch offline-Analyse des Stellverhaltens in unterschiedlichen Zuständen ein wesentliches Abweichen des Modells von der gewünschten Stellstrategie festgestellt werden. Modelle, die eine grundsätzlich falsche Stellstrategie verfolgen, können so vor der ersten Zuschaltung deaktiviert werden.
 3. Wurde das Modell deaktiviert, so kann durch Kombination mehrerer Modelle in einem Ensemble und durch konventionelle Fallbackstrategien ein Schaden für die Anlage im Fall einer falschen Strategie minimiert werden.
-

Alle vorgestellten Ansätze haben als Voraussetzung, dass bereits eine Regelstrategie für die jeweilige Stellgröße existiert. Dies ist allerdings bei der überwiegenden Anzahl der Stellgrößen in den in Kapitel 2 geschilderten Anlagen der Fall³⁰.

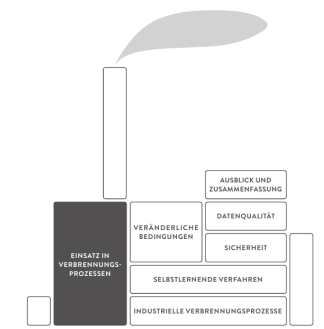
Am Beispiel eines einfachen Cartpole-Simulators sowie an den Daten realer Zement- und Müllverbrennungsanlagen wurde gezeigt, wie die Anwendung aller drei Methoden zu einer deutlichen Verringerung der Wahrscheinlichkeit sicherheitskritischer Regler oder zu einer Erkennung und dem anschließenden Ausschluss dieser Modelle vor der Aktivierung führte.

Die in diesem Kapitel vorgestellten Methoden sind deshalb von erheblicher Bedeutung für den Einsatz von lernfähigen Modellen für bereits geregelte Größen.

Nachdem die in diesem Abschnitt vorgestellten Methoden untersucht wurden und als Prototyp in einer MVA und einem Zementwerk online zum Einsatz kamen, wurden sie in die Software zur Regelung von Prozessen der Firma STEAG Powitec GmbH übernommen. Die langfristigen Resultate der Verfahren im Vergleich mit konventionellen Reglern und älteren NMPC-Verfahren werden in Kapitel 7 dargestellt.

³⁰Wenn für eine Stellgröße bisher noch keine Regelstrategie existierte, muss zuerst eine Explorationsfahrt dieser Stellgröße durchgeführt werden um Daten für ein Lernverfahren zu gewinnen. Wird dann ein solches Modell trainiert und aktiviert, wird die Stellgröße zu einer geregelten Stellgröße. Soll das Lernverfahren später nachtrainiert werden, wie in Abschnitt 5 erläutert, treten dieselben Probleme auf wie bei anderen geregelten Stellgrößen, und die hier vorgestellten Methoden können angewandt werden.

Praktische Untersuchung der selbst- 7. lernenden Regelung eines Verbrennungs- prozesses



In den drei letzten Kapiteln wurden verschiedene Maßnahmen aus unterschiedlichen Bereichen vorgestellt, mit denen sich die Leistung lernfähiger Systeme im Einsatzgebiet der Feuerleistungsführung verbessern lässt. Die dargestellten Ergebnisse basierten dabei in den meisten Fällen auf Berechnungen, die offline auf Daten der Werke oder an Simulatoren durchgeführt wurden. In diesem Kapitel werden die einzelnen Maßnahmen zusammengeführt und ihre Wirksamkeit bei der Verbesserung der Prozessführung analysiert. Im Unterschied zu den vorherigen Kapiteln wurden die Verfahren dabei nicht nur simuliert oder als Prototyp über einen kurzen Zeitraum untersucht sondern kamen in den Anlagen über lange Zeiträume zum Einsatz, um entsprechende Prognosen zu liefern oder den Prozess zu regeln. Die in diesem Kapitel dargestellten Ergebnisse stellen also die reale Veränderung der Prozessführung durch lernfähige Verfahren in den jeweiligen Anlagen dar.

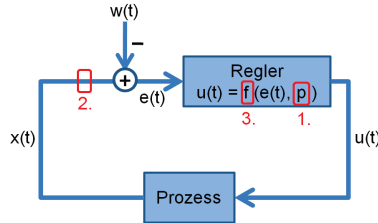


Abbildung 7.1. – Ansatzpunkte zur Anpassung eines Reglers mit lernfähigen oder adaptiven Komponenten. 1.: Änderung der Reglerparameter, 2.: Änderung der Eingangssignale, 3.: Änderung der Reglerfunktion

In Kapitel 3 wurde erläutert, dass es drei verschiedene Methoden gibt um ein lernfähiges Verfahren in ein bestehendes Regelungssystem einzubinden. In Abbildung 7.1 sind diese drei Ansatzpunkte noch einmal dargestellt.

Dabei wurde die Änderung der Eingangssignale durch Kombination aus Prognose und konventionellem Regler (Model Predictive Control) und die Ersetzung einer konventionellen Reglerfunktion durch einen lernfähigen Regler als die in dieser Arbeit im Vordergrund stehenden Methoden definiert. Beide Methoden finden sich auch in diesem Kapitel wieder. MPC-Methoden am Beispiel der Freikalk-Prognose in Zementwerken und der Dampfleistungsprognose in Müllverbrennungsanlagen werden in Abschnitt 7.1 untersucht. Die Ersetzung eines konventionellen Reglers durch ein lernfähiges System, wie das NFQ in verschiedenen Anwendungsbereichen, wird in Abschnitt 7.2 dargestellt.

Neben der Betrachtung der Ergebnisse bei der Verbesserung der Prozessführung ist es wichtig, den Aufwand zu betrachten, der für die Erzielung dieser Ergebnisse beim Training und der Konfiguration der Verfahren durchgeführt werden muss. Eine entsprechende Untersuchung findet sich in Abschnitt 7.3.

Das Kapitel wird abgeschlossen durch ein Fazit in Abschnitt 7.4.

In Abbildung 7.2 ist dargestellt, in welchem Abschnitt auf welchen Anlagen Verfahren zur Verbesserung der Regelqualität im Folgenden vorgestellt werden.

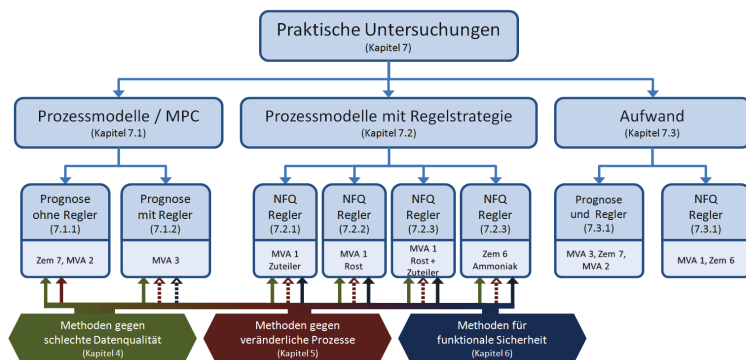


Abbildung 7.2. – Übersicht über die durchgeführten praktischen Versuche auf den jeweiligen Anlagen und die dabei eingesetzten Verfahren. Es ist dargestellt, welche Versuchsergebnisse in welchem Abschnitt der Arbeit erläutert werden (hellblau) und welche Verfahren aus den Kapiteln 4-6 bei dem jeweiligen Versuch zur Anwendung kamen. Ein gestrichelter Anwendungspfeil bedeutet, dass das Verfahren zwar angewendet wurde, aber kaum messbare Auswirkungen hatte, da beispielsweise der Versuchszeitraum von mehreren Wochen für eine Adaptivität, die vor allem auf den Bereich mehrerer Monate ausgelegt ist, zu klein ist.

7.1. Model-Predictive Control und Softsensor

7.1.1. Prozessmodelle ohne direkten Regler

In Kapitel 5 wurden bereits die Ergebnisse verschiedener Versuche zur langfristigen Qualität von Prognosemodellen auf den Daten von Müllverbrennungsanlagen und Zementwerken vorgestellt. Die Versuche wurden offline auf den Daten der Anlagen durchgeführt.

Es kamen dabei verschiedene Verfahren zur Anpassung der Modelle an neue Daten zum Einsatz. Darunter waren Verfahren zur Datenakkumulation (Datenakkumulation (AKK)), Verfahren, die einen festen Datenbereich verwenden (Bereichsauswahl (BAW) und Bereichsauswahl mit Kontrolle (BAWk)) und Verfahren, die statt eines einzelnen Modells die Ergebnisse mehrerer Modelle kombinieren (Ensemble Selection (ES), Ensemble Weighting (EW) und Ensemble Weighting mit m Modellen (EW m)). Weiterhin wurden alle Verfahren in zwei Varianten getestet: Variante 1 nutzt die Gewichte alter Netze als Startinitialisierung für das Nachtraining, Variante 2 initialisiert alle Gewichte vor dem Nachtraining zufällig.

Es stellte sich dabei heraus, dass prinzipiell durch das Verfahren der Datenakkumulation die besten Ergebnisse erzielt werden konnten. AKK stellt allerdings sehr hohe Anforderungen hinsichtlich der verfügbaren Datenmenge und der Rechenleistung der Adaptivität. Kann AKK aufgrund dieser Gründe nicht eingesetzt werden, so bietet sich eines der Ensemble-Verfahren an, die nach AKK die besten Resultate erzielen konnten.

Aufgrund dieser Ergebnisse wurden die Verfahren zur Adaption von Modellen an veränderte Daten in einer Müllverbrennungsanlage und in einem Zementwerk umgestellt. Bisher kam in beiden Anlagen das BAWk2-Verfahren zum Einsatz. Als neues Verfahren wurde das ES1 Verfahren gewählt¹.

¹Aufgrund der Ergebnisse von Kapitel 5 wäre mit dem Verfahren EW m 2 das beste Ergebnis zu erwarten. Die Rahmenbedingungen des Softwaresystems der beiden Anlagen gaben aber den Ausschlag für die Umsetzung des ES1-Verfahrens.

Das alte BAWk2-System und das neue ES1-System wurden parallel mit derselben Basis-Prognose über einen Zeitraum von mehreren Monaten betrieben. Die Ergebnisse dieses Tests sind in Tabelle 7.1 dargestellt.

	BAWk 2	ES 1	Verbesserung in %
Zementwerk 7, Softsensor-Modell 1	0,480	0,447	6,92%
Zementwerk 7, Softsensor-Modell 2	0,493	0,452	8,19%
Zementwerk 7, Softsensor-Modell 3	0,483	0,446	7,72%
Zementwerk 7, Softsensor-Modell 4	0,458	0,439	4,07%
Zementwerk 7, Softsensor-Modell 5	0,513	0,477	7,03%
MVA 2, Prognose-Modell 1	3,26	3,20	1,78%

Tabelle 7.1. – Vergleich unterschiedlicher Adaptionmethoden zur Anpassung von Softsensor- und Prognosemodellen in zwei Anlagen. In Zementwerk 7 liefen fünf unterschiedliche Softsensor-Modelle für Freikalkwerte, auf die jeweils die Adaption angewendet wurde. In MVA 2 kam ein Prognose-Modell für die Dampfleistung zum Einsatz, das von der jeweiligen Adaptionmethode angepaßt wurde. Dargestellt sind jeweils der mittlere Prognosefehler und die Verbesserung des Prognosefehlers durch den ES-Adaptionsansatz. Der Auswertez Zeitraum beträgt bei Zementwerk 7 zehn Monate und bei MVA 2 sechs Monate.

Wie zu sehen ist, konnte durch die Umstellung des Adaptionsverfahrens eine Verbesserung des Prognosefehlers der jeweiligen Freikalk- oder Dampfprognose gegenüber den Prognosen mit dem alten Adaptionsmodell erreicht werden.

Auf den Daten der Anlagen *Zem1* bis *Zem3* wurde ermittelt, welche Auswirkungen eine Kombination der Verfahren der Adaptivität mit den Verfahren der Reduzierung des Rauschens hat. Die Ergebnisse sind in Tabelle 7.2 aufgeführt.

$\bar{e}_{Q50\%}$	Zem1	Zem2	Zem3	\emptyset
Normal	0%	0%	0%	0%
NELLM-BRG	-20,4%	-13,4%	-2,4%	-12,1%
EWm 1	-22,6%	-15,8%	-9,2%	-15,9%
EWm 1 + NELLM-BRG	-32,9%	-21,3%	-12,1%	-22,1%

Tabelle 7.2. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers der einzelnen Verfahren.

Es ist deutlich erkennbar, dass sowohl die Verwendung der Adaptivität in Form des EWm-Verfahrens als auch die Verbesserung der Datenqualität durch das NELLM-BRG-Verfahren eine deutliche Reduktion des Approximationsfehlers mit sich bringen. Werden beide Verfahren kombiniert, so kann diese Verbesserung noch einmal gesteigert werden. Je nach Werk und Datenanzahl kann damit eine relative Verbesserung des Approximationsfehlers um 10% bis 30% gegenüber den bis dahin in den jeweiligen Anlagen verwendeten Prognose- oder Softsensor-Modellen erreicht werden.

7.1.2. Prozessmodelle mit direktem konventionellen Regler

Die im letzten Abschnitt beschriebene Verbesserung bezieht sich auf die Qualität des Approximationsmodells. Welche Auswirkungen ergeben sich aus der Qualität des Modells auf den realen Betrieb der Anlage?

Die reale Auswirkung der Qualität von Softsensor-Modellen auf den Prozess ist schwierig zu bestimmen. Im Fall von Freikalk-Modellen

dienen die Modelle häufig als Entscheidungshilfe für menschliche Operatoren. Ob aufgrund der Modellausgabe ein Stelleingriff erfolgt und bei welcher Stellgröße dieser vorgenommen wird, liegt im Ermessen des jeweiligen Operators. Eine Messung der Verbesserung der Prozessführung durch verbesserte Softsensoren ist deshalb nur indirekt und über sehr langen alternierenden Vergleichszeiträumen möglich. Ein solcher Test konnte im Rahmen dieser Arbeit nicht durchgeführt werden.

Etwas einfacher ist die Bestimmung der Auswirkungen des Modells dagegen, falls die Ausgabe des Modells von einem konventionellen Regler statt einem menschlichen Operator verwendet wird. In diesem Fall ist die Reaktion des Reglers beschränkt auf eine Stellgröße und erfolgt immer mit der gleichen Stärke. Werden alternierend zwei unterschiedliche Modelle verwendet und jeweils die Qualität der Prozessführung gemessen, lassen sich aus einem solchen Test bedeutend schneller und präziser Aussagen ableiten als bei der Führung durch menschliche Operatoren.

In Anlage MVA 3 wurde ein solcher Test im Rahmen dieser Arbeit durchgeführt. Bei dem untersuchten Modell handelt es sich um ein Prognosemodell für die Dampfleistung, welches den Wert der Dampfleistung mit einem Prognosehorizont von zwei Minuten vorhersagen soll. Dieses Prognose wird in Anlage MVA 3 verwendet, um die Rostgeschwindigkeit zu stellen. Um die Auswirkungen der Prognose auf das Regelverhalten zu untersuchen, wurde über einen Testzeitraum von 14 Tagen ein Versuchsprogramm gefahren. Tageweise alternierend wurde jeweils die Dampfprognose oder der reale Messwert der Dampfleistung als Eingabe des Reglersystems verwendet. Als wesentliche Zielgrößen dieser Regelung wurde die mittlere Abweichung der Dampfleistung vom Sollwert und die mittlere Abweichung der O_2 -Emissionen vom Sollwert gemessen. Die Ergebnisse dieses Tests sind in Tabelle 7.3 dargestellt.

	ohne Prognose	mit Prognose	Verbesserung in %
Regelabw. Dampfleistung	$0,78 \frac{t}{h}$	$0,67 \frac{t}{h}$	14,1%
Regelabw. O_2	0,67%	0,59%	11,9%

Tabelle 7.3. – Vergleich der Auswirkungen eines Prognosemodells auf die mittlere Abweichung der Dampfleistung und der O_2 -Emissionen vom jeweiligen Sollwert. Der Auswertezeitraum beträgt 14 Tage, in denen jeweils tageweise zwischen den beiden Modellen umgeschaltet wurde.

Wie zu sehen ist, hat der Einsatz der Prognose einen deutlichen Einfluss auf die Qualität der Prozessführung der beiden Größen Dampfleistung und O_2 -Emissionen. Durch den Einsatz des Prognosemodells konnte die Abweichung der beiden Größen vom Sollwert um über 10% verringert werden. In Abbildung 7.3 ist dargestellt, wie sich dieser Unterschied im Verlauf beider Größen bemerkbar macht.

Dieser Test verdeutlicht allerdings nur, welcher Vorteil sich prinzipiell durch ein Prognosemodell ergibt. In der vorliegenden Arbeit wurden vor allem Verfahren vorgestellt, die die Qualität und Sicherheit bereits vorhandener Modelle erhöhen. Vor diesem Hintergrund ist vor allem die Frage interessant, ob sich eine Verbesserung der Prognosequalität ebenso auf das Prozessverhalten auswirkt. Um dies zu untersuchen, wurde in MVA 3 ein weiterer Test durchgeführt. Statt die Qualität der Regelung mit und ohne Prognosemodell zu testen, wurden zwei unterschiedliche Prognosemodelle verwendet, die wieder alternierend eingesetzt wurden. Prognosemodell 1 ist dasselbe Modell, das auch im obigen Versuch zum Einsatz kam. Prognosemodell 2 hat den gleichen Aufbau wie Prognosemodell 1, wurde aber auf wesentlich weniger Daten trainiert. Auf den Testdaten war deshalb der Fehler von Prognosemodell 2 17% schlechter als bei Prognosemodell 1. Dieser Unterschied entspricht der durchschnittlichen Verbesserung

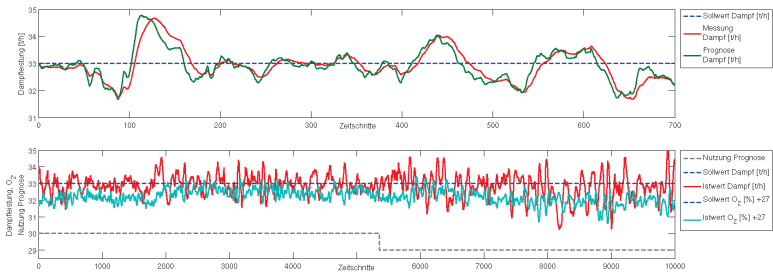


Abbildung 7.3. – *Oben:* Dampfprognose (grün) und reale Dampfleistung (rot) sowie Sollwert der Dampfleistung (blau) in der Anlage MVA 3 über einem Zeitraum von zwei Stunden. Der Prognosehorizont der Dampfprognose beträgt zwei Minuten.

Unten: Auswirkungen der Dampfprognose auf die Dampfleistungsregelung. Dargestellt ist ein Zeitraum von 28 Stunden, während dem im ersten Teil die Prognose aktiv war und im zweiten Teil inaktiv. Regelabweichung und Standardabweichung der Dampfleistung (rot) und der O₂-Emissionen (hellblau) sind im ersten Zeitbereich geringer als im zweiten Zeitbereich.

eines Modells, die durch den Einsatz der in Kapitel 4 und Kapitel 5 erläuterten Verfahren erzielt werden kann. Die Ergebnisse dieses Tests sind in Tabelle 7.4 dargestellt.

Wie zu sehen ist sind die Unterschiede in der Qualität der Prozessführung zwischen beiden Modellen geringer als im vorherigen Versuch. Mit einer Verbesserung der Regelabweichung der Dampfleistung von 4,1% und der O₂-Abweichung von 7,2% ist allerdings trotzdem eine deutliche Verbesserung der Prozessführung zu beobachten.

An der getesteten Anlage MVA 3 konnte durch den Einsatz der in dieser Arbeit vorgestellten Methoden zur Verbesserung der Approximationsqualität von Prozessmodellen die Qualität der Prozessführung gesteigert werden.

	Prognose 2	Prognose 1	Verbesserung in %
Regelabw. Dampfleistung	$0,74 \frac{t}{h}$	$0,71 \frac{t}{h}$	4,1%
Regelabw. O_2	0,55%	0,51%	7,2%

Tabelle 7.4. – Vergleich der Auswirkungen zweier Prognosemodelle auf die mittlere Abweichung der Dampfleistung und der O_2 -Emissionen vom jeweiligen Sollwert. Der Auswertez Zeitraum beträgt 22 Tage, in denen jeweils tageweise zwischen den beiden Modellen umgeschaltet wurde. Der Approximationsfehler von Prognose 2 war im Mittel 17% schlechter als der Fehler von Prognose 1.

7.2. Ersetzung konventioneller Regler durch NFQ-Regler

Im letzten Abschnitt wurde dargestellt, welche Ergebnisse sich bei der Verbesserung der Prozessführung durch den Einsatz lernfähiger Modelle in Kombination mit konventionellen PID-Reglern erzielen lassen. Die zweite in dieser Arbeit untersuchte Möglichkeit zur Nutzung eines lernfähigen Modells besteht darin, auch die Regelstrategie für die Stellgröße durch das Modell zu lernen und entsprechend die Stellgröße direkt zu stellen. In Kapitel 6 wurde erläutert, welche zusätzlichen Maßnahmen in einem solchen Fall erforderlich sind um das Risiko einer falschen Stellstrategie zu verringern. Anhand zweier Stellgrößen in einer MVA und einer Stellgröße eines Zementwerkes wurde gezeigt, wie ohne diese Maßnahmen riskante Stellstrategien entstehen, die nicht auf der Anlage angewendet werden können.

Wurden die Modelle aber mit den in Kapitel 6.2.1 vorgestellten erweiterten Datensätzen trainiert, mit dem in Kapitel 6.4 erläuterten Algorithmus auf Korrektheit überprüft und mit einem Sicherheitsmechanismus, wie in Kapitel 6.5 geschildert, versehen, können sie auf der Anlage eingesetzt werden.

Im Rahmen dieser Arbeit wurden in der Müllverbrennungsanlage MVA 1 und dem Zementwerk Zem 6 entsprechende Modelle implementiert und anschließend im realen Betrieb getestet. In MVA 1 wurden als Stellgrößen, die durch Modelle geregelt werden sollen, die Rostgeschwindigkeit und die Geschwindigkeit des Zuteilers ausgewählt. Beide Stellgrößen sind laut [Thome-Kozmiensky, 1994] in Müllverbrennungsanlagen des Typs Vorschubrost wesentliche Einflussgrößen für die Qualität des Verbrennungsprozesses. In Abbildung 7.4 ist noch einmal der Aufbau des Verbrennungskessels einer Müllverbrennungsanlage zu sehen, in dem beide Stellgrößen markiert wurden.

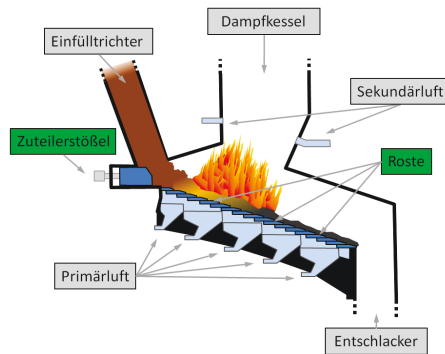


Abbildung 7.4. – Längsschnitt des Verbrennungskessels einer MVA. Bei den Tests der lernfähigen Modelle wurden die grün markierten Stellgrößen Rost und Zuteiler jeweils im normalen Betrieb durch ein lernendes NFQ geregelt.

Modelle für beide Stellgrößen wurden bereits in Kapitel 6 vorgestellt und auf die Sicherheit ihrer Stellstrategie untersucht. Die Untersuchungen in Kapitel 6 erfolgten allerdings offline auf den Daten der Anlage. Nachdem diese Untersuchungen abgeschlossen waren, wurden im Rahmen von Kapitel 7 die jeweils besten Modelle auf der

Anlage installiert und zusammen mit den in Kapitel 6.5 geschilderten Sicherheitsmechanismen in die Feuerleistungsregelung integriert.

7.2.1. Praktische Untersuchungen an der FLR der MVA 1 mittels Zuteiler

Um die Qualität der neuen NFQ-Regelung zu testen, wurde ein Vergleichstest zwischen der alten Regelung der Stellgrößen und den neuen NFQ-Modellen durchgeführt. Bei dem alten System handelt es sich um ein System aus PID-Reglern und einer Dampfleistungsprognose. Es ist baugleich zu dem MPC-System, das in Abschnitt 7.1 für Versuche mit Dampfleistungsprognosen verwendet wurde. Für das Testprogramm wurde tageweise zwischen dem alten System und dem neuen NFQ-System umgeschaltet. Im ersten Test wurde nur der NFQ-Regler für den Zuteiler getestet, während der Rostgeschwindigkeits-NFQ deaktiviert blieb.

Für den Vergleichstest sind vor allem die Regelabweichung, also die Abweichung der Istwerte I von den Sollwerten S , der Dampfleistung, die Regelabweichung der Rosttemperatur und die CO -Emissionen interessant. Laut den in Anhang A.4 geschilderten Analyseverfahren und den Erfahrungen des Anlagenbetreibers sind das die Zielgrößen, die sich durch die Regelung der Zuteilergeschwindigkeit beeinflussen lassen. Aus diesen drei Zielgrößen wurde ein einzelnes Rewardsignal $r_{zuteiler}$ berechnet, welches die "allgemeine Qualität" der Prozessführung bewerten soll. Dieser Gesamtreward, der auch für das Training des eingesetzten NFQ-Systems verwendet wurde, berechnet sich wie folgt:

$$r_{zuteiler} = 0 - |S_{dampf} - I_{dampf}| - \frac{|S_{rost} - I_{rost}|}{20} - \frac{I_{CO}}{5} \quad (7.1)$$

und stellt damit eine Wichtung der drei Zielgrößen Dampfleistungsabweichung $|S_{dampf} - I_{dampf}|$, Rosttemperaturabweichung $|S_{rost} - I_{rost}|$ und Kohlenmonoxid I_{CO} dar. Die Ergebnisse dieses Tests sind in Tabelle 7.5 aufgeführt.

	PID	NFQ+kDP	Verbesserung in %
Regelabweichung Dampfleistung	$0,70 \frac{t}{h}$	$0,75 \frac{t}{h}$	-7,1%
Regelabweichung Rosttemperatur	44,86 °C	38,45 °C	14,3%
CO-Emissionen	$16,82 \frac{mg}{Nm^3}$	$15,93 \frac{mg}{Nm^3}$	5,3%
Gesamt-Reward	-6,29	-5,84	7,2%

Tabelle 7.5. – Ergebnisse der Vergleichstest zwischen altem MPC-System und neuem NFQ-System für die Stellgröße Zuteiler in der Anlage MVA 1. Für den Test wurden beide Systeme jeweils sieben Tage bei täglichen Wechsel im normalen Betrieb der Anlage zur Regelung der Zuteilergeschwindigkeit eingesetzt.

Der Vergleichstest zeigte, dass sich bei der Regelabweichung der Rosttemperatur und den CO-Emissionen eine Verbesserung durch das NFQ-System erzielen ließ. Bei der Regelabweichung der Dampfleistung trat hingegen eine Verschlechterung ein. Betrachtet man diese gewichtete Verrechnung der drei Größen im Reward als Vergleichsmaß, so konnte von dem NFQ-System eine Verbesserung gegenüber dem alten System erzielt werden. Soll ebenfalls eine Reduktion der Regelabweichung der Dampfleistung erzielt werden, so müßte diese Regelabweichung eine stärkere Gewichtung bei der Berechnung des Rewards erhalten und ein neues NFQ-System mit diesem modifizierten Reward trainiert werden.

7.2.2. Praktische Untersuchungen an der FLR der MVA 1 mittels Rostgeschwindigkeit

Nachdem der Test der Zuteilerregelung abgeschlossen war, wurde in der Anlage MVA 1 ein weiterer Vergleichstest für die Regelung der Rostgeschwindigkeit durchgeführt. Die Stellgröße Rostgeschwindigkeit hat vor allem Einfluss auf die Zielgrößen der Regelabweichung der Dampfleistung $|S_{dampf} - I_{dampf}|$ und der O₂-Emissionen $|S_{O_2} - I_{O_2}|$.

Der Gesamt-Reward r_{rost} des NFQ-Systems berechnet sich für die Stellgröße Rostgeschwindigkeit wie folgt:

$$r_{rost} = 0 - |S_{dampf} - I_{dampf}| - 1.5|S_{O_2} - I_{O_2}| \quad (7.2)$$

Für den Test des Rostgeschwindigkeit-NFQs galten die gleichen Randbedingungen wie beim Test der Zuteilersteuerung. Das alte Regelungssystem, bestehend aus PID-Reglern und einer Dampfleistungsprognose, wurde gegenüber dem neuen System bestehend aus einem NFQ getestet. Dieser Test wurde erneut über einem Zeitraum von 14 Tagen durchgeführt, indem beide Systeme tageweise abwechselnd die Rostgeschwindigkeit im normalen Betrieb stellten. Die Ergebnisse dieses Tests sind in Tabelle 7.6 aufgeführt.

	PID	NFQ+kDP	Verbesserung in %
Regelabweichung Dampfleistung	$0,78 \frac{t}{h}$	$0,75 \frac{t}{h}$	3,8%
Regelabweichung O_2	0,42%	0,36%	14,3%
Gesamt-Reward	-1,41	-1,27	9,9%

Tabelle 7.6. – Ergebnisse der Vergleichstests zwischen altem MPC-System und neuem NFQ-System für die Stellgröße Rostgeschwindigkeit in der Anlage MVA 1. Für den Test wurden beide Systeme jeweils sieben Tage bei täglichem Wechsel im normalen Betrieb der Anlage zur Regelung der Rostgeschwindigkeit eingesetzt.

Der Vergleichstest zeigte, dass sich bei beiden Größen eine Verbesserung durch das NFQ-System gegenüber dem bestehenden PID-System erzielen ließ. Bei der gewichteten Verrechnung der Zielgrößen Dampfleistung und der O_2 -Emissionen im Gesamtreward konnte das NFQ-System eine Verbesserung von ca. 10% gegenüber dem alten Regelungssystem erzielen.

7.2.3. Praktische Untersuchungen an der FLR der MVA 1 mittels Zuteiler und Rostgeschwindigkeit

Als letztes wurde in der Anlage MVA 1 ein Vergleichstest durchgeführt, bei dem sowohl die Regelung der Rostgeschwindigkeit als auch die Regelung der Zuteilergeschwindigkeit durch jeweils ein NFQ-System zugleich geführt werden. Als Vergleichssystem wurde erneut das bekannte PID-System genutzt. Zur Bewertung der Gesamtpfomanz wurde folgende Rewardfunktion für den Reward r_{gesamt} genutzt, die die Regelabweichungen der Teilziele Dampfleistung $|S_{dampf} - I_{dampf}|$, Rosttemperatur $|S_{rostt} - I_{rostt}|$, CO -Emissionen I_{CO} und O_2 -Emissionen $|S_{O_2} - I_{O_2}|$ folgendermaßen wichtet:

$$r_{gesamt} = 0 - |S_{dampf} - I_{dampf}| - \frac{|S_{rostt} - I_{rostt}|}{20} - \frac{I_{CO}}{5} - 1.5|S_{O_2} - I_{O_2}| \quad (7.3)$$

Der Test beider Systeme wurde über einen Zeitraum von 22 Tagen durchgeführt, in dem beide, das kombinierte NFQ-System und das alte Regelungssystem, tageweise abwechselnd beide Stellgrößen im normalen Betrieb regelten. Die Ergebnisse dieses Tests sind in Tabelle 7.7 aufgeführt.

Wie zu sehen ist, konnte das NFQ-System durch die Regelung beider Stellgrößen, Rostgeschwindigkeit und Zuteiler, eine Verbesserung bei allen Zielgrößen gegenüber dem konventionellen PID-System erreichen.

	PID	NFQ+kDP	Verbes- serung in %
Regelabweichung Dampf- leistung	$0,82 \frac{t}{h}$	$0,77 \frac{t}{h}$	6,1%
Regelabweichung Rost- temperatur	55,02 °C	42,96 °C	21,91%
CO-Emissionen	$9,72 \frac{mg}{Nm^3}$	$9,66 \frac{mg}{Nm^3}$	0,6%
Regelabweichung O ₂	0,50%	0,45%	10,0%
Gesamt-Reward	-6,27	-5,51	12,12%

Tabelle 7.7. – Ergebnisse der Vergleichstests zwischen altem MPC-System und neuem NFQ-System für die kombinierten Stellgrößen Rostgeschwindigkeit und Zuteiler in der Anlage MVA 1. Für den Test wurden beide Systeme jeweils elf Tage bei täglichen Wechsel im normalen Betrieb der Anlage zur Regelung der Rostgeschwindigkeit und der Zuteilergeschwindigkeit eingesetzt.

7.2.4. Praktische Untersuchungen an der Rauchgasentstickung eines Zementwerkes

Da durch NFQ-Systeme auf der Müllverbrennungsanlage 1 eine Verbesserung der Prozessführung erzielt werden konnte, wurde auch bei einem Zementwerk ein NFQ-System getestet um zu überprüfen, ob sich auch dort eine Verbesserung gegenüber der bisherigen konventionellen Regelung ergibt.

In Zementwerk 6 wurde deshalb ein NFQ-System zur Regelung der Menge des Ammoniakwassers im Rahmen eines SNCR-Verfahrens installiert. Ein SNCR-Verfahren ist ein System zur Entstickung des Rauchgases von Verbrennungssystemen und kann sowohl in Zementwerken als auch in Müllverbrennungsanlagen und Kohlekraftwerken u.a. zum Einsatz kommen. Es wird dabei Ammoniak NH_3 als wässrige Lösung durch Düsen in das heiße Rauchgas gesprüht. Im Temperaturbereich zwischen 900°C und 1000°C reagiert das Ammoniakwassers mit den giftigen NO_x -Verbindungen zu Stickstoff N_2 und Wasser H_2O . Die (vereinfachte²) Reaktionsgleichung dafür lautet:



Das Ammoniakwasser wird in Zementwerk 6 im Bereich des Steigschachtes in das Rauchgas eingedüst. In Abbildung 7.5 ist dieser Teil des Zementofens hervorgehoben.

²Tatsächlich laufen mehrere unterschiedliche Reaktionen gleichzeitig ab. Der Anteil der einzelnen Reaktionen hängt ab vom O_2 - und CO -Anteil im Rauchgas, der Temperatur des Rauchgases und der Anteile von NO und NO_2 im Rauchgas. Eine detaillierte Auflistung der Reaktionen und ihres Zusammenspiels findet sich in [Brouwer et al., 1996]. Ein Beispiel zum Einsatz in der Rauchgasentstickung in Zementwerken findet sich in [Horton et al., 2006] und für Rostfeuerungen in [Burstrom et al., 2010].

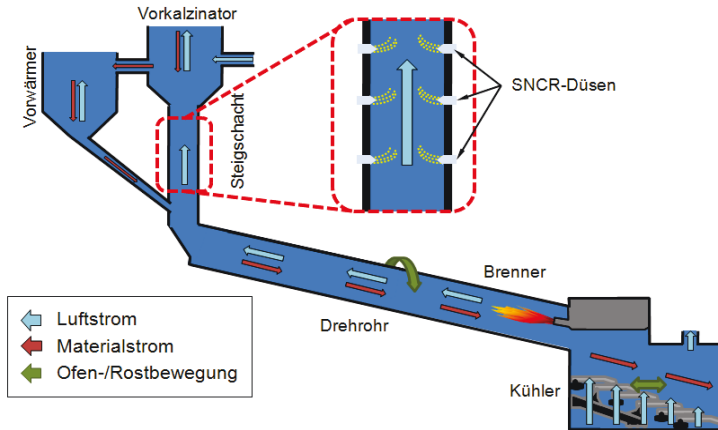


Abbildung 7.5. – Position des Steigschachtes und des SNCR-Verfahrens in Zementwerk 6. Durch Düsen in der Wand des Steigschachtes wird Ammoniakwasser in den heißen Rauchgasstrom injiziert

Für die Effektivität des SNCR-Verfahrens zur Reduktion von NO_x -Verbindungen sind zwei Faktoren wichtig:

- Die Eindüsung des Ammoniakwassers in Rauchgas mit einer Temperatur zwischen 900°C und 1000°C .
- Die Eindüsung der richtigen Menge an Ammoniakwasser.

Wird zu viel Ammoniakwassers eingedüst, reagiert der Ammoniak nicht und muss wieder aus dem Rauchgas gefiltert werden. Wird zu wenig Ammoniakwasser genutzt, wird das giftige NO_x nicht abgebaut, wodurch die Grenzwerte verletzt werden.

Die bisherige Regelung der NO_x -Emissionen durch die Ammoniakwasser-Menge in Zementwerk 6 wurde durch einen PID-Regler durchgeführt. Zum Vergleich wurde ein NFQ-System installiert das über die in Kapitel 4 bis Kapitel 6 erläuterten Erweiterungen

zu Adaptivität, Datenqualität und Sicherheit verfügt. Der Reward r_{gesamt} zum Training des NFQ berechnet sich auf Grundlage der Abweichung des NO_x -Messwertes I_{NO_x} vom NO_x -Sollwert S_{NO_x} :

$$r_{gesamt} = 0 - |S_{NO_x} - I_{NO_x}| \quad (7.5)$$

Der Test beider Systeme wurde über einen Zeitraum von 14 Tagen durchgeführt, in dem beide, das kombinierte NFQ-System und das alte Regelungssystem, tageweise abwechselnd beide Stellgrößen im normalen Betrieb regelten. Die Ergebnisse sind in Tabelle 7.8 aufgeführt.

	PID	NFQ+kDP	Verbes- serung in %
Regelabweichung NO_x	45,77 $\frac{mg}{Nm^3}$	41,29 $\frac{mg}{Nm^3}$	9,8%
Verbrauch NH_4OH	121,96 $\frac{l}{h}$	98,92 $\frac{l}{h}$	18,9%
spez.Verbrauch NH_4OH	2,42	1,94	19,8%

Tabelle 7.8. – Ergebnisse der Vergleichstests zwischen altem MPC-System und neuem NFQ-System für die Stellgrößen NH_4OH -Menge in der Anlage Zem 6. Für den Test wurden beide Systeme jeweils sieben Tage bei täglichem Wechsel im normalen Betrieb der Anlage eingesetzt. Aufgeführt sind die Regelabweichung der NO_x -Emissionen, der Verbrauch an Ammoniakwasser NH_4OH und der spezifische Verbrauch, der als Verbrauch pro Tonne produziertem Klinker berechnet wird.

Durch das NFQ-System konnte in Zementwerk 6 die Abweichung der NO_x -Emissionen vom Sollwert um ca. 10% gegenüber der Regelung mit dem konventionellen PID-System verringert werden. Gleichzeitig wurde der Verbrauch an Ammoniakwasser, das als Reduktionsmittel dient, reduziert.

Der Einsatz des NFQ-Systems kann also auch deutliche Vorteile beim Einsatz zur Regelung von Prozessgrößen in Zementwerken bringen.

7.3. Parametrisierungsaufwand selbstlernender System

In den vorherigen Abschnitten wurden die Ergebnisse beim Einsatz selbstlernender Verfahren zur Regelung des Prozesses in Müllverbrennungsanlagen und Zementwerken vorgestellt. Bis auf wenige Ausnahmen wurden dabei durch die selbstlernenden und adaptiven Systeme bessere Resultate erzielt als durch die bisherigen konventionellen PID-Systeme.

Sollte also jede Stellgröße einer industriellen Feuerleistungsführung durch ein selbstlernendes System geregelt werden? Betrachtet man allein die potenzielle Leistungsfähigkeit eines lernfähigen Systems mit einem hinreichend komplexen Modell (wie z.B. einem MLP oder einem NFQ), so kann diese Frage bejaht werden. Ein komplexes Modell kann prinzipiell die Funktionsgleichung eines PID-Reglers abbilden und noch zusätzliche Informationen nutzen. In den vergangenen Abschnitten wurde gezeigt, dass dieses Potenzial unter den richtigen Bedingungen auch genutzt und eine deutliche Verbesserung der Prozessführung erreicht werden kann.

Neben der Leistungsfähigkeit des Systems unter den richtigen Bedingungen muss aber auch die Häufigkeit dieser richtigen Bedingungen sowie gegebenenfalls die ökonomische Rentabilität betrachtet werden. Vom Mönch William of Occam ist der Grundsatz “Nunquam ponenda est pluralitas sin necessitate” [Thorburn, 1918, Domingos, 1999] was soviel heißt wie “Eine Mehrheit³ darf nie ohne Not zugrunde gelegt werden.” überliefert.⁴ Überträgt man diesen Satz, der oft auch als

³“Mehrheit“ nicht im Sinne von Wahlen oder Abstimmungen sondern im Sinne von “Zusatz“ oder “Erweiterung“.

⁴Statt des erwähnten Satzes wird auch oft “Entia non sunt multiplicanda prae-ter necessitatem”, also “Entitäten dürfen nicht über das Notwendige hinaus vermehrt werden.” Occam zugeschrieben und als “Occam’s razor“ bezeichnet. Laut [Thorburn, 1918] ist aber nur das erste Zitat nachweislich auf Occam zurückzuführen.

”Occam’s Rasiermesser“ bezeichnet wird, auf Modelle oder Lösungsansätze, so bedeutet er sinngemäß ” Existieren zwei Modelle mit gleichem Fehler, sollte das einfachere Modell bevorzugt werden, denn Einfachheit ist ein Vorteil an sich.“ [Domingos, 1999].

Bezogen auf den Vergleich zwischen einem konventionellen PID und einem neuronalen NFQ ist das adaptive NFQ das deutlich komplexere Verfahren. Ein PID-Regler verfügt über drei Parameter (K_P , T_N und T_V), die durch einen Experten eingestellt werden müssen. Ein adaptives Modell in Form eines künstlichen Neuronales Netzes lernt zwar die internen Gewichte selbst aus den verfügbaren Daten, dafür müssen aber die Architektur des Netzwerkes und die Parameter des Lern- und Adaptionsverfahrens durch einen Experten eingestellt werden.

Die Entscheidung für oder gegen den Einsatz eines adaptiven Modells in der Prozessführung sollte also nicht nur nach dem potentiellen Ertrag der Verbesserung des Prozessverhaltens getroffen werden, sondern auch unter Berücksichtigung des Aufwands, der zur Erstellung und Wartung dieses adaptiven Modells notwendig ist.

In den folgenden Abschnitten soll deshalb der Parametrisierungsaufwand für adaptive selbstlernende Systeme mit externer und interner Stellstrategie beschrieben werden. Dadurch kann für künftige Projekte besser abgeschätzt werden, ob sich die Aufsetzung eines selbstlernenden Systems lohnt oder weiter eine konventionelle Regelung verwendet wird.

7.3.1. Adaptive Modelle mit externer Stellstrategie

In Abschnitt 7.1 wurden die Ergebnisse geschildert, die sich erzielen lassen, wenn ein Prognosemodell in Verbindung mit einem konventionellen PID-Regler zur Regelung genutzt wird. Die eigentliche Stellstrategie wird dabei durch den PID-Regler definiert, das adaptive Prognosemodell ermöglicht aber eine vorausschauendere und schnellere Regelung.

Zur Nutzung eines solchen Ansatzes müssen die Parameter des PID-Reglers durch einen Experten eingestellt werden. Sollte bereits ein PID-Regler für das gewünschte Ziel vorhanden sein, kann dieser weiter genutzt werden. Zusätzlich muss ein Prognosemodell für die Regelgröße trainiert und mit einem Verfahren zur adaptiven Anpassung gekoppelt werden.

In dieser Arbeit wurden neuronale Netze in Form von MLP-Netzen als Modell gewählt. Um möglichst gute Trainingsdaten zu erhalten, wurde das in Kapitel 4.3.2 vorgestellte Noise Estimation with Local Linear Models (NELLM)-Verfahren zur Filterung der Trainingsdaten und Bayes'sche Regularisierung (BRG) aus Kapitel 4.3.3 zum Training des Modells verwendet. Die Adaptivität der Prognose wird durch das Ensemble Weighting mit m Modellen (EWM)-Verfahren aus Kapitel 5.4.2 sichergestellt.

Für die MLP-Prognosenetze müssen durch einen Experten die Struktur des Netzes, die Anzahl der Neuronen und die verwendeten Eingabesignale festgelegt werden. Werden zu wenig Neuronen genutzt, kann eine komplexe Funktion nicht ausreichend beschrieben werden, bei zu vielen Neuronen tritt verstärkt Overfitting auf. In dieser Arbeit wurde das initiale Modell mit jeweils fünf verschiedenen Netzarchitekturen trainiert, getestet, und die Architektur mit dem besten Ergebnis gewählt. Steht allerdings genügend Rechenkapazität zur Verfügung, kann dieser Schritt integriert werden in das Ensemble Weighting mit m Modellen (EWM)-Verfahren (siehe Ausblick in Kapitel 9), so dass kein Parametrisierungsaufwand anfällt.

Aufwändig bleibt die Wahl der verwendeten Eingabesignale. In Anhang A werden Verfahren beschrieben, mit denen eine automatische Auswahl möglich ist, es bleibt aber immer ein Experte notwendig um diese Verfahren zu konfigurieren und das Ergebnis zu kontrollieren. Dieser Vorgang muss allerdings nur einmal beim initialen Modell ausgeführt werden.

Das NELLM-Verfahren zur Filterung der Trainingsdaten benötigt zwei Parameter: die Anzahl der nächsten Nachbarn k und den Prozentsatz der Punkte die entfernt werden sollen. Für den Prozentsatz der Punkte hat sich bei allen realen Datensätzen dieser Arbeit 20% zu entfernende Punkte bewährt. Der beste Wert des Parameters k hingegen variiert je nach Datensatz zwischen 5-30. Um den optimalen Wert zu finden, muss ein Testmodell mit verschiedenen Einstellungen von k auf dem initialen Datensatz trainiert werden. Dieser Vorgang wurde in dieser Arbeit manuell durchgeführt, er kann aber automatisiert werden.

Das Levenberg-Marquardt-Trainingsverfahren mit Nutzung von Bayes'sche Regularisierung (BRG) benötigt in der in dieser Arbeit verwendeten Implementierung nur die (maximale) Anzahl der Trainingsschritte als Parameter. 100 Schritte⁵ waren dabei bei allen Tests ausreichend.

Das Ensemble Weighting mit m Modellen (EWm)-Verfahren benötigt drei Parameter: die Anzahl m der genutzten Modelle aus dem Modellpool, die Anzahl der Trainingsdaten und den Abstand zwischen zwei Trainings. In dieser Arbeit wurden stets 15% der verfügbaren Modelle im Pool für die Ausgabe genutzt. Werden zu wenig Modelle genutzt, nähert sich das Ergebnis dem Ensemble Selection (ES)-Verfahren an, werden zu viel Modelle genutzt, ähnelt das Ergebnis dem Ensemble Weighting (EW)-Verfahren. Da beide Verfahren ebenfalls eine deutliche Verbesserung gegenüber einer Nicht-Adaption darstellen, ist dieser Parameter nicht kritisch. Der Abstand zwischen zwei Trainings ist ebenfalls unkritisch. Entscheidend ist allerdings die Anzahl der nötigen Trainingsdaten, welche sich nach der zu approximierenden Funktion richtet und von einem Experten eingerichtet werden muss.

⁵Das Levenberg-Marquardt-Trainingsverfahren in der hier genutzten Version ist ein *batch*-Verfahren. Pro Trainingsschritt wird also der gesamte Trainingsdatensatz verwendet und nicht nur ein Sample.

Zusammengefasst wird für das adaptive Prognosemodell ein Experte benötigt, der für *das initiale Modell* die Auswahl der Eingabekanäle, die Netzwerkkonfiguration, die Anzahl der nächsten Nachbarn k für das NELLM-Verfahren und die Anzahl der genutzten Modelle für das EWm-Verfahren festlegt.

Sind diese Parameter festgelegt, so kann das Verfahren selbständig laufen. Gegebenenfalls ist in regelmäßigen Abständen eine Kontrolle des Modellpools notwendig, die aber bei gutem Human-Machine-Interface (siehe Kapitel 9) auch durch Operatoren ohne Expertenwissen zum Modelltraining erfolgen kann.

7.3.2. Adaptive Modelle mit interner Stellstrategie

In Abschnitt 7.2 werden Ergebnisse aufgeführt, die durch die Nutzung eines NFQ-Systems erzielt wurden. Ein NFQ liefert im Unterschied zu MLP-Prognosenetzen direkt Stellgrößenausgaben und kann ohne zusätzlichen PID-Regler verwendet werden. Dafür muss allerdings auch ein größerer Aufwand bei der Überprüfung der funktionalen Sicherheit des Systems betrieben werden.

Die in dieser Arbeit verwendeten NFQ-Netze wurden ebenso wie die MLPs des vorherigen Abschnittes mit dem Levenberg-Marquardt-Trainingsverfahren mit Nutzung von Bayes'sche Regularisierung (BRG) trainiert. Ihre Adaption an den Prozess erfolgt durch das Ensemble Weighting mit m Modellen (EWm)-Verfahren, wobei allerdings auf den Auswertezwischenräumen im realen Betrieb von jeweils 2-4 Wochen aus Abschnitt 7.2 nur 2-3 Adaptionsschritte durchgeführt wurden.

Für die genannten Trainings- und Adaptionsverfahren ergibt sich derselbe Parametrisierungsaufwand wie im letzten Abschnitt für die Prognosemodelle. Hinzu kommt die Parametrisierung der Sicherheitsmechanismen aus Kapitel 6.2.1 und 6.4.

Der Algorithmus zum Hinzufügen künstlicher Datenpunkte für den Fall geregelter Trainingsdaten (siehe Algorithmus 6) benötigt drei Parameter: die Rate künstlicher Punkte pro Cluster n_{kp} , den Rewar-dabschlag für künstliche Punkte r_{kp} und die Anzahl der Action-Bins n_{action} . Alle drei Parameter sind abhängig vom aktuellen Datensatz zu wählen. Eine automatisierte Einstellung der Parameter wurde in dieser Arbeit nicht vorgenommen, wird aber im Ausblick (siehe Kapitel 9) diskutiert.

Die Parameter für die Prüfung eines trainierten Modells (Kapitel 6.4) werden automatisch aus dem Datensatz ermittelt. Es muss allerdings durch einen Experten festgelegt werden, wie stark eine Regelstrategie von der bisherigen Strategie abweichen darf und welche Messgrößen besondere Beachtung erhalten sollen. Diese Festlegung muss allerdings nur einmal für das initiale Modell durchgeführt werden.

Ein Experte ist ebenso notwendig, um die Sicherheitsbedingungen zur Deaktivierung eines Modells, wie in Kapitel 6.5 beschrieben, zu definieren. Das dafür notwendige Wissen ist allerdings bei den Operatoren der Anlage vorhanden und muss nur mittels einer geeigneten Schnittstelle in das System übertragen werden.

Zur korrekten Parametrisierung der Verfahren sind also ein gewisser Aufwand und ein Verständnis der verwendeten Modelle und Verfahren notwendig. Wird eine aussagekräftige und leicht bedienbare Mensch-Machine-Schnittstelle verwendet, lässt sich ein Teil dieses Aufwandes reduzieren und von den Operatoren vor Ort durchführen. Besonders für das Training und die Parametrisierung des initialen Modells ist allerdings Expertenwissen über den Prozess und die angewandten Verfahren sowie ein gewisser Aufwand unabdingbar.

7.3.3. Hinweise zur Methodenauswahl

In den beiden vorherigen Abschnitten wurde jeweils aufgeführt, welcher Aufwand zur Erstellung und dem Einsatz eines adaptiven Prognosemodells oder eines adaptiven lernfähigen Reglers notwendig ist.

Falls nicht durch das Einsatzgebiet festgelegt ist⁶, welche von beiden Methoden angewendet werden soll, muss eine Entscheidung für eine der beiden Varianten getroffen werden. Im Folgenden werden deshalb einige Kriterien aufgeführt die bei dieser Entscheidung helfen können.

Der Aufwand zur Erstellung eines adaptiven Prognosemodells macht vor allem dann Sinn, wenn

1. ein großer zeitlicher Versatz zwischen der Änderung einer Stellgröße und der Auswirkung dieser Änderung auf die Regelgröße existiert und
2. die Regelgröße sich durch andere Messgrößen in einem Modell mit einem genügenden Zeitversatz vorhersagen lässt.
3. Die Prognose anschließend von einem Regler oder Operator genutzt werden kann, die eigentliche Regelstrategie also bekannt und relativ einfach ist.

Bedingung 1. lässt sich meist durch einen einfachen Stellgrößentest überprüfen, während für Bedingung 2. häufig ein Testmodell erstellt werden muss.

Ein gutes Beispiel für eine Prozessgröße, bei der beide Bedingungen auftreten, ist der Zusammenhang zwischen Stellgröße Rostgeschwindigkeit und Regelgröße Dampfleistung in einer MVA. Eine Änderung der Rostgeschwindigkeit wirkt sich in MVA 3 nach ca. vier Minuten auf die Regelgröße Dampfleistung aus. Durch Nutzung anderer

⁶Zum Beispiel wünschte der Betreiber bei den in dieser Arbeit untersuchten Zementwerken ausdrücklich nur eine Prognose des Freikalkwertes aber keine Regelung von Stellgrößen.

Messgrößen, wie der O_2 -Emissionen, der Rostendtemperatur, der Zuteilergeschwindigkeit u.a., kann ein neuronales Modell gelernt werden, das die Dampfleistung mit einem Prognosehorizont von 1,5 Minuten approximieren kann. Durch diesen Zeitgewinn konnten die in Kapitel 7.1 aufgeführten Ergebnisse erzielt werden.

Die Aufstellung eines adaptiven NFQ-Systems lohnt sich vor allem, wenn möglichst viele der folgenden Bedingungen zutreffen:

1. Wenn Stellgrößen zur Beeinflussung des Prozesses existieren, aber noch keine automatisierten Regler und kein Expertenwissen vorhanden ist, wie durch eine Stellstrategie Ziele der Prozessführung erreicht werden können. Durch ein Testprogramm können Daten gewonnen werden, auf denen anschließend ein NFQ-System trainiert und eine bessere Prozessführung erreicht werden kann.
2. Wenn eine Stellgröße Einfluss auf mehrere Zielgrößen hat. Dieser lässt sich durch einen einzelnen PID-Regler nicht abbilden (siehe Kapitel 2.1.3).
3. Wenn der Zusammenhang zwischen Stell- und Regelgröße nicht-linear oder nichtmonoton ist (siehe Kapitel 2.1.3).
4. Wenn der Zusammenhang zwischen Stell- und Regelgröße sich durch Revisionen oder andere Zeitinvarianzen signifikant ändert.

Treffen hingegen wenige dieser Bedingungen zu und ist der Zusammenhang zwischen Stell- und Regelgröße bekannt und linear, lässt sich häufig durch ein konventionelles PID-System eine ähnliche Prozessgüte mit geringerem Parametrisierungsaufwand erstellen.

7.4. Fazit

In diesem Kapitel wurde dargestellt, welche Ergebnisse lernfähige adaptive Verfahren im tatsächlichen Betrieb in Müllverbrennungsanlagen und Zementwerken zeigen. Das Kapitel führt damit die einzelnen Verfahren zur Datenauswahl und Modellverbesserung (Kapitel 4), zur Adaption an veränderliche Prozesse (Kapitel 5) und zur Funktionalen Sicherheit von Systemen (Kapitel 6) in einem Gesamtsystem zusammen. Während in Kapitel 4-6 die einzelnen Algorithmen erläutert und auf verschiedenen künstlichen und realen Datensätzen **offline** getestet wurden, stand in diesem Kapitel vor allem der praktische Einsatz **online** auf den Anlagen im Vordergrund.

Dazu wurden in drei verschiedenen Anlagen lernfähige adaptive Verfahren im Vergleich zu bisherigen konventionellen Systemen getestet. Tageweise alternierend wurde über einen Zeitraum von mehreren Wochen zwischen dem konventionellen System und dem adaptiven System umgeschaltet, und es wurden belastbare Kennzahlen über die Prozessführung beider Systeme gewonnen.

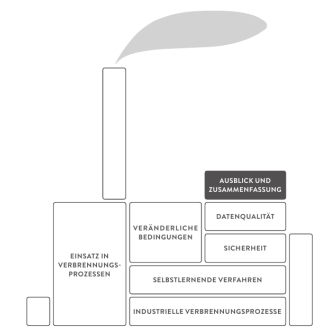
Es zeigte sich, dass sowohl durch die Verwendung von selbstlernenden Prognosemodellen als auch durch lernfähige Modelle, die direkt Stellgrößen beeinflussen, eine Verbesserung gegenüber dem konventionellen System erzielt wird. Im Durchschnitt aller betrachteten Anlagen und Zielgrößen konnte eine Verbesserung von ca. 10% der jeweiligen Zielgröße erreicht werden.

Neben dem Ergebnis des jeweiligen Systems wurde anschließend auch der Aufwand bei der Erstellung und Konfiguration des konventionellen und des adaptiven Systems verglichen. Das adaptive System verfügt über deutlich mehr Parameter und erfordert Wissen über die Prozessdaten und den Lernmechanismus um korrekt eingestellt zu werden. Der Einsatz eines adaptiven Systems lohnt sich deshalb vor allem dann, wenn mehrere Ziele zu beachten sind und der Zusammenhang zwischen der Stellgröße und den Regelgrößen komplex und nichtmonoton ist.

These 11 (*Ergebnisse adaptiver Modelle im Einsatz*)

Lernfähige adaptive Regelverfahren können erfolgreich zur Prozessführung industrieller Verbrennungsprozesse eingesetzt werden. Unter den richtigen Bedingungen erzielen sie deutliche bessere Resultate als konventionelle, nichtlernende Systeme. Ihr Parametrisierungsaufwand ist allerdings deutlich höher als bei konventionellen Systemen, weshalb ihr Einsatz nur bei komplexen Problemen sinnvoll ist.

8. Zusammenfassung



In dieser Arbeit wurde dargestellt, wie lernfähige adaptive Verfahren zum Verbessern der Prozessführung in großen industriellen Verbrennungsprozessen beitragen können. Dazu wurden zuerst die wesentlichen Bestandteile solcher Verbrennungsprozesse dargestellt, und es wurde erläutert, warum konventionelle Regelungsverfahren Teile dieser Prozesse nicht beherrschen können.

Anschließend wurde ein Überblick über verschiedene lernfähige Verfahren gegeben und am Beispiel Neuronaler Netze als lernfähige Modelle erklärt, wie diese prinzipiell in der Lage sind eine bessere Prozessführung zu gewährleisten als konventionelle Regler.

Trotz des Umstandes, dass sowohl die Probleme konventioneller Regler als auch die Vorteile lernfähiger Verfahren seit längerem bekannt sind, werden bislang im industriellen Umfeld lernfähige Verfahren zur Führung von Verbrennungsprozessen nur sehr selten und in sehr beschränktem Umfang eingesetzt.

Im Hauptteil der Arbeit wurde untersucht, was die Ursachen für diesen eingeschränkten Einsatz sind und wie sich das Verhalten sowie

die Lernfähigkeit neuronaler Verfahren verbessern lassen, so dass sich ihre Einsatzfähigkeit vergrößert.

Es wurden dabei drei wesentliche Teilprobleme untersucht:

1. Der Einfluss von fehlerhaften und verrauschten Trainings- und Validierungsdaten auf den Lernvorgang eines datengetriebenen Approximationsmodells.

Dabei konnte gezeigt werden, dass Rauschen auf den Testdaten eine Untergrenze für den Fehler eines Approximationsmodells darstellt. Rauschen auf den Trainingsdaten hat noch größere Auswirkungen und verschlechtert die Approximationsfähigkeit zusätzlich. Die Verschlechterung nimmt dabei zu, je komplexer das Modell ist und je größer die Stärke des Rauschens ist.

Um trotz verrauschter Daten gute Approximationsmodelle zu erhalten, wurden zwei Arten von Verfahren untersucht: Verfahren, die die am meisten verrauschten Punkte (Ausreißer) aus dem Datensatz entfernen und Verfahren, die beim Training von Modellen robust auf Rauschen reagieren. Bei den Verfahren zur Erkennung von Ausreißern schnitt Noise Estimation with Local Linear Models (NELLM) am besten ab, während bei den robusten Trainingsverfahren Bayes'sche Regularisierung (BRG) das beste Resultat erzielte. NELLM [Steege et al., 2012] wurde im Rahmen dieser Arbeit entwickelt, umgesetzt und erfolgreich in Anlagen eingesetzt. BRG wurde im Rahmen dieser Arbeit zum ersten Mal für Anwendungen in der Führung von Verbrennungsprozessen eingesetzt. Durch eine Kombination beider Verfahren konnte der Approximationsfehler der Modelle auf Daten von sieben unterschiedlichen Anlagen um durchschnittlich 7% verbessert werden.

2. Durch verschiedene Prozessänderungen wie zum Beispiel Verschlackungen an den Wänden des Verbrennungskessels, Verschleiß der Anlagenbauteile und Revisionen ändern sich die Merkmale und die Dynamik des Verbrennungsprozesses. In dieser Arbeit wurde untersucht, inwiefern sich lernfähige Modelle an diese Veränderung anpassen können und welche Vor- und Nachteile eine solche Anpassung gegenüber einem statischen Modell hat.

Zur Anpassung eines Modells wurden mehrere Ansätze aus der Fachliteratur erläutert, die sich darin unterscheiden, wie viele Daten und wie viele Modelle genutzt werden, um ein Approximationsmodell zu adaptieren. Diese Ansätze wurden im Rahmen der Arbeit reimplementiert und sowohl auf künstlichen Datensätzen als auch im realen Betrieb der Verbrennungsführung getestet [Steege and Groß, 2012].

Es konnte sowohl auf künstlichen als auch auf realen Daten gezeigt werden, dass ein Adaptieren des Modells bei fast allen Ansätzen zu Vorteilen gegenüber einem statischen Modell führt. Am besten schnitt das System der Datenakkumulation (AKK) ab, welches kontinuierlich Modelle mit allen vorhandenen Daten nachtrainiert. In der Praxis ist allerdings das Verfahren des Ensemble Weighting mit m Modellen (EWm) besser geeignet, da es fast gleich gute Resultate bringt und wesentlich weniger Daten und Trainingszeit aufgewendet werden müssen. Auf den Daten realer Anlagen lässt sich damit eine Verbesserung des Approximationsfehlers von durchschnittlich ca. 20% gegenüber dem nichtadaptierten Modell erreichen.

3. Wird ein durch lernfähige Verfahren erhaltenes Prozessmodell zur Steuerung einer Stellgröße auf der Anlage benutzt, so muss die funktionale Sicherheit dieser Stellstrategie gewährleistet sein. Verfahren, die eine Stellstrategie ausschließlich aufgrund von verfügbaren Daten lernen, scheitern hierbei oft an der Einseitigkeit und Verteilung der Daten, die in bereits geregelten
-

Verbrennungsprozessen vorliegen. Es wurde gezeigt, wie eine solche Verteilung der Daten oft zum Lernen einer falschen Stellstrategie führt.

Um solche Fehler zu vermeiden wurden in dieser Arbeit mehrere sich ergänzende Ansätze entwickelt. Bedeutsam ist dabei vor allem das Einfügen von Expertenwissen in den Lernprozess durch künstliche Datenpunkte [Steege et al., 2010]. Durch Abgleich mit den bereits vorhandenen Daten der Anlage kann gezielt Wissen für unbekannte Bereiche in das Modell integriert werden. Durch dieses Verfahren konnte die Zuverlässigkeit und Einsetzbarkeit neuronaler Regler erheblich erhöht werden.

Weiterhin wurde untersucht, welche Möglichkeiten existieren, um im Falle einer falschen Regelstrategie eines lernfähigen Systems möglichen Schaden von der Anlage abzuwenden. Durch die vorgestellten Regler-Ensembles und Fall-Back-Routinen konnte ein gefahrungsfreier Betrieb der lernfähigen Systeme auch über lange Zeiträume gesichert werden.

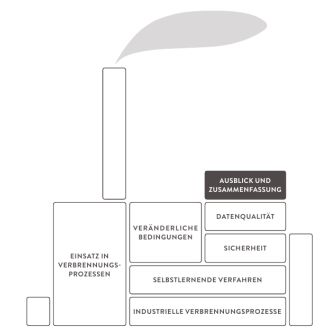
Das wesentliche Alleinstellungsmerkmal dieser Arbeit im Vergleich zu anderen Arbeiten auf dem Gebiet der lernfähigen Prozessmodelle liegt in der erfolgreichen Übertragung dieser fortschrittlichen Konzepte in die industrielle Anwendung mit all ihren hohen Anforderungen bezüglich Verlässlichkeit und Robustheit. Die vorgestellten Verfahren wurden auf verschiedenen Müllverbrennungsanlagen und Zementwerken über mehrere Wochen und Monaten erfolgreich erprobt. Dabei wurden ganz unterschiedliche Stellgrößen, die erheblichen Einfluss auf die Güte des Prozesses haben, untersucht und durch die Modelle im täglichen Betrieb gestellt.

Es konnten Schwachstellen lernfähiger Systeme aufgezeigt werden, welche einen Einsatz zur tatsächlichen Prozessregelung erschweren und erklären, warum neuronale Modelle bisher so selten zur Regelung von Prozessgrößen in der Verbrennungsindustrie eingesetzt wurden.

Für viele dieser Schwachstellen konnten im Rahmen dieser Arbeit Lösungsmöglichkeiten aufgezeigt werden, die es ermöglichen lernfähige Systeme dauerhaft aktiv Stellgrößen regeln zu lassen und so eine erhebliche Verbesserung der Prozessführung gegenüber den bisher eingesetzten konventionellen Systemen zu erreichen.

Durch eine verbesserte Prozessführung wiederum läuft der Feuerungsprozess insgesamt mit geringeren Schwankungen. Es werden dadurch, je nach Anlage, eine effizientere Verbrennung (mehr Leistung oder Produktion bei gleicher Brennstoffzufuhr), geringere Emissionswerte oder ein geringerer Verschleiß der Anlagenbauteile erreicht. Das in der Einleitung dieser Arbeit definierte Problem, dass industrielle Verbrennungsprozesse in Müllverbrennungsanlagen und Zementwerken deutlich schlechter zu steuern sind als andere Verbrennungsprozesse, konnte durch die entwickelten Verfahren verbessert werden. Die Anwendbarkeit lernfähiger Verfahren in diesen herausfordernden Umgebungen wurde durch den praktischen Einsatz in vielen Anlagen gezeigt.

9. Ausblick



In dieser Arbeit wurden Verfahren vorgestellt, mit denen ein lernfähiges und adaptives Modell erfolgreich für den Einsatz in der Feuerleistungsführung großer industrieller Prozesse angewendet wird. Die Ergebnisse zeigen, dass sich mithilfe solcher Modelle eine bessere Prozessführung als mit den bisher eingesetzten konventionellen Systemen erreichen lässt.

Trotz der erzielten Resultate wurden im Rahmen der Arbeit auch Schwachstellen der Verfahren aufgedeckt, welche Raum für weitere Verbesserungen lassen. Wie in Kapitel 7.3 erläutert wurde, ist die Parametrisierung eines adaptiven Modells ein wesentlich komplexerer Vorgang als die Parametrisierung eines einzelnen PID-Reglers. Ebenso ist die Auswertung der Stellstrategie und die Beantwortung der Frage “Warum reagiert der Regler in dieser Situation mit diesem Stellingriff” wesentlich schwerer als bei einem PID-Regler. Neben der Leistung des Reglers sind diese beiden Fragen aber wesentlich für die Akzeptanz des Reglers durch das Personal der Anlage und damit auch für den Einsatz adaptiver Regler in industriellen Prozessen.

Um das Verständnis für die Wirkungsweise des adaptiven Modells zu erhöhen, ist es deshalb erforderlich eine Mensch-Maschine-Schnittstelle (Human Machine Interface (HMI)) zu entwickeln, welche zum einen zur effektiven Parametrisierung der Modelle und zum anderen zur Darstellung ihrer aktuellen Stellstrategie genutzt werden kann. In Abschnitt 9.1 werden die Anforderungen und die mögliche Realisierung einer solchen Schnittstelle skizziert.

Weiterhin wurden in dieser Arbeit als Basis aller adaptiven Modelle stets neuronale Netze in Form von Multi-Layer Perceptrons genutzt. Die vorgestellten Verfahren zur kontinuierlichen Adaptivität der Modelle, zur Verbesserung der Datenbasis und zur Kontrolle der funktionalen Sicherheit lassen sich allerdings auch für andere Basismodelle anwenden. In Abschnitt 9.2 werden andere Basismodelle vorgestellt und ihre Anwendbarkeit diskutiert.

9.1. Parametrisierung und Überwachung mittels HMI

9.1.1. Online Priorisierung von Regelgrößen

In Kapitel 7.2 wurden die Ergebnisse verschiedener Modelle im Betrieb in Müllverbrennungsanlagen und Zementwerken aufgeführt. Oftmals mussten durch die Regler zeitgleich mehrere Regelgrößen berücksichtigt werden. Für das Training des NFQ-Reglers wurden dazu die einzelnen Regelgrößen gewichtet zu einem Rewardsignal verrechnet. Die benutzten Gewichte legen somit die Priorität der einzelnen Regelgrößen fest. Soll diese Priorisierung verändert werden, muss im bisherigen System manuell die Rewardfunktion geändert und anschließend ein neuer Regler trainiert sowie seine funktionale Sicherheit überprüft werden.

Wesentlich besser wäre es, wenn die Priorisierung durch eine Schnittstelle geändert werden kann, ohne dass ein Neutraining und

eine anschließende Überprüfung des Reglers erforderlich ist. In Abbildung 9.1 ist dargestellt, wie eine solche Oberfläche zur Änderung der Priorisierung aufgebaut sein kann.

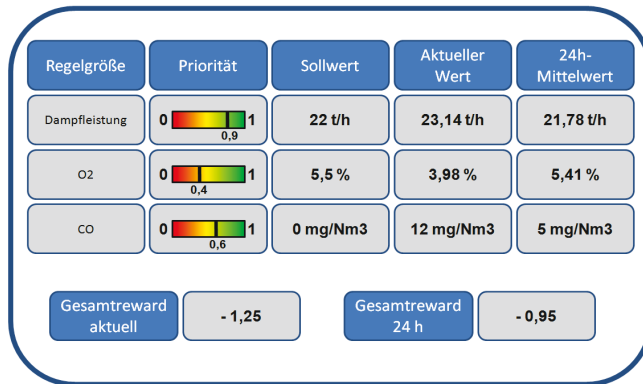


Abbildung 9.1. – Beispiel einer Eingabeoberfläche für die Priorisierung verschiedener Regelgrößen. Zu jeder Regelgröße existiert ein Eingabefeld, in dem die Priorität der jeweiligen Größe zwischen 0 und 1 verändert werden kann. Dem Operator werden außerdem der aktuelle Ist- und Sollwert der Regelgröße, der 24-Stunden Mittelwert sowie der Gesamtward aller verrechneten Regelgrößen angezeigt.

Um diese Funktionalität zu gewährleisten, also die von den Operatoren eingestellten Prioritäten online in den Modellen zu verwenden, sind zwei verschiedene Ansätze möglich:

1. Soll das bisherige NFQ mit einem einzigen Rewardsignal als Modell beibehalten werden, ist eine Veränderung der internen Wichtungen im Rewardsignal ohne Neutraining nicht direkt möglich. Durch einen Umweg kann allerdings trotzdem der Einfluss der einzelnen Regelgrößen geändert werden: Bei den in Kapitel 7.2 verwendeten Modellen wird eine Regelgröße, die Bestandteil des Rewards ist, dem Modell ebenfalls als Eingabegröße präsentiert.

Versieht man diese Eingabegröße mit einem externen Skalierungsfaktor, kann die Abweichung der jeweiligen Regelgröße online verändert werden. Wird beispielsweise der Messwert der Regelabweichung der Dampfleistung künstlich von $1\frac{t}{h}$ auf $1,2\frac{t}{h}$ erhöht, reagiert das Modell stärker auf die Abweichung der Dampfleistung. Der Nachteil dieses Ansatzes besteht darin, dass er eine monotone und lineare Verrechnung der Regelabweichungen innerhalb des Modells voraussetzt. Bei einem NFQ ist dies allerdings nicht der Fall. Möglicherweise reagiert das Modell auf einen künstlich erhöhten Eingangswert also ganz anders als vorhergesehen. Es bleibt zu überprüfen, ob dieser Weg der Priorisierung von Regelgrößen in der Praxis erfolgreich angewendet werden kann, oder ob es besser ist ein neues Modell zu trainieren.

2. Wird statt des neuronalen NFQ-Modells ein anderes Modell verwendet, kann die Priorisierung möglicherweise online geändert werden ohne die oben beschriebenen Nachteile. Ob das möglich ist, hängt davon ab, wie der Reward oder die einzelnen Regelgrößen innerhalb des Modells verarbeitet und gespeichert werden. Wird ein Modell des Gesamtrewards gelernt, ist eine Änderung ohne Neutraining nicht möglich. Werden jedoch innerhalb des Modells die einzelnen Regelgrößen abgebildet und diese in einem nachgelagerten Schritt erst verrechnet, kann die Priorisierung online geändert werden.

Ein Ausgabebildschirm wie in Abbildung 9.1 kann ebenfalls genutzt werden um die Regelstrategie des Modells zu verstehen. Wird ein Modell mit Ansatz 2. trainiert, dann kann neben der tatsächlichen Stellausgabe des Modells auch eine theoretische Stellausgabe für jede einzelne Regelgröße ermittelt werden¹. Gibt man diese “potentiellen” Stellausgaben ebenfalls aus, wird die Regelstrategie des Modells verständlicher für den Operator, was meistens mit einer größeren Akzeptanz des Systems verbunden ist.

¹Also die Stellausgabe, die das Modell tätigen würde, falls nur diese eine Regelgröße berücksichtigt würde, und alle anderen Größen die Priorität Null hätten.

9.1.2. Darstellung aktiver und inaktiver Modelle

Sowohl für die Adaptivität (siehe Kapitel 5) als auch für die funktionale Sicherheit (Kapitel 6) der verwendeten Modelle ist ein Pool mehrerer Modelle erforderlich, aus dem die jeweils besten Modelle für die aktuelle Situation in der Anlage genutzt werden. Im bisherigen System ist für den Operator allerdings kaum ersichtlich, welches Modell aktiv ist und worin die Unterschiede der einzelnen Modelle bestehen. Aufgrund dieser Unkenntnis entsteht oft die Frage, warum das Verfahren in einer Woche anders reagiert als in der anderen und damit verbunden eine generelle Ablehnung des Systems.

Ist hingegen bekannt, wann ein Modell aktiv ist, sind diese Unterschiede leichter zu verstehen. Der Anzeigemodus kann mit einer Möglichkeit zum manuellen Ausschluss von Modellen versehen werden, um so dem Operator die Möglichkeit zu geben, gezielt Modelle zu deaktivieren, die möglicherweise negative Nebenwirkungen haben.

Ist die Darstellung der aktiven und inaktiven Modelle transparent, dann lassen sich aus dem Auswahlmechanismus möglicherweise sogar Rückschlüsse auf den Zustand der Anlage ziehen. Gibt es beispielsweise Modelle, die in einem Zeitraum trainiert wurden, in dem ein bestimmtes Bauteil stark verschlissen war, so werden diese Modelle wieder gute Resultate erzielen, wenn das Bauteil erneut verschleißt. Die Aktivierung eines solchen "Verschleißmodells" könnte also ein Anzeichen sein, dass das jeweilige Bauteil bald erneuert werden muss. In Abbildung 9.2 ist zu sehen wie die Oberfläche für eine solche Modellauswahl aussehen könnte.

9.1.3. Eingabemöglichkeit für Expertenwissen

In Kapitel 6.3 wurde die Möglichkeit vorgestellt, durch das gezielte Einbringen von Expertenwissen die Regelstrategie lernfähiger Regler für seltene Zustände zu verbessern. In dieser Arbeit wurde das Expertenwissen dazu durch Gespräche mit den Operatoren und durch

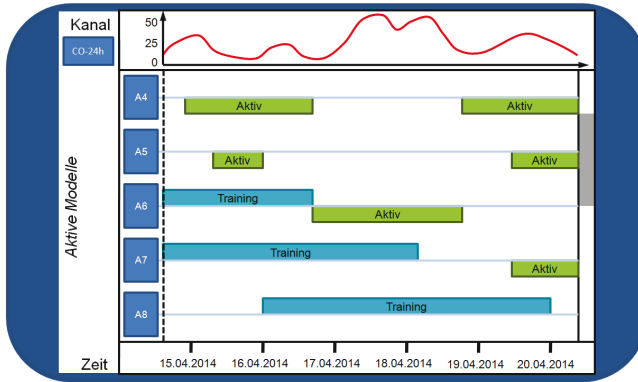


Abbildung 9.2. – Beispiel für den Aufbau einer interaktiven Oberfläche zur Auswertung aktiver Modelle. Im unteren Bereich des Fensters ist die Zeitachse aufgetragen. Im oberen Bereich lassen sich die Verläufe unterschiedlicher Signale anzeigen. Im mittleren Bereich des Fensters wird für jedes Modell des Modellpools angezeigt, aus welchem Zeitbereich die Trainingsdaten stammen und zu welchen Zeiten das Modell aktiv war. Durch Anklicken einzelner Modelle können diese deaktiviert werden, falls sich herausstellt, dass sie ein unerwünschtes Prozessverhalten verursachen.

bereits vorhandenes Prozesswissen gewonnen und dann mittels des in Kapitel 6.3 geschilderten Algorithmus in Daten umgewandelt, die dem Trainingsprozess hinzugefügt wurden.

In der bisherigen Ausführung wird dieses Wissen und die daraus resultierenden Regeln einmal zusammengestellt und dann auf jedes neue Modell angewandt. Dieser Vorgang widerspricht allerdings dem Grundsatz der Adaptivität. Genauso wie sich die Daten einer Anlage ändern, kann sich auch das Expertenwissen und damit die Sonderregeln ändern. Ein Beispiel dafür ist der Einbau von neuen Anlagenelementen während einer Revision, die alte Sonderregeln überflüssig machen oder neue Sonderregeln erfordern.

Ein weiteres Anwendungsszenario sind Störungen in der Anlage, bei denen bis zum nächsten Reparaturtermin die Anlage unter zusätzlichen Randbedingungen gefahren werden muss. Können diese Randbedingungen durch den Operator in das System eingepflegt werden, würde dies die Akzeptanz des Systems durch die Operatoren weiter erhöhen.

Um auf diese Fälle angemessen reagieren zu können, ist es notwendig das Expertenwissen ebenfalls adaptiv zu gestalten. Den Operatoren muss die Möglichkeit gegeben werden, alte Regeln zu verändern oder neue Regeln der Regelbasis hinzuzufügen.

Die Schwierigkeit dabei besteht darin, ein Interface zu erstellen, in dem die Darstellung und Eingabe von Wissen möglichst einfach und intuitiv möglich ist, zugleich aber die Eingabe in verwendbare Regeln und Trainingsdaten umgewandelt werden kann.

Dieses Problem ist allerdings nicht neu. Es ist ein Grundproblem der Mensch-Maschine-Interaktion. In [Knauf, 1993] wird die entsprechende Schnittstelle als Wissenseditor, in [Beierle and Kern-Isberner, 2006] als Wissenserwerbskomponente und in [Altenkrüger and Büttner, 1992] als Wissensakquisitionskomponente bezeichnet. Alle drei Quellen zeigen Wege, mittels derer normales Wissen in eine maschinenverständliche Form gebracht werden kann.

Allerdings lassen sich auf die in dieser Arbeit vorgestellte Anwendung der industriellen Feuerleistungsführung die Standardmethoden zur Eingabe von Wissen aus den folgenden Gründen nur schwer anwenden:

- Wissensbasierte Systeme erwarten oft Wissen in Form von definierten Objekten, Beziehungen zwischen den Objekten, Prädikaten und Aussagen. Experten im Bereich der Feuerleistungsführung denken in Kategorien wie Werten von Messstellen, Bauteilen und Systemgleichungen.
- Das jeweilige Wissen ist oft ungenau bzw. es existiert teilweise

widersprüchliches “Wissen“ bei unterschiedlichen Experten.

- Die in dieser Arbeit vorausgesetzten lernfähigen Systeme erwarten Wissen nicht in der Form von Regeln, wie sie von den Inferenzmaschinen der aufgeführten Quellen verwendet werden, sondern in Form von Datenbeispielen. Beispielsweise könnte es eine formale Regel wie $\forall x(ZS(x) \wedge HH(x) \Rightarrow DH(x))$ geben, die bedeutet “Für alle Anlagenzustände gilt: wenn der Zuteiler schnell fährt und der Heizwert hoch ist, dann steigt der Dampf“. Wie ist dann “alle Anlagenzustände“ in Datenpunkte umzusetzen? Wie müssen die Verhältnisse zwischen der Anzahl der Datenpunkte für unterschiedliche Regeln sein um kein Missverhältnis zwischen den einzelnen Regeln zu erhalten?
- Eine Prüfung der Regeln auf Vollständigkeit und Widersprüchlichkeit muss nicht nur auf den durch Experten aufgestellten Regeln erfolgen. Es müssten auch aus der Datenbasis der Anlage durch Induktion Regeln extrahiert und diese mit den Regeln der Experten auf Widersprüche verglichen werden.

Lösungsmöglichkeiten für diese Probleme sind allerdings denkbar. Objekte können die einzelnen verfügbaren Messstellen sein. Prädikate können beispielsweise Anstiege oder Grenzüberschreitungen sein. Durch die Methoden der Fuzzy-Logik ist es möglich, unscharfes Wissen zu formulieren und in auswertbare Regeln zu übertragen.

Die größte Herausforderung bleibt die Umsetzung dieser Regeln in Datenbeispiele, so dass damit ein sinnvolles Lernen mittels datengetriebener Verfahren möglich ist.

Die Spezifizierung und Umsetzung dieses Konzeptes und der Implementierung einer entsprechenden Oberfläche für den Wissenseditor ist aufwändig. Der Nutzen eines derartigen Systems hängt in bedeutendem Maß zum einen davon ab, ob der Wissenseditor vom Anlagenpersonal akzeptiert und genutzt wird und zum anderen von der Vereinbarkeit zwischen generiertem Wissen und Datenbeispielen des Anlagenbetriebs innerhalb des Lernverfahrens.

9.2. Übertragbarkeit auf andere Basismodelle

In dieser Arbeit werden verschiedenen Verfahren vorgestellt, mit denen die Einsetzbarkeit lernfähiger Modelle zur Prozessregelung in industriellen Verbrennungsanlagen verbessert werden kann. Die verwendeten lernfähigen Modelle waren dabei stets neuronale Netze vom Typ des MLP oder NFQ. Allerdings ist nur eines der vorgestellten Verfahren speziell auf diesen Modelltyp ausgerichtet, nämlich die Bayes'sche Regularisierung (BRG) die mit den Gewichten eines Netzwerkes arbeitet. Alle anderen Verfahren aus den Kapiteln 4-6 arbeiten auf den Daten, die zum Training des Modells verwendet werden, kombinieren mehrere Modelle oder analysieren die Ausgaben eines Modells. Diese Verfahren können auch auf andere Modelltypen angewandt werden, solange diese Modelle zwei Bedingungen erfüllen:

1. Das Modell liefert eine Approximation des funktionalen Zusammenhangs zwischen verschiedenen Eingabegrößen und ein oder mehreren Zielgrößen.
2. Das Modell lernt diese Approximation anhand von Beispieldaten des Prozesses.

In [Bishop, 2006] und anderen Grundlagenbüchern zum Maschinellen Lernen finden sich viele Arten von Modellen, auf die diese Bedingungen zutreffen, und die deshalb prinzipiell statt der in dieser Arbeit verwendeten Neuronalen Netze als Modelle eingesetzt werden können.

Im Folgenden werden drei Arten von Modellen vorgestellt, die prinzipiell auch mit den in dieser Arbeiten vorgestellten Methoden kombiniert werden können.

9.2.1. Lineare Regressionsmodelle

Lineare Regression ist ein Spezialfall der Regressionsanalyse [Backhaus et al., 2006]. Die Grundlagen linearer Regressionsmodelle sind bereits in Abschnitt 3.4.3 dieser Arbeit erklärt. Prinzipiell lassen sich lineare Regressionsmodelle für die in dieser Arbeit erläuterten Anwendungen

ebenso einsetzen und mit den in dieser Arbeit vorgestellten Methoden kombinieren wie neuronale Netze.

Im Unterschied zu neuronalen Netzen verfügen einfache lineare Regressionsmodelle wie in Abschnitt 3.4.3 nicht über zufällige Startinitialisierung. Ein mehrfaches Training des Modells mit denselben Daten führt also nicht zu unterschiedlichen Modellen.

Im Rahmen dieser Arbeit wurden einige Experimente mit linearen Regressionsmodellen durchgeführt. Der durchschnittliche Approximationsfehler der Modelle war vergleichbar mit dem der neuronalen Netze. Allerdings war die Varianz der Ergebnisse deutlich geringer, das heißt lineare Regressionsmodelle führten seltener zu sehr schlechten, aber auch seltener zu sehr guten Approximationsergebnissen.

Das Training linearer Regressionsmodelle in Matlab verlief deutlich schneller als das Training eines neuronalen Netzes. Verbunden mit der Determiniertheit des Ergebnisses, die eine Wiederholung des Trainings auf denselben Daten überflüssig macht, sind lineare Regressionsmodelle somit eine Alternative für neuronale Modelle, solange die zu approximierende Funktion nicht zu komplex ist.

9.2.2. Bayes'sche Modelle

Ein Bayes'sches Modell oder auch Bayes'sches Netz [Bishop, 2006] ist ein gerichteter azyklischer Graph. Die Knoten des Netzes repräsentieren Zufallsvariablen und die Kanten zwischen den Knoten die bedingten Abhängigkeiten dieser Variablen. Bei jedem Knoten wird außerdem die Wahrscheinlichkeitsverteilung der jeweiligen Zufallsvariable gespeichert. Sind bestimmte Zustände bekannt, kann durch die Struktur des Bayes'schen Netzes auf die Wahrscheinlichkeit einer Ausprägung anderer Variablen geschlossen werden.

In [Kalesse, 2008] wird die Implementierung eines solchen Bayes'schen Modells geschildert. Basierend auf dieser Implementierung wurden

bei der Firma des Autors dieser Arbeit, der STEAG Powitec GmbH, Versuche durchgeführt, um Bayes'sche Modelle zur Prognose und Regelung von Verbrennungsprozessen zu nutzen. Die Ergebnisse dieser Versuche sind in [Schaffernicht, 2012] dargestellt. Es zeigte sich dabei, dass der Einsatz des Bayes'schen Systems durchaus gute Resultate bringen kann. Es wurden allerdings zwei Nachteile festgestellt: zum einen ist die Parametrisierung des Bayes'schen Systems sehr aufwändig, zum zweiten erfordert das Bayes'sche System eine sehr lange Berechnungszeit, die es ungeeignet für zahlreiche Stellgrößen macht, bei denen Stelleingriffe im Sekundentakt vorgenommen werden müssen.

Werden diese Nachteile überwunden, können Bayes'sche Modelle auch für andere Bereiche der Feuerungsführung eingesetzt und mit den in dieser Arbeit vorgestellten Methoden kombiniert werden.

9.2.3. Clusteranalyse

Auch mittels Clusteranalyse können Approximationsmodelle zur Nutzung in der Feuerleistungsführung erstellt werden. Ein einfaches Beispiel dafür ist die k-means Clusteranalyse [Lloyd, 1982]. Mittels des Algorithmus werden für eine Menge an Datenpunkten Clusterzentren gebildet. Ein neuer Datenpunkt kann diesen Zentren zugeordnet werden, indem das Zentrum mit der geringsten Euklidischen Distanz im Merkmalsraum zum Datenpunkt gesucht wird.

Eine Approximation eines Zielwertes kann erfolgen, indem jedes Clusterzentrum ein Label erhält über den mittleren Zielwert aller Datenpunkte, die zum jeweiligen Cluster gehören.

Eine Regelstrategie kann extrahiert werden, indem in einem Raum (x_1, \dots, x_n, a, r) bestehend aus Messwerten x_1, \dots, x_n einer Stellgröße a und einer Regelgröße r die Cluster im gesamten Merkmalsraum gebildet werden. Für einen aktuellen Zustand

$(x_1^{Ist}, \dots, x_n^{Ist}, a^{Ist}, r^{Ist})$ wird der Wert der Regelgröße r^{Ist} durch einen Zielwert r^{Soll} ersetzt. Anschließend wird das Euklidisch nächste Clusterzentrum zu $(x_1^{Ist}, \dots, x_n^{Ist}, r^{Soll})$ gesucht, wobei die Dimension

der Stellgröße bei Berechnung der Distanz nicht beachtet wird. Der Wert der Stellgröße im gefundenen Clusterzentrum ist dann der auszuführende Stelleingriff.

Die Firma STEAG Powitec GmbH konnte bereits erste positive Erfahrungen mit solchen Clustermodellen bei der Optimierung von Zementmühlen machen. Vorteil und gleichzeitig Nachteil des Verfahrens ist, dass keine Interpolation für unbekannte Daten stattfindet. Die Ausgabe ist stets abhängig von den existierenden Clusterzentren, die auf realen Daten der Vergangenheit basieren. Die Gefährdung der funktionalen Sicherheit ist deshalb deutlich geringer als bei komplexeren neuronalen Modellen.

Ein Einsatz der Clustermodelle für die Feuerleistungsführung und die Kombination mit den in dieser Arbeit vorgestellten Methoden ist bereits geplant.

A. Merkmalsauswahl

Ein wesentliches Kriterium für die Erstellung von selbstlernenden Prozessmodellen zur Regelung von Stellgrößen, zur Prognose von Messwerten oder von Softsensormodellen ist die Auswahl der richtigen Eingangsgrößen für das Modell.

Werden Messgrößen nicht beachtet, die wesentliche Informationen zum Prozess tragen, oder werden sehr viele Messgrößen ausgewählt, die keine Information zur jeweiligen Zielgröße beinhalten, so ist die Chance ein qualitativ gutes Modell für die entsprechende Zielgröße zu erhalten sehr gering. Besonders bei industriellen Verbrennungsprozessen ist die Auswahl der besten Prozessmessgrößen nicht einfach. Je nach Ausstattung der Anlage mit Sensoren können durchaus 100-200 Messungen vorliegen. Gibt es dabei komplexere Sensoren wie Kameras, Schwingungssensoren oder Mikrophone, die nicht nur einen Messwert, sondern einen Vektor oder eine Matrix an Messwerten liefern, steigt die Anzahl verfügbarer Sensoren stark an. Weiterhin ist oft nicht nur der aktuelle Wert eines Sensors von Bedeutung, sondern auch die Veränderung dieses Wertes innerhalb der letzten Zeitschritte bzw. Messungen der Vergangenheit, deren Auswirkungen erst nach einer gewissen Totzeit und Verzögerung an anderen Stellen im Prozess wirksam werden.

Um aus diesen Informationskanälen relevante Kanäle zu finden, existieren Verfahren der Merkmalsselektion sowie der Merkmalstransformation [Torkkola, 2002]. Bei der Merkmalsselektion wird eine Untermenge relevanter Kanäle aus der Menge der verfügbaren Kanäle gebildet. Ein Kanal wird entweder als relevant oder als irrelevant eingestuft und entsprechend berücksichtigt.

Bei der Merkmalstransformation können Kanäle gewichtet verrechnet werden und so aus mehreren Kanälen ein Kanal entstehen, welcher eine ähnliche Menge an Informationen beinhaltet.

Für die in dieser Arbeit angewandten Modelle sind die Verfahren der Merkmalsselektion und Merkmalstransformation von großer Bedeutung. Trotzdem werden sie im Hauptteil der Arbeit kaum oder nur am Rande erwähnt. Dies geschah aus zwei Gründen. Zum einen ist der Bereich der Merkmalsauswahl ein eigener Forschungsbereich mit vielen existierenden Verfahren für unterschiedliche Anwendungen. Eine ausführliche Beschreibung dieser Verfahren und ihrer Anwendung auf Verbrennungsprozesse hätte den Rahmen dieser Arbeit gesprengt. Zum anderen existiert mit [Schaffernicht, 2012] bereits eine Arbeit, in der die relevanten Verfahren für die hier geschilderte Anwendung ausführlich untersucht wurden. Deshalb sollen in diesem Anhang nur die wesentlichen Grundzüge der Verfahren zur Merkmalsauswahl geschildert werden, die zum Erstellen der Modelle verwendet wurden.

Für die Erstellung von Softsensor- und Prognosemodellen in dieser Arbeit wurde zur Merkmalsauswahl vor allem die Transinformation (Mutual Information (MI)) und darauf aufbauend das Residual Mutual Information (RMI) Verfahren genutzt. Beide Verfahren erzielten in [Schaffernicht, 2012] gute Ergebnisse bei der Auswahl von relevanten Kanälen für die Modellbildung bei Verbrennungsprozessen.

Für die Erstellung von NFQ-Regelmodellen im Bereich der Müllverbrennung, wie in Kapitel 6 dargestellt, wurden außerdem Verfahren zur automatischen Problemdekomposition angewendet. Im Folgenden wird die Funktionalität dieser drei genannten Verfahren sowie die Bedeutung von Totzeiten und Verzögerungen jeweils kurz dargestellt.

A.1. Totzeit und Verzögerung

Unter einer Totzeit versteht man in der Regelungstechnik eine Zeit die vergeht, bevor sich die Änderung einer Eingangsgröße in der Änderung einer Ausgangsgröße bemerkbar macht [Lutz and Wendt, 2002]. Ein Beispiel dafür ist das Verhalten im Mülltrichter einer Müllverbrennungsanlage. Über den Müllkran wird Müll in den Trichter gegeben. Bis dieser Müll durch den Trichter gerutscht ist und im Verbrennungsraum ankommt, vergeht eine Totzeit.

Eine Verzögerung ist ähnlich einer Totzeit. Während sich bei einer reinen Totzeit Eingangs- und Ausgangssignal im Verlauf nicht unterscheidet, wird das Ausgangssignal bei einer Verzögerung zeitliche geglättet [Lutz and Wendt, 2002]. Ein Sprung in einer Eingangsgröße äußert sich also nicht in einem ebensolchen Sprung in der Ausgangsgröße sondern in einem Ansteigen des Signals über einen bestimmten Zeitraum.

Totzeiten und Verzögerungen kommen bei industriellen Verbrennungsprozessen häufig vor und erschweren sowohl die Regelung des Prozesses als auch die Merkmalsauswahl. In Abbildung A.1 ist die Reaktion eines Systems auf einen Sprung in der Stellgröße mit Einbeziehung von Totzeit und Verzögerung dargestellt.

Der negative Einfluss von Totzeiten und Verzögerungen auf die Merkmalsauswahl lässt sich gut in Abbildung A.2 erkennen. Dargestellt sind vier Signale. Signal a dient als Basissignal. Signal b ist identisch mit Signal a. Signal c ist ebenfalls identisch zu Signal a, hat aber eine Totzeit von fünf Zeitschritten und eine Verzögerungszeit von 10 Zeitschritten. Signal d hat eine Totzeit von 10 Zeitschritten und eine Verzögerungszeit von 20 Zeitschritten. Unter dem Verlauf der Signale sind jeweils zwei Signale als Streudiagramm dargestellt. Die meisten Merkmalsauswahl-Verfahren, die den Zusammenhang zweier Kanäle analysieren, nutzen als Ausgangspunkt dieses Streudiagramm.

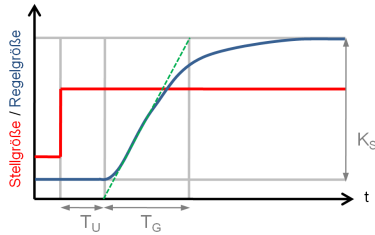


Abbildung A.1. – Idealisierte Antwort einer Regelgröße (blau) auf den Sprung einer Stellgröße (rot). Die Wendetangente (grün) wird genutzt um die Verzugszeit/Totzeit T_U und die Ausgleichszeit/Verzögerung T_G zu bestimmen.

Während der Zusammenhang zwischen Signalkanal a und Signalkanal b sehr eindeutig ist, sind die Streudiagramme zwischen Signal a und Signal c und vor allem zwischen Signal a und Signal d wesentlich weniger eindeutig. Würde zwischen den jeweiligen Signalen die Transinformation oder der Korrelationskoeffizient bestimmt, so ist der berechnete Zusammenhang zwischen Signal a und c sowie zwischen Signal a und d deutlich geringer als zwischen Signal a und b. Umgekehrt würde Signal a also möglicherweise nicht als wichtiges Merkmal bei einer Merkmalsauswahl für Kanäle zur Prognose für Signal d in Betracht kommen.

Um diesem Effekt zu umgehen muss vor einer Merkmalsauswahl eine Totzeit-/ Verzögerungszeitbereinigung vorgenommen werden. Da beide Zeiten bei realen Verbrennungsprozessen in großen Anlagen meist nicht bekannt sind, müssen sie geschätzt werden.

In dieser Arbeit wurde deshalb innerhalb der Merkmalsauswahl jeder potentielle Kanal mit Filtern für unterschiedliche Totzeiten und Verzögerungszeiten gefiltert. Eine Analyse der Zusammenhänge zwischen zwei Kanälen fand dann zwischen dem Zielkanal und allen Filterungen des in Frage kommenden Kanals statt. Auf diese Weise

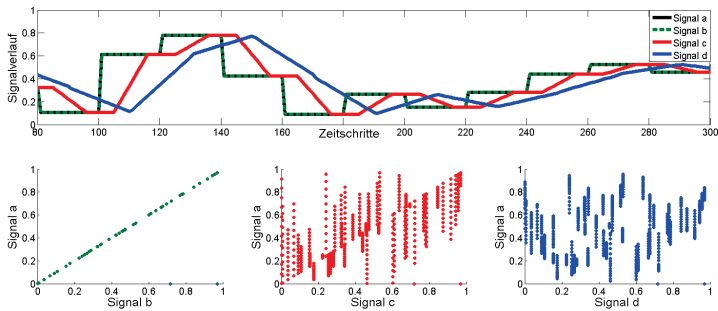


Abbildung A.2. – *Oben:* Verlauf von drei Signalen mit unterschiedlichen Tot- und Verzögerungszeiten im Vergleich zu Signal a. *Unten:* Streudiagramme der drei Signale jeweils im Bezug zu Signal a. Der erkennbare Zusammenhang zwischen den Signalen wird kleiner, je größer die Tot- und Verzögerungszeiten sind.

konnten Totzeiten und Verzögerungen kompensiert werden. Ein Nachteil ist allerdings, dass die Anzahl zu untersuchender Kanäle stark steigt. Gibt es in einer Anlage 10 Messsignale, die potentiell für das Approximationsmodell einer Regelgröße in Frage kämen, aber gleichzeitig 15 mögliche Tot-/Verzögerungszeiten je Kanal, so müssen effektiv 150 verschiedene Kanäle auf ihren Zusammenhang mit der Regelgröße untersucht werden.

Es ist deshalb wichtig ein effektives und schnelles Analyseverfahren für die Merkmalsauswahl zu verwenden. Die folgenden Verfahren sind dazu geeignet, unterscheiden sich aber im Ansatz (Filterverfahren oder Wrapperverfahren) und in ihrer Einsetzbarkeit für verschiedene Fragestellungen. Neben den aufgeführten Verfahren existieren andere Verfahren, die in dieser Arbeit nicht genutzt wurden, potentiell aber ebenfalls geeignet sind. In [Sorjamaa et al., 2007] sowie in [Schaffernicht, 2012] finden sich ausführliche Darstellungen der unterschiedlichen Verfahren und ihrer Anwendbarkeit für das Erstellen von Modellen zur Approximation von Signalverläufen.

A.2. Mutual Information

Die Transinformation (im Folgenden nur mit dem englischen Begriff Mutual Information und der Abkürzung MI benannt) bezeichnet die *Information*, die eine Variable Y über eine andere Variable X enthält. Zum Verständnis der MI ist es notwendig zuerst den Begriff der *Information* im Kontext der MI zu erläutern. Die in diesem Abschnitt verwendeten Definitionen finden sich auch in [Schaffernicht, 2012], Kapitel 3.2 und gehen zurück auf die Arbeiten von [Shannon, 1948] und [Cover and Thomas, 2006].

Information ist dabei eng verbunden mit dem Begriff der *Entropie*. Die Entropie $H(X)$ einer diskreten Zufallsvariable X ist definiert durch:

$$H(X) = - \sum_x p(x) \log p(x) \quad (\text{A.1})$$

wobei $p(x) = \text{Prob}(X = x)$ die Wahrscheinlichkeit ist, dass X die Ausprägung x aus der Menge der möglichen Ausprägungen hat. Die Entropie $H(X)$ ist dann Maß für die Unsicherheit der Zufallsvariable und damit auch für den Erkenntnisgewinn, den man durch Messung des derzeitigen Wertes hat. Deutlich wird dies an einem Beispiel: hat die Variable X nur eine Ausprägung x_1 und diese Ausprägung tritt immer mit der Wahrscheinlichkeit $p(x_1) = 1$ auf, so erhält man durch Messung der Variable keine neue Information. Die Entropie von X ist gleich Null. Die Entropie wird dagegen maximal, wenn X viele Ausprägungen $x_1 \dots x_n$ hat, die alle mit der gleichen Wahrscheinlichkeit auftreten können. Es ist dann schwer vorauszusehen, welche Ausprägung X hat, und eine Messung bringt einen Informationsgewinn über die Ausprägung von X .

Existiert nicht nur eine Variable, sondern mehrere, so lässt sich eine bedingte Entropie $H(X|Y)$ für zwei Variablen X und Y wie folgt definieren:

$$H(X|Y) = - \sum_x \sum_y p(x, y) \log p(x|y). \quad (\text{A.2})$$

$H(X|Y)$ ist dabei die verbleibende Unsicherheit über die Ausprägung von X , falls die Ausprägung von Y bekannt ist. Verringert sich die Unsicherheit über die Ausprägung von X durch Beobachten von Y , dann heißt das, dass in Y eine Information über die Ausprägung von X enthalten ist.

Daraus ergibt sich die Definition der Mutual Information (MI) $I(X; Y)$, welche genau diese Verringerung der Unsicherheit beinhaltet:

$$I(X; Y) = H(X) - H(X|Y) \quad (\text{A.3})$$

Die Transinformation $I(X; Y)$ einer Variable Y zu einer Variable X ist also die Entropie $H(X)$ der Variable X abzüglich der bedingten Entropie $H(X|Y)$ zwischen den Variablen X und Y .

Bildlich kann dieser Zusammenhang wie in Abbildung A.3 dargestellt werden.

Übertragen auf die Feuerleistungsführung in großen Anlagen lässt sich die Transinformation als Maß für den Informationsgehalt eines Messwertes über einen anderen ausdrücken. $I(m_{Dampf}; \phi_{O_2})$ ist demzufolge die Menge an Information, die im Messwert des Sauerstoffanteils ϕ_{O_2} über den Messwert der Dampfleistung m_{Dampf} enthalten ist. Je größer der Wert der Transinformation ist, desto mehr ist ein Kanal geeignet um zur Modellbildung für ein Modell des zweiten Kanals genutzt zu werden.

Um den Wert der Transinformation $I(X; Y)$ für zwei Messgrößen X und Y zu bestimmen, werden allerdings die Werte der Wahrscheinlichkeitsverteilungen $p(x)$, $p(y)$ und $p(x, y)$ benötigt. Diese Verteilungen sind in der Praxis meist nicht bekannt und müssen geschätzt werden. In [Schaffernicht et al., 2009b] werden verschiedene Verfahren zur Schätzung der Wahrscheinlichkeitsverteilungen, der Berechnung der Transinformation und der Nutzung dieser Transinformation zur Selektion von Kanälen für die Modellbildung verglichen. Die Untersuchung brachte allerdings kein klares Ergebnis zugunsten eines

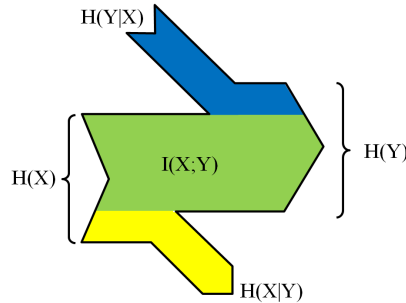


Abbildung A.3. – Die Transinformation $I(X;Y)$ ist die Information, die sich aus der Kenntnis der Variablen Y für die Variable X ableiten lässt. Es ist also die Menge an Entropie, die beide Variablen gemeinsam haben. Der Teil der Entropie von X , der sich nicht in Y findet, wird Verlustentropie $H(X|Y)$ genannt. Der Teil der Entropie von Y , der nichts mit X gemein hat, wird als Fehlinformation $H(Y|X)$ bezeichnet. Ist die Transinformation $I(X;Y) = 0$, ist in der Variable Y also keine Information über X enthalten, gelten die beiden Variablen als unabhängig.

bestimmten Verfahrens sondern zeigte, dass je nach Problemstellung und Verteilung der beteiligten Kanäle unterschiedliche Verfahren am besten abschnitten. In dieser Arbeit wird für die Schätzung der Transinformation der Histogrammbasierte Ansatz gewählt, da er eine geringe Berechnungszeit hat.

Zur Selektion von Kanälen aus einer Menge Y_1, \dots, Y_n für die Erstellung eines Modells des Kanals X wird also die Transinformation $I_i(X; Y_i), i = 1 \dots n$ geschätzt und anschließend die m Kanäle mit dem höchsten Wert der Transinformation als Eingaben des Modells genutzt. Dieses Filterverfahren zur Merkmalsauswahl kann unabhängig von der Art der Modelle, die anschließend genutzt werden, verwendet werden.

Die paarweise Schätzung der Transinformation hat allerdings einen bedeutenden Nachteil: es können nur Zusammenhänge zwischen zwei

Kanälen erkannt werden. Ergibt sich der Wert eines Kanals allerdings erst durch das Zusammenwirken von mehr als zwei Kanälen, wird dieser Zusammenhang durch die paarweise Transinformation möglicherweise nicht erkannt. Es ist zwar möglich die Transinformation auch für mehrere Kanäle zu schätzen, der Rechenaufwand und die Menge der benötigten Beispieldaten steigt dabei allerdings exponentiell.

Um trotzdem solche Zusammenhänge zu erkennen wurde in dieser Arbeit das Residual Mutual Information (RMI) Verfahren verwendet.

A.3. Residual Mutual Information

Wie bereits im letzten Abschnitt erläutert, ist das MI mit der Bestimmung der paarweisen Transinformation nur bedingt geeignet um Zusammenhänge zu erkennen, die sich nur aus dem Zusammenwirken von mehreren Kanälen ergeben. In Abbildung A.4 ist ein Beispiel für einen solchen Zusammenhang erläutert. Jeder der dort genannten Merkmalskanäle X_1, \dots, X_n hat einzeln betrachtet nur eine geringe Transinformation zum Zielkanal Y . Werden sie aber alle zusammen betrachtet beschreiben sie den Zielkanal hingegen vollständig.

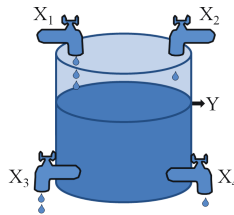


Abbildung A.4. – Es existieren zwei Zuläufe X_1 und X_2 , sowie zwei Abläufe X_3 und X_4 an einem Wasserbehälter. Soll der aktuelle Wasserstand Y geschätzt werden, sind dazu alle vier Messwerte der Zu- und Abläufe notwendig. Jedes Signal X_1, \dots, X_n für sich alleine hat aber nur eine geringe Transinformation mit dem Zielwert Y .

Um solche Abhängigkeiten zu finden könnte zum einen die Verbundtransinformation zwischen allen Kanälen berechnet werden. Der Aufwand dafür steigt allerdings exponentiell mit der Anzahl der vorhandenen Kanäle und lässt sich auf den Kanalmengen realer Anlagen nicht durchführen.

Für diese Arbeit wurde deshalb das Residual Mutual Information (RMI) Verfahren [Schaffernicht et al., 2009a, Schaffernicht, 2012] genutzt, welches durch einen iterativen Auswahlprozess unter günstigen Bedingungen¹ Zusammenhänge zwischen mehreren Kanälen erkennt. Das RMI Verfahren zur Auswahl von Kanälen ist im Unterschied zur Auswahl per MI keine reines Filterverfahren, sondern ein Filter-Wrapper-Verfahren. Es verwendet während des iterativen Auswahlprozesses ein Modell, und die Entscheidung über die Auswahl der Kanäle wird maßgeblich durch das Approximationsergebnis des Modells mit beeinflusst. In jedem Schritt wird das *Residuum*, also die Abweichung der Approximation von jedem Zielwert, bestimmt. Dieses Residuum wird dann genutzt, um einen Kanal auszuwählen, dessen Transinformation mit dem Residuum am größten ist. Anschließend wird ein neues Modell mit den bisher ausgewählten Kanälen trainiert und erneut das Residuum zum Zielkanal bestimmt. Der iterative Ablauf dieses Algorithmus wird in Abbildung A.5 gezeigt.

Das in [Schaffernicht et al., 2009a, Schaffernicht, 2012] vorgestellte Verfahren wurde für die Anwendung in dieser Arbeit an zwei Stellen erweitert. Zum einen wird das Training des Modells in jedem Iterationsschritt mehrfach durchgeführt. Für den nächsten Schritt wird dann das bisher geringste Residuum verwendet um das Risiko einer falschen Kanalauswahl aufgrund einer schlechten Initialisierung des Trainingsmodells zu minimieren. Als zweite Änderung können einmal ausgewählte Kanäle im nächsten Iterationsschritt auch wieder aus der Menge der gewählten Kanäle entfernt werden, falls das Approximationsergebnis nicht besser wird.

¹Voraussetzung dafür ist, dass zumindest eine geringe Transinformation der einzelnen Kanäle zum Ziel besteht. In den praktischen Versuchen dieser Arbeit war das allerdings immer der Fall.

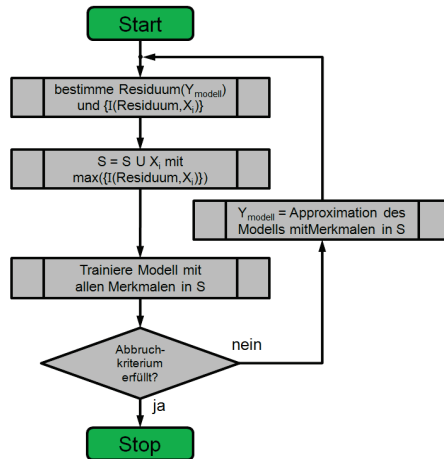


Abbildung A.5. – RMI Verfahren zur Merkmalsselektion. In jeder Iteration wird das Residuum zwischen der Approximation des Modells und dem Zielkanal berechnet. Anschließend wird die Transinformation $I(\text{Residuum}, X_i)$ zwischen dem Residuum und allen Kanälen X_i berechnet, der Kanal mit der größten Transinformation zur Menge der ausgewählten Kanäle S hinzugefügt und ein neues Approximationsmodell trainiert. Dieser Vorgang wird wiederholt bis das Abbruchkriterium (z.B. eine maximale Anzahl an Iterationen oder ein bestimmter Approximationsfehler) erreicht ist.

A.4. Automatische Problem Dekomposition

Die bisher erläuterten Verfahren zur Merkmalsauswahl dienen stets dazu zu einem definierten Zielkanal Merkmale zu finden, mit denen sich ein gutes Approximationsmodell des Zielkanals trainieren lässt. Der Zielkanal ist dabei im Allgemeinen eine Regelgröße, die später durch einen neuronalen Regler geregelt werden soll.

Die Merkmalsauswahl ist allerdings nur eine Teilaufgabe in dem Prozess eine Feuerleistungsführung für eine industrielle Anlage zu erstellen. Ein mindestens ebenso wichtiger Aspekt ist eine Auswahl zu treffen, welche Regelgrößen durch welche Stellgrößen beeinflusst werden sollen. In Kapitel 2.1.1 wurde der Aufbau einer Müllverbrennungsanlage erläutert. Es wurde dabei deutlich, dass viele Stellgrößen existieren um den Prozess zu beeinflussen und gleichzeitig oft mehrere Regelgrößen zu beachten sind.

Theoretisch könnte mithilfe eines neuronalen Netzes oder vergleichbarer Multiple-Input-Multiple-Output (MIMO)-Modelle ein monolithischer Regler trainiert werden, der alle Stellgrößen unter Beachtung aller Regelgrößen stellt. In der Praxis lässt sich ein solches Modell aufgrund der großen Menge an Kanälen und Zielen aber nicht trainieren. Das Gesamtproblem der optimalen Feuerleistungsführung muss also in kleinere Einzelprobleme zerlegt werden. Das heißt es muss festgelegt werden, welche Stellgrößen zur Regelung welcher Regelgrößen verwendet werden, welche Priorität die einzelnen Regelgrößen dabei jeweils haben und welche anderen Messgrößen in den jeweiligen Modellen beachtet werden müssen. Dieser Vorgang wird im Rahmen dieser Arbeit Problem Dekomposition genannt. Soll er nicht vollständig durch einen Experten geschehen sondern automatisch mithilfe von Datenanalysen erfolgen, dann ist es eine automatische Problem Dekomposition. In Abbildung A.6 ist der Prozess einer solchen Problem Dekomposition grafisch dargestellt.

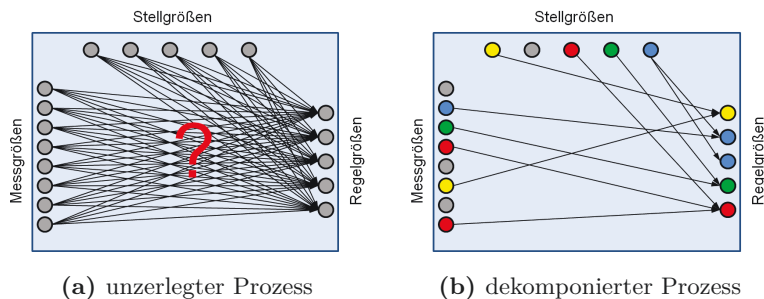


Abbildung A.6. – Problemdekomposition: ein komplexer Prozess mit mehreren Mess-, Stell- und Regelgrößen, dessen genaue Zusammenhänge unbekannt sind, soll in kleine Teilprobleme, die jeweils nur wenige Größen beinhalten, zerlegt werden. Dabei muss für jede Regelgröße mindestens eine Stellgröße gefunden werden.

Das Vorgehen bei einer solchen Problem-Dekomposition ist ähnlich dem der Merkmalsauswahl in den letzten beiden Kapiteln. Allerdings müssen nun mehrere Zielgrößen berücksichtigt werden. Im Rahmen der Arbeit [Schaffernicht, 2012] und dieser Arbeit wurden dabei verschiedene Verfahren untersucht. Grundsätzlich konnten drei verschiedene funktionsfähige Ansätze gefunden werden:

1. **Problemzerlegung aufgrund der Transinformation:** Für alle Variablen des Prozesses wird jeweils die Transinformation zweier Variablen untereinander bestimmt². Anschließend wird ein Graph erstellt. Die Variablen sind die Knoten des Graphen, die Transformationen sind die Gewichte der Verbindungen zwischen den Knoten. In diesem Graphen können dann Schnitte durchgeführt werden, um Teilmengen und damit Teilprobleme zu erzeugen. Eine Nebenbedingung bei der Aufteilung ist, dass jedes Teilproblem mit einem Knoten des Typs Regelgröße auch mindestens einen Knoten des Typs Stellgröße enthalten muss. Eine

²Da jeweils Zweierkombinationen berechnet werden, müssen bei 20 Kanälen also $\binom{20}{2} = 190$ Transformationen berechnet werden.

ausführliche Erläuterung dieses Verfahrens und der durchgeführten Experimente findet sich in [Torke, 2011].

2. **Co-Evolutionäre bottom-up Struktursuche:** Es werden zufällig Teilmengen der Variablen des Prozesses erstellt. Innerhalb dieser Teilmengen werden Modelle für Regelgrößen trainiert und anschließend die Güte der Modelle bewertet. Ein evolutionärer Algorithmus verändert anschließend die Zusammensetzung der Teilmengen und sucht iterativ nach einer Aufteilung, in der die Approximationsgüte am besten wird. Diese Aufteilung wird dann als beste Zerlegung in Teilprobleme angenommen. Erläuterungen zu diesem Verfahren und den Experimenten finden sich in [Kaltenhäuser, 2012, Kaltenhäuser et al., 2013].
3. **Zerlegung durch Clusteranalyse:** Ähnlich wie bei der evolutionären Suche arbeitet dieses Verfahren iterativ. Es wird ein Subraum aus den mehrdimensionalen Datenpunkten ausgewählter Mess- und Stellgrößen gebildet. Dieser Subraum wird mittels des k -means-Clusterverfahrens [Lloyd, 1982] in k Cluster aufgeteilt. Anschließend wird für jede Regelgröße bestimmt, wie groß die Standardabweichung der jeweiligen Regelgröße innerhalb der jeweiligen Cluster ist. Je geringer dieses Maß, desto größer der Zusammenhang zwischen Regelgröße und dem derzeitigen Subraum. Dieses Maß wird dann genutzt um mittels Feature Forward Selection und Backward Subset Elimination eine möglichst optimale Merkmalsmenge zu finden. Eine Veröffentlichung des Verfahrens und der Resultate wird derzeit angestrebt.

Alle drei Verfahren erzeugten auf den Daten einer Müllverbrennungsanlage und eines Kraftwerkes plausible Zerlegungen des Gesamtprozesses in Teilprozesse. Da die Verfahren iterativ arbeiten und auf großen Kanalmengen zum Einsatz kommen, benötigen sie allerdings eine erhebliche Rechenzeit.

Vor dem Einsatz der Verfahren sollte also, ähnlich wie bereits in Kapitel 7.3 beschrieben, eine Abwägung des Parametrisierungs- und Rechenaufwandes gegenüber dem möglichen Nutzen durchgeführt werden. Der Nutzen ist dann groß, wenn wenig Expertenwissen über die Zusammenhänge zwischen einzelnen Stell- und Regelgrößen vorhanden ist. Sind die Zusammenhänge hingegen eindeutig oder lassen sie sich durch wenige Experimente ermitteln, ist der Nutzen durch Einsatz der Verfahren eher begrenzt.

Für die in dieser Arbeit durchgeführten Experimente kam Verfahren 3. zum Einsatz um die Eingabe- und Stellgrößen für die Regler in Kapitel 7.2.1-7.2.3 zu bestimmen. Für die Wahl der Eingabegrößen bei allen anderen Experimenten wurden die in Anhang A.2 und A.3 beschriebenen Verfahren genutzt.

B. Detaillierte Ergebnisse und Verfahrensparameter

In diesem Abschnitt werden zusätzliche Informationen und Ergebnisse zu den Versuchen der Arbeit aufgeführt. Sie dienen vor allem dazu die Nachvollziehbarkeit der Versuche zu erhöhen und können zum Vergleich genutzt werden, falls zu einem späteren Zeitpunkt weitere Versuche folgen sollen.

Soweit möglich sind bei allen Ergebnissen Verlinkungen zu den jeweiligen Abschnitten im Hauptteil der Arbeit angegeben.

B.1. Müllverbrennungsanlagen und Zementwerke

In dieser Arbeit wurden Versuche in verschiedenen Müllverbrennungsanlagen und Zementwerken durchgeführt. Im Hauptteil der Arbeit sind die Anlagen jeweils nur durch ein Kürzel repräsentiert. In der folgenden Tabelle sind zu jeder Anlage auch der Standort und die Produktion des Jahres 2013 angegeben.

Bei den Zementwerken wurden zwei Klassen unterschieden: Zem1-Zem7 sind Anlagen von denen lange Datenbereiche vorlagen und in denen im Rahmen dieser Arbeit Versuche durchgeführt wurden. Zema-ZemD hingegen bezieht sich auf Anlagen in denen von der Firma STEAG Powitec GmbH nur eine Versuchsreihe zur Messung der Varianz des Freikalkwertes durchgeführt wurde ohne das notwendigerweise weitere Daten erhoben oder Regelungs- und Prognosemodelle installiert wurden.

Anlagen-kürzel	Standort	Kapazität	Typ
MVA1	Nordrhein-Westfalen	ca. 27 $\frac{t}{h}$ Müll	3*Vorschubrost, Festlastregelung
MVA2	Spanien	ca. 30 $\frac{t}{h}$ Müll	1*Rückschubrost, Festlastregelung
MVA3	Nordrhein-Westfalen	ca. 27 $\frac{t}{h}$ Müll	3*Vorschubrost, Festlastregelung
Zem1	Spanien	ca. 1,3 $\frac{Mt}{a}$ Zement	1*Drehrohrofen
Zem2	Schleswig-Holstein	ca. 1,4 $\frac{Mt}{a}$ Zement	2*Drehrohrofen
Zem3	Bayern	k.A.	1*Drehrohrofen
Zem4	Tschechien	ca. 1 $\frac{Mt}{a}$ Zement	1*Drehrohrofen
Zem5	Nordrhein-Westfalen	k.A.	1*Drehrohrofen
Zem6	Rheinland-Pfalz	ca. 0,8 $\frac{Mt}{a}$ Zement	2*Drehrohrofen
Zem7	Schleswig-Holstein	ca. 1,4 $\frac{Mt}{a}$ Zement	2*Drehrohrofen
ZemA	USA	ca. 4 $\frac{Mt}{a}$ Zement	1*Drehrohrofen
ZemB	Bayern	k.A.	1*Drehrohrofen
ZemC	Tschechien	ca. 1 $\frac{Mt}{a}$ Zement	1*Drehrohrofen
ZemD	Nordrhein-Westfalen	k.A.	1*Drehrohrofen

Tabelle B.1. – Charakterisierung der in dieser Arbeit für Versuche verwendeten Anlagen.

B.2. Cartpole-Versuchsumgebung

B.2.1. Physik und Parameter des Simulators

Die Cartpole-Simulationsumgebung ist ein sehr bekannter Benchmark für die Güte von Regelungen oder von lernfähigen Prozessmodellen. Das Wort "Cartpole" setzt sich zusammen aus den englischen Wörtern "cart" für Wagen und "pole" für einen Stab. Aufgabe der Simulation ist es einen Stab, der an einem Gelenk auf einem Wagen montiert ist, in einer aufrechten Position zu balancieren. Dazu kann mittels eines Motors eine Kraft auf den Wagen ausgeübt werden. Neben der Aufgabe, den Stab in einer aufrechten Position zu halten, kommt manchmal auch das Problem hinzu, dass die Strecke, auf der der Wagen sich bewegen kann, begrenzt ist oder der Wagen auf eine bestimmte Position der Strecke fahren soll. Das Gesamtproblem ist auch unter dem Namen "inverted pendulum" oder "broom balancing" zu finden. In Abbildung B.1 ist der Aufbau des Simulationsexperimentes dargestellt.

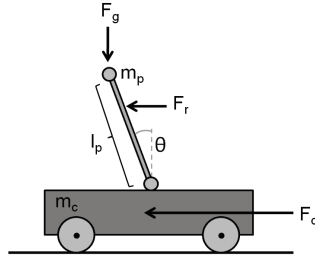


Abbildung B.1. – Modell der Cartpole-Simulationsumgebung.

Die physikalischen Bedingungen der Simulation sind wie folgt: Der Wagen hat eine Masse m_c . Auf dem Wagen ist ein Stab der Länge l_p mit einer Masse m_p montiert. Der Stab kann sich durch ein Gelenk in der Fahrtrichtung des Wagens bewegen. Die aktuelle Auslenkung des Stabes wird durch den Winkel θ beschrieben. Bei einem Winkel $\theta = 0\text{rad}$ steht der Stab aufrecht, bei einem Winkel von $\theta = [90^\circ, -90^\circ] = [\frac{1}{2}\pi, -\frac{1}{2}\pi] = [1.57\text{rad}, -1.57\text{rad}]$ ist der Stab umgefallen¹. Auf den Stab wirkt die Gravitationskraft F_g , die Beschleunigungskraft des Wagens F_c und eine zufällige seitliche Windkraft² F_r .

¹Bei manchen Quellen kann der Stab nach unten durchschwingen, und es sind größere Winkel möglich. Dieses Problem wird dann allerdings meist als “Swing-Up-Pole“ bezeichnet.

²Die zufällige Windkraft F_r macht das Balance-Problem schwieriger und fehlt in einigen Publikationen.

Die Bewegung des Wagens und des Stabes lassen sich laut [Barto et al., 1983] durch die folgenden Formeln beschreiben:

$$F = F_c + F_r \quad (\text{B.1})$$

$$\ddot{x} = \frac{F - m_p l (\ddot{\theta} \cos(\theta) - \dot{\theta}^2 \sin(\theta))}{m_c + m_p} \quad (\text{B.2})$$

$$\ddot{\theta} = \frac{g \sin(\theta)(m_c + m_p) - (F + m_s l \dot{\theta}^2 \sin(\theta)) \cos(\theta)}{\frac{4}{3} l (m_c + m_p) - m_p l \cos^2(\theta)} \quad (\text{B.3})$$

Dabei steht x für die Position, \dot{x} für die Geschwindigkeit³ und \ddot{x} für die Beschleunigung des Wagens. θ , $\dot{\theta}$ und $\ddot{\theta}$ sind Winkel, Winkelgeschwindigkeit⁴ und Winkelbeschleunigung des Stabes.

Bei den obigen Formeln wurde die Reibungskraft nicht berücksichtigt. Zum einen ist die Reibung recht gering und beeinflusst das Problem kaum [Geva and Sitte, 1993], zum anderen werden die Formeln durch Reibung sehr viel komplizierter⁵, weshalb die Reibung bei vielen publizierten Versuchen vernachlässigt wurde.

³Die Position und die Geschwindigkeit des Wagens lassen sich durch $\dot{x}_t = \dot{x}_{t-1} + \ddot{x}_t \cdot \delta t$ und $x_t = x_{t-1} + \dot{x}_t \cdot \delta t$ berechnen, falls die Anfangsposition und die Anfangsgeschwindigkeit des Wagens bekannt sind.

⁴Winkel und Winkelgeschwindigkeit lassen sich wie bei der Position des Wagens aus der Beschleunigung ermitteln.

⁵Laut [Florian, 2007] ist die Reibung in der ursprünglichen Veröffentlichung von [Barto et al., 1983] auch physikalisch nicht korrekt.

Bei den Versuchen in dieser Arbeit wurden folgende Werte für die einzelnen Parameter gewählt:

Parameter	Zeichen	Wert	Wert in [Barto et al., 1983]
Erdanziehung	g	$9,81 \frac{m}{s^2}$	$9,81 \frac{m}{s^2}$
Masse Stab	m_p	$2,0kg$	$0,1kg$
Masse Wagen	m_c	$8,0kg$	$1,0kg$
Länge Stab	l_p	$1,0m$	$1,0m$
Weglänge links/rechts	x_{max}	$\pm\infty$	$\pm 2,4m$
Regelkraft	F_c	$5 \frac{kg \cdot m}{s^2}$	$10 \frac{kg \cdot m}{s^2}$

Die Masse von Stab und Wagen sind in dieser Arbeit höher und die Regelkraft schwächer als in der Arbeit von [Barto et al., 1983]. Dadurch ist das System träger, und die Regelstrategie muss vorausschauend sein. Im Folgenden werden einige Publikationen aufgeführt, in denen das Cartpole-Problem ebenfalls untersucht wurde, und es wird ein Vergleich zu den Versuchen in dieser Arbeit gezogen.

Erste Publikationen zur Lösung des Cartpole-Problems mittels lernender Verfahren finden sich schon in [Michie and Chambers, 1968] sowie in [Widrow and Smith, 1963, Widrow, 1987]. Die bekannteste Publikation, die auch für die Popularität des Cartpole-Experimentes sorgte, ist allerdings [Barto et al., 1983] von Barto, Sutton und Anderson, die darin die Anwendbarkeit von Actor-Critic Methoden auf dieses Problem demonstrierten. Aufbauend darauf wurde die Cartpole-Simulation als Benchmark in vielen anderen Publikationen genutzt. In [Riedmiller, 2005a, Riedmiller et al., 2007b] wird zum Beispiel ein NFQ benutzt um die Regelstrategie des Simulators zu lernen. Andere Reinforcement Learning Ansätze finden sich unter anderem in [Pazis and Lagoudakis, 2009, Dimitrakakis and Lagoudakis, 2008]. In [Riedmiller, 2005b] wurde schließlich sogar ein reales physikalisches Cartpole gebaut und eine Regelstrategie am realen Modell gelernt. Cartpole-Ansätze wurden schließlich so populär, dass auf der Konferenz "Neural Information Processing Systems" (NIPS)

2005 ein eigener Workshop "Lernverfahren am Cartpole-Simulator" durchgeführt wurde [Dutech et al., 2005]. Trotz dieser vielen Beispiele für erfolgreiche Lernstrategien ergeben sich aus der Menge an Quellen auch einige Probleme. Fast alle Autoren verwenden unterschiedliche Parameter der Cartpole-Umgebung, und jede Publikation verwendet ein anderes Qualitätsmaß um die Güte der Regelung festzulegen.

[Barto et al., 1983, Michie and Chambers, 1968, Riedmiller, 2005a] nutzen als Maßstab für die Güte des Systems die Zeitschritte bis zum Umfallen des Stabes in einem Simulationslauf. [Riedmiller et al., 2007b] nutzt ein Signal das beschreibt, wie lange das System innerhalb definierter Grenzen für Stabwinkel und Position blieb. Die Qualität in [Pazis and Lagoudakis, 2009] berechnet sich als $r = -((\frac{2\theta}{\pi})^2 + \dot{\theta}^2 + (\frac{F}{50})^2)$. [Dimitrakakis and Lagoudakis, 2008] gibt die Prozentzahl an "erfolgreichen" Durchgängen an, also Durchgängen, bei denen der Stab mehr als 1000 Schritte nicht um fiel. Die Ansätze in [Dutech et al., 2005] verwenden einen Reward, bei dem unterschieden wird zwischen drei unterschiedlich guten Bereichen. [Geva and Sitte, 1993] verwendet den Auslenkungswinkel des Stabes in rad.

Ebenso unterschiedlich sind die Publikationen, sofern es die Berücksichtigung der Reibung, die Länge des maximal verfügbaren Weges und die Parameter des Wagens betrifft. In [Geva and Sitte, 1993] wird ein Überblick über 12 weitere Publikationen zum Cartpole-Simulator gegeben, in denen dieses Problem ebenfalls auftaucht und die Autoren diese Unterschiedlichkeiten bei der Versuchsdurchführung bemängeln.

Wesentlich wichtiger für diese Arbeit ist allerdings eine Gemeinsamkeit der bisher genannten Quellen: alle lernfähigen Ansätze verwenden zum Training einen Datensatz, der aus zufälligen Beispieldaten besteht [Barto et al., 1983, Michie and Chambers, 1968, Riedmiller, 2005a, Pazis and Lagoudakis, 2009] oder verwenden eine iterative Lernstrategie, bei der ein schlechter Controller durch Ausprobieren mit der Zeit immer besser wird [Riedmiller, 2005b, Dutech et al., 2005, Dimitrakakis and Lagoudakis, 2008]. Auch die in [Geva and Sitte, 1993] aufge-

fürten Ansätze lassen sich in diese beiden Klassen unterteilen. Keiner der untersuchten Ansätze benutzt hingegen einen Datensatz, der ausschließlich von einem bereits geregelten System stammt und versucht auf diesem eine bessere Strategie zu finden ohne ein Umfallen des Stabes zu bewirken. Ein solches Szenario wird erstmals in dieser Arbeit untersucht.

B.2.2. Versuche zum Lernen auf geregelten Daten

In Kapitel 6 werden in Abschnitt 6.2.2 die Ergebnisse verschiedener Versuche mit dem Cartpole-Simulator dargestellt.

Für diese Versuche wurde ein Cartpole mit den im letzten Abschnitt aufgeführten Parametern simuliert. Alle Experimente wurden in Matlab durchgeführt. Ein Simulationschritt entsprach physikalisch 0,1 Sekunden tatsächlicher Zeit. Der benutzte PID-Regler verwendete folgende Parameter:

$$K_P = -2,8 \quad T_N = \infty \quad T_V = 0,35 \quad (\text{B.4})$$

Der Eingabewert des PID-Reglers war die aktuelle Abweichung des Stabwinkels von der aufrechten Position.

Die Parameter des PID sind bewusst nicht ideal gewählt. Es lassen sich für das Cartpole-Problem PID-Parameter finden, die den Stab besser balancieren als der hier gewählte PID. Ziel der Versuche war aber nicht einen möglichst guten PID-Regler für dieses Problem zu finden, sondern zu prüfen ob und unter welchen Bedingungen ein NFQ-Regler ein besseres Resultat erzielen kann. Dazu sollte der PID-Regler nicht optimal parametrisiert sein, da sich sonst kein besseres Resultat erzielen lässt.

Für die Versuche mit dem NFQ-Regler wurde ein MLP mit zwei Hidden-schichten genutzt. In der ersten Hidden-Schicht befanden sich vier und in der zweiten Hiddenschicht drei Neuronen. Der Reward wurde

festgelegt als absolute Abweichung des Winkels von der aufrechten Position:

$$r = -1 \cdot |\theta| \tag{B.5}$$

Der Zustand $s_t = (\theta_t, \dot{\theta}_t, \ddot{x}_t)$ des Systems wurde definiert als aktueller Winkel, Winkelgeschwindigkeit und Beschleunigung des Wagens. Stellgröße a_t war die auf den Wagen wirkende Kraft.

Alle NFQs wurden trainiert mit dem Levenberg-Marquardt-Algorithmus und dem Verfahren der Bayesschen Regularisierung. Der Trainingsdatensatz bestand aus jeweils 500 Samples. Der Parameter γ des NFQ wurde auf Null gesetzt. Mit einem höheren γ lassen sich bessere Resultate erzielen, allerdings steigt die Trainingsdauer damit sehr stark an. An der gezeigten Grundproblematik des Lernens auf geregelten Daten ändert ein höheres γ nichts.

Für das Hinzufügen der virtuellen Punkte wie in Algorithmus 6 wurde der k-Means-Cluster-Algorithmus mit 100 Clusterzentren verwendet. Es wurden pro Cluster $n_{kp} = 1,5$ künstliche Datenpunkte für jeden realen Datenpunkt eingefügt, falls das entsprechende Cluster nicht vollkommen abgedeckt war. Das Histogramm zur Entscheidung über die Abdeckung des Aktionsraumes verfügte über $n_{action} = 11$ gleichgroße Bins. Der Reward der künstlichen Datenpunkte wurde mit einem Abschlag von $r_{kp} = -0.2$ belegt.

Alle Versuche wurden 50 Mal wiederholt. In jedem Versuch wurde eine andere zufällige Startinitialisierung der MLP-Netze gewählt. Alle drei NFQ-Netze wurden mit dieser Startinitialisierung trainiert um eine möglichst große Vergleichbarkeit der Versuche zu gewährleisten. Die Ergebnisse der Versuche sind in Abschnitt 6.2.2 dargestellt.

B.3. Umgang mit schlechter Datenqualität

B.3.1. Gammatest-Pseudocode

Algorithmus 9 Gammatest

Eingabe: Datensatz $S = \{s_i | i = 1, \dots, n; s_i = (\underline{x}_i, y_i); \underline{x}_i \in \mathbb{R}^m; y_i \in \mathbb{R}\}$ bestehend aus Eingabewerten \underline{x} und Zielwerten y

Anzahl nächster Nachbarn k

Ausgabe: geschätztes Rauschen $var(\xi)$

$\overline{dx}_j \leftarrow 0$

$\overline{dy}_j \leftarrow 0$

$dx(s_i, s_j) = \|x_i - x_j\|_2 \forall s_i, s_j \in S, i \neq j$ // bestimme euklidischen Abstand aller Punkte im Eingaberaum

$dy(s_i, s_j) = |y_i - y_j| \forall s_i, s_j \in S, i \neq j$ // bestimme Abstand aller Zielwerte

for all $s_i \in S$ **do**

$N_i \leftarrow \{s_j | j = 1, \dots, k; s_j \in S, i \neq j, s_j \text{ ist } k\text{-nächster Nachbar von } s_i\}$

for $j = 1$ to k **do**

$\overline{dx}_j \leftarrow \overline{dx}_j + \frac{dx(s_i, s_j)^2}{m-1}$ // mittlerer quadratischer Abstand im Eingaberaum

$\overline{dy}_j \leftarrow \overline{dy}_j + \frac{dy(s_i, s_j)^2}{2(m-1)}$ // mittlerer quadratischer Abstand der Ziele

end for

end for

$\overline{dy}_j = a \cdot \overline{dx}_j + b$ // definiere Geradengleichung

// Bestimme a, b mit linearer Regression über \overline{dy}_j und \overline{dx}_j

return $var(\xi) \leftarrow b$

B.3.2. Abschätzung des Rauschens für verschiedene Datensätze

In Kapitel 4.2.1 werden in Abbildung 4.5 die Ergebnisse des Gamma-tests bei der Abschätzung des Rauschens für verschiedene künstliche Datensätze dargestellt. Die den Grafiken zugrunde liegenden Ergebnisse werden im Folgenden in Tabelle B.2 bis B.4 aufgeführt.

Der in Abschnitt 4.2.1 durch den Algorithmus 9 vorgestellte Gammatest benötigt nur einen Parameter: die Anzahl der verwendeten nächsten Nachbarn. Die “richtige” Anzahl der nächsten Nachbarn ist sowohl von der Komplexität des Datensatzes als auch von der Größe des Datensatzes abhängig. In Tabelle B.2 bis B.4 sind deshalb die Ergebnisse bei der Abschätzung des Rauschens jeweils für verschiedene Datensatzgrößen und verschiedene Werte der Anzahl der nächsten Nachbarn aufgeführt. Jede Tabelle enthält die Ergebnisse für Datensatzgrößen zwischen 50 und 10000 Datenpunkten und jeweils 20 bis 100 nächste Nachbarn.

Das künstlich hinzugefügte “tatsächliche Rauschen” ist bei allen Versuchen mittelwertfrei und hat eine Varianz von $var(\xi) = 0,1$. Das Zielsignal (ohne Rauschen) hat eine Varianz von $var(y) = 1$.

Samples	$var(\xi)$	γ 20 Nachbarn	γ 35 Nachbarn	γ 50 Nachbarn	γ 100 Nachbarn
50	0,1	0,086	0,100	0,067	/
100	0,1	0,087	0,091	0,102	0,099
200	0,1	0,087	0,095	0,096	0,103
500	0,1	0,086	0,094	0,096	0,102
1000	0,1	0,086	0,094	0,097	0,101
2000	0,1	0,086	0,094	0,097	0,101
5000	0,1	0,086	0,094	0,097	0,101
10000	0,1	0,086	0,094	0,097	0,100

Tabelle B.2. – Ergebnisse des Gammatests für die Zielfunktion $y = x_1 + x_2$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.2.1.

Samples	$var(\xi)$	γ 20 Nachbarn	γ 35 Nachbarn	γ 50 Nachbarn	γ 100 Nachbarn
50	0,1	0,090	0,127	0,124	/
100	0,1	0,088	0,102	0,106	0,128
200	0,1	0,085	0,097	0,097	0,112
500	0,1	0,087	0,095	0,098	0,103
1000	0,1	0,087	0,094	0,097	0,101
2000	0,1	0,086	0,094	0,097	0,101
5000	0,1	0,086	0,093	0,097	0,101
10000	0,1	0,086	0,094	0,097	0,101

Tabelle B.3. – Ergebnisse des Gammatestes für die Zielfunktion $y = (x_1 + x_2)^2$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.2.1.

Samples	$var(\xi)$	γ 20 Nachbarn	γ 35 Nachbarn	γ 50 Nachbarn	γ 100 Nachbarn
50	0,1	0,592	0,937	1,074	/
100	0,1	0,255	0,525	0,778	1,119
200	0,1	0,124	0,214	0,337	0,784
500	0,1	0,092	0,114	0,137	0,276
1000	0,1	0,087	0,100	0,106	0,141
2000	0,1	0,086	0,095	0,099	0,111
5000	0,1	0,086	0,094	0,097	0,102
10000	0,1	0,086	0,093	0,097	0,101

Tabelle B.4. – Ergebnisse des Gammatestes für die Zielfunktion $y = \sin(x_1) + \sin(x_2)$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.2.1.

B.3.3. Auswirkungen zusätzlicher Eingabekanäle auf die Abschätzung des Rauschens

In Kapitel 4.2.1 wurde untersucht, wie sich zusätzliche Eingabekanäle, die keinen direkten Bezug zum Zielkanal haben, auf die Abschätzung des Rauschens durch den Gammatest auswirken. Die Ergebnisse dieser Versuche sind in Abbildung 4.6 dargestellt. In der unten folgenden Tabelle B.5 sind die numerischen Ergebnisse der jeweiligen Tests aufgeführt.

Bei allen Versuchen wurde der Gammatest mit jeweils 50 nächsten Nachbarn gestartet. Die Versuche wurden mit unterschiedlicher Datensatzgröße und 1-8 zusätzlichen Eingabekanälen durchgeführt. Das künstlich hinzugefügte "tatsächliche Rauschen" ist bei allen Versuchen mittelwertfrei und hat eine Varianz von $var(\xi) = 0,1$. Das Zielsignal (ohne Rauschen) hat eine Varianz von $var(y) = 1$.

Samples	$var(\xi)$	γ orig. Inp.	γ 1 zus. Inp.	γ 2 zus. Inp.	γ 3 zus. Inp.	γ 8 zus. Inp.
50	0,1	0,125	0,160	0,159	0,173	0,132
100	0,1	0,106	0,114	0,106	0,113	0,100
200	0,1	0,097	0,100	0,108	0,114	0,090
500	0,1	0,098	0,095	0,096	0,094	0,084
1000	0,1	0,097	0,096	0,092	0,088	0,078
2000	0,1	0,097	0,093	0,091	0,089	0,074
5000	0,1	0,097	0,094	0,090	0,086	0,070

Tabelle B.5. – Ergebnisse des Gammatestes für die Zielfunktion $y = (x_1 + x_2)^2$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.2.1. Bei allen Versuchen verwendete der Gammatest 50 nächste Nachbarn.

B.3.4. Auswirkungen des Rauschens und der Datensatzgröße auf die Approximationsfähigkeit

In Kapitel 4.3 wurde untersucht, wie sich Rauschen auf einem Datensatz auf die Approximationsfähigkeit datengetriebener Lernverfahren auswirkt. In Abbildung 4.8 sind dazu die durchschnittlichen Approximationsfehler eines MLP und eines linearen Regressionsmodells bei unterschiedlichem Rauschen dargestellt. Die numerischen Werte zu dieser Darstellung sind in Tabelle B.6 aufgeführt.

Das für die Approximation verwendete MLP besaß zwei Hidden-schichten mit fünf Neuronen in der ersten und zwei Neuronen in der zweiten Hiddenschicht. Jedes MLP wurde 100 Schritte mit dem Leven-berg-Marquardt-Algorithmus trainiert. Der verfügbare Datensatz wurde in Trainings-, Validierungs- und Testdaten aufgeteilt. Das Training wurde beendet, wenn entweder der Validierungsfehler anstieg oder die maximale Anzahl an Trainingsschritten erreicht wurden. Der ermittelte Fehler bezieht sich auf den Fehler auf den Testdaten.

Das Lineare Regressionsmodell (LM) wurde jeweils auf denselben Daten trainiert. Dazu wurde die Methode der kleinsten Quadrate mit den Daten der vier Eingabekanäle verwendet.

Jeder Versuch wurde mit beiden Approximationsmodellen je 100 Mal wiederholt. Die in Tabelle B.6 aufgeführten Approximationsfehler sind der Median über den 100 Versuchen.

Eine weitere Untersuchung wurde durchgeführt um herauszufinden, ob ein Zusammenhang zwischen der Anzahl der Neuronen im MLP und der Anfälligkeit des MLP gegenüber verrauschten Daten besteht. In Kapitel 4.3 sind die Ergebnisse dieser Untersuchung in Abbildung 4.9 dargestellt. Die zugehörigen numerischen Werte sind in Tabelle B.7 aufgeführt.

$\frac{\text{var}(\xi)}{\text{var}(y)}$	MLP $n = 100$	LM $n =$ 100	MLP $n = 500$	LM $n =$ 500	MLP $n =$ 1000	LM $n =$ 1000
0%	0,061	0,027	0,000	0,026	0,000	0,025
5%	0,242	0,032	0,023	0,027	0,009	0,026
10%	0,310	0,039	0,035	0,028	0,017	0,026
20%	0,452	0,045	0,068	0,029	0,031	0,027
40%	0,652	0,066	0,127	0,032	0,048	0,029
60%	0,735	0,090	0,236	0,035	0,082	0,030
80%	0,788	0,093	0,256	0,039	0,101	0,032
100%	0,886	0,118	0,255	0,044	0,146	0,035

Tabelle B.6. – Mittlerer Approximationsfehler eines MLP und eines linearen Regressionsmodells für die Zielfunktion $y = x_1 + x_2 + x_3x_4 + x_4 + x_1x_3 + \xi$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.3.

Die verwendeten Trainingsverfahren sind die gleichen wie im vorherigen Abschnitt. Es wurden allerdings drei verschiedene Netzarchitekturen verwendet. $[5,2]$ bezeichnet ein MLP mit zwei Hiddenschichten mit fünf Neuronen in der ersten und zwei Neuronen in der zweiten Hiddenschicht. $[3]$ und $[1]$ bezeichnen dementsprechend MLPs mit nur einer Hiddenschicht mit jeweils drei bzw. einem Hidden-Neuron.

Die Größe des Beispieldatensatzes betrug bei allen Versuchen 100 Samples. Jeder Versuch wurde 100 Mal wiederholt. Die aufgeführten Werte sind der Median aus den Ergebnissen dieser jeweils 100 Versuche.

B.3.5. Approximationsfähigkeit bei Verwendung von Verfahren zur Filterung von Outliern

In Kapitel 4 wurde festgestellt, dass sich Rauschen negativ auf die Approximationsfähigkeit Neuronaler Modelle auswirkt und dass das Approximationsergebnis verbessert werden kann, wenn die am stärksten verrauschten Punkte (Outlier) aus dem Trainingsdatensatz entfernt werden. In Kapitel 4.3.1 werden verschiedene Verfahren vorgestellt, mit denen solche Outlier erkannt werden sollen. Diese Verfahren werden anschließend auf verschiedene reale Datensätze aus dem Bereich der

$\frac{\text{var}(\xi)}{\text{var}(y)}$	MLP [5,2]	LM	MLP [3]	MLP [1]
0%	0,061	0,027	0,017	0,027
5%	0,242	0,032	0,070	0,046
10%	0,310	0,039	0,101	0,068
20%	0,452	0,045	0,216	0,092
40%	0,652	0,066	0,348	0,203
60%	0,735	0,090	0,448	0,263
80%	0,788	0,093	0,564	0,356
100%	0,886	0,118	0,660	0,477

Tabelle B.7. – Mittlerer Approximationsfehler verschiedener MLP-Architekturen und eines linearen Regressionsmodells für die Zielfunktion $y = x_1 + x_2 + x_3x_4 + x_4 + x_1x_3 + \xi$. Die zugehörigen Grafiken und Erläuterungen finden sich in Kapitel 4.3.

Zementherstellung und der Müllverbrennung angewandt und mit den bisherigen Approximationsergebnissen verglichen. Die Ergebnisse im Hauptteil der Arbeit (Tabelle 4.6 bis 4.11) enthalten dabei die relativen Verbesserungen der Approximationsergebnisse im Vergleich zum Standard-Verfahren. In Tabelle B.8 sind dagegen die absoluten Fehler aufgeführt um die Nachvollziehbarkeit und Vergleichbarkeit der Ergebnisse bei möglichen späteren Tests zu gewährleisten.

Die Tests wurden mit den Daten realer Werke durchgeführt. Aus den im jeweiligen Werk verfügbaren Sensormessungen⁶ wurden durch die in Kapitel A aufgeführten Verfahren sowie Expertenwissen Eingabekanäle für das MLP ausgewählt. Die Architektur des MLP wurde durch Experimente mit verschiedenen Netzwerk-Strukturen an der jeweiligen Anlage festgelegt. Für die Versuche wurden folgende Konfigurationen verwendet:

- **Zem1:** *Zeitraum:* 1 Jahr und 10 Monate; *Abtastung:* 1 Sample je Minute / 1 Freikalkwert alle 2 Stunden; *Eingabekanäle:* 6; *Hidden-Neuronen:* 7;
- **Zem2:** *Zeitraum:* 2 Jahre; *Abtastung:* 1 Sample je Minute / 1 Frei-

⁶Die genaue Bezeichnung der einzelnen Werke und der verwendeten Messwerte kann bei Bedarf bei der Firma STEAG Powitec GmbH erfragt werden.

kalkwert alle 30 Minuten; *Eingabekanäle*: 7; *Hidden-Neuronen*: 3;

- **Zem3**: *Zeitraum*: 2 Jahre; *Abtastung*: 1 Sample je Minute / 1 Freikalkwert jede Stunde; *Eingabekanäle*: 9; *Hidden-Neuronen*: 7;
- **Zem4**: *Zeitraum*: 1 Jahr und 7 Monate; *Abtastung*: 1 Sample je Minute / 1 Freikalkwert alle 30 Minuten *Eingabekanäle*: 7; *Hidden-Neuronen*: 3;
- **Zem5**: *Zeitraum*: 1 Jahr und 6 Monate; *Abtastung*: 1 Sample je Minute / 1 Freikalkwert alle 2 Stunden *Eingabekanäle*: 7; *Hidden-Neuronen*: 7;
- **MVA1**: *Zeitraum*: 1 Jahr und 3 Monate; *Abtastung*: 2 Samples je Minute; *Eingabekanäle*: 7; *Hidden-Neuronen*: 5, 3;
- **MVA2**: *Zeitraum*: 1 Jahr; *Abtastung*: 2 Samples je Minute; *Eingabekanäle*: 7; *Hidden-Neuronen*: 3;

Bei Zementwerken existieren zwei Abtastungen: zum einen die normalen Messwerte, die meist minütlich abgetastet werden. Zum anderen die per Labor bestimmten Freikalkwerte, die wesentlich seltener (0,5 bis 4 pro Stunde) vorliegen. Für die Trainingssamples die das MLP benötigt, sind immer Freikalkwerte notwendig, weshalb die längere Abtastrate entscheidend ist. Bei Müllverbrennungsanlagen werden für die Prognose keine Labormesswerte benötigt, weshalb die Abtastung wesentlich schneller erfolgt.

Aus diesen Basisdaten wurden für jedes Werk die Daten für das Training eines Netzwerkes ausgewählt. Dazu wurde zufällig ein Anfangszeitpunkt festgelegt. Ausgehend von diesem Anfangszeitpunkt wurden bei Zementwerken (je nach Abtastung) 200-500 Samples ausgewählt. Dies entspricht einem Zeitraum von 15-30 Tagen. Bei Müllverbrennungsanlagen wurden 1500 Samples ausgewählt, was einem Zeitraum von 12 Stunden entspricht. Auf diesen Daten wurden mit den in Tabelle B.8 aufgeführten und in Kapitel 4.3.1 erklärten Verfahren MLP-Prognosemodelle trainiert. Die verwendeten MLP

erhielten vor Beginn des Trainings eine zufällige Initialisierung. Diese Initialisierung wurde allerdings für alle Verfahren übernommen um eine möglichst große Vergleichbarkeit der Ergebnisse zu gewährleisten.

Die Trainingsdaten wurden nach folgendem Schema aufgeteilt: die letzten 20% wurden zum Testen und zur Bestimmung der Fehler genutzt. Weitere 20% wurden zur Bestimmung des Validierungsfehlers genutzt. Von den verbleibenden 60% wurden bei den Verfahren, die Outlier entfernen (NELLM, NELLM-BRG, LOF, NELLM-INI) die 20% am meisten verrauschten Datenpunkte entfernt.

Jeder Test wurde bei Zementwerken jeweils 500 Mal und bei Müllverbrennungsdaten je 50 Mal mit anderen Startzeitpunkten wiederholt. In Tabelle B.8 ist der Median des Fehlers aus diesen Versuchen angegeben.

Bei den Verfahren NELLM, NELLM-BRG, LOF, NELLM-INI wurden jeweils 20 nächste Nachbarn zur Bestimmung der Outlierness genutzt. Alle MLPs wurden jeweils 100 Schritte mit dem Levenberg-Marquardt-Algorithmus oder dem Verfahren nach BRG trainiert.

$\overline{mse}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5	MVA1	MVA2
Normal	0,679	1,184	0,718	1,539	0,973	0,386	0,808
BRG	0,598	1,140	0,671	1,578	0,857	0,383	0,809
NELLM	0,621	1,106	0,640	1,497	0,901	0,378	0,794
NELLM-BRG	0,574	1,122	0,619	1,519	0,853	0,372	0,798
LOF	0,782	1,071	0,767	1,397	0,978	0,391	0,826
LMLS	0,664	1,203	0,708	1,547	0,968	0,386	0,814
LLM	0,628	1,444	0,730	2,414	0,941	0,458	0,983
INI	0,651	1,175	0,669	1,526	0,934	0,375	0,794
NELLM-INI	0,617	1,091	0,657	1,490	0,908	0,374	0,787

Tabelle B.8. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers pro Verfahren. Im Unterschied zu Tabelle 4.6-4.11 sind hier die absoluten Fehlerwerte und nicht die prozentuale Verbesserung gegenüber dem Standardverfahren angegeben.

B.4. Veränderliche Prozesse

In Kapitel 5 wird analysiert, welche Auswirkungen eine Veränderung des Prozesses auf die Approximationsleistung von Modellen hat. Diese Veränderung wird “concept drift“ genannt. Es wird gezeigt, dass es vorteilhaft ist Modelle an neue Daten anzupassen. Gleichzeitig wird aber auch gezeigt, dass eine Anpassung negative Auswirkungen haben kann, wenn alte Zusammenhänge verlernt werden oder durch Widersprüche in den Daten gar kein Zusammenhang mehr approximiert werden kann.

In Abschnitt 5.2 wird ein Testscenario vorgestellt, mit dem die Auswirkungen von concept drift auf die Approximationsleistung untersucht werden kann. Durch Veränderung des Parameters α innerhalb des Testscenario können unterschiedliche Fälle von concept drift simuliert werden. Für die Untersuchungen dieser Arbeit wurden drei zeitliche Verläufe für α vorgesehen:

- ein mehrfaches Ansteigen und Zurückgehen des Parameters α , ähnlich einem Sägezahn, in Datensatz kD1,
- ein kontinuierliches Ansteigen von α , ohne ein Zurückgehen auf den Anfangswert in Datensatz kD2 und
- keine Veränderung des Parameters α in Datensatz kD3.

Außer diesen künstlichen Datensätzen werden reale Datensätze aus fünf verschiedenen Zementwerken und zwei unterschiedlichen Müllverbrennungsanlagen verwendet.

In den Abschnitten 5.3 bis 5.5 werden verschiedene Verfahren vorgestellt, mit denen Modelle an veränderliche Prozesse angepasst werden können. Zum einen sind dies Verfahren, die nur mit jeweils einem Modell arbeiten (ohne Adaption (OA), Datenakkumulation (AKK), Bereichsauswahl (BAW) und Bereichsauswahl mit Kontrolle (BAWk)), zum anderen auch Verfahren die einen Pool aus mehreren Modellen verwenden (Ensemble Selection (ES), Ensemble Weighting (EW) und Ensemble Weighting mit m Modellen (EW m)). Bei allen Verfahren

wird untersucht ob es vorteilhaft ist die Parameter der alten/aktiven Modelle als Startwert zu benutzen (Verfahren 1) oder die Modelle bei jedem Training neu zu initialisieren (Verfahren 2).

Für alle Tests wurden MLPs als Approximationsmodelle genutzt, die mit dem Levenberg-Marquardt Algorithmus trainiert wurden. 75% der Trainingsdaten wurden zum Training des Modells und 25% zur Validierung genutzt. Die in Kapitel 5 und den unten folgenden Tabellen ermittelten Fehler stammen von Testdaten, die nicht zum Training desselben Modells verwendet wurden. Für die künstlichen Datensätze kD1-kD3 wurden MLP mit drei Hiddenneuronen und fünf Eingangsneuronen verwendet. Jeder künstliche Datensatz enthielt jeweils 4000 Datenpunkte.

Die realen Datensätze und die auf den Datensätzen trainierten Modelle entsprachen den in Abschnitt B.3.5 bereits vorgestellten Modellen.

In den Abschnitten 5.3 bis 5.5 werden die Ergebnisse der einzelnen Verfahren auf den jeweiligen Datensätzen vorgestellt und erläutert. Um die Effekte der einzelnen Adaptionsmechanismen besser vergleichen zu können sind dabei nur die relativen Verbesserungen des Fehlers im Vergleich zum Standardverfahren ohne Adaption des Modells aufgeführt. In Tabelle B.9 und Tabelle B.10 sind für die Nachvollziehbarkeit der Experimente allerdings auch die absoluten Fehler aufgeführt. Die absoluten Fehler auf den Daten der einzelnen Zementwerke unterscheiden sich in ihrer Höhe, da je nach Werk unterschiedliche Messwerte mit verschiedener Genauigkeit vorliegen, so dass die Prognose des Freikalkwertes mit unterschiedlicher Genauigkeit vorgenommen werden kann.

$\bar{e}_{Q50\%}$	Zem1	Zem2	Zem3	Zem4	Zem5	\bar{Zem}
OA	0,512	0,722	1,362	0,704	1,016	0,865
AKK 1	0,446	0,659	0,818	0,636	0,826	0,677
AKK 2	0,426	0,604	0,732	0,635	0,792	0,638
BAW 1	0,721	0,687	0,857	0,710	0,866	0,768
BAW 2	0,558	0,655	0,876	0,708	0,881	0,736
BAWk 1	0,881	0,662	0,848	0,716	0,828	0,787
BAWk 2	0,534	0,681	0,831	0,678	0,892	0,723
ES 1	0,475	0,665	0,783	0,682	0,888	0,699
ES 2	0,474	0,658	0,753	0,681	0,874	0,688
EW 1	0,450	0,625	0,844	0,649	0,829	0,679
EW 2	0,532	0,720	0,860	0,672	0,803	0,717
EWm 1	0,417	0,625	0,752	0,632	0,825	0,650
EWm 2	0,420	0,615	0,746	0,630	0,817	0,646
Rev.	3	4	0	2	3	

Tabelle B.9. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 200 Versuchen pro Verfahren und Datensatz. Zusätzlich aufgeführt sind auch die Anzahl der Revisionen im Testzeitraum bei Zementwerken. Die Zahl \mathcal{Z} bezeichnet Verfahren, bei denen neu initialisierte Modelle verwendet werden, während bei Verfahren mit 1 die Gewichte der alten Netze als Startwert genutzt werden.

$\bar{e}_{Q50\%}$	kD1	kD2	kD3	$k\bar{D}$
OA	0,306	0,336	0,135	0,259
AKK 1	0,221	0,217	0,126	0,188
AKK 2	0,288	0,284	0,207	0,260
BAW 1	0,204	0,157	0,130	0,164
BAW 2	0,309	0,277	0,255	0,280
BAWk 1	0,195	0,156	0,129	0,160
BAWk 2	0,247	0,166	0,134	0,182
ES 1	0,190	0,167	0,132	0,163
ES 2	0,192	0,169	0,133	0,165
EW 1	0,229	0,232	0,136	0,199
EW 2	0,481	0,530	0,452	0,488
EWm 1	0,189	0,168	0,129	0,162
EWm 2	0,192	0,171	0,130	0,164
Rev.	3	0	0	

Tabelle B.10. – Median $\bar{e}_{Q50\%}$ des Approximationsfehlers über je 200 Versuchen pro Verfahren und Datensatz. Zusätzlich aufgeführt sind die Anzahl der Parameter-Rücksetzungen bei den künstlichen Datensätzen.

Die Parameter für die Adaptionenverfahren sind abhängig von der Abtastrate und der Menge der Trainingsdaten pro Zeiteinheit. Für die einzelnen Adaptionenverfahren wurden (neben den schon beschriebenen Parametern zu Aufbau und Training des MLP-Modells) folgende Parameter gewählt:

- t_{min} : Zement und kD: 50-100; MVA: 800; verwendet von BAW, BAWk, ES, EW, EWm
- t_{max} : Zement und kD: 50-200; MVA: 1200; verwendet von BAW, BAWk, ES, EW, EWm
- $t_{max,train}$: Zement und kD: 50; MVA: 200; verwendet von BAWk, ES, EW, EWm
- $t_{max,test}$: Zement und kD: 50; MVA: 200; verwendet von BAWk, ES, EW, EWm
- m von P^m : das Maximum von 5 und der 15% der Anzahl der Netze im aktuellen Pool P ; verwendet von EWm

In Tabelle B.11 ist die Standardabweichung der Fehler aus Tabelle B.9 aufgeführt. Die Standardabweichung bezieht sich dabei auf die Veränderung des $\bar{e}_{Q50\%}$ -Approximationsfehlers während der je 200 Versuche pro Datensatz. Es ist deutlich erkennbar, dass die Standardabweichungen auf realen Daten größer ist als auf künstlichen Daten. Gleichzeitig wird deutlich, dass sich mit Ensemble-Verfahren eine deutlich geringere Standardabweichung des Fehlers erreichen lässt als mit Verfahren, die jeweils nur ein Modell verwenden.

In den Tabellen B.11 und B.9 werden die Werte dargestellt, die auf Zementdaten erreicht wurden. Die Werte auf den MVA-Datensätzen sind in Tabelle B.12 und Tabelle B.13 dargestellt. Sie bestätigen die Werte, die auf Zementdaten erreicht wurden.

$std(\bar{\epsilon})$	Zem1	Zem2	Zem3	Zem4	Zem5	kD1	kD2	kD3
OA	0,199	0,077	1,458	2,344	0,580	0,011	0,010	0,004
AKK 1	0,051	0,067	0,258	0,128	0,019	0,005	0,004	0,002
AKK 2	0,006	0,015	0,024	0,009	0,013	0,009	0,008	0,010
BAW 1	0,307	0,754	0,307	0,421	3,425	0,005	0,003	0,003
BAW 2	0,045	0,026	0,067	0,035	0,035	0,013	0,013	0,011
BAWk 1	1,238	0,595	0,380	1,663	0,781	0,005	0,003	0,002
BAWk 2	0,039	0,051	0,172	0,048	0,051	0,009	0,003	0,003
ES 1	0,028	0,027	0,086	0,037	0,052	0,005	0,004	0,003
ES 2	0,028	0,027	0,085	0,044	0,050	0,006	0,004	0,003
EW 1	0,012	0,018	0,105	0,029	0,045	0,008	0,006	0,005
EW 2	0,030	0,035	0,200	0,068	0,025	0,012	0,009	0,013
EWm 1	0,015	0,019	0,095	0,028	0,028	0,005	0,003	0,003
EWm 2	0,014	0,023	0,090	0,018	0,070	0,005	0,004	0,003
Rev.	3	4	0	2	3	3	0	0

Tabelle B.11. – Standardabweichung $std(\bar{\epsilon})$ des Approximationsfehlers über je 200 Versuchen pro Verfahren und Datensatz. Zusätzlich aufgeführt sind auch die Anzahl der Revisionen im Testzeitraum bei Zementwerken bzw. die Anzahl der Parameter-Rücksetzungen bei den künstlichen Datensätzen. Die Zahl 2 bezeichnet Verfahren, bei denen neu initialisierte Modelle verwendet werden, während bei Verfahren mit 1 die Gewichte der alten Netze als Startwert genutzt werden.

$\bar{\epsilon}_{Q50\%}$	MVA1	MVA2	\overline{MVA}
OA	0,236	0,499	0,368
AKK 1	0,210	0,412	0,311
AKK 2	0,208	0,407	0,308
BAW 1	0,233	0,439	0,336
BAW 2	0,232	0,438	0,335
BAWk 1	0,229	0,417	0,323
BAWk 2	0,236	0,487	0,362
ES 1	0,228	0,452	0,340
ES 2	0,236	0,458	0,347
EW 1	0,217	0,434	0,326
EW 2	0,387	0,606	0,497
EWm 1	0,214	0,433	0,324
EWm 2	0,216	0,436	0,326
Rev.	2	1	

Tabelle B.12. – Median $\bar{\epsilon}_{Q50\%}$ des Approximationsfehlers über je 40 Versuchen pro Verfahren und MVA-Datensatz. Im Unterschied zu Tabelle in Kapitel 5.5 sind hier die absoluten Fehlerwerte angegeben.

$std(\bar{\epsilon})$	MVA1	MVA2	\overline{MVA}
OA	0,027	0,227	0,127
AKK 1	0,013	0,004	0,009
AKK 2	0,001	0,002	0,002
BAW 1	0,045	4,106	2,076
BAW 2	0,005	0,004	0,005
BAWk 1	0,093	0,030	0,062
BAWk 2	0,027	0,489	0,258
ES 1	0,116	0,014	0,065
ES 2	0,033	0,289	0,161
EW 1	0,014	0,019	0,017
EW 2	0,022	0,037	0,030
EWm 1	0,007	0,011	0,010
EWm 2	0,009	0,143	0,076
Rev.	2	1	

Tabelle B.13. – Standardabweichung $std(\bar{\epsilon})$ des Approximationsfehlers über je 40 Versuchen pro Verfahren und MVA-Datensatz.

B.5. Sicherheit und geregelte Prozesse

B.5.1. Parameter und Aufbau der Simulation

In Kapitel 6 wurde festgestellt, dass in großen Verbrennungssystemen besondere Anforderungen an die Führung von Stellgrößen bestehen. Im Besonderen gilt dies für Stellgrößen, die bereits durch einen automatischen Regler oder einen menschlichen Operator gestellt werden. Eine Anforderung ist dabei, dass die Stellstrategie des neuen Verfahrens nicht allzu stark von der bisherigen Stellstrategie abweichen soll. Zur Erläuterung dieser Problematik wurden mit dem Cartpole-Simulator verschiedene Szenarien simuliert. Weiterhin wurden in Abschnitt 6.2.3 und 6.2.4 Versuche mit Stellgrößen von Müllverbrennungsanlagen und Zementwerken vorgenommen um das Stellverhalten zu testen.

Wie in den vorherigen Abschnitten auch wurden die Tests mit den Daten realer Werke durchgeführt. Durch die in Kapitel A aufgeführten Verfahren wurden die notwendigen Eingabekanäle des NFQ ausgewählt. Die Architektur des NFQ wurde durch Erfahrungen mit verschiedenen Strukturen an der jeweiligen Anlage festgelegt. Für die Versuche wurden folgende Konfigurationen verwendet:

- Stellgröße **Cartpole-Kraft**: *Simulationsschritte*: 50/500 Schritte; *Eingabekanäle*: 4; *Hidden-Neuronen*: [4 3]; *Versuchswiederholungen*: 50;
 - Stellgröße **Rostgeschwindigkeit**: *Zeitraum*: 6 Tage; *Abtastung*: 6 Samples je Minute; *Eingabekanäle*: 8; *Hidden-Neuronen*: [4 3]; *Versuchswiederholungen*: 100;
 - Stellgröße **Zuteilergeschwindigkeit**: *Zeitraum*: 13 Tage; *Abtastung*: 1 Sample je Minute; *Eingabekanäle*: 10; *Hidden-Neuronen*: [4 3]; *Versuchswiederholungen*: 50;
 - Stellgröße **Ammoniakwasser-Menge**: *Zeitraum*: 11 Tage; *Abtastung*: 6 Samples je Minute; *Eingabekanäle*: 7; *Hidden-Neuronen*: [4 3]; *Versuchswiederholungen*: 100;
-

- Stellgröße **Energieeintrag**: *Zeitraum*: 6 Monate; *Abtastung*: 1 Sample je Minute; *Eingabekanäle*: 8; *Hidden-Neuronen*: [4 3]; *Versuchswiederholungen*: 50;

Der Datenbereich für die Stellgröße Energieeintrag ist wesentlich länger gewählt als für die anderen Stellgrößen. Dies hängt mit der Frequenz der effektiven Änderung der Stellgröße zusammen. Während die Stellgrößen Rostgeschwindigkeit und Ammoniakwasser-Menge alle 10 Sekunden und die Stellgröße Zuteilergeschwindigkeit jede Minute durch automatisierte Regler geändert werden, wird die Stellgröße Energieeintrag im Durchschnitt nur alle 4 Stunden von einem Operator geändert.

Für die Generierung der künstlichen Datenpunkt nach Algorithmus 6 wurden bei allen Verfahren folgende Parameter genutzt: Rate künstlicher Punkte pro Cluster $n_{kp} = 2,0$, Rewardabschlag für künstliche Punkte $r_{kp} = 0,4$ und Anzahl Action-Bins $n_{action} = 20$. Für die Clusterung des Zustandes wurde ein k-Means-Clusterer mit (je nach Menge der Daten) 50-250 Clustern genutzt.

Die verwendeten NFQ erhielten vor Beginn des Trainings eine zufällige Initialisierung. Diese Initialisierung wurde allerdings für alle Verfahren übernommen um eine möglichst große Vergleichbarkeit der Ergebnisse zu gewährleisten.

Als Trainingsverfahren diente bei allen Tests der Levenberg-Marquardt-Algorithmus unter Verwendung von Bayesscher Regularisierung. Von den verfügbaren Daten wurden 65% zum Training und 25% zur Validierung genutzt. Die verbleibenden 10% dienten als Testdaten und zur Bestimmung der in Kapitel 6 aufgeführten Zahlen.

B.5.2. Erweiterte Versuchsergebnisse

In Tabelle 6.1 bis Tabelle 6.6 werden die Ergebnisse verschiedener Trainingsverfahren auf den im letzten Abschnitt vorgestellten Datensätzen analysiert. In Kapitel 6 wurden stets die Mediane der jeweils berechneten Gütekriterien der einzelnen Untersuchungen dargestellt. Es ist

allerdings nicht nur interessant den Median, also die häufigste Ausprägung, sondern auch andere Quantile zu betrachten. Das 10%-Quantil $Q_{0,1}$ und das 90%-Quantil $Q_{0,9}$ können Auskunft darüber geben, wie groß die Abweichung im besten und im schlechtesten Fall ist. Der erste Fall ist interessant, falls mittels eines Algorithmus viele Modelle gleichzeitig trainiert werden und anschließend das Beste ausgewählt werden kann. Der zweite Fall ist interessant um zu ermitteln, wie groß das Risiko "im schlimmsten Fall" ist eine abweichende Strategie zu erhalten. In den folgenden Tabellen B.14-B.18 sind deshalb auch die 10% - und 90%-Quantile zu allen durchgeführten Versuchen aufgeführt.

	$Q_{0,1}(\bar{e})$	$Q_{0,5}(\bar{e})$	$Q_{0,9}(\bar{e})$
PID-Regler	0,053	0,094	0,128
NFQ, Zufalls-Daten	0,039	0,068	0,117
NFQ, PID-Daten	0,657	0,767	0,878
NFQ, PID-Daten + kDP	0,065	0,109	0,213
idealer Regler	0,021	0,033	0,043

Tabelle B.14. – Winkelabweichungen der Cartpole-Simulation mit unterschiedlichen Reglern. Jede Simulation über je 50 Simulationschritte wurde 50 Mal durchgeführt. Der aufgeführte Fehler entspricht dem 10%, 50%- und 90%-Quantil über 50 Simulationen.

Die erste Untersuchung behandelt die Veränderung der Qualität der Regelstrategie am Cartpole-Simulator bei unterschiedlichen Trainingsverfahren. \bar{e} steht für die mittlere Winkelabweichung der Cartpole-Simulation vom Sollwert.

An den Ergebnissen des Cartpole-Simulators fällt auf, dass ein mit geregelten Daten trainierter NFQ ohne künstliche Datenpunkte selbst bei der Auswertung der 10% besten Versuche keine sinnvolle Regelstrategie findet. Dies spricht dafür, dass ein Training auf bereits geregelten Datenpunkten mit diesem Trainingsverfahren nicht zum Erfolg führen kann. Es müssen bessere Datenbeispiele gefunden werden (wie die

Ergebnisse mit Zufallsdaten in der dritten Tabellenzeile zeigen), die Datenbeispiele um künstliche Punkte ergänzt werden oder ein anderer Regelungsansatz gewählt werden.

Für die Datensätze realer Anlagen ist eine ideale Stellstrategie wie beim Cartpole-Simulator nicht bekannt. Ebenso wenig liegt ein explorativer Datensatz mit zahlreichen unterschiedlichen Aktionen in unterschiedlichen Zuständen vor. Die Ansätze "idealer Regler" und "NFQ, Zufalls-Daten" konnten deshalb auf realen Datensätzen nicht getestet werden.

Für die Experimente auf den Anlagen war es ebenfalls nicht möglich im selben Zustand verschiedene Aktionen zu probieren und deren Auswirkungen zu studieren. Ein direkter Vergleich der Regelstrategien wie beim Cartpole-Simulator entfällt also. Stattdessen wurde die Regelstrategie des NFQ mit der Strategie des aktiven PID-Systems verglichen. Im gewählten Zeitraum liefen die Anlagen stabil, es wird also davon ausgegangen, dass die Strategie des PID-Systems korrekt war.

Je größer die Abweichung der NFQ-Strategie, desto größer ist die Gefahr einen kritischen Systemzustand zu erzeugen. Für die Auswertung wurde deshalb zum einen die mittlere Abweichung $a_{diff} = |a_{NFQ} - a_{PID}|$ bestimmt und zum anderen die prozentuale Anzahl der kritischen Aktionen a_{krit} . Eine Aktion gilt als kritisch, wenn sie um mehr als 50% des Stellbereiches von den Aktionen des PID-Systems abweicht.

Eine Auswertung der Mediane aller Kriterien erfolgte bereits in Kapitel 6. An dieser Stelle wird auf die 10%- und 90%-Quantile eingegangen.

Bei allen vier Versuchen (Rostgeschwindigkeit, Zuteilergeschwindigkeit, Ammoniakwasser und Energieeintrag) erzielt auch bei Betrachtung der 10%- und 90%-Quantile das NFQ+kDP-Verfahren wesentlich bessere Werte als ein NFQ ohne erweiterten Trainingsdatensatz. Das 10%-Quantil der Anzahl der kritischen Aktionen ist bei

Approx. Zielgröße	$Q_{0,1}(\bar{e})$	$Q_{0,5}(\bar{e})$	$Q_{0,9}(\bar{e})$
NFQ, PID-Daten	0,065	0,167	0,303
NFQ, PID-Daten + kDP	0,087	0,226	0,343
kritische Aktionen	$Q_{0,1}(a_{krit})$	$Q_{0,5}(a_{krit})$	$Q_{0,9}(a_{krit})$
NFQ, PID-Daten	12,68%	35,27%	67,40%
NFQ, PID-Daten + kDP	0%	3,10%	19,34%
Abweichung $a_{diff} =$ $ a_{NFQ} - a_{PID} $	$Q_{0,1}(\overline{a_{diff}})$	$Q_{0,5}(\overline{a_{diff}})$	$Q_{0,9}(\overline{a_{diff}})$
NFQ, PID-Daten	0,230	0,358	0,600
NFQ, PID-Daten + kDP	0,093	0,128	0,276

Tabelle B.15. – Ergebnisse der Simulation des Rostgeschwindigkeit-Reglers auf Daten von MVA 1. Aufgeführt sind der mittlere Fehler bei der Approximation der Dampf-Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen des aktiven PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 100 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 100 Simulationen.

allen vier realen Datensätzen unter 0,3%. Kombiniert man mehrere Modelle und schließt Aktionen aus, die jeweils der Mehrzahl der Modelle widersprechen, kann man mit dem NFQ+kDP-Verfahren kritische Aktionen somit gut vermeiden.

Auch bei dem 90%-Quantil der Anzahl der kritischen Aktionen ist ein deutlicher Unterschied zwischen den Verfahren mit und ohne Trainingdatensatzerweiterung erkennbar. Bei den Stellgrößen Zuteilergeschwindigkeit und Energieeintrag sind weniger als 1% der Aktionen beim NFQ+kDP-Verfahren kritisch. Ohne künstliche Datenpunkte erhöht sich dieser Wert auf 78% bzw. 61%.

Bei Auswertung des 90%-Quantils der Stellgrößen Rostgeschwindigkeit und Ammoniakwasser ergeben sich hingegen auch für das NFQ+kDP-Verfahren erhöhte Werte von 19% bzw. 15% kritischer Aktionen. Es ist bei diesen Stellgrößen also wahrscheinlich, dass trotz des NFQ+kDP-Verfahrens Modelle aktiviert werden, welche ungünstige Aktionen wählen. Aus diesem Grund wurden die in Abschnitt 6.4.1 bis 6.5 vorgestellten Verfahren zur Kontrolle und Absicherung aktiver Modelle entwickelt, die auch in solchen Fällen eine zusätzliche funktionale Sicherheit bieten.

Approx. Zielgröße	$Q_{0,1}(\bar{e})$	$Q_{0,5}(\bar{e})$	$Q_{0,9}(\bar{e})$
NFQ, PID-Daten	0,125	0,133	0,140
NFQ, PID-Daten + kDP	0,132	0,145	0,198
kritische Aktionen	$Q_{0,1}(a_{krit})$	$Q_{0,5}(a_{krit})$	$Q_{0,9}(a_{krit})$
NFQ, PID-Daten	27,30%	63,67%	78,00%
NFQ, PID-Daten + kDP	0%	0%	0,50%
Abweichung $a_{diff} = a_{NFQ} - a_{PID} $	$Q_{0,1}(\overline{a_{diff}})$	$Q_{0,5}(\overline{a_{diff}})$	$Q_{0,9}(\overline{a_{diff}})$
NFQ, PID-Daten	0,370	0,530	0,599
NFQ, PID-Daten + kDP	0,053	0,068	0,112

Tabelle B.16. – Ergebnisse der Simulation des Zuteilergeschwindigkeit-Reglers auf Daten von MVA 1. Aufgeführt sind der mittlere Fehler bei der Approximation des Rewards \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen des aktiven PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 50 Mal durchgeführt, dargestellt ist jeweils der Median des Fehlers über 50 Simulationen.

Approx. Zielgröße	$Q_{0,1}(\bar{e})$	$Q_{0,5}(\bar{e})$	$Q_{0,9}(\bar{e})$
NFQ, PID-Daten	0,359	0,368	0,381
NFQ, PID-Daten + kDP	0,366	0,387	0,550
kritische Aktionen	$Q_{0,1}(a_{krit})$	$Q_{0,5}(a_{krit})$	$Q_{0,9}(a_{krit})$
NFQ, PID-Daten	1,61%	5,72%	19,75%
NFQ, PID-Daten + kDP	0,27%	0,52%	15,91%
Abweichung $a_{diff} =$ $ a_{NFQ} - a_{PID} $	$Q_{0,1}(\bar{a}_{diff})$	$Q_{0,5}(\bar{a}_{diff})$	$Q_{0,9}(\bar{a}_{diff})$
NFQ, PID-Daten	0,136	0,209	0,279
NFQ, PID-Daten + kDP	0,058	0,068	0,260

Tabelle B.17. – Ergebnisse der Simulation des Ammoniakwasser-Reglers auf Daten von Zementwerk 6. Aufgeführt sind der mittlere Fehler bei der Approximation der *NOx*-Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den Aktionen des aktiven PID $a_{diff} = |a_{NFQ} - a_{PID}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 100 Mal durchgeführt, dargestellt ist jeweils das 10%- 50% und 90%-Quantil der Ergebnisse.

Approx. Zielgröße	$Q_{0,1}(\bar{e})$	$Q_{0,5}(\bar{e})$	$Q_{0,9}(\bar{e})$
NFQ, Anlagen-Daten	0,040	0,043	0,056
NFQ, Anlagen-Daten + kDP	0,081	0,088	0,132
kritische Aktionen	$Q_{0,1}(a_{krit})$	$Q_{0,5}(a_{krit})$	$Q_{0,9}(a_{krit})$
NFQ, Anlagen-Daten	18,73%	40,11%	61,13%
NFQ, Anlagen-Daten + kDP	0,21%	0,22%	0,71%
Abweichung $a_{diff} =$ $ a_{NFQ} - a_{real} $	$Q_{0,1}(\bar{a}_{diff})$	$Q_{0,5}(\bar{a}_{diff})$	$Q_{0,9}(\bar{a}_{diff})$
NFQ, Anlagen-Daten	0,284	0,438	0,513
NFQ, Anlagen-Daten + kDP	0,103	0,110	0,284

Tabelle B.18. – Ergebnisse der Simulation des Energieeintrags auf Daten von Zementwerk 2. Aufgeführt sind der mittlere Fehler bei der Approximation der Ofeneinlaß-Temperatur-Regelabweichung \bar{e} , die mittlere Abweichung der simulierten Aktionen des NFQ von den durchgeführten Aktionen $a_{diff} = |a_{NFQ} - a_{real}|$ und die prozentuale Anzahl an kritischen Aktionen a_{krit} . Alle Versuche wurden 50 Mal durchgeführt, dargestellt ist jeweils das 10%- 50% und 90%-Quantil der Ergebnisse.

C. Verwandte Untersuchungen an Verbrennungsprozessen

Im Fachgebiet Neuroinformatik und kognitive Robotik der TU Ilmenau wurde neben dieser Arbeit auch die Dissertation von Erik Schaffernicht [Schaffernicht, 2012] angefertigt. Da sich beide Arbeiten mit der automatischen und adaptiven Regelung von Verbrennungsprozessen beschäftigen und einige gemeinsame Publikationen [Steeger et al., 2010, Kaltenhäuser et al., 2013] im Rahmen beider Arbeiten entstanden, soll in diesem Abschnitt auf die Unterschiede und Gemeinsamkeiten beider Arbeiten eingegangen werden.

In [Schaffernicht, 2012] werden verschiedene Beiträge zu dem ganzheitlichen Konzept einer kognitiven Architektur für die intelligente Feuerungsführung untersucht. Der Fokus der Arbeit liegt auf den Bereichen der Merkmalsextraktion und Merkmalsselektion, des Reinforcement Learning und des Lernmanagement. Der Bereich der Merkmalsextraktion und -selektion nimmt dabei den größten Teil der Arbeit ein. Merkmalsselektion ist der Vorgang aus einer großen Menge an verfügbaren Informationskanälen diejenigen Kanäle zu finden, welche die meisten oder “nützlichsten” Informationen für ein bestimmtes Ziel enthalten. Merkmalsextraktion ist der Vorgang aus mehreren Informationskanälen einen neuen Informationskanal zu berechnen, der die nützlichen Informationen der anderen Kanäle in komprimierter Form darstellt.

Ein einfaches Beispiel dafür ist die Principal Component Analysis (PCA) mit welcher die komplexe Information eines Bildes in wenige Kanäle transformiert werden kann¹.

Im Rahmen von [Schaffernicht, 2012] wurden zahlreiche Untersuchungen durchgeführt um möglichst effiziente Verfahren zu finden, mit denen die nützlichen Kanäle aus einer großen Menge an Kanälen gefunden werden können. Schließlich wurde das Residual Mutual Information (RMI) Verfahren als bestes Verfahren für die Auswahl der nützlichsten Kanäle zum Training eines Modells entwickelt.

Das RMI-Verfahren wurde auch in dieser Arbeit genutzt um die Kanäle für sämtliche Netztrainings auf realen Anlagendaten auszuwählen. Das RMI-Verfahren und der gesamte Prozess der Merkmalsauswahl wird hier allerdings nur kurz in Anhang A.3 erläutert, da die wesentlichen Entwicklungen und Untersuchungen von Erik Schaffernicht vorgenommen und in [Schaffernicht, 2012] dargestellt wurden. In [Schaffernicht, 2012] werden weiterhin zahlreiche andere Verfahren der Merkmalsselektion und -transformation wie Schätzung mittels Chow-Liu-Bäumen, Verbundtransinformation, gewichtete RMI und andere vorgestellt, entwickelt und untersucht, auf die im Rahmen der vorliegenden Arbeit nicht eingegangen wird.

Der Bereich des Reinforcement Learnings wird in [Schaffernicht, 2012] wesentlich kürzer behandelt. Es werden dabei vor allem drei selbstlernende Verfahren untersucht: Gaus'sche Prozesse, Evolutionäres Lernen mit Rekurrenten Netzen (CoSyNE) und das NFQ. Alle drei Verfahren werden auf verschiedenen Simulatoren getestet. Zur Anwendung im Kraftwerk kommt letztlich vor allem das CoSyNE-Verfahren.

¹Ein Bild der Größe 100 Mal 100 Pixel mit jeweils einem Grauwert von 0-255 entspricht 10.000 Informationskanälen. Durch eine PCA, kann ein solches Bild in 10-12 PCA-Kanäle transformiert werden ohne die für den entsprechenden Prozess wichtigen Information zu verlieren.

Während der Forschung auf dem Gebiet des Reinforcement Learnings wurde ausführlich die Problematik der Reward-Dekomposition untersucht und letztlich die Ergebnisse dieser Forschung in [Kaltenhäuser et al., 2013] veröffentlicht. Die Forschung dazu wurde von den Masterstudenten Robert Kaltenhäuser und Paul Rudi Torke durchgeführt und zu gleichen Teilen von Erik Schaffernicht und mir betreut und weitergeführt. Da die Thematik die wesentlichen Punkte dieser Arbeit aber nur am Rande berührt, wird hier nur in Anhang A.4 kurz darauf eingegangen. Das CoSyNE-Verfahren, das in [Schaffernicht, 2012] zur Kontrolle von Prozessen im Kraftwerk eingesetzt wurde, kam im Rahmen dieser Arbeit nicht zum Einsatz. Das Evolutionäre Training Rekurrenter Netze erwies sich als zu komplex und fehleranfällig um es auf den hier untersuchten bereits geregelten und äußerst sicherheitskritischen Prozessen einzusetzen.

Der direkte Anwendungsbereich und der Bezug zur Anwendung stellt auch die größte Unterscheidung zwischen beiden Arbeiten dar. [Schaffernicht, 2012] untersucht viele Problemstellungen sehr systematisch an vielen unterschiedlichen Simulatoren und mit Bezug auf andere Problemstellungen kognitiver Systeme wie z.B. Gesichtserkennung oder Lokalisierung. Versuche (offline und online) mit realen Verbrennungsprozessen werden am Beispiel eines Kohlestaub-Kraftwerkes durchgeführt. Als Stellgrößen dienen dabei Luftverteilungen zwischen verschiedenen Sekundärlufteindüsungen, die bisher nicht geregelt wurden. Mithilfe der oben genannten Merkmalsselektion und des CoSyNE-Verfahrens konnte eine Stellstrategie gefunden werden, welche eine Verbesserung der Effizienz bei gleichzeitiger Minderung des Stickoxidausstoßes ermöglicht.

Anhand der Anwendung wird allerdings auch der wesentliche Unterschied beider Arbeiten deutlich: in [Schaffernicht, 2012] werden bisher unregelte Stellgrößen untersucht. Es ist möglich auf diesen Stellgrößen einen Explorationsalgorithmus laufen zu lassen ohne der Anlage wesentlich zu schaden. Sollte von dem lernfähigen

Regler eine “falsche” Stellstrategie erlernt werden, so wird diese Strategie nicht unmittelbar zu einem kritischen Schaden führen. Weiterhin existiert wenig oder kein Vorwissen, wie die Stellgröße in Abhängigkeit verschiedener Zielgrößen zu fahren ist, und es existiert kein konventionelles Regelsystem, welches die Stellgröße bereits steuert.

In der vorliegenden Arbeit geht es hingegen um eine andere Gruppe von Stellgrößen. Die in dieser Arbeit untersuchten Stellgrößen sind im Allgemeinen bereits durch ein konventionelles System geregelt. Es existiert grundlegendes Wissen darüber, wie die Stellgröße zu fahren ist. Wird die Stellgröße komplett anders bzw. “falsch“ gefahren, kann dies unmittelbar zu einem wirtschaftlichen oder technischen Schaden führen. Selbst wenn dies nicht passiert, wird das lernfähige System vom Operator deaktiviert, und wieder mit konventionellen Systemen gefahren. Während in der Arbeit von [Schaffernicht, 2012] das lernfähige Modell als einzig aktiver Regler betrachtet wird, wird in dieser Arbeit auch die Kombination von lernfähigem Modell und Regler in Form von MPC oder eines Softsensors untersucht.

Die wesentlichen Merkmale beider Arbeiten sind in Table C.1 noch einmal zusammengefasst:

	[Schaffernicht, 2012]	vorliegende Arbeit
Fokus der Arbeit	Merkmalsextraktion und -selektion, kognitive Gesamtarchitektur	Adaptive Regelung
Anlagendaten/-versuche	1 Kohlekraftwerk	3 MVA, 7 Zementwerke
Stellgrößen	Sekundärluft Verteilung	Müllzuteiler, Rostgeschwindigkeit, SNCR-Ammoniakwasser, Dampfleistung-Prognose, Freikalk-Softsensor
Beispieldaten	nicht geregelt	bereits geregelt
Sicherheitskritisch	i.A. nein	i.A. ja
weitere Anwendungsszenarien	Mountain-Car-Simulator, Gitterwelt-Simulator, Schätzung Nutzerinteresse, Schätzung Emotionen, Klassifikation MLR-Datensätze	Cartpole-Simulator

Tabelle C.1. – Darstellung der wesentlichen Unterschiede zwischen der Arbeit [Schaffernicht, 2012] und der vorliegenden Arbeit.

Literaturverzeichnis

- [Abbeel and Ng, 2005] Abbeel, P. and Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, pages 1–8.
- [Abu-Mostafa, 1994] Abu-Mostafa, Y. S. (1994). Learning from hints. *Journal of Complexity*, 10(1):165–178.
- [Alsop, 2005] Alsop, P. (2005). *The Cement Plant Operations Handbook*. Tradeship Publications, 4 edition.
- [Altenkrüger and Büttner, 1992] Altenkrüger, D. and Büttner, W. (1992). *Wissensbasierte Systeme*. Springer.
- [An, 1996] An, G. (1996). The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.*, 8(3):643–674.
- [Andrews et al., 1995] Andrews, R., Diederich, J., and Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389.
- [Auel and Kühn, 2007] Auel, W. and Kühn, M. (2007). Vorschubrost „Koch-Rosttechnologie“: Feuerungskonzept und Betriebserfahrungen.
- [Backhaus et al., 2006] Backhaus, K., Erichson, B., Plinke, W., and Weiber, R. (2006). *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*. Springer-Lehrbuch. Springer.
-

- [Barnett and Lewis, 1994] Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data*. John Wiley & Sons, 3 edition.
- [Bartlett, 1999] Bartlett, P. (1999). For valid generalization, the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems*, 9:134–140.
- [Barto et al., 1983] Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuron-like adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics*, 13:835–846.
- [Beckmann and Spiegel, 2006] Beckmann, M. and Spiegel, W. (2006). Optimierung von Abfallverbrennungsanlagen. *Optimierung der Abfallverbrennung*, pages 209–264.
- [Beierle and Kern-Isberner, 2006] Beierle, C. and Kern-Isberner, G. (2006). *Methoden wissensbasierter Systeme*. Springer, 3 edition.
- [Bellman, 1957] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- [Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1 edition.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Bozoki and Friedman, 1994] Bozoki, E. and Friedman, A. (1994). Neural networks and orbit control in accelerators. In *The Fourth European Conference For Particle Accelerators*, pages 1589–1591.
- [Breunig et al., 2000] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, pages 93–104. ACM.
-

-
- [Brouwer et al., 1996] Brouwer, J., Heap, M., Pershing, D., and Smith, P. (1996). A model for prediction of selective noncatalytic reduction of nitrogen oxides by ammonia, urea, and cyanuric acid with mixing limitations in the presence of co. In *Symposium (International) on Combustion*, volume 26, pages 2117–2124.
- [Burstrom et al., 2010] Burstrom, P., Lundstrom, S., Marjavaara, D., and Toyra, S. (2010). Cfd-modelling of selective non-catalytic reduction of nox in grate-kiln plants. *Progress in Computational Fluid Dynamics, An International Journal*, 10(5):284–291.
- [Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.
- [Dan Foresee and Hagan, 1997] Dan Foresee, F. and Hagan, M. (1997). Gauss-newton approximation to bayesian learning. In *Neural Networks, 1997., International Conference on*, volume 3, pages 1930–1935.
- [Dimitrakakis and Lagoudakis, 2008] Dimitrakakis, C. and Lagoudakis, M. G. (2008). Rollout sampling approximate policy iteration. *Machine Learning*, 72:157–171.
- [Dittmar and Pfeiffer, 2004] Dittmar, R. and Pfeiffer, B. (2004). *Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure*. Oldenbourg Wissensch.Vlg.
- [Domingos, 1999] Domingos, P. (1999). The role of occam’s razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4):409–425.
- [Dutech et al., 2005] Dutech, A. et al. (2005). Nips workshop: Reinforcement learning benchmarks and bake-offs ii. In *Advances in Neural Information Processing Systems, NIPS 2005, Vancouver, Canada*, volume 18.
-

- [Effenberger, 1999] Effenberger, H. (1999). *Dampferzeugung*. VDI-Buch. Springer.
- [Eisenbach, 2009] Eisenbach, M. (2009). Rewarddekomposition für Multiagentensysteme bei komplexen Regelungsprozessen. Master's thesis, FG Neuroinformatik und Kognitive Robotik, TU Ilmenau.
- [Elwell and Polikar, 2011] Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531.
- [Epple, 2009] Epple, B. (2009). *Simulation von Kraftwerken und wärmetechnischen Anlagen*. Springer Wien.
- [Evans et al., 2002] Evans, D., Jones, A. J., Evans, D., and Jones, A. J. (2002). A proof of the gamma test. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 458(2027):2759–2799.
- [Florian, 2007] Florian, R. V. (2007). Correct equations for the dynamics of the cart-pole system. Technical report, Center for Cognitive and Neural Studies.
- [Fortuna et al., 2007] Fortuna, L., Graziani, S., Rizzo, A., and Xibilia, M. G. (2007). *Soft sensors for monitoring and control of industrial processes*. Advances in Industrial Control. London: Springer.
- [Geibel, 2001] Geibel, P. (2001). Reinforcement learning with bounded risk. In *ICML*, pages 162–169.
- [Geibel and Wysotzki, 2005] Geibel, P. and Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.(JAIR)*, 24:81–108.
- [Geman et al., 1992] Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.
-

-
- [Geva and Sitte, 1993] Geva, S. and Sitte, J. (1993). A cartpole experiment benchmark for trainable controllers. *IEEE Control Systems Magazine*, 13:40–51.
- [Goldhan, 2012] Goldhan, J. (2012). Hauptseminar: Recherche zu Methoden zum Einbinden von Expertenwissen in automatische Lernverfahren.
- [Griebe, 2007] Griebe, S. (2007). *Modellierung der Verbrennung von Braunkohle in einer Zykloidfeuerung unter Berücksichtigung der Schadstoffemissionen CO, NOX und SO2*. Cuvillier Verlag.
- [Grossberg, 1987] Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63.
- [Görner et al., 1999] Görner, K., Klasen, T., and Kümmel, J. (1999). Numerische Berechnungen und Optimierung der MVA Bonn. *VDI-Berichte*, 1492:331–336.
- [Hagan and Menhaj, 1994] Hagan, M. T. and Menhaj, M. B. (1994). Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6):989–993.
- [Hans et al., 2008] Hans, A., Schneegaß, D., Schäfer, A. M., and Udluft, S. (2008). Safe exploration for reinforcement learning. In *ESANN 2008, 16th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23-25, 2008, Proceedings*, pages 143–148.
- [He, 2011] He, H. (2011). *Self-Adaptive Systems for Machine Intelligence*. John Wiley & Sons.
- [Heger, 1994] Heger, M. (1994). Consideration of risk in reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, volume 105, page 111. Citeseer.
- [Ho et al., 2009] Ho, K., Leung, C.-s., and Sum, J. (2009). On weight-noise-injection training. In Köppen, M., Kasabov, N., and Coghill,
-

- G., editors, *Advances in Neuro-Information Processing*, volume 5507 of *Lecture Notes in Computer Science*, pages 919–926. Springer Berlin Heidelberg.
- [Holcim, 2004] Holcim (2004). *Reference Guide for Thermal Process Technology Kiln Systems*. Holcim Group Support Ltd.
- [Holmstrom and Koistinen, 1992] Holmstrom, L. and Koistinen, P. (1992). Using additive noise in back-propagation training. *Neural Networks, IEEE Transactions on*, 3(1):24–38.
- [Horton et al., 2006] Horton, J., Linero, A., and Miller, F. (2006). Use of snr to control emissions of oxides of nitrogen from cement plants. In *Cement Industry Technical Conference, 2006. Conference Record. IEEE*, pages 316–344.
- [Hurwitz, 1964] Hurwitz, A. (1964). On the conditions under which an equation has only roots with negative real parts. *Selected Papers on Mathematical Trends in Control Theory*, pages 70–82.
- [Imboden and Koch, 2003] Imboden, D. M. and Koch, S. (2003). *Systemanalyse: Einführung in die mathematische Modellierung natürlicher Systeme*. Springer.
- [Ingimundarson and Hägglund, 2001] Ingimundarson, A. and Hägglund, T. (2001). Robust tuning procedures of dead-time compensating controllers. *Control Engineering Practice*, 9(11):1195–1208.
- [Irmer, 2011] Irmer, D. (2011). Auswirkungen verrauschter und unzuverlässiger Datensätze auf automatische Lskypeernverfahren. Master’s thesis, Technische Universität Ilmenau.
- [Jain and De Silva, 1999] Jain, L. C. and De Silva, C. (1999). *Intelligent Adaptive Control: Industrial Applications*. The CRC Press International Series on Computational Intelligence. CRC PressINC.
- [Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.
-

-
- [Kaiser et al., 1996] Kaiser, M., Dillmann, R., and Rogalla, O. (1996). Communication as the basis for learning in multi-agent systems. In *ECAI'96 Workshop on Learning in Distributed AI Systems*.
- [Kalesse, 2008] Kalesse, S. (2008). Entwicklung eines Framework zum effizienten Inferieren, Lernen und Planen in probabilistischen Modellen. Master's thesis, FG Neuroinformatik und Kognitive Robotik, TU Ilmenau.
- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.
- [Kaltenhäuser, 2012] Kaltenhäuser, R. (2012). Evolutionäre Problemlösung mit Hilfe von neuronalen Netzen. Master's thesis, FG Neuroinformatik und Kognitive Robotik, TU Ilmenau.
- [Kaltenhäuser et al., 2013] Kaltenhäuser, R., Schaffernicht, E., Steege, F.-F., and Groß, H.-M. (2013). Evolutionary computation based system decomposition with neural networks. In *Proc. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, pages 191–196.
- [Klinkenberg, 2004] Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300.
- [Knauf, 1993] Knauf, R. (1993). *Logische Programmierung und Wissensbasierte Systeme*. Verlag Shaker.
- [Knorr et al., 2000] Knorr, E. M., Ng, R. T., and Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237–253.
- [Kolter and Maloof, 2003] Kolter, J. Z. and Maloof, M. A. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. In *IEEE International Conference on Data Mining (ICDM2003)*, pages 123–130.
-

- [Kolter and Maloof, 2007] Kolter, J. Z. and Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *Journal on Machine Learning Research*, 8:2755–2790.
- [Kriegel et al., 2011] Kriegel, H.-P., Kröger, P., Schubert, E., and Zimek, A. (2011). Interpreting and unifying outlier scores. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, Mesa, Arizona, USA*, pages 13–24. SIAM Omnipress.
- [Kurd and Kelly, 2003] Kurd, Z. and Kelly, T. (2003). Safety lifecycle for developing safety critical artificial neural networks. In *Computer Safety, Reliability, and Security*, pages 77–91. Springer.
- [Lampinen et al., 1999] Lampinen, J., Litkey, P., and Hakkarainen, H. (1999). Selection of training samples for learning with hints. In *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, volume 2, pages 1438–1441.
- [Legendre, 1805] Legendre, A.-M. (1805). Nouvelles méthodes pour la détermination des orbites des comètes.
- [Leskens et al., 2010] Leskens, M., van't Veen, P., van Kessel, R., Bosgra, O., and Van den Hof, P. (2010). Improved economic operation of mswc plants with a new model based pid control strategy. In *International Symposium on Dynamics and Control of Process Systems*, pages 641–646.
- [Liano, 1996] Liano, K. (1996). Robust error measure for supervised neural network learning with outliers. *Neural Networks, IEEE Transactions on*, 7(1):246–250.
- [Liu et al., 2005] Liu, Y., Cukic, B., Fuller, E., Gururajan, S., and Yeramalla, S. (2005). Novelty detection for a neural network-based online adaptive system. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 2, pages 117–122.
-

- [Liu et al., 2004] Liu, Y., Yerramalla, S., Fuller, E., Cukic, B., and Gururajan, S. (2004). Adaptive control software: can we guarantee safety? In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, volume 2, pages 100–103. IEEE.
- [Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137.
- [Lutz and Wendt, 2002] Lutz, H. and Wendt, W. (2002). *Taschenbuch der Regelungstechnik*. Verlag Harri Deutsch, 4 edition.
- [Lyapunov, 1992] Lyapunov, A. M. (1992). The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534.
- [MacKay, 1991] MacKay, D. J. (1991). Bayesian interpolation. *Neural Computation*, 4:415–447.
- [MacKay, 2003] MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.
- [Mann et al., 2005] Mann, H., Schiffelgen, H., and Frieriep, R. (2005). *Einführung in die Regelungstechnik: Analoge und digitale Regelung, Fuzzy-Regler, Regel-Realisierung, Software*. Hanser, 10 edition.
- [Marquardt, 1963] Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441.
- [Martinetz et al., 1995] Martinetz, T., Protzel, P., Gramckow, O., and Sörgel, G. (1995). Neural Network Control for Steel Rolling Mills. In Kappen, B. and Gielen, S., editors, *Neural Networks: Artificial Intelligence and Industrial Applications*, pages 280–286, Heidelberg. Springer.
- [McCloskey and Cohen, 1989] McCloskey, M. and Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *The psychology of learning and motivation*, 24:109–165.
-

- [Michie and Chambers, 1968] Michie, D. and Chambers, R. (1968). Boxes: An experiment in adaptive control. *Machine intelligence*, 2(2):137–152.
- [Murray and Edwards, 1993] Murray, A. and Edwards, P. (1993). Synaptic weight noise during multilayer perceptron training: fault tolerance and training improvements. *Neural Networks, IEEE Transactions on*, 4(4):722–725.
- [Nelder and Mead, 1965] Nelder, J. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.
- [Niyogi et al., 1998] Niyogi, P., Girosi, F., and Poggio, T. (1998). Incorporating prior information in machine learning by creating virtual examples. *Proceedings of the IEEE*, 86(11):2196–2209.
- [Nyquist, 1932] Nyquist, H. (1932). Regeneration theory. *Bell System Technical Journal*, 11(1):126–147.
- [Papadimitriou et al., 2003] Papadimitriou, S., Kitagawa, H., Gibbons, P., and Faloutsos, C. (2003). Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326.
- [Pazis and Lagoudakis, 2009] Pazis, J. and Lagoudakis, M. G. (2009). Binary action search for learning continuous-action control policies. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 793–800, New York, NY, USA. ACM.
- [Polikar et al., 2001] Polikar, R., Upda, L., Upda, S. S., and Honavar, V. (2001). Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 31(4):497–508.
-

- [Pomerleau, 1990] Pomerleau, D. (1990). Rapidly adapting artificial neural networks for autonomous navigation. In Lippmann, R., Moody, J. E., and Touretzky, D. S., editors, *Advances in Neural Information Processing Systems 3, NIPS*, pages 429–435. Morgan Kaufmann.
- [Poulsen, 2007] Poulsen, O. (2007). Bild: Municipal solid waste during combustion in a moving grate incinerator capable of handling 15 tonnes per hour waste. public domain liscence.
- [Protzel et al., 2000] Protzel, P., Tagscherer, M., and Fazlija, N. (2000). Stabilität und Plastizität Neuronaler Netze bei Kontinuierlichem Lernen. In *Workshop SOAVE 2000 Selbstorganisation von adaptivem Verhalten*, pages 164–173.
- [Rau, 2003] Rau, M. (2003). *Nichtlineare modellbasierte prädiktive Regelung auf Basis lernfähiger Zustandsraummodelle*. PhD thesis, Technische Universität Münschen, Fachgebiet Elektrische Antriebssysteme.
- [Riedmiller, 2005a] Riedmiller, M. (2005a). Neural fitted q iteration - first experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML 2005, 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005, Proceedings*, volume 3720 of *Lecture Notes in Computer Science*, pages 317–328. Springer.
- [Riedmiller, 2005b] Riedmiller, M. (2005b). Neural reinforcement learning to swing-up and balance a real pole. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3191–3196. IEEE.
- [Riedmiller et al., 2007a] Riedmiller, M., Montemerlo, M., and Dahlkamp, H. (2007a). Learning to drive a real car in 20 minutes. In *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, pages 645–650.
- [Riedmiller et al., 2007b] Riedmiller, M., Peters, J., and Schaal, S. (2007b). Evaluation of policy gradient methods and variants on the
-

- cart-pole benchmark. In *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, pages 254–261. IEEE.
- [Rodrigueza and Pereza, 2005] Rodrigueza, M. and Pereza, D. (2005). First principles model based control. In *European Symposium on Computer-Aided Process Engineering*, pages 1285–1291.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Reviews*, 65:386–408.
- [Schaffer, 1993] Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, 13:135–143.
- [Schaffernicht, 2012] Schaffernicht, E. (2012). *Lernbeiträge im Rahmen einer kognitiven Architektur für die intelligente Prozessführung*. PhD thesis, Technische Universität Ilmenau, Fachgebiet Neuroinformatik und kognitive Robotik.
- [Schaffernicht et al., 2009a] Schaffernicht, E., Möller, C., Debes, K., and Gross, H.-M. (2009a). Forward feature selection using residual mutual information. In *ESANN*, pages 583–588.
- [Schaffernicht et al., 2009b] Schaffernicht, E., Stephan, V., and Gross, H.-M. (2009b). Adaptive feature transformation for image data from non-stationary processes. In *Artificial Neural Networks–ICANN 2009*, pages 735–744. Springer.
- [Schlimmer and Granger, 1986] Schlimmer, J. C. and Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning*, 1:317–354.
- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656.
-

-
- [Si et al., 2004] Si, J., Barto, A. G., Powell, W. B., and Wunsch, D. (2004). *Handbook of Learning and Approximate Dynamic Programming*. Wiley-IEEE Press.
- [Sietsma and Dow, 1991] Sietsma, J. and Dow, R. J. (1991). Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79.
- [Sorjamaa et al., 2007] Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomput.*, 70(16-18):2861–2869.
- [Steege and Groß, 2012] Steege, F.-F. and Groß, H.-M. (2012). Comparison of long-term adaptivity for neural networks. In *22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11-14, Proceedings Part II*, volume 7553 of *Lecture Notes in Computer Science*, pages 50–57. Springer.
- [Steege et al., 2010] Steege, F.-F., Hartmann, A., Schaffernicht, E., and Groß, H.-M. (2010). Reinforcement learning based neural controllers for dynamic processes without exploration. In *20th International Conference on Artificial Neural Networks, Thessaloniki, Greece, September 15-18, Proceedings, Part II*, volume 6353 of *Lecture Notes in Computer Science*, pages 222–227. Springer.
- [Steege et al., 2012] Steege, F.-F., Stephan, V., and Groß, H.-M. (2012). Effects of noise-reduction on neural function approximation. In *Proc. 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, pages 73–78.
- [Stefánsson et al., 1997] Stefánsson, A., Končar, N., and Jones, A. J. (1997). A note on the gamma test. *Neural Computing & Applications*, 5(3):131–133.
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.
-

- [Taylor, 2006] Taylor, B. J. (2006). *Methods and procedures for the verification and validation of artificial neural networks*. Springer Publishing Company, Incorporated.
- [Thome-Kozmiensky, 1994] Thome-Kozmiensky, K. J. (1994). *Thermische Abfallbehandlung*. EF-Verlag für Energie und Umwelttechnik, 1 edition.
- [Thorburn, 1918] Thorburn, W. M. (1918). The myth of occam's razor. *Mind*, 27(107):345–353.
- [Torke, 2011] Torke, P. R. (2011). Automatische Problemdekomposition zur Zerlegung homöostatischer Regelsysteme. Master's thesis, FG Neuroinformatik und Kognitive Robotik, TU Ilmenau.
- [Torkkola, 2002] Torkkola, K. (2002). Learning feature transforms is an easier problem than feature selection. In *ICPR (2)*, pages 104–107.
- [Tsymbal, 2004] Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College: Dublin.
- [Tsymbal et al., 2008] Tsymbal, A., Pechenizkiy, M., Cunningham, P., and Puuronen, S. (2008). Dynamic integration of classifiers for handling concept drift. *Information Fusion*, 9(1):56–68.
- [Walter et al., 1990] Walter, J., Riter, H., and Schulten, K. (1990). Nonlinear prediction with self-organizing maps. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, volume 1, pages 589–594.
- [Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.
- [Welch and Bishop, 1995] Welch, G. and Bishop, G. (1995). An introduction to the kalman filter.
- [Weller, 1985] Weller, W. (1985). *Lernende Steuerungen*. VEB Verlag Technik Berlin.
-

- [Werbos, 1988] Werbos, P. (1988). Backpropagation: past and future. In *Neural Networks, 1988., IEEE International Conference on*, pages 343–353.
- [Widmer and Kubat, 1996] Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101.
- [Widrow, 1987] Widrow, B. (1987). The original adaptive neural net broom-balancer. In *1987 IEEE International Symposium on Circuits and Systems*, volume 2, pages 351–357.
- [Widrow and Smith, 1963] Widrow, B. and Smith, F. W. (1963). Pattern-recognizing control systems. In *Computer and Informations Sciences Symposium Proceedings*, pages 288–317.
- [Wolf, 2005] Wolf, C. (2005). *Erstellung eines Modells der Verbrennung von Abfall auf Rostsystemen unter Berücksichtigung der Vermischung - ein Beitrag zur Simulation von Abfallverbrennungsanlagen*. PhD thesis, Universität Duisburg-Essen.
- [Zelkowski, 1986] Zelkowski, J. (1986). *Kohleverbrennung - Brennstoff, Physik und Theorie, Technik*. VGB Kraftwerkstechnik GmbH, 1 edition.
- [Zell, 1997] Zell, A. (1997). *Simulation neuronaler Netze*. Oldenbourg, München [u.a.], 2 edition.
- [Ziegler and Nichols, 1942] Ziegler, J. and Nichols, N. (1942). Optimum settings for automatic controllers. *trans. ASME*, 64(11).
- [Zur et al., 2009] Zur, R. M., Jiang, Y., Pesce, L. L., and Drukker, K. (2009). Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. *Medical physics*, 36(10):4810–4818.
-

Symbolverzeichnis

β	Parameter einer linearen Regression, Seite 46
η	Datenpunkt bestehend aus (s, a, s', r) , Seite 118
γ	Diskontierungsfaktor, Seite 44
μ	Erwartungswert einer Zufallsvariablen/Verteilung, Seite 65
π	Policy eines Agenten/Modells, Seite 44
σ	Standardabweichung einer Verteilung/Messreihe, Seite 65
\mathbf{x}	Vektor von Ist-/Eingabewerten, Seite 42
\mathbf{y}	Vektor von Ausgabewerten, Seite 42
ξ	Rauschen, Seite 52
ζ	Gewicht der Ausgabe eines Modells aus einem Pools von Modellen, Seite 100
a	Aktion eines Agenten/Modells, Seite 44
C	Cluster aus mehreren Datenpunkten, Seite 118
dx	Abstand zweier Datenpunkte im Eingaberaum, Seite 56
dy	Abstand zweier Datenpunkte im Zielwertraum, Seite 56
E	Erwartungswert, Seite 44
e	Fehlerwert, Seite 15
$f(x)$	Funktion abhängig von der Variable x , Seite 46

- $H(X)$ Entropie einer diskreten Zufallsvariable X , Seite 180
- $I(X; Y)$ Transinformation / Mutual Information einer Variable Y zu einer Variable X , Seite 181
- K_P Verstärkungsfaktor eines PID-Reglers, Seite 15
- K_S Verstärkung einer Strecke, Seite 35
- lof local outlier factor (LOF-Verfahren), Seite 69
- lrd local reachability-distance (LOF-Verfahren), Seite 68
- mse Mean-Squared Error / mittlerer quadratischer Fehler, Seite 60
- N Approximationsmodell, Seite 86
- n_{action} Anzahl der Aktions-Bins in einem Aktionshistogramm, Seite 119
- N_{init} erstes/intiales Approximationsmodell unter mehreren Modellen die im Laufe der Adaption entstehen, Seite 86
- n_{kp} Rate künstlicher Punkte pro Cluster, Seite 119
- P Pool von Modellen N , Seite 98
- $p(x)$ Wahrscheinlichkeit ist das X die Ausprägung x hat, Seite 180
- $p(x|y)$ Bedingte Wahrscheinlichkeit von x falls y bekannt ist, Seite 180
- $Q(s, a)$ Q-Wert für eine Zustands-Aktions-Kombination, Seite 44
- $Q^\pi(s, a)$ erwarteter Q-Wert für eine Zustands-Aktions-Kombination unter der Policy π , Seite 44
- r Reward, Seite 44
- r_{kp} Rewardabschlag für den Reward künstlicher Datenpunkte, Seite 119
-

rd	reachability-distance (LOF-Verfahren), Seite 68
S	Datensatz / Menge an Zustands-Aktions-Tupeln, Seite 45
s	Zustand eines Systems, Seite 44
std	Standardabweichung einer Messreihe, Seite 53
t	Zeit, Seite 15
T_G	Ausgleichszeit einer Strecke (Verzögerung), Seite 35
T_N	Nachstellzeit eines PID-Reglers, Seite 15
T_U	Verzugszeit einer Strecke (Totzeit), Seite 35
T_V	Vorhaltzeit eines PID-Reglers, Seite 15
u	Reaktion eines Reglers, Seite 15
var	Varianz einer Verteilung/Messreihe, Seite 53
w	Sollwert, Seite 15
w_{ij}	Gewicht von Neuron j zu Neuron i in einem neuronalen Netzwerk, Seite 42
x	Ist-/Eingabewert eines Modells, Seite 15
X_{Q05}	5%-Quantil einer Menge X an Werten, Seite 60
X_{Q50}	50%-Quantil (Median) einer Menge X an Werten, Seite 60
X_{Q90}	90%-Quantil einer Menge X an Werten, Seite 60
y	Ausgabewert eines Modells, Seite 42
$y_i^{(l)}$	Ausgabe des Neurons i der Schicht l , Seite 42
$z_i^{(l)}$	Aktivierung des Neurons i der Schicht l , Seite 42
\bar{O}	Mittelwert einer Messreihe, Seite 53

Abkürzungsverzeichnis

ADP	Adaptive Dynamic Programming	48
AKK	Datenakkumulation	134
BAW	Bereichsauswahl	138
BAWk	Bereichsauswahl mit Kontrolle	140
BImSchG	Bundesimmissionsschutzgesetz	12
BRG	Bayes'sche Regularisierung	113
ES	Ensemble Selection	158
EW	Ensemble Weighting	158
EWm	Ensemble Weighting mit m Modellen	158
HMI	Human Machine Interface	266
INI	Input Noise Injection	113
LMLS	Least Mean Log Squares	112
LLM	Local Linear Model	106
LOF	Local Outlier Factor	104
LVQ	Learning Vector Quantization	63
kDP	künstliche Datenpunkte	184
MI	Mutual Information	278
MIMO	Multiple-Input-Multiple-Output	288
MLP	Multi-Layer Perceptron	63
MPC	Model Predictive Control	56
MSE	Mean Squared Error	109

MVA	Müllverbrennungsanlage	9
MRAC	Model-Referenced Adaptive Control	52
NELLM	Noise Estimation with Local Linear Models	106
NFQ	Neural Fitted Q-Iteration	63
OA	ohne Adaption	143
PID	Proportional-Integral-Differential	21
Rev.	Revision	130
RMI	Residual Mutual Information	278
SISO	Single-Input-Single-Output	22
SNCR	Selective Non-Catalytic Reduction	202
SOM	Self-Organizing Map	63
STR	Self-Tuning Regulator	52
WNI	Weight Noise Injection	113

Thesen

These 1 (PID-Regler und Nichtmonotone Prozesse (S. 28))

Ein einzelner PID-Regler kann einen Prozess, dessen Charakteristik nichtmonoton ist, nicht optimal regeln. Der PID-Regler verstärkt in bestimmten Situationen sogar die Abweichung des Prozesses vom Sollwert.

These 2 (Veränderung von Verbrennungsprozessen (S. 37))

Die Charakteristik eines industriellen Verbrennungsprozesses ändert sich mit zunehmender Zeitdauer. Durch Verschlackungen und Abnutzungen ändern sich sowohl die Wertebereiche der erhaltenen Messwerte als auch die Zusammenhänge zwischen den einzelnen Messgrößen.

These 3 (Weiterverwendung alter Reglerstruktur (S. 50))

Die komplette Ersetzung eines vorhandenen Feuerleistungssystems durch ein monolithisches lernfähiges System ist nicht durchführbar, da die vorhandenen Risiken für den Anlagenbetrieb zu groß sind. Stattdessen sollen vorhandene Reglerstrukturen um lernfähige oder adaptive Komponenten erweitert werden.

These 4 (*Rauschen und Approximationsleistung (S. 97)*)

Rauschen auf einem Trainingsdatensatz wirkt sich negativ auf die Approximationsleistung eines neuronalen Modells aus. Je größer das Rauschen ist, desto größer wird der Approximationsfehler auf einem Testdatensatz.

These 5 (*Rauschen für Validierungsdaten(S. 101)*)

Wird ein Approximationsmodell erstellt und werden zum Training des Modells Validierungs- und Trainingsdaten benötigt, so ist es vorteilhaft, verrauschte Punkte im Validierungsdatensatz zu verwenden und unverrauschte im Trainingsdatensatz.

These 6 (*Robuste Trainingsverfahren (S. 118)*)

Robuste Erweiterungen der Trainingsverfahren Neuronaler Modelle wie Bayes'sche Regularisierung und die Auswahl der Trainingsdaten nach dem NELLM-Verfahren verbessern die Approximationsleistung neuronaler Modelle auf verrauschten Daten.

These 7 (*Catastrophic Interference (S. 124)*)

Ein neuronales Modell kann sich an neue Daten anpassen, wenn ihm diese präsentiert werden. Gleichzeitig können dadurch aber bereits gelernte Zusammenhänge verloren werden. Sind alte und neue Daten widersprüchlich, kann die Approximation sogar komplett fehlschlagen.

These 8 (*Adaption einz. Mod. und Datenmenge (S. 150)*)

Je stärker die concept drift ist und je häufiger Änderungen im Prozess vorkommen, desto mehr Nutzen bringt ein Bereichsauswahl-Verfahren. Ist die concept drift hingegen schwach, die zu approximierende Funktion sehr komplex oder stehen pro Modell nur wenige Datenbeispiele zur Verfügung, dann werden bessere Resultate erzielt, wenn ein Verfahren der Datenakkumulation benutzt wird.

These 9 (*Adaption von Modell-Ensembles (S. 160)*)

Kann statt eines einzelnen Modells ein Pool von Modellen verwendet werden, so ist es vorteilhaft, nur die besten Modelle des Pools zu kombinieren. Die dadurch erstellte Approximation ist besser als die Approximation des besten Modells und besser als die linear gewichtete Kombination aller Modelle.

These 10 (*Lernen auf Nicht-Explorativen Daten (S. 182)*)

Datengetriebenes Lernen einer optimalen Stellstrategie in einem Zustands-Aktionsraum ist nicht möglich, falls keine Exploration der Stellstrategie erfolgte, für jeden Zustand s_i also stets dieselbe Aktion a_j ausgeführt wurde.

These 11 (*Ergebnisse adaptiver Mod. im Einsatz (S. 257)*)

Lernfähige adaptive Regelverfahren können erfolgreich zur Prozessführung industrieller Verbrennungsprozesse eingesetzt werden. Unter geeigneten Bedingungen erzielen sie deutlich bessere Resultate als konventionelle, nichtlernende Systeme. Ihr Parametrisierungsaufwand ist allerdings deutlich höher als bei konventionellen Systemen, weshalb ihr Einsatz nur bei komplexen Problemen sinnvoll ist.