Erstellung von Prädiktionsmodellen für die Vorhersage des subjektiven Fahrkomforts von PKW bei Einzelhindernisüberfahrten

Von der Fakultät für Maschinenbau
der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
zur Erlangung des akademischen Grades eines Doktor-Ingenieurs
genehmigte

DISSERTATION

vorgelegt von

Niklas Fischer-von Rönn
aus Hamburg

Hamburg 2014

Erstgutachter: **Univ.-Prof. Dr.-Ing. M. Meywerk**
Zweitgutachter: **Univ.-Prof. Dr.-Ing. K. Krüger**

Tag der mündlichen Prüfung: 17. Dezember 2014

Danksagung

Die vorliegende Arbeit enstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Fahrzeug- und Antriebssystemtechnik (IFAS) der Helmut-Schmidt-Universität, Universität der Bundeswehr Hamburg.

An erster Stelle danke ich Herrn Prof. Dr.-Ing. M. Meywerk für die fachliche Betreuung im Rahmen der Promotion sowie sein Vertrauen in mich und meine Arbeit. Herrn Prof. Dr.-Ing. K. Krüger möchte ich für die Übernahme des Zweitgutachtens und den daraus resultierenden organisatorischen Aufwendungen ganz herzlich danken.
Weiterhin gilt mein Dank allen ehemaligen Kollegen vom IFAS, wobei ich die große Unterstützung von Herrn Dipl.-Ing. (FH) Armin Penske bei der Durchführung der Messkampagnen im Rahmen des Projektes *OfeS*, sowie den intensiven und äußerst fruchtbaren fachlichen Austausch mit Herrn Dr.-Ing. W. Tomaske, Herrn Dr.-Ing. B. Fuhr sowie Herrn Dipl.-Ing. (FH) A. Sonnenberg besonders hervorheben möchte.
Vor dem Hintergrund des Projektes *OfeS* danke ich ebenfalls allen damaligen Teilnehmern aus den Reihen von Audi, Magna Steyr und dem ViF.

Zu guter Letzt gilt mein ganz persönlicher Dank meinen Eltern und meinem Bruder sowie meiner Frau Eileen und meiner Tochter Julin für die Geduld und das Verständnis sowie die Bestärkung und die Unterstützung ganz besonders während der Fertigstellung der Arbeit und der abschließenden Vorbereitungen.

Hamburg, im Dezember 2014

Niklas Fischer-von Rönn

Inhaltsverzeichnis

1 Einleitung 1
- 1.1 Problemstellung . . . 1
- 1.2 Stand der Forschung und Technik . . . 2
- 1.3 Zielsetzung . . . 4
- 1.4 Gliederung der Arbeit . . . 4

2 Erhebung der Datenbasis 7
- 2.1 Fahrversuche . . . 7
  - 2.1.1 Messfahrzeug . . . 7
  - 2.1.2 Hindernisse . . . 8
  - 2.1.3 Komfortbewertung . . . 8
  - 2.1.4 Durchführung . . . 9
- 2.2 Erfassung und Verarbeitung der Messdaten . . . 9

3 Transformation der Messdaten 15
- 3.1 Zusammenhang von Skala und Frequenz . . . 21
- 3.2 Wahl des Transformations-Parameters γ . . . 23
- 3.3 Einflussbereiche zeitlich getrennter Ereignisse . . . 27
- 3.4 Numerische Berechnung der Transformation . . . 34

4 Extrahierung komfortrelevanter Merkmale 41
- 4.1 Tableau-Separierung . . . 41
- 4.2 Verfahren der Tableau-Approximation . . . 46

5 Prädiktionsmodelle 53
- 5.1 Unterteilung und Beispiele . . . 54
- 5.2 Grundbegriffe . . . 54
  - 5.2.1 Training . . . 55
  - 5.2.2 Überanpassung . . . 56
  - 5.2.3 Data-Division . . . 58
  - 5.2.4 Regularisierung . . . 60
  - 5.2.5 K-Kreuzvalidierung . . . 62
- 5.3 Bewertung der Modellqualität . . . 62
- 5.4 Lineare Regressionsmodelle . . . 64
  - 5.4.1 Standardisierung . . . 67

5.4.2 Regularisierte lineare Regression 69
5.4.3 Kombination linearer Regressionsmodelle 73
5.5 Künstliche Neuronale Netze 76
5.5.1 Feed-Forward-Struktur 77
5.5.2 Backpropagation und gradientenbasierte Trainingsverfahren 82
5.5.3 Bayessche Regularisierung 85
5.5.4 Paralleles und sequentielles Lernen 87
5.5.5 Selbstorganisierende Karten nach Kohonen 88
5.6 Zusammenfassung 94

6 Auswahl von Merkmalen 97
6.1 Begriffe und Einordnung der Methoden 97
6.2 Suchmethoden der Merkmalsauswahl 99
6.3 Bewertung von Merkmalsauswahlen 100
6.4 Der Feature-Browser von OfeS-Tool 101
6.4.1 Merkmalsauswahl mit dem Feature-Browser 103
6.4.2 Bewertung der Merkmalskombinationen mit Kohonen-Netzen . . . 108
6.4.3 Modellerstellung mit künstlichen neuronalen Netzen 109

7 Merkmalsauswahl und Modellerstellung mit OTFS 119
7.1 Das OTFS-Verfahren 119
7.1.1 Kombination der einzelnen Modelle 120
7.1.2 Such-Strategie 120
7.1.3 Bewertung von Auswahlen 121
7.2 Ermittlung und Bewertung von Merkmalskombinationen 121
7.2.1 Erstellung von Vorhersagemodellen 129
7.3 Analyse der linearen OTFS-Modelle 136

8 Zusammenfassung und Ausblick 143

A Anhang 147
A.1 Ergebnisse der SOM-Bewertung 147
A.1.1 Feature-Browser-Auswahl 147
A.1.2 OTFS-Auswahl 151
A.2 Regressorlisten der linearen OTFS-Modelle 155

Literaturverzeichnis 167

1 Einleitung

1.1 Problemstellung

Der Schwingungskomfort von PKW ist fester Bestandteil des täglichen Fahrerlebnisses und stellt somit ein wichtiges Kriterium beim Fahrzeugkauf dar. Es liegt also im Interesse der Fahrzeughersteller, komfortmindernde Einflüsse so weit wie möglich zu unterdrücken. Im Fall des Schwingungskomforts werden diese Komfortminderungen ausgehend von Straßenanregungen über das Fahrwerk in das Fahrzeug eingeleitet. Aus diesem Grunde ist es wünschenswert, bereits zu einem frühen Zeitpunkt des Entwicklungsprozesses Aussagen über den Fahrkomfort treffen zu können. Da zu dieser Zeit jedoch noch keine fahrbaren Prototypen des Fahrzeugs existieren, können diese Aussagen nicht direkt, wie sonst üblich, durch Testfahrer getroffen werden.
Um Komfortbeurteilungen auch ohne Testfahrer durchführen zu können, bedarf es somit Methoden, die die Vorhersage subjektiver Komforteindrücke aus simulierten oder gemessenen, objektiven Daten ermöglichen. Diese Vorhersagemöglichkeit ist daher Gegenstand zahlreicher Publikationen. So werden viele dieser Veröffentlichungen in längsdynamische [Alb05],[Sch00], querdynamische [Rie00],[Jür96] und vertikaldynamische [Haz94],[Rer86] Betrachtungen aufgeteilt.
Zum Teil kommen dabei spezielle Fahrmanöver zum Einsatz. Außerdem werden bestimmte unebene Fahrbahnen befahren, um den Schwingungskomfort von Fahrbahnerregungen beurteilen zu können.
Das Vorgehen vieler Arbeiten gleicht sich vom Prinzip her. Es werden dabei von Fahrern abgegebene Komfortnoten in Beziehung gesetzt zu bestimmten Kenngrößen, die aus Messsignalen extrahiert werden. Bei stochastischen Anregungen durch die Fahrbahn kommen dabei häufig Kenngrößen zum Tragen, die aus Leistungsdichte-Spektren bestimmter Beschleunigungssignale gewonnen werden [Kos02],[Mit04],[Sta]. Auch andere Kenngrößen werden eingesetzt [Cuc93],[Sta09].
Der Einsatz der Fourier-Transformation eignet sich bei Einzelhindernissen nur bedingt, da die zeitliche Abfolge der Beschleunigungsänderungen durch die Fourier-Transformation nicht erfasst werden kann. Daher wird im Rahmen dieser Arbeit ein bis heute wenig untersuchter Ansatz [Lee00],[Lee01][Mis09],[GW09] vorgestellt, bei dem entsprechende Beschleunigungssignale wavelet-transformiert werden.
Die Wavelet-Transformation löst den Frequenzinhalt der zu transformierenden Signale zeitlich auf, so dass zu bestimmten Zeitintervallen Aussagen über die Frequenzinhalte getroffen werden können. Im Gegensatz zur Fourier-Transformation erlaubt die Wavelet-Transformation zum Beispiel eine Unterscheidung zwischen Anregungen, die durch das

Überfahren der Vorderachse und durch das Überfahren der Hinterachse von Einzelhindernissen entstehen.
Die Daten für die Untersuchungen wurden im Rahmen des Forschungsprojektes *Optimierung des fahrbahnerregten Schwingungskomforts* (OfeS) in Zusammenarbeit mit der Audi AG, Magna Steyr Fahrzeugtechnik und dem Kompetenzzentrum Virtuelles Fahrzeug (ViF) der TU-Graz erhoben [Fis10],[Mey10],[FvR13]. Als Versuchsfahrzeug diente hierbei ein Audi A4 TFSI (Bj. 2008). Wesentliche Teile der erhobenen objektiven und subjektiven Daten obliegen daher der Geheimhaltung. Entsprechend werden die betroffenen Ergebnisse in der gesamten Arbeit nicht absolut sondern nur bezogen auf eine feste Basiskonfiguration des Fahrzeugs oder gar nicht angegeben.
Das Projekt OfeS wurde gefördert durch das „COMET K2 Forschungsförderungs-Programm “des Österreichischen Bundesministeriums für Verkehr, Innovation und Technologie, des Österreichischen Bundesministeriums für Wirtschaft, Familie und Jugend, der Österreichischen Forschungsförderungsgesellschaft mbH, des Landes Steiermark sowie der Steirischen Wirtschaftsförderung.

1.2 Stand der Forschung und Technik

Die subjektive Bewertung des Fahrkomforts wird im konkreten Fall über vier Ankerpunkte charakterisiert. Nach [Hei02] lauten diese

- Fahrbahn
- Fahrzeugzustand
- Bewertungskriterien
- Notenskala

Der Fahrzeugzustand kann Beladung, Bereifung sowie die Anzahl der Insassen berücksichtigen. Für die Vergleichbarkeit und Aussagekraft der Komfortmessungen ist es von höchster Wichtigkeit, dass der Fahrzeugzustand genauestens dokumentiert wird.
Als Notenskala für die subjektive Fahrkomfortbewertung hat sich der Notenbereich von eins (geringster Komfort) bis zehn (höchster Komfort) nach dem Bewertungssystem von Aigner [Aig82] durchsetzen können (vgl. Abbildung 1.1), wobei die Bedeutung der einzelnen Notenstufen je nach Fahrzeughersteller oder Zulieferer individuell variiert. Auch die Bewertungskriterien werden häufig von den Herstellern intern festgelegt. Wichtig bei ihrer Definition ist in erster Linie eine klare Trennbarkeit, da in der Regel innerhalb einer Messfahrt mehrere Kriterien parallel bewertet werden müssen. Einen umfangreichen Katalog möglicher Bewertungskriterien findet man ebenfalls in dem Werk von Heissing und Brandl [Hei02].
Für die Spezifizierung der Anregung durch die Fahrbahn stehen drei Klassen zur Verfügung.

	nicht annehmbar				Grenz-fall	annehmbar				
Bewertungs-Index	1	2	3	4	5	6	7	8	9	10
Bewertung des Verhaltens der Aggregate	Produktions-Verwertung				Grenz-fall	gerade noch annehmb.	aus-reichend	gut	sehr gut	ausge-zeichnet
	schlecht			Kunden-beschw.						
Festgestellt von	allen Kunden	Durchschnittskunden			kritischen Kunden			ausgebildeten Beobachtern		nicht wahr-nehmbar
Geräusche Vibrationen Rauheit Schütteln	nicht annehmbar			unange-nehm	Verbesse-rung er-forderlich	mäßig	leicht	sehr leicht	Spuren	keine
Achsen-bewertung	E	D			C			B		A

Abbildung 1.1: Bewertungssystem mit einer zehn-stelligen Skala (nach [Aig82])

Der allgemeinste und am meisten untersuchte Fall ist dabei die Klasse der stochastischen Anregungen. Hier sind Fahrbahnen enthalten, die im Wesentlichen eben sind, aber stochastisch verteilte Defekt aufweisen. Für diese Art der Anregung schlägt die ISO-Norm 8608 [ISO95] die Darstellung über das Leistungsdichte-Spektrum der Straße vor. Ausgehend hiervon können durch Berechnung und Simulation (Fahrzeugmodelle), Prüfstandsversuche (4-Poster, Fahrsimulatoren) oder Realfahrten die Leistungsdichte-Spektren der Beschleunigungen um den Fahrer herum ermittelt werden.
Gängige Verfahren bewerten Letztere mit zuvor definierten Bewertungsfunktionen [Cuc93], [ISO97]. Diese berücksichtigen die Empfindlichkeit des Menschen für die Einleitung (längs oder vertikal sowie Körperteil) und die Stärke (Frequenz und Amplitude) der Schwingung bezüglich des zugrunde liegenden Bewertungskriteriums (Stuckern, Zittern, Prellen etc.). Die Weiterverarbeitung dieser bewerteten Schwingstärken führt dann schließlich zu objektiven Komfortbewertungen.
Die weiteren relevanten Fahrbahn-Anregungsklassen sind die harmonische Anregung und die Einzelhindernisse. Harmonische Anregungen können ähnlich wie die stochastischen Unebenheiten behandelt werden, wobei die Bewertungskriterien in dem Fall in der Regel den Bereich der tieffrequenten Phänomene wie die Aufbaufederung abdecken.
Gänzlich anders verhält es sich bei der Bewertung von Einzelhindernis-erregten Fahrzeugschwingungen. Im Gegensatz zu den stochastischen und harmonischen Anregungen sind solche Schwingungen instationär und, in der Regel, von sehr kurzer Dauer. Sie lassen sich somit nicht auf vernünftige Weise durch eine rein spektrale Darstellung repräsentieren, wodurch die Anwendung der zuvor genannten Methoden hier nicht möglich ist. Stattdessen wird häufig in das duale Extrem der reinen zeitlichen Darstellung gewechselt [Gri96], [JÖ5]. Hier werden Kennzahlen wie der VDV (eng.: vibration dose value), der RMS-Wert (eng.: root mean square) und der MTVV (eng.: maximum transient vibration value) aus den

geeignet gefilterten zeitlichen Beschleunigungsverläufen ermittelt, die mit der subjektiven Komfortbewertung korrelieren [ISO97]. Methoden, die auf einer simultanen Analyse in Zeit und Frequenz beruhen, sind in diesem Bereich noch nicht etabliert.
Zur Erhebung der objektiven und subjektiven Daten für Untersuchungen des Schwingungskomforts kommen sowohl Realversuche mit Serienfahrzeugen oder Prototypen [Sta09] als auch spezielle Simulatoren zum Einsatz [Ada07].

1.3 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung einer Methode zur Bewertung des Schwingungskomforts von PWK bei Einzelhindernisüberfahrten unter besonderer Berücksichtigung des Zeit-Frequenz-Verhaltens der komfortrelevanten Fahrzeugschwingungen. Letzteres soll unter Verwendung der kontinuierlichen komplexen Wavelet-Transformation gemessener Beschleunigungssignale analysiert und durch eine speziell für diese Untersuchungen zu entwickelnde Methode der Merkmalsextrahierung auf komfortrelevante Kennzahlen abgebildet werden. Aus diesen sind schließlich durch geeignete Vorhersagemodelle Approximationen der Komfortnoten abzuleiten.
Grundlage aller Untersuchungen sind Daten, die im Rahmen mehrerer Kampagnen von Komfortmessungen auf dem Audi-Testgelände bei Neustadt an der Donau erhoben wurden. Sie umfassen die objektiven Beschleunigungsdaten an ausgewählten Positionen eines Audi A4 Avant, der durch gezielte Manipulationen in verschiedenen Varianten vorlag, und die subjektiven Komfortbewertungen eines geschulten Testfahrers, dessen Bewertungsverhalten zu approximieren anvisiert wurde.
Die Eignung des Verfahrens für den spezifischen Fall sollte sich auf Fahrzeuge übertragen lassen, die in den Grundeigenschaften (Fahrzeugklasse, Radstand, Antriebstrang etc.) nicht wesentlich von den Varianten des Testfahrzeugs abweichen.

1.4 Gliederung der Arbeit

Der Weg von der Zielsetzung über die Entwicklung, und Implementierung bis zur Erprobung des gesamten Verfahrens wird in klar voneinander getrennte Entwicklungsstufen unterteilt.
Im ersten Schritt wird die Erhebung der Daten sowie deren unmittelbare Weiterverarbeitung und Organisation beschrieben. Als Ergebnis dieses Abschnitts entsteht eine Datenbasis mit objektiven, den Fahrzustand beschreibenden, Daten und zugehörigen subjektiven Daten in Form von Komfortbewertungen bezüglich unterschiedlicher Kriterien. Diese Datenbasis repräsentiert den Ausgangspunkt der darauf aufsetzenden Entwicklung des Komfortbewertungsverfahrens und setzt gleichzeitig den Rahmen des Gültigkeitsbereiches der Methode fest.
In dem darauf folgenden Schritt wird eine geeignete Transformation der objektiven Daten hergeleitet, die den Anforderungen der simultanen Zeit-Frequenz-Analyse der Zielsetzung

genügt und als Ergebnis eine Darstellung der objektiven Messdaten in Form später genauer spezifizierter Wavelet-Tableaus liefert.
Im Anschluss daran wird zunächst für die Wavelet-Tableaus eine Auftrennung der Messignale in die Bereiche Vorderachse und Hinterachse vorgestellt, die aufgrund der verwendeten Bewertungskriterien geschehen muss. Dann wird für die jeweiligen Teil-Tableaus eine Methode zur Abbildung der Daten auf komfortrelevante Merkmale entwickelt, die die Auflösung der Beschleunigungen in Zeit und Frequenz nutzt. Durch geometrische Approximationen der Wavelet-Tableaus werden dabei charakteristische Eigenschaften auf beschreibende Merkmale abgebildet, die als Ergebnis dieser Entwicklungsstufe die endgültige Repräsentation der objektiven Daten darstellen. Der Teil der Analyse und Verarbeitung der objektiven Daten endet mit diesem Meilenstein.
Der nächste Entwicklungsschritt befasst sich mit der Abbildung der Merkmale auf Komfortnoten, die die Approximation der subjektiven Komfortnoten als Ziel hat. Hierfür werden Vohersagemodelle sowie Trainingsmethoden vorgestellt und angepasst. Als Ergebnis werden zwei Typen von Regressionsmodellen identifiziert, die für die Vorhersage geeignet sind. Ebenso werden die zugehörigen Trainingsmethoden und Maßnahmen zur Minderung des Risikos einer Überanpassung ermittelt.
Mit der Einschränkung der Wahl der Vorhersagemodelle kann dann in der vorletzten Entwicklungsstufe die Merkmalsauswahl vorgenommen werden. Hierzu werden nach einer Einführung in die grundlegende Problematik die angewandten Methoden vorgestellt. Unter Zuhilfenahme der Ergebnisse der vorherigen Entwicklungsstufen werden mit der Methode des Feature-Browsers schließlich Merkmalsauswahlen ermittelt und unter Verwendung künstlicher neuronaler Netze und bayes'scher Regularisierung zur Vorhersage des achsspezifischen Fahrkomforts für drei verschiedene Einzelhindernisse herangezogen. Mit diesen Ergebnissen endet ein Zweig des Verfahrens.
Im Rahmen des Projekts OfeS (vgl. Abschnitt 1.1) wurde von dem Autor in Matlab die Berechnungs-Software *OfeS-Tool* erstellt. Sie deckt mit ihrer Funktionalität alle Schritte dieses ersten Zweiges ab und bietet dabei eine komfortable grafische Benutzeroberfläche.
Der zweite Zweig stellt unter Anderem eine Alternative zu der Verwendung künstliche neuronaler Netze dar und wird in Abschnitt 7 vorgestellt. Er nutzt die Ergebnisse vorheriger Abschnitte für die Entwicklung und Implementierung des eingebetteten Merkmalsauswahl-Verfahrens *OTFS* (OfeS-Tool-Feature-Selection). Es kann als Alternative zu dem Feature-Browser für die Wahl geeigneter Merkmalsauswahlen alleine herangezogen werden, stellt aber gleichzeitig bereits eine Komfortvorhersage auf Basis dieser Auswahl und unter Verwendung linearer Regressionsmodelle dar.
Die Methode wird zum Vergleich mit dem ersten Zweig zur Modellerstellung angewandt. Eine Abwägung beider Zweige in Hinblick auf Güte der Vorhersage und Transparenz der internen Struktur beschließt die Entwicklung des Verfahrens und diese Arbeit.
Für jede der dargestellten Entwicklungsstufen werden zum umfassenden Verständnis theoretische Grundlagen benötigt, die sich auf natürliche Weise fundamental voneinander unterscheiden. Es wurde daher darauf verzichtet, für deren Vermittlung einen allgemeinen Abschnitt vorzusehen. Stattdessen werden zu Beginn jeder Entwicklungsstufe die im

Kontext der spezifischen Problemstellung relevanten Grundlagen vorgestellt, erläutert und nahtlos für die Entwicklung der maßgeblichen Funktionalität des Verfahrens herangezogen.

2 Erhebung der Datenbasis

Die Basis aller Untersuchungen im Rahmen dieser Arbeit bilden Daten, die über Fahrversuche in Kooperation mit den Projektpartnern erhoben wurden. Die Fahrversuche selber bilden den ersten Teil dieses Kapitels. Im zweiten Teil werden grundlegende Verarbeitungsschritte beschrieben, die unmittelbar auf den erhobenen Daten aufsetzen.

2.1 Fahrversuche

Die Fahrten für die Komfortmessungen wurden in insgesamt drei Kampagnen im November 2009, im Juli 2010 und im Juni 2011 durchgeführt. Der Testfahrer, dessen Komfortbewertungsverhalten im Rahmen dieser Arbeit abgebildet werden soll, nahm an sämtlichen Kampagnen teil und stellt eine wesentliche Komponente der Versuchsreihen dar. Die weiteren Komponenten sind zum Einen das Fahrzeug und die Hindernisse, zum Anderen die Festlegung der subjektiven Komfortbewertung.

2.1.1 Messfahrzeug

Das untersuchte Testfahrzeug war ein Audi A4 Avant (Bj. 2008) mit einem 1,8l TFSI Frontantrieb und einem Radstand von 2,81m. Die Abmessung der Serienbereifung war 200/50 R 17, der Druck 2,0 bar.
Um Varianzen in der Komfortbeurteilung zu erzwingen, wurden, ausgehend von dieser Basisvariante, Manipulationen an relevanten Positionen an dem Fahrzeug vorgenommen. Hierzu gehörten

- Motorlager
- Getriebelager
- Führungslenkerlager
- Hilfsrahmenlager
- Bereifung und Luftdruck (Vorderachse)
- Bereifung und Luftdruck (Hinterachse)

Durch verschiedene Kombinationen wurden in der ersten und zweiten Kampagne jeweils zehn, in der dritten sogar 20 Fahrzeug-Varianten erstellt. Somit konnten über die drei Messkampagnen insgesamt 40 verschiedene Variationen des Basisfahrzeugs vermessen werden.

2.1.2 Hindernisse

Für die Durchführung der Fahrversuche wurde von Audi das Testgelände bei Neustadt an der Donau zur Verfügung gestellt. Es bietet Fahrbahnabschnitte verschiedenster Anregungsniveaus. Darunter befindet sich mit der schnellen Zustandsstrecke auch ein spezieller Abschnitt mit den Einzelhindernissen, die für die Fahrversuche herangezogen wurden.
Die Einzelhindernisse lassen sich in die Klassen negative Kante, positive Kante und U-Rille einstufen. Von den Kanten existieren jeweils sechs Ausprägungen, von denen wiederum drei jeweils gleich dimensioniert sind. Die U-Rillen sind fünfmal in der gleichen Dimensionierung vertreten. Die durchgeführten Versuche beschränken sich auf die Hindernisse

- mittlere positive Kante ($mposK$)
- mittlere negative Kante ($mnegK$)
- U-Rille (UR)

Die Kanten haben jeweils eine Höhe von 20mm. Bei den U-Rillen handelt es sich um in die Fahrbahn eingelassene U-Profile. Alle Hindernistypen sind um 90° gegen die Fahrtrichtung ausgerichtet, so dass bei der Überfahrt beide Fahrzeug-Spuren gleich angeregt werden.

2.1.3 Komfortbewertung

Die Bewertung des Fahrkomforts erfolgte in Noten auf der üblichen Skala von eins für den geringsten und zehn für den maximal erreichbaren Komfort (vgl. Abschnitt 1.2). Bei der Quantisierung der Noten wurden Teilnoten bis zu dem Achtel einer Note zugelassen, so dass die Bewertung insgesamt 73 verschiedene diskrete Notenstufen annehmen kann.
Der in dieser Arbeit betrachtete Begriff des Fahrkomforts vor dem Hintergrund der Einwirkung mechanischer und akustischer Schwingungen auf den Fahrer entspricht dem Komplement des häufig anzutreffenden Begriffs des Diskomforts in Folge dieser Schwingungen [Gri96],[ISO97]. Hertzberg (1958) hatte diesen Zugang vorgeschlagen, der sich daraufhin gerade bei der Sitzentwicklung durchsetzen konnte [Bub00]. Bei dieser Herangehensweise existieren keine direkten Einflüsse, die den Komfort erhöhen, sondern nur solche, die im Allgemeinen eine Minderung des Komforts zur Folge haben und nach Möglichkeit unterdrückt werden sollten. Der Zustand maximalen Komforts entspricht damit einem Fall, in dem der Fahrer vollständig entkoppelt ist von den Fahrzeugschwingungen und somit keine komfortrelevanten Einflüsse seitens der Straße oder des Fahrzeugs auf den Fahrer existieren.
Bewertet wurden, neben dem Gesamteindruck der Variante, zum Teil auch Komforteindrücke von Teilsystemen wie Sitz in x- und z-Richtung, Lenkrad, Fußraum und Akustik. Jede Bewertung wurde dabei jeweils differenziert nach Vorder- und Hinterachse abgegeben.
Der Testfahrer wurde instruiert, den Komfort auf Grundlage von Schwingungen und Akustik zu bewerten. Er ist ferner speziell dafür ausgebildet, Empfindungen und Phänomene auszublenden, die in der Datenerfassung keine Berücksichtigung finden. Daher wird von der

in den Zeitverläufen der aufgezeichneten Messkanäle enthaltenen Information angenommen, dass sie eine Vorhersage des Fahrkomforts gestattet.
Da die Bewertungen der Teilsysteme aus organisatorischen Gründen nicht durchgängig über alle Messkampagnen abgegeben werden konnten, wurden für die folgenden Untersuchungen die Bewertungen auf den Gesamteindruck eingeschränkt. Dadurch können die Möglichkeiten des vorgestellten Verfahrens unter Berücksichtigung der maximal möglichen Anzahl verfügbarer Datensätze aufgezeigt werden. Dabei sind die wesentlichen Schritte und Maßnahmen auf einer höheren Abstraktionsebene angesiedelt, so dass die analoge Erstellung von Vorhersagemodellen für speziellere Bewertungskriterien mit Kenntnis der vorliegenden Arbeit ohne Weiteres möglich sein sollte - vorausgesetzt, es liegt hierfür eine entsprechende Datenbasis vor.

2.1.4 Durchführung

Die Versuche wurden in drei Messkampagnen von insgesamt vier Testfahrern absolviert. Aus organisatorischen Gründen konnte allerdings nur einer dieser Fahrer an allen Versuchen teilnehmen. Da für das Anlernen der Vorhersagemodelle aber eine möglichst große Datenbasis benötigt wird, wurde mit höchster Priorität die Abbildung des Bewertungsverhaltens dieses Fahrers durch ein mathematisches Verfahren anvisiert (vgl. Abschnitt 1.3).
Die Fahrer fuhren mit den verschiedenen Fahrzeugvarianten stets die gleiche Teststrecke ab, innerhalb der nacheinander die drei Hindernis-Typen aus Abschnitt 2.1.2 in jeweils dreifacher (negative und positive Kante), bzw. fünffacher (U-Rille) Ausführung auftreten. Die Einzelhindernisse wurden jeweils mit einer Fahrzeuggeschwindigkeit von 40km/h im dritten Gang überfahren. Die über den CAN-Bus aufgezeichneten Geschwindigkeitsverläufe konnten bestätigen, dass diese Vorgabe auch durchweg eingehalten wurde.
Vor den Messfahrten wurde die Strecke mit einem stets unveränderten Vergleichsfahrzeug abgefahren, gegen das bewertet werden sollte. Nach der Fahrt wurden die Noten sowie die Versuchsbedingungen in einen Bewertungsbogen eingetragen. Neben den Noten konnten ebenfalls Bemerkungen zum Komfortempfinden niedergeschrieben werden.

2.2 Erfassung und Verarbeitung der Messdaten

In den Fahrversuchen wurden zahlreiche potentiell komfortrelevante Signale aufgezeichnet. Für die Erfassung der Beschleunigungssignale wurden dabei triaxiale piezoelektrische Sensoren verwendet. Hiervon wurden drei am Lenkrad an den Positionen 9 Uhr, 12 Uhr und 3 Uhr (vgl. Abbildung 2.1a) angebracht. Weitere drei Beschleunigungsaufnehmer wurden vorne links und rechts so wie hinten links an den Sitzschienen verbaut (vgl. Abbildungen 2.1d und 2.1b). Für die Messung der Beschleunigungen im Fußraum wurde schließlich auch dort jeweils links und rechts ein entsprechender Sensor platziert (vgl. Abbildung 2.1c).
Nicht in Abbildung 2.1 dargestellt sind die zwei Beschleunigungsaufnehmer, die auf der Fahrerseite am vorderen und am hinteren Radträger angebracht wurden. Über sie wurden die vertikalen Beschleunigungen der Radträger fahrerseitig erfasst, die zwar nicht für den

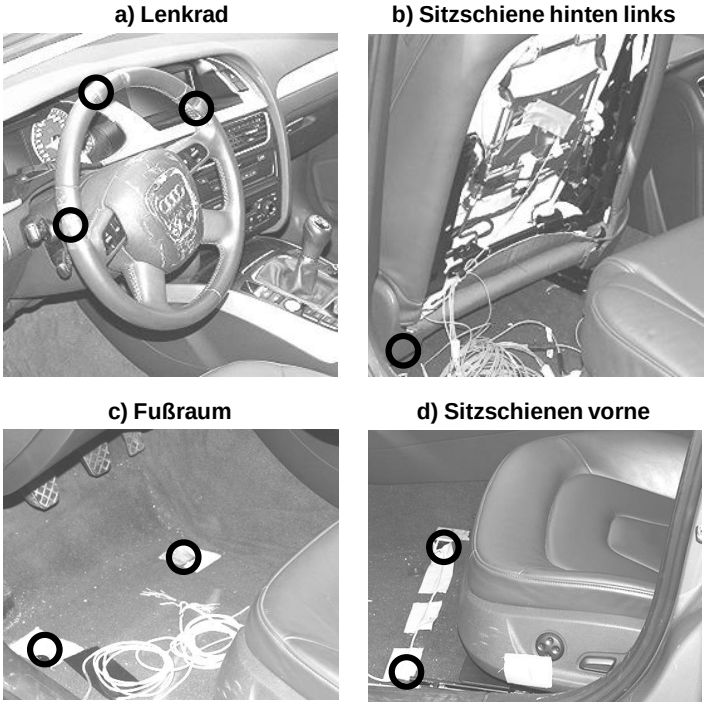

Abbildung 2.1: Positionen der triaxialen Beschleunigungsaufnehmer

Fahrkomfort herangezogen wurden, sich aber bei der zeitlichen Trennung der einzelnen Achsüberfahrten als überaus nützlich erwiesen (vgl. Abschnitt 4.1). Für die Schalldruck-Messungen kamen außerdem zwei Mikrofone zum Einsatz, die jeweils links und rechts von der Kopfstütze angebracht waren. Weitere Daten wurden über den CAN-Bus des Fahrzeugs aus dem Motor- und dem ESP-Steuergerät aufgezeichnet. Sie spielten im Wesentlichen eine unterstützende Rolle bei Interpretation und Einordnung der Daten vor dem Hintergrund der Versuchsdurchführung. Für die synchrone Aufzeichnung aller Daten wurden mehrere QuantumX Messverstärker der Firma HBM verwendet.
Im Rahmen der Voruntersuchungen wurde die Fülle der heranzuziehendene Messdaten immer weiter eingeschränkt. Eine erste Vorauswahl der komfortrelevanten Signale ist durch die nach Subsystemen differenzierten Tabellen 2.1 bis 2.5 gegeben.
Um den Umfang der anzustellenden Untersuchungen möglicher Zusammenhänge weiter zu reduzieren und die Möglichkeit ausschließen zu können, dass zufällige Korrelationen für unplausible Zusammenhänge ihren Weg in das Prädiktionsverfahren finden, wurde diese Auswahl weiter eingeschränkt auf die in Tabelle 2.6 dargestellte Auswahl komfortrelevanter Signale.
Die Algorithmen für die Verarbeitung der Beschleunigungssignale wurden für eine Abtastfrequenz von 500 Hz ausgelegt. Höher abgetastete Signale wurden mit dieser Frequenz unterabgetastet. Die spektrale Analyse der akustischen Signale hatte keine relevanten

Nr.	Signal	Einheit
1	Längsbeschleunigung Fußraum links	m/s^{-2}
2	Vertikalbeschleunigung Fußraum links	m/s^{-2}
3	Längsbeschleunigung Fußraum rechts	m/s^{-2}
4	Vertikalbeschleunigung Fußraum rechts	m/s^{-2}

Tabelle 2.1: Messkanäle zu Subsystem Fußraum

Nr.	Signal	Einheit
1	Längsbeschleunigung Lenkrad neun Uhr	m/s^{-2}
2	Vertikalbeschleunigung Lenkrad neun Uhr	m/s^{-2}
3	Längsbeschleunigung Lenkrad 12 Uhr	m/s^{-2}
4	Vertikalbeschleunigung Lenkrad 12 Uhr	m/s^{-2}
5	Längsbeschleunigung Lenkrad drei Uhr	m/s^{-2}
6	Vertikalbeschleunigung Lenkrad drei Uhr	m/s^{-2}

Tabelle 2.2: Messkanäle zu Subsystem Lenkrad

Nr.	Signal	Einheit
1	Schalldruck Fahrerohr links	Pa
2	Schalldruck Fahrerohr rechts	Pa

Tabelle 2.3: Messkanäle zu Subsystem Akustik

Nr.	Signal	Einheit
1	Vertikalbeschleunigung Sitzschiene vorne links	m/s^{-2}
2	Vertikalbeschleunigung Sitzschiene vorne rechts	m/s^{-2}
3	Vertikalbeschleunigung Sitzschiene hinten links	m/s^{-2}
4	Vertikalbeschleunigung Sitzfläche	m/s^{-2}
5	Vertikalbeschleunigung Sitzlehne	m/s^{-2}

Tabelle 2.4: Messkanäle zu Subsystem Sitz (vertikale Richtung)

Nr.	Signal	Einheit
1	Längsbeschleunigung Sitzschiene vorne links	m/s^{-2}
2	Längsbeschleunigung Sitzschiene vorne rechts	m/s^{-2}
3	Längsbeschleunigung Sitzschiene hinten links	m/s^{-2}
4	Längsbeschleunigung Sitzfläche	m/s^{-2}
5	Längsbeschleunigung Sitzlehne	m/s^{-2}

Tabelle 2.5: Messkanäle zu Subsystem Sitz (Längsrichtung)

Nr.	Signal	Einheit
1	Längsbeschleunigung Sitzschiene vorne links	m/s^{-2}
2	Vertikalbeschleunigung Sitzschiene vorne links	m/s^{-2}
3	Längsbeschleunigung Lenkrad 12 Uhr	m/s^{-2}
4	Vertikalbeschleunigung Lenkrad 12 Uhr	m/s^{-2}
5	Vertikalbeschleunigung Fußraum rechts	m/s^{-2}
6	Schalldruck Fahrerohr rechts	Pa

Tabelle 2.6: Auswahl der Messkanäle für die Bewertung des Gesamtfahrzeug-Komforts

Anteile jenseits der 200Hz Grenze ergeben, so dass auch hier eine Unterabtastung mit 500Hz berechtigt ist und verwendet wurde.
Zusätzlich wurde zur Rauschunterdrückung ein Butterworth-Tiefpass-Filter zweiter Ordnung auf die Signale angewandt. Die Grenzfrequenz des verwendeten Filters lag bei 100Hz für die Beschleunigungssignale und bei 250Hz für die Schalldruckpegel.
Unter Berücksichtigung der Frequenzbereiche, in denen bekannte Aspekte der Komfortminderung auftreten, wurde der zu untersuchende Bereich für die Beschleunigungsdaten auf das Frequenzband von 3Hz bis 60Hz und für die akustischen Daten auf das Frequenzband von 10Hz bis 240Hz festgelegt (vgl. Tabelle 2.7).
Ein weiterer wichtiger Schritt der Verarbeitung des Messdaten ist die automatisierte Ermittlung der Zeitfenster, in denen eine Einzelhindernisüberfahrt stattgefunden hat. Bei der Extrahierung relevanter Zeitabschnitte kommt es vor dem Hintergrund der Einzelhindernisse bereits an dieser Stelle im Prozess auf eine recht genaue Wahl der Grenzen an. Deswegen wurde speziell für diese Aufgabe ein Algorithmus entworfen, der aus jeweils einem größeren zeitlichen Abschnitt zu mehreren Einzelhindernisüberfahrten, beispielsweise fünf nacheinander überfahrenen U-Rillen, die entsprechende Anzahl von Zeitabschnitten ermittelt, die jeweils nur eine und insgesamt jede Fahrt beinhalten. Innerhalb dieser Einzeldaten sind die Signale annähernd gleich ausgerichtet und gut geeignet für die weitere Bearbeitung bei einzelner Betrachtung.
Da sämtliche Messdaten zeitsynchron erfasst werden, reicht es tatsächlich, nur die zeitlichen Grenzen der einzelnen Abschnitte zu ermitteln und diese auf alle Kanäle anzuwenden. Der Algorithmus benötigt eine Vorgabe der Länge T der einzelnen Abschnitte in Sekunden. Dann muss sichergestellt werden, dass die zu ermittelnden Zeitfenster auch alle für eine einzelne Überfahrt relevanten Daten enthalten. Daher werden dem Algorithmus alle

Signal Typ	Start-Frequenz	Stop-Frequenz	N_f
Beschleunigung	3Hz	60Hz	300
Schalldruck	10Hz	240Hz	500

Tabelle 2.7: Herangezogene Frequenzbänder der betrachteten Signaltypen bestehend aus N_f äquidistant verteilten Frequenzen-Werten

relevanten Messkanäle übergeben. Als weiterer Parameter wird die sogenannte relative Überlappung als reelle Zahl zwischen null und eins übergeben. Sie drückt eine Abwägung zwischen Genauigkeit der ermittelten Grenzen und dem Berechnungsaufwand aus. Schließlich muss die Anzahl n der Einzelhindernisüberfahrten in dem genannten Zeitabschnitt vorab bekannt gegeben werden. Optional kann die Fensterfunktion explizit vorgegeben werden, mit der die Daten zu den Rändern der kleineren Abschnitte hin abgeschwächt werden.

Der Algorithmus unterteilt zunächst den großen Zeitabschnitt in soviele Teilabschnitte der Länge T, wie möglich. Der vorgegebene Parameter der relativen Überlappung steuert dabei die Anzahl der zu ermittelnden Bereiche. Ist er null, so wird der gesamte Zeitbereich in disjunkte Abschnitte aufgeteilt. Für positive Werte kleiner eins wird ein jedes Zeitfenster so an das unmittelbar davor liegende angeknüpft, dass sich bezüglich beider Fenster eine Überlappung einstellt, deren Länge bezogen auf T dem genannten Parameter der relativen Überlappung entspricht.

Die durch diese Abtastung des großen Abschnitts insgesamt ermittelten Zeitbereiche werden nicht weiter verfeinert sondern beinhalten bereits die zu wählenden Zeitfenster vollständig. Daher führt ein kleiner Wert bei der relativen Überlappung zwar zu deutlich kürzeren Rechenzeiten einerseits aber andererseits auch zu deutlich ungenaueren Ergebnissen bezüglich der zeitlichen Grenzen und der relativen Ausrichtung der Teilsignale innerhalb dieser Grenzen. Bei Werten über 0,9 hingegen steigt die Anzahl der insgesamt ermittelten Fenster extrem an und damit auch die Rechenzeit und der Speicherbedarf.

Die Probleme einer großen Anzahl potentiell geeigneter Fenster entstehen vor allem dadurch, dass jedes dieser Fenster überprüft und bewertet wird, um aus allen Bereichen nur die n relevantesten auszuwählen. Bei dieser Überprüfung wird jeweils die Energie der additiv überlagerten Signale innerhalb der gegebenen Grenzen ermittelt. Damit die Teilsignale innerhalb der extrahierten Zeitfenster einheitlich und bezüglich der Inhalte möglichst zentriert ausgerichtet sind, werden sie für die Ermittlung der Signalenergie mit einer symmetrischen Fensterfunktion gewichtet, die zu den Rändern hin abfällt. Wie oben bereits erwähnt, kann dieses explizit vorgegeben werden. Das standardmäßig verwendete Kaiser-Fenster [Kai80] erwies sich allerdings als universell geeignet und wurde durchweg verwendet.

Im Anschluss werden die untersuchten Bereiche entsprechend ihres ermittelten Energieniveaus absteigend in einer Liste angeordnet. Dann werden alle Bereiche aus dieser Liste entfernt, die mit einem Zeitfenster überlappt sind, dass weiter oben in der Liste eingestuft ist, also über eine höhere Signalenergie verfügt. Durch diesen Schritt wird verhindert, dass in der letzten Auswahl mehrere Fenster zu dem gleichen Ereignis enthalten sind.

Schließlich werden aus der verbleibenden Aufstellung die ersten n Zeitfenster ausgewählt, in die zeitlich korrekte Reihenfolge gebracht und zurückgegeben. Abbildung 2.2 zeigt die Anwendung des Algorithmus auf ein Messsignal mit fünf Einzelhindernissen. Die ermittelten Zeitfenster sind durch die gestrichelten Rahmen gekennzeichnet.

Nach der Verarbeitung der Messdaten einer Fahrt inklusive der Fensterung stehen die Zeitschriebe aller Kanäle für jedes Hindernis einzeln für die weitere Verarbeitung zur

Verfügung. Zusammen mit den zugehörigen Komfortnoten bilden sie die elementaren Datensätze zu den einzelnen Versuchsfahrten. Sämtliche in diesem Kapitel vorgestellten Verarbeitungsschritte stehen gesammelt in OfeS-Tool (vgl. Abschnitt 1.4) zur Verfügung. Dabei werden die einzelnen Zwischenstufen der Datensätze stets mit gespeichert.

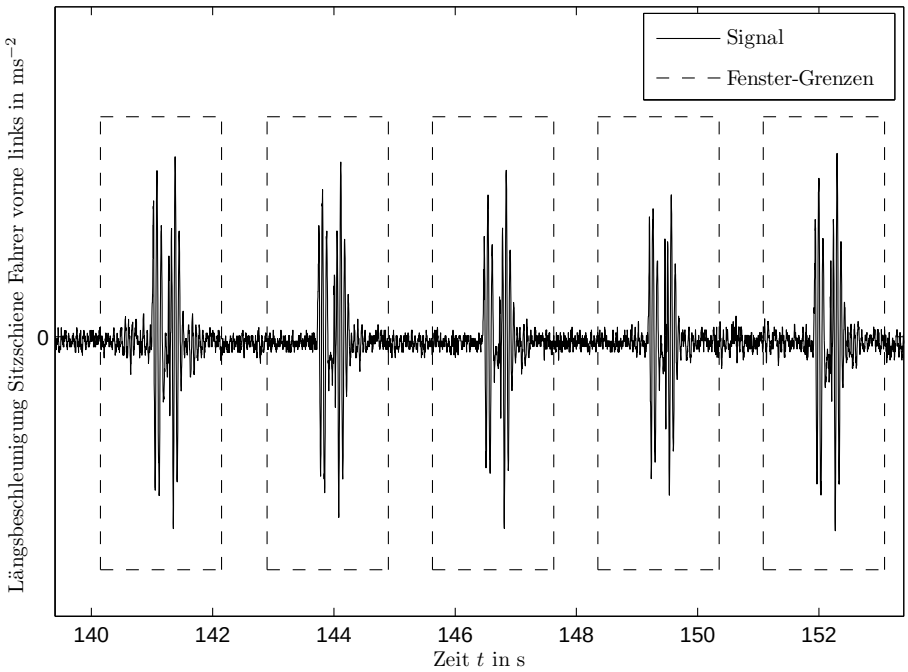

Abbildung 2.2: Anwendung des Fensterungs-Algorithmus auf ein Beschleunigungssignal mit fünf aufeinander folgenden Einzelhindernissen

3 Transformation der Messdaten

In diesem Kapitel wird die Datentransformation vorgestellt, die im weiteren Verlauf auf die entsprechend Kapitel 2 aufgezeichneten und nachbearbeiteten Daten angewandt wird. Seit mehreren Jahrzehnten werden bereits Korrelationen zwischen Beschleunigungssignalen von Fahrzeugkomponenten in der Reichweite des Fahrers und dem Fahrkomfort untersucht (vgl. Kapitel 1). Als Ergebnis wurden Methoden entwickelt, die eine Vorhersage des Fahrkomforts im Falle stochastischer Fahrbahnanregungen näherungsweise gestatten. In diesem Fall lässt sich die Anregung durch die unebene Fahrbahn durch Fourierreihen oder Fourier-Integrale beschreiben. Dies hat den Vorteil, dass ebenso die Beschleunigungen des Fahrzeugs durch Fourier-Integrale beschrieben werden können. Die Bestimmung des Fahrkomforts kann somit vollständig im Frequenzbereich geschehen. Grundlage hierfür liefern in der Regel bewertete Schwingstärken, die durch die Integration gewichteter Leistungsdichte-Spektren aufgezeichneter Beschleunigungssignale erhoben werden. Die Gewichtung geschieht unter Verwendung spezieller Bewertungsfunktionen [Cuc93], [Gri96], [ISO97],[Mit04].
Die genannten Methoden sind nur anwendbar, wenn die Anregung durch die Straße nichtdeterministisch ist und der Fahrer im Zuge der Bewertung die erfahrenen Eindrücke einer längeren Fahrt berücksichtigt und gewissermaßen mittelt.
Die Beschleunigungen der Karosserie, des Sitzes, des Fußraums und des Lenkrads jeweils in Längs- und Vertikalrichtung können als die Ausgänge dynamischer Systeme angesehen werden. Der Eingang dieser Systeme ist die Anregung durch die Straßenunebenheiten. Im Fall von Einzelhindernissen ist diese Anregung stark instationär, und somit sind es auch die Beschleunigungen der Fahrzeugkomponenten. Sie können daher nicht auf vernünftige Weise ausschließlich im Frequenzbereich untersucht werden. Da also in diesem Fall die Fourier-Transformation nicht angebracht ist, werden die Daten einer Zeit-Frequenz-Transformation in Form der kontinuierlichen komplexen Wavelet-Transformation unterzogen.
Bei der kontinuierlichen Wavelet-Transformation werden verschobene und getreckte oder gestauchte Kopien einer festen Wellenform, dem sogenannten Mutterwavelet [Lou97], zur Korrelation mit dem zu transformierenden Signal herangezogen. Das Mutterwavelet ist mittelwertfrei und klingt sowohl im Zeit- als auch im Frequenzbereich rasch ab (vgl. zum Beispiel Abbildung 3.1c). Dies folgt aus der sogenannten Admissibilitätsbedingung, die jedes Wavelet erfüllen muss:

$$0 < \int_{\mathbb{R}} \frac{|\hat{\varphi}(f)|^2}{|f|} df < \infty \tag{3.1}$$

In (3.1) ist $\hat{\varphi}(f)$ die Fourier-Transformation des zeitlich definierten Mutterwavelets $\varphi(t)$. Vor dem Zusammenhang physikalischer Größen wie Zeit t und Frequenz f werden diese im Folgenden als dimensionslos betrachtet, wobei zu Grunde gelegt wird, dass dies durch implizites Teilen durch die entsprechende SI-Einheit im Vorfeld geschehen ist. Somit ist die Zeit t im Rahmen dieser Arbeit also zu verstehen als Zeit geteilt durch Sekunden und die Frequenz f als Frequenz geteilt durch Hertz. Die Einheiten der weiteren relevanten physikalischen Größen können Tabelle 2.6 entnommen werden.
Die Anwendung der Transformation auf ein Eingangssignal $a(t)$ erfolgt durch die komplexe kontinuierliche Wavelet-Transformation, die durch

$$\mathcal{W}[a](s,\tau) := \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} a(t)\ \varphi^*\left(\frac{t-\tau}{s}\right) dt \tag{3.2}$$

definiert ist.
In (3.2) bezeichnet $\mathcal{W}[a](s,\tau)$ den komplexen Waveletkoeffizienten des Signals $a(t)$ zur Zeit τ auf der Skala s. $\varphi^*(t)$ bezeichnet dabei das komplex konjugierte Mutterwavelet. Der Verschiebungsparameter τ gestattet die Analyse des Signals in einem bestimmten Zeitintervall, wohingegen der Skalierungsparameter s durch Strecken oder Stauchen des Wavelets die Analyse auf verschiedenen Skalen ermöglicht.
Trotz der großen Verbreitung reellwertiger Mutterwavelets [Dau92], wird in dieser Arbeit auf komplexe Wavelets zurückgegriffen, um somit den Signalen innewohnende Phaseninformationen zu erhalten.
Im Anschluss an die Transformation werden aus den gewonnenen komplexen Waveletkoeffizienten sogenannte Wavelet-Tableaus konstruiert, die die Absolutbeträge der Koeffizienten beinhalten. Ein solches Wavelet-Tableau ist daher stets reellwertig und positiv in allen Einträgen.
Wohingegen der Kern der Fourier-Transformationen als komplexe harmonische Schwingung fest definiert ist, muss im Fall der Wavelet-Transformation ein geeignetes Mutterwavelet aus unendlich vielen möglichen Wavelets gewählt werden. Bei dieser Wahl eines für ein spezifisches Problem am besten geeigneten Wavelets muss bedacht werden, dass die bestmögliche Auflösung einer Zeit-Frequenz-Analyse auf natürliche Weise beschränkt ist [Dah95]. Daher beinhaltet eine derartige Entscheidung auch stets die Anforderung einer Zeit-Frequenz-Auflösung, die für die Analyse des zugrunde liegenden Signals geeignet ist.
Vor dem Hintergrund der Vohersage von Fahrkomfort sind die relevanten Frequenzbänder und die Abtastraten der Signale bekannt. Die natürlichste Herangehensweise für die Analyse dieser Signale bezüglich einer festen Frequenz wäre daher die Berechnung des Skalarprodukts des Signals mit einer harmonischen Schwingung eben dieser Analyse-Frequenz. Diese Idee liegt der Fourier-Transformation zugrunde und ist für eine Zeit-Frequenz-Analyse ungeeignet, da dieser Kern im Zeitbereich nicht abfällt und somit keinerlei Zeitlokalisierung aufweist.
Als Maßnahme um diesem auf natürliche Weise motivierten Weg dennoch folgen zu können,

muss ein glatter Abfall des Kerns im Zeitbereich erzwungen werden, wobei gleichzeitig der Abfall im Frequenzbereich zu einem gewissen Grad erhalten bleiben muss. Hier drängt sich als Fensterfunktion die der Normalverteilung zugrunde liegende Gauß-Funktion auf, da sie als Eigenfunktion der Fourier-Transformation sowohl im Zeit- als auch im Frequenzbereich einen glatten Abfall mit zunehmender Distanz von der Stelle des Maximums aufweist.
Die Multiplikation der Harmonischen Schwingung mit der Gaußschen Fensterfunktion führt zu einem Prototypen

$$\varphi_{f_0,\alpha}(t) := \exp\left[-\alpha^2 t^2 + i2\pi f_0 t\right] \tag{3.3}$$

für das Mutterwavelet.
Die freien Parameter f_0 und α wurden hierbei für die gezielte Beeinflussung der Analyse-Frequenz sowie der Zeit-Frequenz-Auflösung eingeführt.
In der Literatur ist die Wellenform $\varphi_{f_0,\alpha}$ aus (3.3) in ähnlicher Form sowohl als Gabor-Wavelet als auch als komplexes Morlet Wavelet zu finden [Ber05]. Tatsächlich war es D. Gabor, der als erster die Gaußsche Fensterfunktion aufgrund ihrer guten Lokalisierungseigenschaften in Zeit und Frequenz für eine Kurz-Zeit Fourier-Transformation [All77] verwendete. Hieraus entstand die sogenannte Gabor Transformation [Gab46], die als Ursprung des bekannten komplexen Wavelets angesehen werden kann, dass nach seinem Erfinder J. Morlet benannte wurde [Gou84]. In Anbetracht dieser Entwicklung wurde entschieden, den Term im Folgenden als Gabor-Wavelet zu bezeichnen, ohne damit die wertvolle Arbeit von Morlet degradieren zu wollen.
Das Gabor Wavelet wird in den Bereichen der Zeit-Frequenz-Analyse häufig verwendet und hat eine gewichtige Bedeutung in der digitalen Bildverarbeitung [Ari06],[Man92],[Zha09].
Wendet man die Fourier-Transformation auf (3.3) an, so erhält man

$$\begin{aligned}\hat{\varphi}_{f_0,\alpha}(f) :=& \int_{-\infty}^{\infty} \varphi_{f_0,\alpha}(t)\ \exp\left[-i\ 2\pi f t\right]\ dt \\ =& \sqrt{\frac{\pi}{\alpha^2}}\ \exp\left[-\frac{\pi^2}{\alpha^2}\ (f_0 - f)^2\right]\end{aligned} \tag{3.4}$$

was erneut eine Gaußsche Fensterfunktion ist.
Die Frequenz, für die $\hat{\varphi}_{f_0,\alpha}(f)$ den Maximalwert annimmt, wird im Kontext der Wavelet-Analyse auch als zentrale Frequenz f_c von $\varphi_{f_0,\alpha}$ bezeichnet [Naj]. Im vorliegenden Fall ergibt sich unmittelbar $f_c = f_0$. Die zentrale Frequenz ist hier also direkt an einen konstruktiven Parameter gebunden, und nur an diesen. Sie lässt sich somit unabhängig von anderen Entwurfskriterien direkt vorgeben.
Da $\hat{\varphi}_{f_0,\alpha}(f)$ eine Gaußsche Fensterfunktion ist, wird das Signal nicht wie bei der Fourier-Transformation bezüglich einer einzigen Frequenz analysiert sondern bezüglich eines ganzen Frequenzbandes. Für jede Wahl einer Signifikanz-Schranke S mit $0 < S < \sqrt{\frac{\pi}{\alpha^2}}$ ist die

Menge

$$I\left(S\right) = \{f \ \epsilon \ \mathbb{R} : \hat{\varphi}_{f_0,\alpha}(f) \geq S\}$$

der Frequenzen, für die $\hat{\varphi}_{f_0,\alpha}(f)$ bezüglich S signifikant von Null verschieden ist, ein kompaktes Intervall, dessen Mittelwert die zentrale Frequenz f_c ist. Die Breite des Intervalls kann als Maß für die Frequenzunschärfe herangezogen werden und hängt von der Wahl von α ab.
Zwecks Normalisierung von (3.4) wird (3.3) mit $\frac{\alpha}{\sqrt{\pi}}$ multipliziert. In Hinblick auf die Admissibilitätsbedingung (3.1) fällt auf, dass (3.3) nicht notwendigerweise mittelwertfrei ist, da

$$\hat{\varphi}_{f_0,\alpha}(0) = \exp\left[-\frac{\pi^2 f_0^2}{\alpha^2}\right].$$

Aufgrund dieser Verletzung von (3.1), kann die Wellenform (3.3) streng genommen nicht als Wavelet bezeichnet werden. Allerdings lässt sich dieses Problem durch die Einführung eines Korrektur-Terms e^{ξ} für einen noch zu bestimmenden Wert ξ beheben. Hierdurch ändert sich der Mittelwert der in der Wellenform enthaltenen harmonischen Schwingung:

$$\varphi_{f_0,\alpha}(t) := \frac{\alpha}{\sqrt{\pi}} \exp\left[-\alpha^2 t^2\right] \left(\exp\left[i \ 2\pi f_0 t\right] - \exp\left[\xi\right]\right) \tag{3.5}$$

Zur Bestimmung des Wertes ξ wird die Fourier-Transformation von (3.5) bestimmt:

$$\hat{\varphi}_{f_0,\alpha}(f) = \exp\left[-\frac{\pi^2}{\alpha^2} \ (f_0 - f)^2\right] - \exp\left[\xi - \frac{\pi^2 f^2}{\alpha^2}\right] \tag{3.6}$$

Man erhält für $f = 0$ durch Exponentenvergleich

$$\xi = -\frac{\pi^2 f_0^2}{\alpha^2}. \tag{3.7}$$

In Abbildung 3.1 ist die normalisierte Wellenform (3.3) im Zeitbereich (a) und im Frequenzbereich (b) dargestellt. Dass (3.3) die Admissibilitätsbedingung nicht erfüllt, kann in (b) klar nachvollzogen werden. Ebenfalls enthalten in Abbildung 3.1 ist das Wavelet (3.5). Auch hier wurde die Darstellung für Zeitbereich (c) und Frequenzbereich (d) getrennt. Die Auswirkung des Korrektur-Terms e^{ξ} in (3.5) wird hier deutlich, da der Mittelwert der Wellenform verschwindet. Allerdings hat diese Maßnahme auch eine Verschiebung der zentralen Frequenz f_c gegenüber f_0 zur Folge (d). Dieses Problem wird im Folgenden behandelt.
Zunächst wird zur Reduzierung der freien Parameter der Wellenform (3.5) durch die

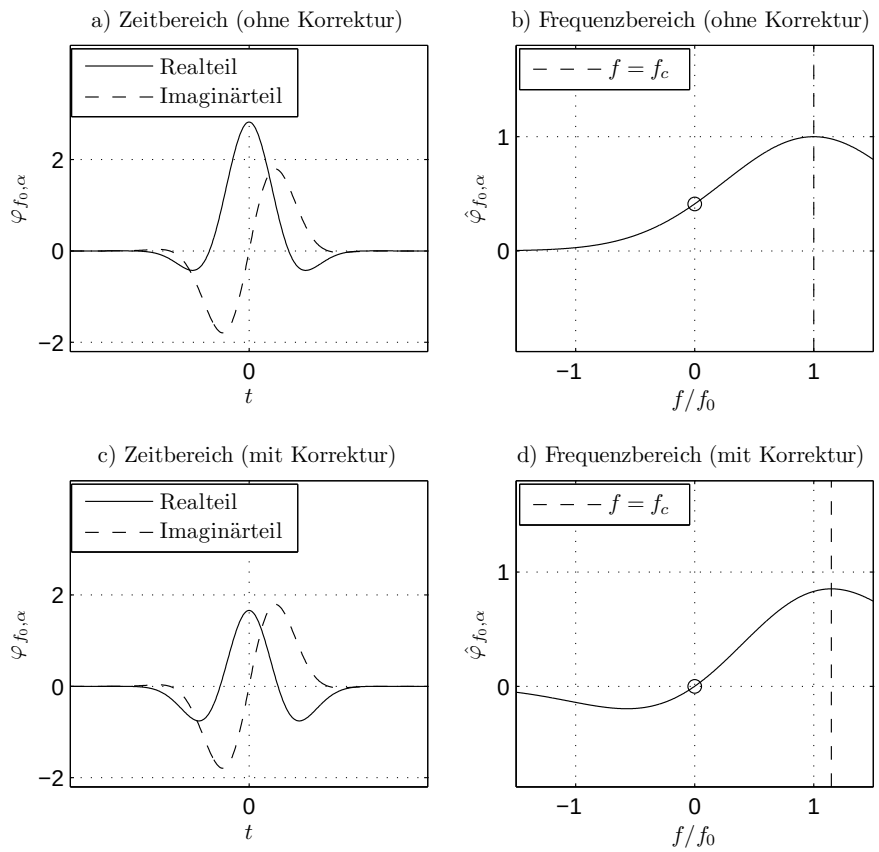

Abbildung 3.1: Die Wellenform (3.5) mit und ohne Korrektur-Term im Zeit- und im Frequenzbereich

Einführung eines neuen Parameters

$$\gamma := \frac{f_0}{\alpha} \tag{3.8}$$

die Fenstergröße an f_0 gekoppelt.
Unter Berücksichtigung von (3.7) und (3.8) erhält man nun für (3.5) und (3.6)

$$\varphi_{f_0,\gamma}(t) := \frac{f_0}{\gamma\,\sqrt{\pi}} \exp\left[-\frac{f_0^2}{\gamma^2}\,t^2\right]\left(\exp\left[i\;2\pi f_0 t\right] - \exp\left[-\pi^2\gamma^2\right]\right) \tag{3.9}$$

und

$$\hat{\varphi}_{f_0,\gamma}(f) = \exp\left[-\frac{\pi^2\gamma^2}{f_0^2}\ (f_0 - f)^2\right]\left(1 - \exp\left[-2\pi^2\gamma^2\frac{f}{f_0}\right]\right). \tag{3.10}$$

Die Einführung des Korrektur-Terms bewirkt eine Verschiebung der zentralen Frequenz relativ zu f_0, die nur von dem Parameter γ abhängt. Das Maximum von (3.10), und damit f_c, ist die Lösung der nichtlinearen Gleichung

$$\left(\frac{f_c}{f_0}\right)\left(1 - \exp\left[-2\pi^2\gamma^2\ \left(\frac{f_c}{f_0}\right)\right]\right) = 1. \tag{3.11}$$

Für kleine Werte von γ steigt das Verhältnis f_c/f_0 nichtlinear an wohingegen sich für große γ-Werte asymptotisch die Gleichheit $f_c = f_0$ einstellt. Durch numerisches Lösen von (3.11) für kleinere γ-Werte konnte ein Polynom identifiziert werden, das dieses Verhältnis für für $\gamma \geq 0{,}02$ gut approximiert:

$$\frac{f_c}{f_0} \approx \lambda(\gamma) := 1 + 0{,}018\left(\gamma^2 + 0{,}333\gamma + 0{,}005\right)^{-1{,}44} \tag{3.12}$$

Die numerische Lösung von (3.11) und die Approximation (3.12) sind in Abbildung 3.2 dargestellt.

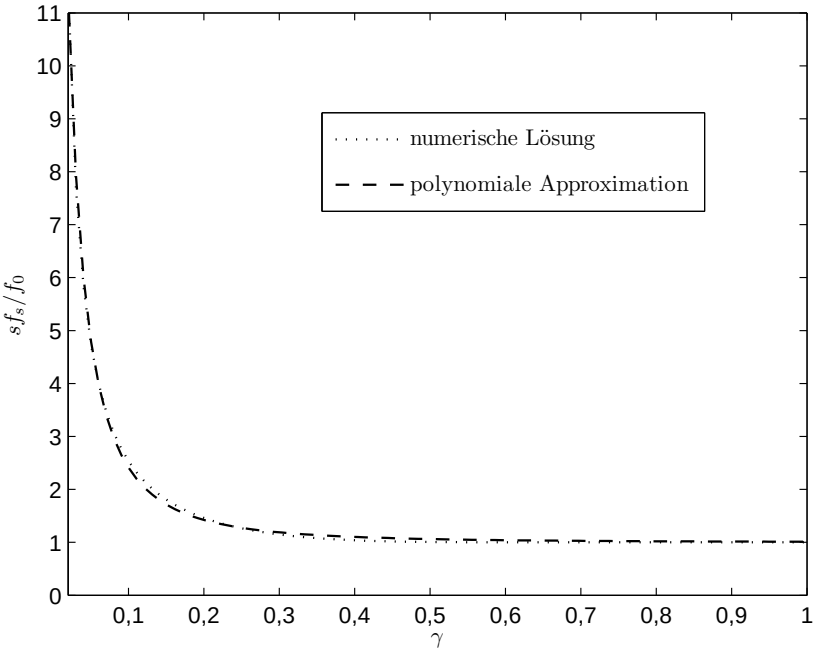

Abbildung 3.2: Numerische Lösung von (3.11) und polynomiale Approximation (3.12) von $\frac{f_c}{f_0}$

3.1 Zusammenhang von Skala und Frequenz

Bei der Einführung und Motivation der Datentransformation wurde die Wavelet-Transformation als Instrument der Zeit-Frequenz-Analyse bezeichnet. Tatsächlich liefern die in (3.2) berechneten Waveletkoeffizienten aber keine direkt in Zeit und Frequenz aufgelöste Darstellung des transformierten Signals, sondern sogenannte Skalogramme [Ari95], [Dau92].

Die berechneten Koeffizienten liefern Informationen zu Signalinhalten an einer Position τ und auf einer Skala s. Während die Position τ gleichzusetzen ist mit dem gewünschten Zeitpunkt, an dem das Analyse-Wavelet zu zentrieren ist, verhält sich die Skala s umgekehrt proportional zu einer Frequenz f. Für eine Signaldarstellung in Zeit und Frequenz muss also eine Vorschrift gefunden werden, mit der die Skalen ermittelt werden können, die einer derartigen Darstellung bezüglich eines fest vorgegebenen Frequenzintervalls entsprechen.

Aus der Definition (3.2) der komplexen kontinuierlichen Wavelet-Transformation wird ersichtlich, dass für eine spezifische Skala s und $\varphi_{f_0,\gamma}(t)$ als Mutterwavelet die Berechnung der Waveletkoeffizienten durch eine Faltung des Signals $a(t)$ mit der komplexen Konjugation

von

$$\varphi^s_{f_0,\gamma}(t) := \frac{1}{\sqrt{s}}\,\frac{f_0}{\gamma\sqrt{\pi}}\exp\left[-\frac{f_0^2t^2}{\gamma^2s^2}\right]\left(\exp\left[\frac{i2\pi f_0 t}{s}\right]-\exp\left[-\pi^2\gamma^2\right]\right)$$

erfolgen kann. Wegen

$$\varphi^s_{f_0,\gamma}(t) = \sqrt{s}\;\varphi_{f_0/s,\gamma}(t) \tag{3.13}$$

gilt auf Grund der Linearität des Fourier-Integrals auch

$$\hat{\varphi}^s_{f_0,\gamma}(f) = \sqrt{s}\;\hat{\varphi}_{f_0/s,\gamma}(f). \tag{3.14}$$

Unter Beachtung von (3.11) kann man nun weiter folgern, dass durch die Wahl der Skala s das Spektrum $\hat{\varphi}^s_{f_0,\gamma}(f)$ des Faltungskerns $\varphi^s_{f_0,\gamma}(f)$ genau dann sein Maximum bei einer Frequenz f_s annimmt, wenn der Zusammenhang

$$\left(\frac{sf_s}{f_0}\right)\left(1-\exp\left[-2\pi^2\gamma^2\;\left(\frac{sf_s}{f_0}\right)\right]\right)=1$$

erfüllt ist.
Unter Verwendung der Approximation (3.12) gestattet dies schließlich die Ermittlung der jeweiligen Skala s, die, bezogen auf die Wavelet-Transformation (3.2), eine Analyse des Signals $a\,(t)$ zu einem um f_s zentrierten Frequenz- und einem um τ zentrierten Zeitintervalls bewirkt:

$$s\,(f_s) := \frac{f_0}{f_s}\lambda(\gamma) \tag{3.15}$$

Wendet man (3.15) auf (3.13) und (3.14) an, so erhält man schließlich den Faltungskern

$$\varphi^{s(f_s)}_{f_0,\gamma}(t) := \sqrt{\frac{f_0 f_s}{\gamma^2\pi\lambda\,(\gamma)}}\exp\left[-\frac{f_s^2}{\gamma^2\;\lambda^2(\gamma)}t^2\right]\left(\exp\left[\frac{i2\pi f_s}{\lambda(\gamma)}t\right]-\exp\left[-\pi^2\gamma^2\right]\right) \tag{3.16}$$

und dessen Fourier-Transformierte

$$\hat{\varphi}^{s(f_s)}_{f_0,\gamma}(f) = \sqrt{\lambda\,(\gamma)\,\frac{f_0}{f_s}}\exp\left[-\frac{\pi^2\gamma^2}{f_s^2}\;(f_s-f\;\lambda(\gamma))^2\right]\left(1-\exp\left[-2\pi^2\gamma^2\;\frac{f\;\lambda(\gamma)}{f_s}\right]\right). \tag{3.17}$$

Man erkennt in (3.16) und (3.17), dass der Parameter f_0 keinen Einfluss auf die Zeit-Frequenz-Auflösung der Transformation hat. Er tritt lediglich als konstanter Faktor auf. Zum Zwecke der Normierung von (3.9) bietet sich hier o.B.d.A die Festlegung $f_0 = \gamma\;\sqrt{\pi}$ an, die für den weiteren Verlauf auch verwendet wird.
Hiermit ergibt sich schließlich als Endversion des zu verwendenden Mutterwavelets die

Wellenform

$$\varphi_\gamma(t) := \exp\left[-\pi\ t^2\right]\left(\exp\left[i\ 2\sqrt{\pi^3}\gamma t\right] - \exp\left[-\pi^2\gamma^2\right]\right) \tag{3.18}$$

mit der Fourier-Transformierten

$$\hat{\varphi}_\gamma(f) = \exp\left[-\pi\ \left(\gamma\ \sqrt{\pi} - f\right)^2\right]\left(1 - \exp\left[-2\sqrt{\pi^3}\gamma f\right]\right).$$

Die Skala $s = s\left(f_s\right)$, für die im Rahmen der Waveletanalyse das Signal bezüglich eines Intervalls um die beliebig aber fest gewählte Frequenz $f_s > 0$ untersucht wird, ist dabei über die Vorschrift

$$s\left(f_s\right) := \frac{\gamma\ \sqrt{\pi}}{f_s}\lambda(\gamma) \tag{3.19}$$

zu ermitteln.
Mit (3.2) und (3.19) lässt sich somit das Wavelet-Tableau

$$W\left[a\right]\left(f,t\right) := \left|\mathcal{W}\left[a\right]\left(s\left(f\right),t\right)\right| \tag{3.20}$$

definieren, dass einer zweckmäßigen Darstellung der Leistung des Signals a in der Zeit-Frequenz-Ebene entspricht.
Als letzter Schritt für die Festlegung der Datentransformation verbleibt vorerst die Wahl des Parameters γ.

3.2 Wahl des Transformations-Parameters γ

In den letzten Abschnitten wurde die Datentransformation der gemessenen Signale vorgestellt, wobei der Transformations-Parameter γ als einziger Freiheitsgrad für die Zeit-Frequenz-Auflösung des Mutterwavelets (3.18) bislang offen geblieben ist. Wie dieser sinnvoll zu wählen ist, wird im Folgenden erörtert.
Die gemessenen Daten werden bezüglich (3.2) und (3.18) transformiert. Bezeichnet N_t die Anzahl der Zeitschritte und N_f die Anzahl der relevanten Frequenz-Werte, so führt die Bestimmung des Wavelet-Tableaus (3.20) zu einer Wavelet-Tableau-Matrix $\mathbf{W} \in \mathbb{R}_+^{N_f \times N_t}$, die aus den Absolutbeträgen der komplexen Waveletkoeffizienten besteht. Welche Frequenzen dabei im Rahmen dieser Arbeit für welche Signaltypen herangezogen wurden, kann Tabelle 2.7 entnommen werden.
Die Absicht hinter der verwendeten Datentransformation war, die Anteile der Signaldaten sichtbar zu machen, die für Änderungen des subjektiv empfundenen Fahrkomforts verantwortlich sind. Vor diesem Hintergrund wird für einen festen Kanal k die Teilmenge

$$D_k \subset \left\{(\mathbf{W},\ y)\ \mid \mathbf{W} \in \mathbb{R}_+^{N_f \times N_t}, y \in [1{,}10] \subset \mathbb{R}\right\}$$

als Datenbasis der experimentell ermittelten objektiven und subjektiven Daten aus $|D_k|$ Experimenten definiert.
Für jeden Datensatz $(\mathbf{W}_k^i, y^i) =: d_k^i \in D_k$, $i = 1,2,\ldots |D_k|$, von objektiven und subjektiven Daten ist $\mathbf{W}_k^i$ die $N_f \times N_t$-Matrix, die die Absolutbeträge der Waveletkoeffizienten der transformierten Signaldaten des k-ten Kanals beinhaltet. y^i ist die Komfortnote, die bezüglich der Fahrzeugvariante aus dem i-ten Experiment vom Testfahrer abgegeben wurde. Wie bereits erwähnt ist dabei die bestmögliche Note die Zehn, die schlechteste die Eins (vgl. Abschnitt 2.1.3).
Das jeweils bewertete Kriterium sei in dieser Betrachtung außen vor belassen, da o.B.d.A die komplette Verarbeitungskette von den Messdaten bis hin zum Schätzwert für die Komfortnote für jedes zu bewertende Kriterium individuell erfolgen kann.
Die Idee für die Bestimmung des Parameters γ besteht jetzt darin, für jedes Paar von unterschiedlichen Datensätzen d_k^i und d_k^j die relativen Abstände

$$\delta_{i,j}^{\mathbf{W}} := 2\,\frac{\left\|\mathbf{W}_k^i - \mathbf{W}_k^j\right\|_2}{\left\|\mathbf{W}_k^i\right\|_2 + \left\|\mathbf{W}_k^j\right\|_2}$$

und

$$\delta_{i,j}^{y} := 2\,\frac{\left|y_k^i - y_k^j\right|}{22 - y_k^i - y_k^i}$$

zu ermitteln. Hierbei ist $\|\cdot\|_2$ die Frobenius-Norm auf $\mathbb{R}^{N_f \times N_t}$ [Hor90].
Für den relativen Abstand $\delta_{i,j}^{y}$ werden die Komfortnoten y_k^i, die per Definition mit steigendem Komfort zunehmen, mittels der Vorschrift $y_k^i \mapsto 11 - y_k^i$ gedreht. So gehen schließlich kleinere Komfortnoten mit größerem Komfort einher. Da nach der in dieser Arbeit angenommenen Einordnung des Komfortbegriffs (vgl. Abschnitt 2.1.3) davon auszugehen ist, dass der subjektiv empfundene Fahrkomfort mit zunehmenden Beschleunigungsamplituden abnimmt, führt diese Anpassung zu einer positiven Korrelation zwischen $\delta_{i,j}^{\mathbf{W}}$ and $\delta_{i,j}^{y}$.
Für eine gut geeignete Darstellung der komfortrelevanten Messdaten nach der Transformation in Form der $\mathbf{W}_k^i$ würde man daher annehmen, dass

$$\delta_{i,j}^{y} \sim \delta_{i,j}^{\mathbf{W}} \quad \forall i,j = 1,\ldots,|D_k|$$

Da aber die Daten-Darstellung $\mathbf{W}_k^i = \mathbf{W}_k^i(\gamma)$ von dem Transformations-Parameter γ abhängt, kann umgekehrt die Wahl von γ entsprechend der Darstellung $\mathbf{W}_k^i(\gamma)$ gewählt werden, die die Kostenfunktion

$$E_k(\gamma) = \frac{1}{2}\sum_{i=1}^{|D_k|}\sum_{j=i+1}^{|D_k|}\left(\delta_{i,j}^{y} - \beta\,\delta_{i,j}^{\mathbf{W}(\gamma)}\right)^2, \tag{3.21}$$

mit

$$\delta_{i,j}^{\mathbf{W}(\gamma)} := 2\,\frac{\left\|\mathbf{W}_k^i(\gamma) - \mathbf{W}_k^j(\gamma)\right\|_2}{\left\|\mathbf{W}_k^i(\gamma)\right\|_2 + \left\|\mathbf{W}_k^j(\gamma)\right\|_2},$$

minimiert.
Der Skalar β wurde dabei eingeführt, um die grundsätzliche Diskrepanz bei dem Vergleich der euklidischen Norm auf zwei verschieden dimensionierten Vektorräumen zu kompensieren. Um das größtmögliche Potential zur Minimierung von (3.21) dem Parameter γ zukommen zu lassen, wird

$$\beta = \min_{\beta}\; E_k\left(\gamma\right) = \frac{\sum\limits_{i=1}^{|D_k|}\sum\limits_{j=i+1}^{|D_k|} \delta_{i,j}^{y}\;\delta_{i,j}^{\mathbf{W}(\gamma)}}{\sum\limits_{i=1}^{|D_k|}\sum\limits_{j=i+1}^{|D_k|} \left(\delta_{i,j}^{\mathbf{W}(\gamma)}\right)^2}$$

gesetzt.
Aus einer statistischen Sichtweise können die verfügbaren experimentell ermittelten Daten als kleine Stichprobe einer sehr viel größeren Grundpopulation möglicher Zustände angesehen werden.
Der Ansatz, E_k in (3.21) bezüglich aller verfügbaren Datensätze zu minimieren, wird als gierig (eng.: greedy) bezeichnet [Man02]. Man muss sich dessen bewusst sein, dass im Allgemeinen kein Wert γ^* existiert, der E_k für jeden möglichen Datensatz Null werden lässt. Stattdessen ist davon auszugehen, dass E_k einen zufälligen, von Null verschiedenen Fehler beinhaltet, der nicht von γ abhängt.
Diese Unabhängigkeit ist dabei allgemeingültig und kann daher nicht von jeder beliebigen aber festen Stichprobe von Daten erfüllt werden. Es ist sogar zu erwarten, dass bei ausschließlicher Betrachtung einer Datenbasis wie im obigen Fall der Parameter γ mit dem zufälligen Anteil mehr oder weniger stark korreliert ist.
Wenn dies der Fall ist, so wird die Minimierung von E_k *um jeden Preis* und einzig unter der Berücksichtigung einer kleinen Stichprobe von Daten Nutzen aus dieser Korrelation ziehen. Die Folge wäre eine Anpassung des freien Parameters γ an den eigentlich nicht durch γ erklärbaren Restanteil von der Kostenfunktion, wodurch die Aussagekraft von γ für den allgemeinen Fall deutlich verfälscht wird.
Vor dem Hintergrund der Vorhersagemodelle wird dieses Problem als Überanpassung (eng.: overfitting) bezeichnet (vgl. Abschnitt 5.2.2). Die Folge ist im Allgemeinen ein unzureichendes Generalisierungsvermögen der zugrunde liegenden Modelle, da die Anpassung der freien Parameter übermäßig bezüglich einer festen und unzureichend repräsentativen Stichprobe geschehen ist.
Um eine Überanpassung von γ so weit wie möglich zu vermeiden, werden nicht alle verfügbaren Datensätze für die Minimierung von der Kostenfunktion (3.21) herangezogen.

Stattdessen wird eine sogenannte K-Kreuzvalidierung implementiert (vgl. Abschnitt 5.2.5). In der Praxis werden hierbei die insgesamt verfügbaren Daten zufällig auf K disjunkte und näherungsweise gleich große Teilmengen $D_k^1,...,D_k^K$ verteilt. Anschließend wird für $i = 1,...,K$ der Parameter γ^i als das Minimum von (3.21) bezüglich

$$D_k = \dot{\bigcup_{j \neq i}} D_k^j$$

bestimmt. Der verbleibende Rest von E_k wird in diesem Kontext als der Trainings-Fehler

$$e_{train}^i := E_k\left(\gamma^i\right)$$

bezeichnet. Dagegen bezeichnet der Wert von (3.21) bei Anwendung auf $D_k = D_k^i$ den Validierungs-Fehler e_{val}^i für γ^i.
Nach den K Kreuzvalidierungs-Durchläufen werden die dabei erhaltenen Trainings- und Validierungsfehler normalisiert, um eine Vergleichbarkeit trotz der unterschiedlichen Anzahl der verwendeten Datensätze zu gestatten:

$$\tilde{e}_{val}^i := K \frac{e_{val}^i}{\sum\limits_{j=1}^{K} e_{val}^j},$$

$$\tilde{e}_{train}^i := K \frac{e_{train}^i}{\sum\limits_{j=1}^{K} e_{train}^j}$$

Schließlich wird $\gamma^* = \gamma^{i^*}$ entsprechend des kleinsten kombinierten Fehlers

$$\tilde{e}_k^* = \tilde{e}_k^{i^*} = \min_i \left\{\max\left\{\tilde{e}_{train}^i, \tilde{e}_{val}^i\right\}\right\}$$

gesetzt.
Diese Prozedur lässt sich leicht erweitern, indem man die Werte γ^* und $\tilde{e}_k^*$ über mehrere K-Kreuzvalidierungen hinweg ermittelt. Der endgültige Wert für γ ist dann entsprechend des Wertes γ^* zu wählen, der mit dem kleinsten kombinierten Fehler $\tilde{e}_k^*$ aus allen im Rahmen der verschiedenen Kreuzvalidierungsrunden ermittelten kombinierten Fehlern korrespondiert.
Da der Schritt der zufälligen Partitionierung der gesamten Datenbasis in jeder Runde wiederholt wird, erhöht diese Maßnahme die Chance, einen geeigneten Wert für γ zu erhalten, der einen kleinen Wert für die Kostenfunktion (3.21) für alle verfügbaren Datensätze erreicht, und gleichzeitig eine Überanpassung an die verwendeten Daten weitestgehend vermeidet.

Im Rahmen der Untersuchungen wurden 50 Runden einer 4-Kreuzvalidierung bei einer Datenmenge von $|D_k| = 41$ verfügbaren Datensätzen für die sechs betrachteten Kanäle (vgl. Tabelle 2.6) und die drei verschiedenen Hindernis-Typen verwendet (vgl. Abschnitt 2.1.2).
Eine frühere Untersuchung der berechneten Waveletkoeffizienten hatte dabei gezeigt, dass für die aufgezeichneten Daten und die betrachteten Frequenzbereiche der Parameter γ in dem Intervall $I_\gamma = [0{,}3\ ;\ 3]$ liegen sollte. Werte oberhalb von drei führten zu einer sehr geringen zeitlichen Auflösung. Für Werte unterhalb von 0,3 hingegen war die Frequenz-Information beinahe vollständig verlorenen gegangen.
Die endgültigen γ-Werte, die mit dieser Methode ermittelt wurden, sind in Tabelle 3.1 dargestellt.

		Kanal					
		1	**2**	**3**	**4**	**5**	**6**
mnegK	**vorne**	*3,00*	*0,30*	*0,30*	*0,30*	*0,30*	*3,00*
	hinten	*0,30*	*1,05*	*0,30*	*1,90*	*0,50*	*0,30*
mposK	**vorne**	*3,00*	*3,00*	*0,30*	*3,00*	*0,70*	*1,90*
	hinten	*0,30*	*1,05*	*0,30*	*1,90*	*0,50*	*0,30*
UR	**vorne**	*3,00*	*3,00*	*3,00*	*3,00*	*0,30*	*0,30*
	hinten	*3,00*	*0,30*	*3,00*	*0,45*	*0,80*	*3,00*

Tabelle 3.1: Berechnete γ-Werte

3.3 Einflussbereiche zeitlich getrennter Ereignisse

Wie in Abschnitt 3.1 erläutert, gestattet das in (3.20) definierte Wavelet-Tableau eine Auflösung der Signalinhalte in Zeit und Frequenz. Das Wavelet-Tableau selber wird dabei über die Wavelet-Transformation (3.2) der Signaldaten bezüglich geeignet gewählter Skalen ermittelt.
Auf natürliche Weise ist die zeitliche Auflösung der Daten in dieser Darstellung abhängig von der Skala, bezüglich der die Daten transformiert werden. Dies wird zum Einen konstruktiv genutzt, um über die Variation der Skalen die Signalanteile zu verschiedenen Frequenzen darzustellen. Zum Anderen aber bewirkt der Skalenparameter eine Verschiebung der Zeit- und der Frequenzauflösung.
Zur Veranschaulichung dieses Sachverhalts betrachte man Abbildung 3.3. Sie zeigt den Faltungskern der Wavelet-Transformation für das hergeleitete Wavelet zu den Skalen $s = 0{,}8$ und $s = 2{,}2$. Der Parameter γ spielt in diesem Zusammenhang eine untergeordnete Rolle und wurde auf $\gamma = 1{,}3$ gesetzt.
Der Einfluss des Skalenparameters s auf die Transformation (3.2) wird deutlich sichtbar. Für größere Werte von s wird der Faltungskern im Zeitbereich gestreckt, für kleinere gestaucht. Dadurch ändert sich, wie gewünscht, die Frequenz der enthaltenen harmonischen Schwingung. Für kleinere Skalen wird diese folglich größer, für große Skalen kleiner. Dies ist

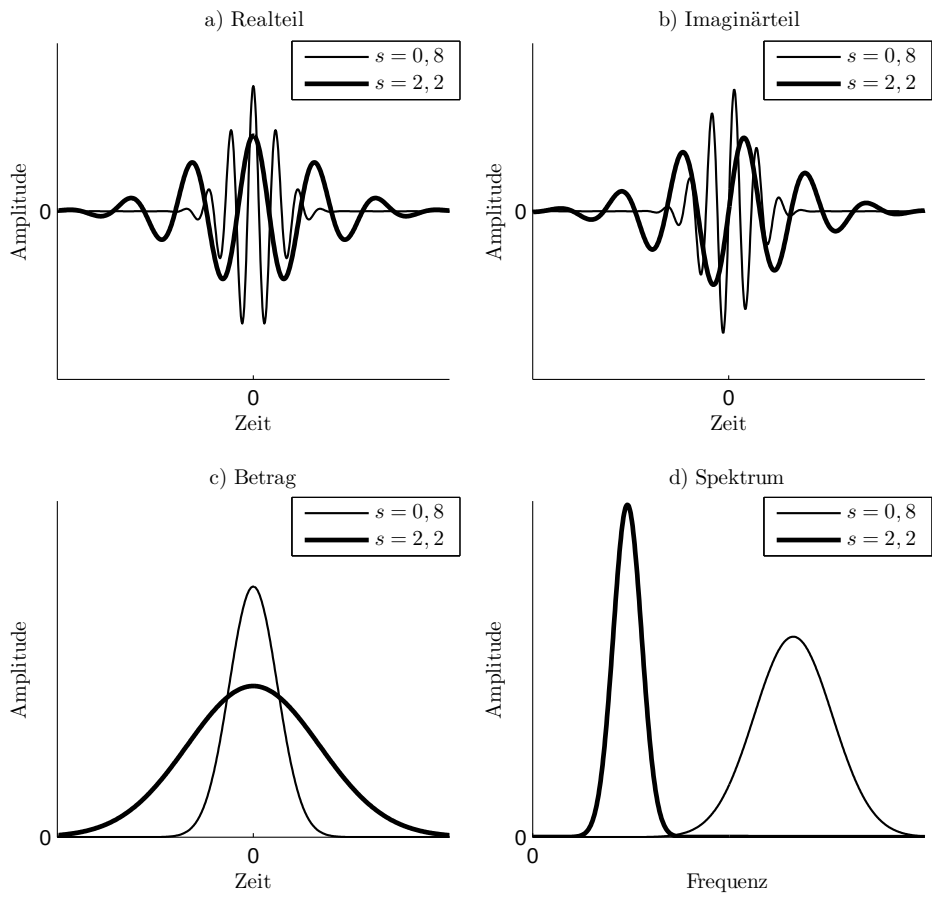

Abbildung 3.3: Einfluss der Skala auf Zeit- und Frequenzlokalisierung des Faltungskerns der Wavelet-Transformation

in Abbildung 3.3 sowohl in den Zeitschrieben der Teilabbildungen (a) und (b) erkennbar, als auch in der spektralen Ansicht der Teilabbildung (d), wo sich jeweils ein anderes Frequenzband für den Träger des Faltungskerns einstellt.
Ebenfalls in Teilabbildung (d) erkennbar ist aber, dass die Breite der Frequenzbereiche für kleinere Skalen größer und für größere Skalen kleiner ist. Diese Breite lässt sich als Unschärfe der Frequenzlokalisierung des Faltungskerns bezeichnen. Im Fall einer reinen harmonischen Schwingung im Zeitbereich ist diese infinitisimal klein, was der bestmöglichen Frequenzlokalisierung entspricht.
Der Transformationskern ist im Zeitbereich eine gewichtete harmonische Schwingung. Die Gaußglocken-förmige Gewichtungsfunktion verkleinert den Bereich, in dem diese Schwingung signifikant von Null verschieden ist. Je größer dieser Bereich ist, um so mehr ähnelt der Transformationskern einer harmonischen Schwingung und um so kleiner wird die

Frequenzunschärfe. Anders herum wird die Frequenzunschärfe größer, wenn dieser Bereich kleiner wird, da sich dadurch der Transformationskern von der harmonischen Schwingung weg in Richtung einer eher stoßförmigen Wellenform verändert.
Teilabbildung (c) zeigt den Betrag des Faltungskerns im Zeitbereich, der gerade der Gaußglocken-förmigen Gewichtungsfunktion im Zeitbereich entspricht. Durch die zuvor erwähnte Streckung in Folge größerer Skalenwerte wird neben der Frequenz der harmonischen Schwingung auch die Breite des Faltungskerns im Zeitbereich größer. Dies entspricht einer Zunahme der Unschärfe der Zeitlokalisierung des Faltungskerns, die, analog zum oben genannten Beispiel, dann bestmöglich ist, wenn die Breite des Betrages des Faltungskerns im Zeitbereich infinitisimal klein ist, also ein Dirac-Impuls. Je größer der Bereich ist, desto mehr Einfluss haben Signalanteile in der Umgebung des jeweils zu transformierenden Abtastwertes auf das Ergebnis der Transformation.
Zusammenfassend lässt sich sagen, dass die Wavelet-Transformation keine feste Zeit- und Frequenzauflösung besitzt. Für ein festes γ und dem gewünschten Frequenzbereich entsprechende Skalenwerte werden hochfrequente Signalanteile mit höherer Präzision in der Zeitlokalisierung transformiert. Die dazugehörige Frequenz kann nur mit geringerer Genauigkeit aufgelöst werden. Umgekehrt verhält es sich mit niederfrequenten Signalanteilen, bei denen die Frequenz mit höherer Genauigkeit aufgelöst, die exakte zeitliche Lokalisierung hingegen nur wage angegeben werden kann.
Im Folgenden werden die Konsequenzen der Unschärfe der Zeitlokalisierung näher untersucht. Wie bereits erwähnt, liegt die bestmögliche Zeitauflösung nur in der zeitlichen Darstellung selber vor, die dem Grenzfall einer infinitisimal kleinen Skala entspricht. Vergrößert man die Skala, so vergrößert man auch den Einflussbereich des Faltungskerns und bezieht Abtastwerte in die Transformation mit ein, die in der Umgebung der aktuell betrachteten Abtastzeit liegen.
Im Fall der Einzelhindernisüberfahrten enthält ein Beschleunigungssignal mit den einzelnen Achsüberfahrten zwei Ereignisse, die als jeweils für sich abgeschlossen und voneinander getrennt analysiert werden müssen. Hierzu muss bekannt sein, wie groß bei fester Skala s der Zeitbereich im Wavelet-Tableau ist, der von einem einzigen Abtastwert beeinflusst wird.
Die zu transformierenden Signale liegen in abgetasteter Form als Zeit-Werte-Paare vor. Ein Signal Y der Länge N lässt sich also vorerst nur als eine Menge

$$Y = \{(t_1,y_1)\,,(t_2,y_2)\,,\ldots,(t_N,y_N)\}$$

mathematisch formulieren, wobei die Nummerierung entsprechend der Anordnung

$$t_1 < t_2 < \ldots < t_N$$

der Abtastzeiten gewählt ist.
In der vorliegenden Arbeit werden dabei nur solche Signale betrachtet, die mit einer

konstanten Periode

$$\Delta t = t_2 - t_1 = \ldots = t_N - t_{N-1}$$

abgetastet vorliegen.
Diese zeitdiskrete Darstellung ist für eine zeitkontinuierliche Datentransformation ungeeignet. Da über die Signalzustände zwischen den Abtastzeiten t_i und $t_i + \Delta t$ keine Informationen vorliegen, werden die daraus resultierenden Lücken im Signal nach dem einfachsten denkbaren Fall aufgefüllt. Dieser besteht in der Annahme, dass in den, bedingt durch die fest vorgegebene Abtastrate, nicht erfassten Zeiträumen auch keine relevanten Änderungen des Signals auftreten können. Ohne diese Voraussetzung wäre die Aussagekraft und damit der Nutzen der aufgezeichneten Daten mehr als fraglich.
Der Signalwert $y(t)$ wird demnach über die gesamte jeweilige Zeitspanne von t_i bis unmittelbar vor $t_i + \Delta t$ als der jeweils letzte abgetastete Signalwert y_i angenommen. Somit ist durch

$$y(t)\Big|_{t\in[t_i,t_i+\Delta t)} = y_i, \quad i = 1,2,\ldots,N$$

bereits eine zeitkontinuierliche Darstellung der aus dem Signal Y abgeleiteten Funktion $y(t)$ gegeben. Diese Darstellung ist allerdings noch bereichsweise und somit nicht in geschlossener Form für den gesamten Definitionsbereich $t \in [t_1,t_N + \Delta t)$ formuliert. Abhilfe kann auf elegante Weise durch die Indikator-Funktion (Kronecker-Symbol)

$$\delta_M(x) := \begin{cases} 1, \text{ falls } x \in M \\ \\ 0 \text{ sonst} \end{cases}$$

geschaffen werden, wonach $y(t)$ als die Summe

$$y(t) = \sum_{i=1}^{N} \delta_{[t_i,t_i+\Delta t)}(t)\ y_i \tag{3.22}$$

geschlossen angegeben werden kann.
Da sich die einzelnen Bereichsintervalle nicht überschneiden, geht in die Summe (3.22) stets genau ein Abtastwert von Y effektiv ein. Die jeweils $N-1$ übrigen Summanden werden durch die Indikator-Funktion zu Null gesetzt.
Durch die Linearität des Integrals ergibt sich für diese Signalform die Wavelet-Transformation

gemäß (3.2) zu

$$\begin{aligned}
\mathcal{W}[y](s,\tau) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}} \left[\sum_{i=1}^{N} \delta_{[t_i,t_i+\Delta t)}(t)\ y_i\right] \varphi^*\left(\frac{t-\tau}{s}\right) dt \\
&= \sum_{i=1}^{N} \left[\frac{y_i}{\sqrt{s}} \int_{-\infty}^{\infty} \delta_{[t_i,t_i+\Delta t)}(t)\ \varphi^*\left(\frac{t-\tau}{s}\right) dt\right] \\
&= \sum_{i=1}^{N} y_i \left[\frac{1}{\sqrt{s}} \int_{t_i}^{t_i+\Delta t} \varphi^*\left(\frac{t-\tau}{s}\right) dt\right]. \qquad (3.23)
\end{aligned}$$

Die Aussage zu den Summanden in (3.22) gilt für (3.23) analog.
Für die Bewertung des Einflusses des Abtastwertes zur Abtastzeit t_i auf das Wavelet-Tableau wird die leicht variierte Zeitreihe $\tilde{y}(t)$ betrachtet, bei der nur der Wert zur Abtastzeit t_i gegenüber $y(t_i)$ mit der Störung Δy additiv überlagert ist. Dann lässt sich $\mathcal{W}[\tilde{y}](s,\tau)$ zerlegen in die Darstellung

$$\mathcal{W}[\tilde{y}](s,\tau) = \mathcal{W}[y](s,\tau) + \Delta\mathcal{W}[y](s,\tau),$$

wobei das Skalogramm $\Delta\mathcal{W}[y](s,\tau)$ gegeben ist durch

$$\Delta\mathcal{W}[y](s,\tau) := \Delta y \left[\frac{1}{\sqrt{s}} \int_{t_i}^{t_i+\Delta t} \varphi^*\left(\frac{t-\tau}{s}\right) dt\right]. \qquad (3.24)$$

Aus der Definition des Wavelet-Tableaus folgt, wegen

$$\left|\, |\mathcal{W}[\tilde{y}](s,\tau)| - |\mathcal{W}[y](s,\tau)| \,\right| \leq |\Delta\mathcal{W}[y](s,\tau)|,$$

die Abschätzung

$$|W[\tilde{y}](f,t) - W[y](f,t)| \leq \Delta W[y](f,t). \qquad (3.25)$$

Das Wavelet-Tableau $\Delta W[y](f,t)$ ist also für jedes zulässige Parameter-Tupel (f,t) eine obere Schranke für die durch die Störung Δy im Zeitbereich hervorgerufene absolute Änderung des Wavelet-Tableaus.
Die komplexen Waveletkoeffizienten in dem Skalogramm (3.24) hängen sowohl von der Störung Δy selber als auch von der Abtastperiode Δt des Signals ab. In einer ersten Betrachtung wird von einer kontinuierlichen Abtastung, also einer infinitisimal kleinen und konstanten Abtastperiode, ausgegangen. Im Grenzübergang vereinfacht sich dann der

Ausdruck für $\Delta\mathcal{W}[y](s,\tau)$ zu

$$\begin{aligned}\Delta\mathcal{W}[y](s,\tau) =& \Delta y \left[\frac{1}{\sqrt{s}}\ \varphi^*\left(\frac{t_i-\tau}{s}\right)\right]\\ =& \frac{\Delta y}{\sqrt{s}}\ \exp\left[-\pi\left(\frac{t_i-\tau}{s}\right)^2\right]\\ & \left\{\exp\left[-i\ 2\sqrt{\pi^3}\gamma\ \frac{t_i-\tau}{s}\right]-\exp\left[-\pi^2\gamma^2\right]\right\}.\end{aligned}$$

Für die Beträge der Skalogramm-Einträge gilt in diesem Fall

$$\begin{aligned}|\Delta\mathcal{W}[y](s,\tau)| =& \frac{|\Delta y|}{\sqrt{s}}\ \exp\left[-\pi\ \left(\frac{t_i-\tau}{s}\right)^2\right]\\ & \sqrt{1+\exp\left[-\pi^2\ \gamma^2\right]\ \left(\exp\left[-\pi^2\ \gamma^2\right]-2\ \cos\left[2\ \sqrt{\pi^3}\ \gamma\ \frac{t_i-\tau}{s}\right]\right)}.\end{aligned}$$

Das zugehörige Wavelet-Tableau lautet schließlich

$$\begin{aligned}\Delta W[y](f,t) =& |\Delta y|\ \exp\left[-f^2\left(\frac{t_i-t}{\lambda\gamma}\right)^2\right]\\ & \sqrt{\frac{f}{\lambda\gamma\sqrt{\pi}}\ \left(1+\exp\left[-\pi^2\ \gamma^2\right]\ \left(\exp\left[-\pi^2\ \gamma^2\right]-2\ \cos\left[\frac{2\pi f\,(t_i-t)}{\lambda}\right]\right)\right)}.\end{aligned}$$

Definiert man durch C_{thresh} eine obere Schranke für vernachlässigbar kleine absolute Änderungen des Wavelet-Tableaus, so lässt sich durch die Bestimmung der Grenzkurven $\Delta W[y](f,t) = C_{\text{thresh}}$ der beeinflusste Bereich im Tableau darstellen.
In Abbildung (3.4a), bzw. Abbildung (3.4b), ist die Grenzkurve $|\Delta\mathcal{W}[y](s,\tau)| = C_{\text{thresh}}$, bzw. $\Delta W[y](f,t) = C_{\text{thresh}}$, und der durch die Störung beeinflusste grau hinterlegte Bereich des Skalograms, bzw. Wavelet-Tableaus, dargestellt. Als Parameter wurden für diese Darstellung $\gamma = 0{,}3$ und $C_{\text{thresh}} = 0{,}2\ |\Delta y|$ gesetzt. Die zur Störung Δy gehörige Abtastzeit wurde o.B.d.A. zu $t_i = 0{,}3$ s gewählt.
Man erkennt deutlich, dass die zeitliche Ausdehnung des Einflussbereiches für große Skalen, bzw. tiefe Frequenzen, zunimmt, und für kleine Skalen, bzw. hohe Frequenzen, abnimmt. Die stark unterschiedlichen Verläufe der Grenzkurven ergeben sich dadurch, dass sich die Skala s nach (3.19) umgekehrt proportional zu der Frequenz f verhält.

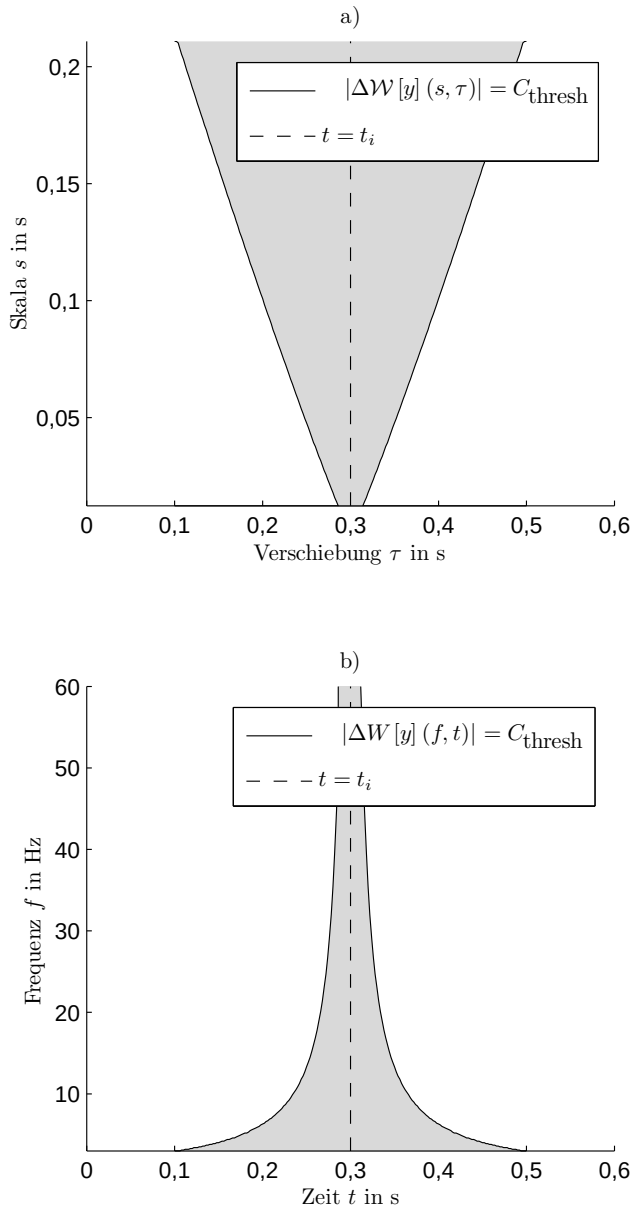

Abbildung 3.4: Grenzen und Bereiche des Einflusses einer Störung des Zeitsignals um Δy zur Zeit $t_i = 0{,}3$ s für $\gamma = 0{,}3$ und $C_{\text{thresh}} = 0{,}2\ |\Delta y|$

3.4 Numerische Berechnung der Transformation

Für die Transformation

$$\mathcal{W}\left[y\right](s,\tau) := \frac{1}{\sqrt{s}} \int\limits_{-\infty}^{\infty} y\left(t\right)\ \varphi^{*}\left(\frac{t-\tau}{s}\right) dt \tag{3.26}$$

der mit der Periode Δt abgetasteten Signaldaten $y\left(t\right)$ unter Verwendung des in Abschnitt 3.1 entwickelten Wavelets $\varphi\left(t\right)$ wird im Folgenden ein numerisches Verfahren hergeleitet. Zunächst ist anzumerken, dass die Transformationsvorschrift (3.26) im Wesentlichen in einer komplexen Integration besteht. Diese kann durch die Zerlegung

$$\begin{aligned}\int\limits_{-\infty}^{\infty} y\left(t\right)\ \varphi^{*}\left(\frac{t-\tau}{s}\right) dt = &\int\limits_{-\infty}^{\infty} y\left(t\right)\ \mathrm{Re}\left\{\varphi\left(\frac{t-\tau}{s}\right)\right\} dt \\ &- j \int\limits_{-\infty}^{\infty} y\left(t\right)\ \mathrm{Im}\left\{\varphi\left(\frac{t-\tau}{s}\right)\right\} dt\end{aligned}$$

auf die komplexe Summe zweier rein reeller Integral-Transformationen zurückgeführt werden und wird numerisch auch auf diese Weise ermittelt. Vorgestellt wird daher nur ein Transformationsalgorithmus für ein reelles Wavelet $\tilde{\varphi}\left(t\right)$.
Stellt man $y\left(t\right)$ wie in (3.22) dar, so erhält man, ausgehend von (3.23),

$$\mathcal{W}\left[y\right](s,\tau) = \frac{1}{\sqrt{s}} \sum_{i=1}^{N} y_i \left[\int\limits_{-\infty}^{t_i+\Delta t} \tilde{\varphi}\left(\frac{t-\tau}{s}\right) dt - \int\limits_{-\infty}^{t_i} \tilde{\varphi}\left(\frac{t-\tau}{s}\right) dt\right].$$

Für die folgenden Schritte wird o.B.d.A $t_1 = 0s$ angenommen, so dass gilt:

$$t_i = \Delta t\ \left(i-1\right), \quad i = 1,2,...,N$$

Für die Verschiebung τ wird die gleiche Diskretisierung wie für t angesetzt:

$$\tau = \Delta t\ k, \quad \text{mit } k \in \mathbb{N} \text{ und } 0 \leq k \leq N-1$$

Eine wichtige Erkenntnis besteht dann darin, dass der wesentliche Aufwand der Transformation von $y\left(t\right)$ bezüglich der Skala s in der Bestimmung der Integrale

$$I_i := \int\limits_{-\infty}^{\Delta t\ (i-1)} \tilde{\varphi}\left(\frac{t-\Delta t\ k}{s}\right) dt, \qquad i = 1,2,\ldots,N+1 \tag{3.27}$$

besteht. Diese hängen zwar von der Abstastperiode Δt von $y(t)$ ab, sind aber unabhängig von den tatsächlichen Abtastwerten $y_1, y_2, \ldots, y_N$.
Sind die Integrale $I_1, \ldots, I_{N+1}$ also einmal bestimmt, so reduziert sich der Rest der numerischen Transformation bezüglich der Skala s von $y(t)$ und allen weiteren Signalen, die auf die gleiche Weise abgetastet wurden, auf die Berechnung von Skalarprodukten. Vor dem Hintergrund, dass die Transformation auf viele synchron erfasste und identisch abgetastete Datenkanäle angewendet werden soll ergibt sich hierdurch eine signifikante Verringerung des Gesamtaufwands.
Das Integral 3.27 lässt sich durch Anwendung der Substitution umschreiben zu

$$I_i := s \int_{-\infty}^{\frac{\Delta t\ (i-k-1)}{s}} \tilde{\varphi}(x)\, dx. \tag{3.28}$$

Ist nun die Stammfunktion F von $\tilde{\varphi}$ mit $\lim_{x \to -\infty} F(x) = 0$ (da $\tilde{\varphi}$ ein Wavelet ist) bekannt, so lässt sich 3.28 schreiben als

$$I_i := s\ F\left(\frac{\Delta t\,(i-k-1)}{s}\right). \tag{3.29}$$

Die gesamte Transformation lässt sich dann darstellen zu

$$\mathcal{W}[y](s, \Delta t\ k) = \sqrt{s}\ \sum_{i=1}^{N} y_i \left[F\left(\frac{\Delta t\,(i-k)}{s}\right) - F\left(\frac{\Delta t\,(i-k-1)}{s}\right)\right].$$

Definiert man nun

$$F^j_{\Delta t, s} := F\left(j\frac{\Delta t}{s}\right)$$

und

$$V^j_{\Delta t, s} := \sqrt{s}\ \left(F^j_{\Delta t, s} - F^{j-1}_{\Delta t, s}\right),$$

so ergeben sich die Waveletkoeffizienten für die Skala s zu

$$\mathcal{W}[y](s, \Delta t\ k) = \sum_{i=1}^{N} y_i\ V^{i-k}_{\Delta t, s}.$$

Sie lassen sich demnach auch über das Matrix-Vektor-Produkt

$$\mathbf{w}_s[y] := \begin{bmatrix} \mathcal{W}[y](s,(N-1)\,\Delta t) \\ \mathcal{W}[y](s,(N-2)\,\Delta t) \\ \vdots \\ \mathcal{W}[y](s,\Delta t) \\ \mathcal{W}[y](s,0) \end{bmatrix}$$

$$= \begin{bmatrix} y_1 & \dots & y_N & 0 & \dots & \dots & 0 \\ 0 & y_1 & \dots & y_N & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & y_1 & \dots & y_N & 0 \\ 0 & \dots & \dots & 0 & y_1 & \dots & y_N \end{bmatrix} \begin{bmatrix} V_{\Delta t,s}^{2-N} \\ V_{\Delta t,s}^{3-N} \\ \vdots \\ V_{\Delta t,s}^{N-1} \\ V_{\Delta t,s}^{N} \end{bmatrix} \tag{3.30}$$

berechnen.
Hieraus wird ersichtlich, dass sich der Vektor $\mathbf{w}_s[y]$ der Waveletkoeffizienten des Signals $y(t)$ zur Skala s als diskrete Faltung (mit zero-padding) des Vektors $\mathbf{v}_{\Delta t,s} = \left(V_{\Delta t,s}^{2-N},\dots,V_{\Delta t,s}^{N}\right)^{\mathrm{T}}$ mit dem Vektor $\mathbf{y} = (y_1,\dots,y_N)^{\mathrm{T}}$ der Abtastwerte berechnet.
Für die weitere Verarbeitung der Waveletkoeffizienten sei darauf hingewiesen, dass diese in dem Vektor $\mathbf{w}_s[y]$ für absteigende τ-Werte, beginnend bei $(N-1)\,\Delta t$, eingetragen sind.
Die Darstellung (3.30) bereitet den Weg für eine deutlich schnellere Berechnung der Waveletkoeffizienten über Algorithmen, die für die Berechnung der diskreten Faltung eine schnelle Fourier-Transformation heranziehen [Tei94],[Rad10].
Bestimmt man die Vektoren $\mathbf{v}_{\Delta t,s}$ für alle in dem Vektor $\mathbf{s} = [s_1,\dots,s_M]^{\mathrm{T}}$ enthaltenen Skalen und fasst diese in der Matrix

$$\mathbf{WV}_{\tilde{\varphi},\mathbf{s},\Delta t,N} = [\mathbf{v}_{\Delta t,s_1},\dots,\mathbf{v}_{\Delta t,s_M}]$$

zusammen, so sind durch $\mathbf{WV}_{\tilde{\varphi},\mathbf{s},\Delta t,N}$ alleine alle Informationen gegeben, um die Wavelet-Transformation beliebiger (Teil-)Signale der Länge N mit Abtastperiode Δt bezüglich der Skalen in $\mathbf{s}$ und des reellen Wavelets $\tilde{\varphi}$ nach der Vorschrift (3.30) zu bestimmen.
Für die Transformation bezüglich des komplexen Wavelets φ werden somit die Matrizen $\mathbf{WV}_{\mathrm{Re}\{\varphi\},\mathbf{s},\Delta t,N}$ und $\mathbf{WV}_{\mathrm{Im}\{\varphi\},\mathbf{s},\Delta t,N}$ benötigt. Die resultierenden komplexen Waveletkoeffizienten werden dann über die entsprechende komplexe Summe der jeweils bestimmten reellen Koeffizienten $\mathbf{w}_s[y]$ ermittelt.
Im Allgemeinen ist die Stammfunktion F in (3.29) nicht bekannt. Eine Möglichkeit, $\mathbf{v}_{\Delta t,s}$ dennoch zu bestimmen, besteht darin, das Integral (3.28) für jede entsprechende obere Grenze numerisch zu approximieren. Anstelle dieser Möglichkeit wird in dieser Arbeit ein Ansatz verfolgt, in dem die unbekannte Stammfunktion F geschlossen durch einen kubischen Spline näherungsweise ermittelt wird.

Allgemein hat ein interpolierender kubischer Spline $S_f(x)$ einer Funktion $f(x)$ zu gegebenen äquidistanten Stützstellen $x_1, \ldots, x_n$ und zugehörigen Funktionswerten $f_1 = f(x_1), \ldots, f_n = f(x_n)$ die Darstellung

$$S_f(x) = \begin{cases} P_1(x), & \text{für } x < x_1 \\ P_i(x), & \text{für } x_i \leq x < x_{i+1}, \quad i = 1, \ldots, n-1. \\ P_{n-1}(x), & \text{für } x \geq x_n \end{cases}$$

Die kubischen Polynome

$$P_i(x) = a_i (x - x_i)^3 + b_i (x - x_i)^2 + c_i (x - x_i) + d_i$$

verfügen jeweils über vier freie Parameter. Um diese geeignet festzusetzen, werden zunächst zwei Anforderungen an den Spline gestellt:

1. $S_f(x)$ interpoliert $f(x)$ in allen Knoten $(x_1, f_1), \ldots, (x_n, f_n)$
2. $S_f(x)$ ist auf ganz $\mathbb{R}$ mindestens zweimal stetig differenzierbar

Aus der ersten Anforderung ergeben sich für $i = 1, \ldots, n-1$ die Bedingungen

$$P_i(x_i) = f_i \tag{3.31}$$

und

$$P_i(x_{i+1}) = f_{i+1}. \tag{3.32}$$

Die zweite Anforderung stellt an den inneren Knoten $(i = 2, \ldots, n-1)$ die Bedingungen

$$P_i'(x_i) = P_{i-1}'(x_i) \tag{3.33}$$

und

$$P_i''(x_i) = P_{i-1}''(x_i). \tag{3.34}$$

Durch diese Bedingungen werden $4n - 6$ der insgesamt $4n - 4$ Freiheitsgrade gesperrt. Für die beiden verbleibenden Freiheitsgrade werden weitere Bedingungen benötigt, sogenannte Randbedingungen [Sch11].
In dem hier vorliegenden Fall soll allerdings nicht $\tilde{\varphi}(x)$ interpoliert, sondern die Stammfunktion $\int_{-\infty}^{x} \tilde{\varphi}(\xi)\, d\xi$ durch einen kubischen Spline approximiert werden. Nach Definition ist $\tilde{\varphi}(x)$ das Produkt einer (geringfügig verschobenen) harmonischen Schwingung und der Skalierungsfunktion $\exp\left[-\pi\, x^2\right]$.
Die Skalierungsfunktion erreicht niemals den Wert Null, weswegen aus analytischer Sicht für die Approximation der Stammfunktion ganz $\mathbb{R}$ berücksichtigt werden müsste. Numerisch ist allerdings bereits für $|x| > 15{,}41$ der Wert der Skalierungsfunktion so klein,

dass er beispielsweise von Matlab (als Gleitkommazahl mit doppelter Genauigkeit) nicht mehr als von Null verschiedener Wert dargestellt werden kann. Bereits für $|x| > 2{,}5$ ist $\exp\left[-\pi\, x^2\right] < 10^{-10}$ praktisch vernachlässigbar klein. Für die weiteren Betrachtungen wird daher der Träger des Wavelets $\tilde{\varphi}\,(x)$ auf das offene Intervall

$$I_T = \left(-\sqrt{\frac{1}{\pi}\ \log\frac{1}{\varepsilon_{\text{tol}}}}\ ;\ \sqrt{\frac{1}{\pi}\ \log\frac{1}{\varepsilon_{\text{tol}}}}\right)$$

beschränkt, wobei die Toleranzgrenze $0 < \varepsilon_{\text{tol}} < 1$ frei gewählt werden kann.
Über I_T wird nun ein äquidistantes Stützstellen-Gitter

$$x_1 = -\sqrt{\frac{1}{\pi}\ \log\frac{1}{\varepsilon_{\text{tol}}}},\ x_2,\ \ldots\ ,\ x_{n-1},\ x_n = \sqrt{\frac{1}{\pi}\ \log\frac{1}{\varepsilon_{\text{tol}}}}$$

der Feinheit

$$\begin{aligned} h &= x_2 - x_1 = \ldots = x_n - x_{n-1} \\ &= \sqrt{\frac{1}{\pi}\ \log\frac{1}{\varepsilon_{\text{tol}}}}\ \frac{2}{n-1} \end{aligned} \tag{3.35}$$

gelegt, über dem die Stammfunktion von $\tilde{\varphi}\,(x)$ mit Träger-Beschränkung durch einen kubischen Spline zu approximieren ist.
Durch ε_{tol} und n sind hier bereits zwei Parameter zur Beeinflussung der Approximationsgüte gegeben. Für $\varepsilon_{\text{tol}} \to 0$ wird der Fehler auf Grund der Beschränkung des Trägers von $\tilde{\varphi}\,(x)$ geringer und für $n \to \infty$ nimmt der Fehler auf Grund der abschnittsweisen Polynom-Approximation ab.
Natürliche Schranken für beide Parameter sind der Speicherbedarf sowie die Rechenzeit. Bedingt durch (3.35) zieht bei fester Feinheit h eine Anpassung von ε_{tol} nach unten stets eine Anpassung von n nach oben nach sich. Die Parameter ε_{tol} und n sind also durch h aneinander gekoppelt.
Während im Falle der Interpolation einer Funktion $f\,(x)$ mittels kubischer Splines neben den Stützstellen $x_1, \ldots, x_n$ auch die entsprechenden Funktionswerte vorliegen, ist dies hier nicht der Fall. Dafür sind allerdings die Werte $f_1' = \tilde{\varphi}\,(x_1), \ldots, f_n' = \tilde{\varphi}\,(x_n)$ der ersten Ableitung bekannt, so dass die Bedingungen (3.50) bis (3.53) sinnvoll ersetzt werden können durch

$$\begin{array}{lrl} (i) & P_i'\,(x_i) = \tilde{\varphi}\,(x_i)\,, & i = 1, \ldots, n-1 \\ (ii) & P_i'\,(x_{i+1}) = \tilde{\varphi}\,(x_{i+1})\,, & i = 1, \ldots, n-1 \\ (iii) & P_i\,(x_i) = P_{i-1}\,(x_i)\,, & i = 2, \ldots, n-1 \\ (iv) & P_i''\,(x_i) = P_{i-1}''\,(x_i)\,, & i = 2, \ldots, n-1. \end{array}$$

Zur Vervollständigung wird am linken Rand

$$P_1(x_1) = 0 = P_1''(x_1)$$

gefordert.
Durch die Bedingungen (i) bis (iv) ergeben sich die Freiheitsgrade der P_i für $i = 1$ zu

$$\begin{aligned} a_1 &= \frac{\tilde{\varphi}(x_2)}{3h^2}, \\ b_1 &= 0, \\ c_1 &= 0, \\ d_1 &= 0 \end{aligned}$$

und für $i = 2, \ldots, n-1$ zu

$$\begin{aligned} a_i =& \frac{1}{3h}\left(\frac{\tilde{\varphi}(x_{i+1}) - \tilde{\varphi}(x_i)}{h} - 2b_{i-1}\right) - 2a_{i-1}, \\ b_i =& 3a_{i-1}h + b_{i-1}, \\ c_i =& \tilde{\varphi}(x_i) \end{aligned}$$

sowie

$$d_i = ((a_{i-1}h + b_{i-1})\,h + c_{i-1})\,h + d_{i-1}.$$

Hierbei wurde bezüglich höherer Potenzen der Feinheit h mit dem Horner-Schema [Hor19] bereits eine Darstellung gewählt, die direkt für die Implementierung in Matlab, Java oder einer höheren Programmiersprache geeignet ist.
Man erhält, unter Erweiterung der Koeffizienten-Folge durch

$$a_n = b_n = c_n = 0, \quad d_n = ((a_{n-1}\ h + b_{n-1})\ \ h + c_{n-1})\ \ h + d_{n-1},$$

die Approximation der Stammfunktion des Wavelets $\tilde{\varphi}(x)$ mit Träger-Beschränkung zu

$$\tilde{F}(x) := \begin{cases} 0, & \text{für } x < x_1 \\ ((a_i(x - x_i) + b_i)(x - x_i) + c_i)(x - x_i) + d_i, & \text{für } x_i \leq x < x_{i+1}, \\ & \text{mit } i = 1, \ldots, n-1 \\ d_n, & \text{für } x \geq x_n \end{cases}.$$

Durch die angepassten Bedingungen (i) und (ii) wird klar, dass das hier beschriebene Vorgehen identisch ist mit der Interpolation des Wavelets $\tilde{\varphi}(x)$ mit Beschränkung des Trägers durch einen quadratischen Spline, dessen Stammfunktion analytisch korrekt durch

$F(x)$ gegeben ist. Durch diesen Zugang könnten zunächst Parallelen zu der numerischen Integration über die zusammengesetzte Simpson-Regel [Smy97] gezogen werden. Daher sei an dieser Stelle darauf hingewiesen, dass eine interpolierende quadratische Spline-Kurve über dem gesamten Definitionsbereich eine stetig differenzierbare Funktion ist, wohingegen bei der zusammengesetzten Simpson-Regel die abschnittsweise definierten quadratischen Interpolationspolynome im Allgemeinen nicht einmal zu einer stetigen Funktion führen.
Das vorgestellte Verfahren rundet die Beschreibung der Datentransformation ab und liefert zugleich einen konstruktiven Zugang. Eine Implemetierung des Verfahrens ist in OfeS-Tool hinterlegt. Mit den Ergebnissen dieses Kapitels werden die objektiven Anteile der einzelnen Datensätze um die entsprechenden Wavelet-Tableaus erweitert, so dass sie nun in einer Form vorliegen, in der komfortrelevante Anteile besser zugänglich sind.

4 Extrahierung komfortrelevanter Merkmale

Die Transformation der Messdaten in Kapitel 3 hatte zum Ziel, die enthaltenen Strukturen, die für den Schwingungskomfort relevant sind, sichtbar zu machen. Gleichzeitig bringt sie mit den Wavelet-Tableaus eine starke Erhöhung des Datenumfangs mit sich. Für die Vorhersage des Fahrkomforts über entsprechende Prädiktionsmodelle im weiteren Verlauf wird eine Reduzierung dieser gesamten Daten auf eine kleine Menge beschreibender Werte benötigt.
Der Prozess der Abbildung der transformierten Messdaten auf die Eingangsgrößen eines geeigneten Vorhersagemodells unterteilt sich in zwei wesentliche Schritte [Guy06]:

1. Merkmalsextrahierung
2. Merkmalsauswahl

Die Merkmalsextrahierung beschäftigt sich mit der Ermittlung von Kennzahlen als alternative Darstellung der Daten. Sie beschreiben in der Regel Eigenschaften der Daten, die charakteristisch für deren Beurteilung sind und in denen stark unterschiedlich zu beurteilende Daten entsprechend variieren. Sie stellen die potentiell geeigneten Eingänge des Modells auf der Datenebene dar.
Bei der Merkmalsauswahl wird aus der Fülle dieser Kennzahlen eine für das spezifische Modell geeignete Auswahl von Merkmalen ermittelt. Sie lässt sich auf einer höheren Abstraktions-Ebene beschreiben, auf der die Bindung an die Struktur des Vorhersagemodells stärker einzustufen ist als der Bezug zu der Struktur der zugrunde liegenden spezifischen Daten. Durch Sie wird die grundlegende Eignung der Merkmale für die Vorhersage auf Datenebene erweitert zu einer entsprechenden Eignung auf Modellebene. Die Merkmalsauswahl wird daher im Anschluss an die Beschreibung der Vorhersagemodelle in Kapitel 6 behandelt.

4.1 Tableau-Separierung

In Kapitel 3 wurde die individuelle Transformations-Vorschrift für die Signaldaten der einzelnen Messkanäle hergeleitet. Abbildung 4.1a zeigt beispielhaft die Zeitreihe der gemessenen Vertikalbeschleunigung der Sitzschiene vorne links bei Überquerung der mittleren positiven Kante für eine feste Fahrzeugvariante. Das Wavelet-Tableau als Ergebnis der Anwendung der Datentransformation auf die Zeitreihe ist in Abbildung 4.1b dargestellt.
Entsprechend zeigen die Abbildungen 4.2 und 4.3 jeweils die Zeitreihe und das zugehörige Wavelet-Tableau der Vertikalbeschleunigung der Sitzschiene vorne links bei Überquerung

der mittleren negativen Kante und der U-Rille für das gleiche Fahrzeug.
Um eine möglichst gute Vergleichbarkeit zu gewährleisten, wurde für die Abbildungen (4.1) bis (4.3) jeweils der gleiche Messkanal gewählt. Nichtsdestotrotz wurden die übrigen Messkanäle aus Tabelle 2.6 in gleicher Weise für die Untersuchungen berücksichtigt.

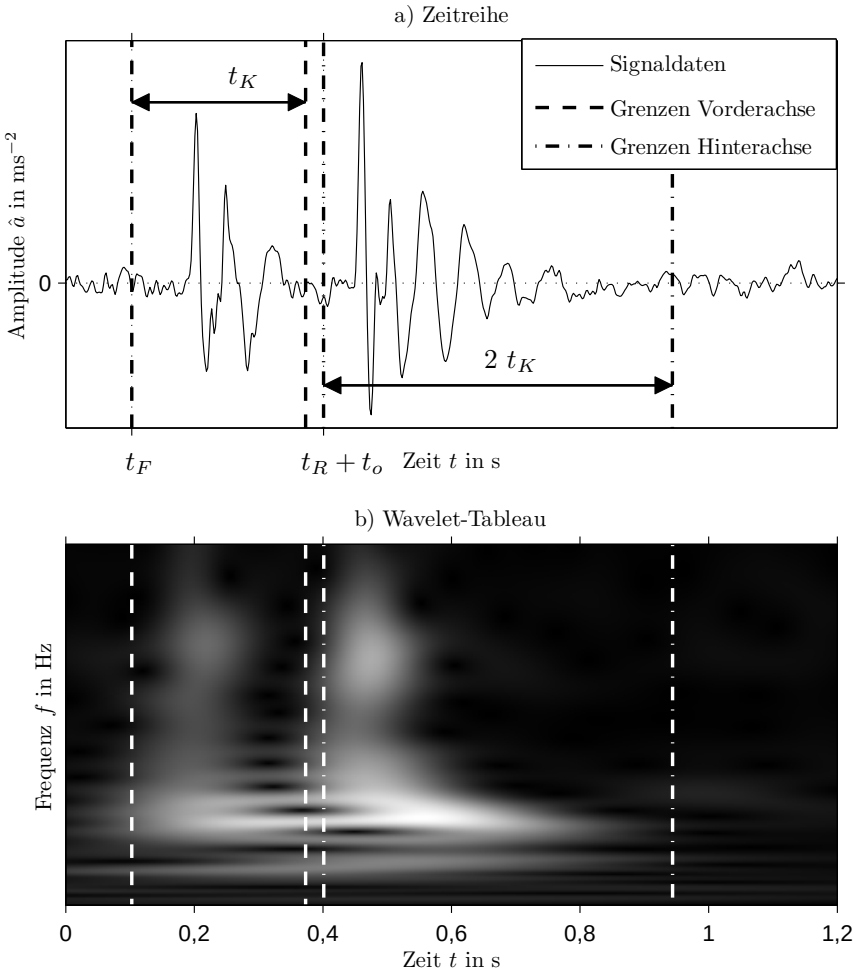

Abbildung 4.1: Zeitreihe und Wavelet-Tableau der Vertikalbeschleunigung der Sitzschiene vorne links bei Überquerung der mittleren positiven Kante zusammen mit den Grenzen für Vorder- und Hinterachse

Dank der zeitlichen Auflösung der Daten können die Anteile der Vorder- und der Hinterachsanregung im Tableau mit dem bloßen Auge grob unterschieden werden. Da der Fahrkomfort für die beiden Achsen differenziert bewertet wird, musste eine Methode ermittelt werden, die eine klare und für alle Daten konsistente Trennung der Signaldaten in Vorderachs- und Hinterachsanteil ermöglicht.

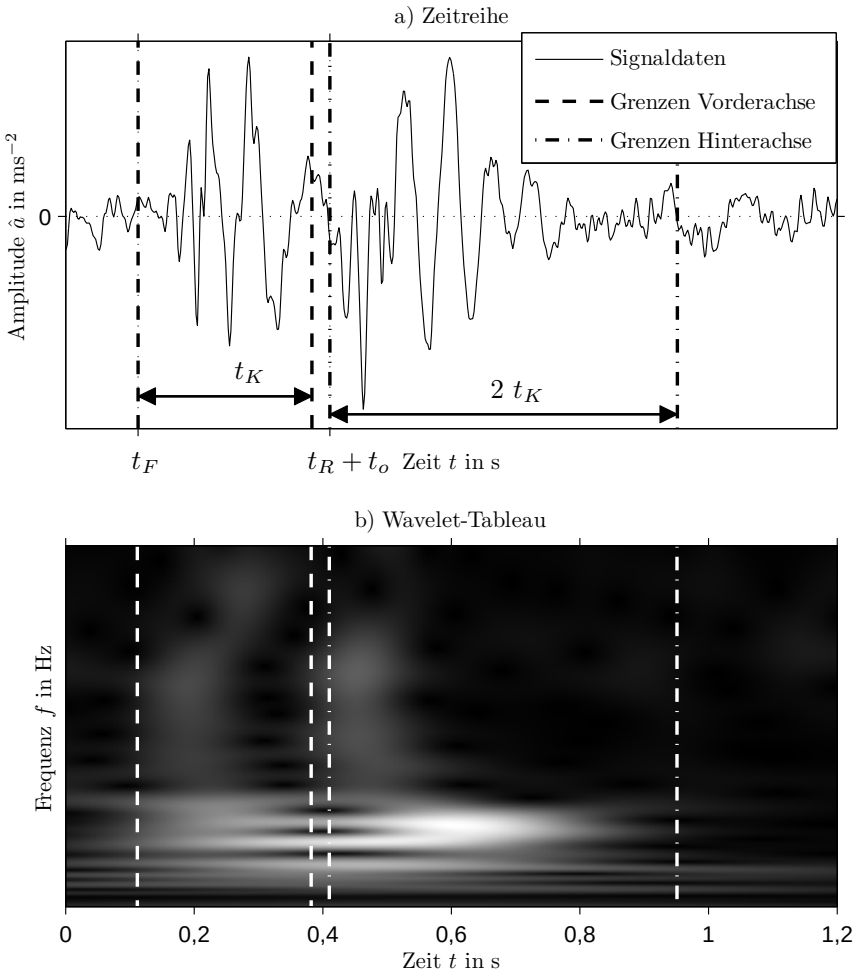

Abbildung 4.2: Zeitreihe und Wavelet-Tableau der Vertikalbeschleunigung der Sitzschiene vorne links bei Überquerung der mittleren negativen Kante zusammen mit den Grenzen für Vorder- und Hinterachse

Diese Methode nutzt den Umstand aus, dass die Signaldaten über alle Messkanäle zeitsynchron erfasst werden. In diesem Zusammenhang sind die Vertikalbeschleunigungen der beiden Radträger auf der Fahrerseite gut geeignet, um die jeweiligen Zeitpunkte zu ermitteln, an denen die entsprechende Achse das Hindernis überquert. Daher werden die Wavelet-Tableaus $W\left[a_F\right](f,t)$ und $W\left[a_R\right](f,t)$, die jeweils aus der Transformation der Vertikalbeschleunigung $a_F\left(t\right)$ des vorderen fahrerseitigen Radträgers und $a_R\left(t\right)$ des hinteren fahrerseitigen Radträgers entstehen, jeweils mit einem Speziellen Kern-Tableau

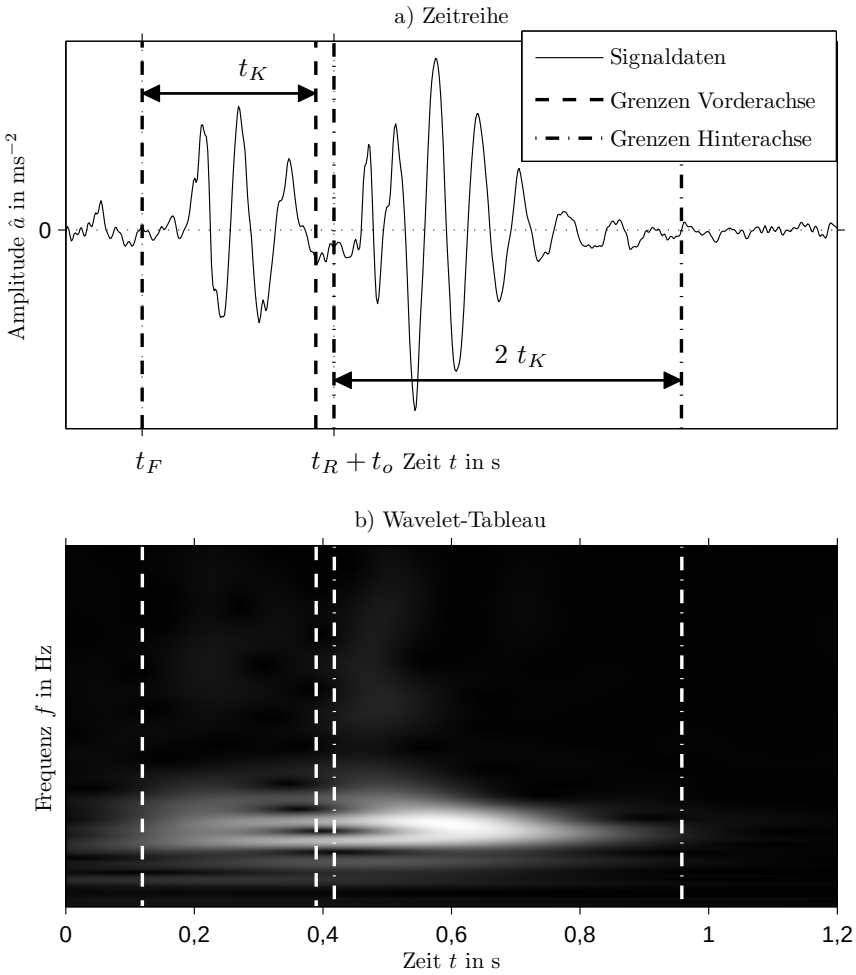

Abbildung 4.3: Zeitreihe und Wavelet-Tableau der Vertikalbeschleunigung der Sitzschiene vorne links bei Überquerung der U-Rille zusammen mit den Grenzen für Vorder- und Hinterachse

$K\,(f,t)$, dessen Träger im Zeitbereich das Intervall $[0,t_K]$ ist, gefaltet. Man erhält

$$F(t) = \int_0^{t_K} \int_{\mathbb{R}_+} W\,[a_F]\,(f,t+\tau)\;\; K\,(f,\tau)\,df\,d\tau$$

und

$$R(t) = \int_0^{t_K} \int_{\mathbb{R}_+} W\left[a_R\right](f,t+\tau)\ \ K\left(f,\tau\right) df d\tau$$

Der Zeitpunkt t_F, an dem $F\left(t\right)$ maximal wird, wird als Zeitpunkt der Überquerung angenommen. Das Wavelet-Tableau für die Vorderachse wird dann zeitlich durch t_F und $t_F{+}t_K$ begrenzt (vgl. Abbildung 4.1).
Für die Hinterachse wird analog der Zeitpunkt t_R herangezogen, für den $R\left(t\right)$ maximal wird. Außerdem ist ein optionaler Zeit-Offset t_o vorgesehen um Überschneidungen mit Inhalten zu minimieren, die noch der Vorderachsüberfahrt zuzuordnen sind. Darüber hinaus ist das Zeitfenster für die Hinterachse doppelt so groß wie das für die Vorderachse, um mögliche Nachschwing-Effekte berücksichtigen zu können. Insgesamt ist das Wavelet-Tableau der Hinterachse daher zeitlich begrenzt durch $t_R + t_o$ und $t_R + t_o + 2\ t_K$ (vgl. Abbildung 4.1).
Um den Zeit-Offset t_o geeignet zu bestimmen, werden Informationen über die Ausmaße des Bereichs des Wavelet-Tableaus benötigt, der durch einen einzelnen Signal-Wert beeinflusst wird. Wie bereits in Abschnitt 3.3 erwähnt, liegen im Fall der messtechnischen Erfassung des Signals a dessen Daten in diskreten Abtastwerten $a_0{,}a_1,\ldots,a_N$ mit einer festen Abtastperiode Δt vor. Die Wavelet-Transformation (3.2) kann in diesem Fall, analog zu (3.23), in der Form

$$\mathcal{W}\left[a\right](s,\tau) = \sum_{i=0}^{N} \frac{a_i}{\sqrt{s}} \int_{i\ \Delta t}^{(i+1)\ \Delta t} \varphi^*\left(\frac{t-\tau}{s}\right) dt$$

geschrieben werden, wobei o.B.d.A $t_1 = 0$ angenommen wird. Gemäß (3.25) lässt sich hieraus folgern, dass die absolute Beeinflussung des Tableau-Eintrages $W\left[a\right](f,t)$ zur Zeit t und zu der Frequenz f durch den Abtastwert a_i des Signals zur Zeit $i\ \Delta t$ nach oben beschränkt ist durch

$$\Delta W\left[a\right](f,t) = \left|a_i\right| \sqrt{\frac{f}{\lambda\gamma\sqrt{\pi}}} \left| \int_{i\ \Delta t}^{(i+1)\ \Delta t} \varphi^*\left(\frac{(\tau-t)\ f}{\lambda\gamma\sqrt{\pi}}\right) d\tau \right|.$$

Sei nun $t_F = i_F\ \Delta t$ und $t_F + t_K = (i_F + i_K + 1)\ \Delta t$. Dann ist der letzte Signalwert, der zu der Vorderachsüberfahrt gehört, gerade $a_{i_F+i_K}$. Für eine feste Frequenz f ist der Einfluss dieses Signalwerts auf das Wavelet-Tableau für Bereiche mit $t \geq z\ \Delta t$ nach oben beschränkt durch

$$\varepsilon\left(f,z\right) := \left|a_{i_F+i_K}\right| \sqrt{\frac{f}{\lambda\gamma\sqrt{\pi}}} \left| \int_{(i_F+i_K)\ \Delta t}^{(i_F+i_K+1)\ \Delta t} \varphi^*\left(\frac{(t-z\ \Delta t)\ f}{\lambda\gamma\sqrt{\pi}}\right) dt \right|. \tag{4.1}$$

Sind nun eine Frequenz f^* und eine obere Schranke $\varepsilon_{max} \geq \varepsilon\,(f^*,z)$ für den verbleibenden Einfluss vorgegeben, so kann über (4.1) der passende Zeit-Offset z bestimmt werden.
Um diesen Offset für unterschiedliche Datensätze beizubehalten muss der Wert $a_{i_F+i_K}$ in (4.1) durch eine passende obere Schranke für den Absolutwert des letzten, der Vorderachse zugehörigen, Signalwerts ersetzt werden.
Der resultierende Zeit-Offset t_o ergibt sich somit zu

$$t_\mathrm{o} = \max\left\{0,t_F + t_K - t_R + z\ \Delta t\right\}.$$

Schließlich bleibt anzumerken, dass die Zeitdifferenz zwischen t_R und t_F wesentlich von dem Radstand, der Ausgestaltung des Hindernisses und nicht zuletzt der Geschwindigkeit abhängt, mit dem das Hindernis überfahren wird. Daher kann der Bereich für ein zulässiges $\varepsilon_{max} \geq \varepsilon\,(f^*)$ stark eingeschränkt sein.
Die Wavelet-Tableaus der Radträgerbeschleunigungen werden über das Wavelet in (3.18) für $\gamma = 2$ berechnet. Dieser Wert bewahrte in allen untersuchten Fällen die benötigte Zeitauflösung und lieferte stets stabile Ergebnisse für die zeitliche Trennung in Vorderachs- und Hinterachsanteil.
Die bis hierhin beschriebenen Verarbeitungsschritte liefern eine detailliertere Darstellung der gemessenen Signaldaten, in der die den jeweils zwei Achsüberquerungen zugehörigen Anteile bereits getrennt sind. Diese Darstellung erleichtert die Identifizierung komfortrelevanter Informationen, die im Folgenden in Form von Merkmalen aus den getrennten Wavelet-Tableaus abgeleitet werden.

4.2 Verfahren der Tableau-Approximation

Die Forderungen an die Merkmale der objektiven Daten sind, dass sie in einer Anzahl benötigt werden, die deutlich unter der Anzahl der Wavelet-Tableau-Einträge der transformierten Signale liegt, dass sie die komfortrelevanten Teile der zugrunde liegenden Signaldaten hinreichend beschreiben und dass sie stabil sind gegen kleine Störungen in der zeitlichen Ausrichtung der Wavelet-Tableaus. Letztere könnten zum Beispiel durch kleine Abweichungen bei der Tableau-Separierung oder in der Geschwindigkeit auftreten, mit der das Hindernis im Experiment überfahren wurde.
Versuche zu Merkmalsextrahierungs-Methoden aus frühen Ansätzen, bei denen die Tableau-Daten über einem regulären vorgegebenen Gitter gemittelt wurden (vgl. [Fuh12]), erwiesen sich als nicht stabil genug gegen diese Art von Störungen. Andere klassische Methoden aus der digitalen Bildverarbeitung, wie etwa die Ermittlung invarianter Momente [Nix08], konnten die zugrunde liegenden Daten nicht mehr in angemessener Anzahl hinreichend beschreiben. Daher musste im Rahmen dieser Arbeit eine neue Merkmalsextrahierungs-Methode entwickelt werden, die auf den gegebenen Tableau-Daten basiert.
Da die Entwicklung des Signals in Zeit und Frequenz im Mittelpunkt des Interesses steht, sollte diese Methode die Möglichkeit bieten, Trends in den Tableau-Daten zu erkennen, ohne dass die zugehörige Position in Zeit und Frequenz explizit vorgegeben werden muss.

Um hingegen die Vergleichbarkeit der extrahierten Merkmale über Datensätze zu verschiedenen Experimenten hinweg zu erhalten, muss die relevante Region zeitlich auf irgendeine Weise definiert werden.
Die Lösung war in diesem Fall eine Approximation der Wavelet-Tableaus mit einem parametrierten Muster, wobei die geometrischen Informationen, die in Form der sich einstellenden Parameterwerte vorliegen, als Merkmale herangezogen werden.
Die Parameter des Musters sollten dessen Skalierung und Verschiebung sowie die Nachbildung von Trends in der Zeit-Frequenz-Ebene ermöglichen. Gleichzeitig sollte das Muster selber einfach genug sein um hindernisübergreifend für alle relevanten Wavelet-Tableaus gleichermaßen zuverlässig herangezogen werden zu können. Zusätzlich sollte die Übertragung von den Muster-Parametern auf die entsprechenden geometrischen Eigenschaften so direkt wie möglich sein.
Unter Berücksichtigung all dieser Zielkriterien fiel die Wahl für das parametrierte Muster auf eine angepasste Gaußsche Glockenkurve auf dem $\mathbb{R}^2$:

$$G\left(f,t\right) = a \cdot \exp\left[-\left(\frac{x\left(f,t\right)}{r_b}\right)^2 - \left(\frac{y\left(f,t\right)}{r_s}\right)^2\right] \tag{4.2}$$

Hierbei ist

$$\begin{bmatrix} x\left(f,t\right) \\ y\left(f,t\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\varphi\right) & \sin\left(\varphi\right) \\ -\sin\left(\varphi\right) & \cos\left(\varphi\right) \end{bmatrix} \cdot \begin{bmatrix} \left(t - m_t\right) \\ \left(f - m_f\right) \end{bmatrix}.$$

Die Niveaulinie für $G = \frac{a}{e}$ ist eine Ellipse im $\mathbb{R}^2$ mit den beiden Halbachsen r_b und r_s, die um den Winkel φ gedreht und um $\left(m_t,m_f\right)^{\mathrm{T}}$ aus dem Nullpunkt verschoben ist.
Das Muster ist in Abbildung 4.4 dargestellt und besitzt die folgenden sechs Freiheitsgrade:

(1) die Amplitude $p_1 = a$

(2) die Position des Zentrums auf der Zeitachse $p_2 = m_t$

(3) die Position des Zentrums auf der Frequenzachse
$p_3 = m_f$

(4) die Länge der größeren Halbachse $p_4 = r_b$

(5) die Länge der kürzeren Halbachse $p_5 = r_s \leq r_b$

(6) den skalierten Verdrehwinkel $p_6 = \frac{\varphi}{\pi}$ mit $\varphi \in [0,\pi) \subset \mathbb{R}$

Für $\varphi = 0$ verläuft die längere Halbachse parallel zu der Zeitachse und die kürzere Halbachse parallel zur Frequenzachse.
Durch die Symmetrie der Ellipse ist diese Definition des Musters gleichermaßen eindeutig und vollständig. Man erkennt, dass verschiedene geometrische Aspekte, wie die Entwicklung in Zeit und Frequenz (φ), die Position in Zeit und Frequenz $\left(m_t,m_f\right)$, die Ausmaße der im

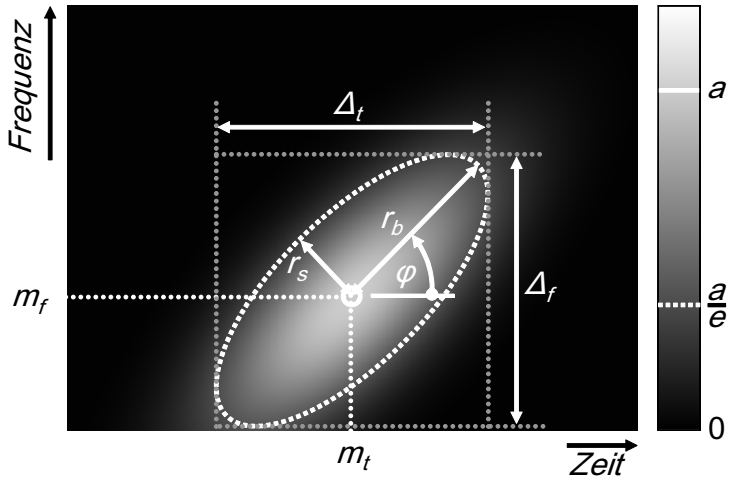

Abbildung 4.4: Höhenlinien-Darstellung des parametrierten Musters für die Tableau-Approximation

Wesentlichen okkupierten Zeit- und Frequenz-Intervalle (r_b und r_s in Kombination mit φ) sowie das Energie-Niveau des Stoßes selber (a), nicht nur in dem Muster enthalten sondern auch direkt mit den entsprechenden Muster-Parametern verknüpft sind.
Die Parameter $p_1, p_2, \ldots, p_6$ werden im Zuge der bestmöglichen Anpassung des Musters an ein entsprechendes Wavelet-Tableau $W[a](f,t)$ bestimmt. Hierzu wird das Integral der quadrierten Abweichungen über der Zeit-Frequenz-Ebene

$$E_I = \int_{\mathbb{R}_+} \int_{\mathbb{R}} \left(G(f,t) - W[a](f,t)\right)^2 dtdf \tag{4.3}$$

minimiert.
Da hier ein Maß für die absolute Abweichung minimiert wird, findet eine gute Approximation (und damit verbunden die Ableitung relevanter Information) nur in Bereichen von $\mathbb{R}_+ \times \mathbb{R}$ statt, in denen das Wavelet-Tableau große Beträge aufweist. Jedoch sind die Bereiche großer Waveletkoeffizienten-Beträge alleine nicht notwendigerweise auch die maßgeblich komfortrelevanten Datenanteile.
Um weitere Informationen über das Signal zu erhalten, wird die Merkmalsextrahierung um einen zweiten Approximations-Schritt erweitert, in dem die leicht variierte Zielfunktion

$$E_{II} = \int_{\mathbb{R}_+} \int_{\mathbb{R}} \left(\frac{G(f,t) - W[a](f,t)}{W[a](f,t)}\right)^2 dtdf \tag{4.4}$$

minimiert wird. Diese letzte der beiden Optimierungen liefert eher qualitative Informationen der Entwicklung in Zeit und Frequenz, da hier Bereiche absoluter Abweichungen, die durch generell kleinere Waveletkoeffizienten des Signals einen geringeren Pegel aufweisen, höher gewichtet werden. Um in (4.4) eine Division durch Null zu vermeinden, wird bereits bei der Ermittlung der Wavelet-Tableaus zu jedem Eintrag der positive Wert $w_{\min}$ addiert. In den vorliegenden Untersuchungen wurde $w_{\min} = 10^{-3}$ gewählt.
In beiden Fällen wird die jeweilige Zielfunktion durch die Implementierung von Verfahren der sequentiellen quadratischen Programmierung (SQP-Verfahren, [Bog95]) minimiert. Da SQP-Verfahren bekannt dafür sind, lokale Optima als Ergebnis zu liefern [Gei02], wird der am besten geeignete Startpunkt für die Optimierung im Vorfeld bestimmt. Dies wird durch entsprechende Auswertungen der Zielfunktion (4.3) bzw. (4.4) für mehrere Parameter-Tupel erreicht, die über einen 10^6 vollständig faktoriellen Plan [Mon12] ermittelt werden. Dabei werden für jeden der sechs Parameter zehn mögliche Werte berücksichtigt, die äquidistant in dem jeweils zulässigen Parameterbereich verteilt sind. Der Parametervektor, der unter allen auftretenden Kombinationen die herangezogene Zielfunktion minimiert, wird als Startpunkt für das SQP-Verfahren zur weiteren Verfeinerung verwendet.
Während der Optimierung wird die strenge eindeutige Definition des Musters zu einem gewissen Grad gelockert, da der Vorgang selber eine kontinuierliche Suche im Parameter-Raum beinhaltet, der selbst Teilmenge des $\mathbb{R}^6$ ist. Es wird daher während der Optimierung $p_6 \in \mathbb{R}$ und p_4,$p_5 \in [0,r_{max}] \subset \mathbb{R}$ ohne weitere Einschränkungen zugelassen, wobei r_{max} eine obere Schranke für beide Halbachse ist, die von der verwendeten Abtastungen und den Zeit- und Frequenzbereichen abhängt.
Im Anschluss an die Ermittlung der optimalen Parameter $(p_1, \ldots ,p_6)$ wird die Eindeutigkeit des zugehörigen Musters durch die folgenden Nachbearbeitungsschritte wieder hergestellt:

(i) falls $p_4 < p_5$, dann:
$p_h \to p_4$
$p_4 \to p_5$
$p_5 \to p_h$
$p_6 \to p_6 + 0{,}5$

(ii) bestimme $k \in \mathbb{Z}$ so, dass $p_6 + k \in [0,1)$

(iii) $p_6 \to p_6 + k$

Der erste Schritt stellt die Eindeutigkeits-Bedingung durch Drehung des Musters um $\pm\frac{\pi}{2}$ was durch Vertauschen von r_b und r_s kompensiert wird, wieder her. Die Schritte zwei und drei werden eingeführt, um sicher stellen zu können, dass $\varphi \in [0,\pi)$.
Bei der Anwendung auf existierende Daten sind diese als eine Wavelet-Tableau-Matrix $\mathbf{W} = (w_{i,j}) \in \mathbb{R}_+^{N_f \times N_t}$ hinterlegt, die N_f diskrete Frequenzen und N_t diskrete Zeitpositionen beinhaltet.

Die jeweilige Zielfunktion kann daher durch die diskreten Approximationen

$$E_I \approx \sum_{i=1}^{N_f} \sum_{j=1}^{N_t} (G_{i,j} - w_{i,j})^2 \tag{4.5}$$

von (4.3) und

$$E_{II} \approx \sum_{i=1}^{N_f} \sum_{j=1}^{N_t} \left(\frac{G_{i,j} - w_{i,j}}{w_{i,j}} \right)^2 \tag{4.6}$$

von (4.4) berechnet werden.
Dabei ist $G_{i,j}$ die numerische Manifestation des Musters (4.2), ausgewertet auf einem diskreten $N_f \times N_t$-Gitter des Gebietes $[0,1] \times [0,1] \subset \mathbb{R}^2$. Abbildung 4.5 zeigt die Approximationen des Wavelet-Tableaus aus Abbildung 4.1. In (a) und (b) ist die jeweils erste Approximation von Vorder- und Hinterachse dargestellt. Die Zweite Approximation von Vorder- und Hinterachse zeigen (c) und (d). Für die untersuchten Daten lässt sich allgemein angeben, dass die Kombination der Merkmale aus beiden Approximationen das zu Grunde liegende Wavelet-Tableau auf gut geeignete Weise zu repräsentieren vermag.
Mit Kenntnis des Parametersatzes für das Muster, dass der jeweils besten Approximation des Wavelet-Tableaus entspricht, können noch mehr Merkmale für die weitere Beschreibung der Daten hergeleitet werden:

- die projizierte zeitliche Dauer Δ_t der Ellipse
- die projizierte Bandbreite Δ_f der Ellipse
- die Residuen der Tableau-Approximation E_I und E_{II}
- der Sinus des Verdrehwinkels $\sin(\varphi)$
- der Cosinus des Verdrehwinkels $\cos(\varphi)$
- der Mittelwert der Einträge des Wavelet-Tableaus $\overline{\mathbf{W}}$

Die projizierte zeitliche Dauer bzw. Frequenz-Bandbreite werden aus der Projektion der Ellipse auf die Zeit- bzw. die Frequenzachse abgeleitet (vgl. Abbildung 4.4). Die Residuen sind die jeweiligen verbleibenden Restwerte in (4.5) und (4.6).
Tabelle 4.1 zeigt eine Zusammenfassung der mit dieser Methode extrahierten Merkmale für eine der Approximationen. Dieser erweiterte Merkmalsatz drückt nun den Frequenzbereich (Größe und Position), den Trend der zeitabhängigen Frequenzänderungen, den Spitzenwert und die Energie des Stoßes, seine Dauer und, unter Verwendung der Residuen in Kombination mit dem Mittelwert, die Komplexität der Zeit-Frequenz-Darstellung des Signals aus.
Trotz der Verfügbarkeit wird das Zeitlokalisierungs-Merkmal nicht für die alternative

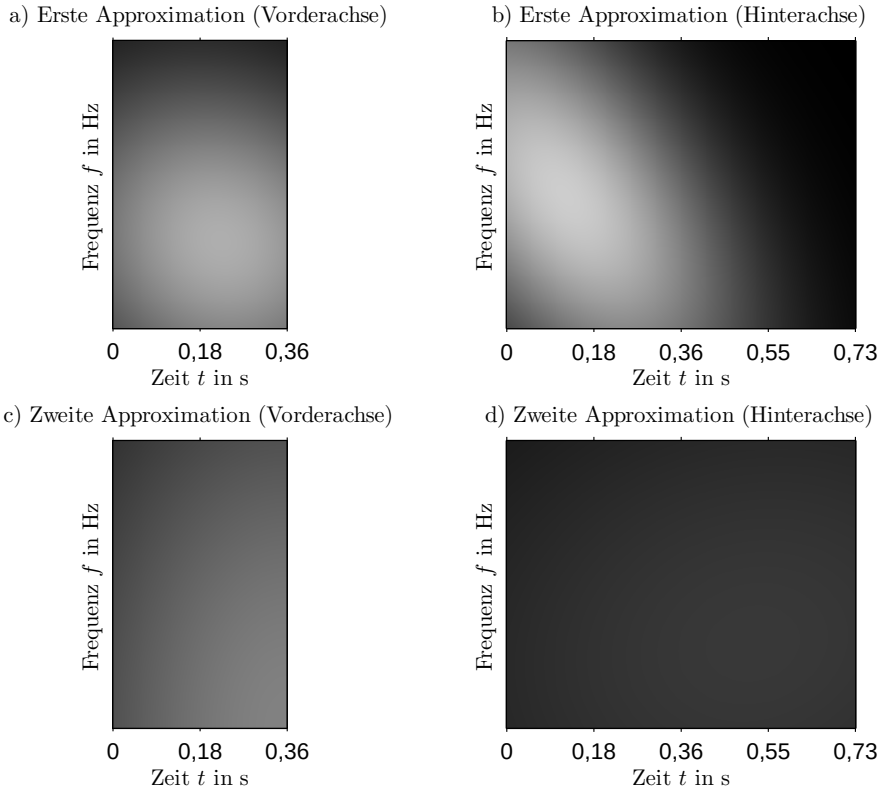

Abbildung 4.5: Erste und zweite Approximation des Wavelet-Tableaus aus Abbildung 4.1b für Vorder- und Hinterachse

Darstellung der Signaldaten verwendet, da es sehr empfindlich gegenüber kleinen Abweichungen bei der Tableauausrichtung und aus dem Tableau-Trennungs-Algorithmus ist. Bereits kleine Störungen können potentiell Streuungen in den Daten hervorrufen, die nicht notwendiger Weise durch komfortrelevante Änderungen an dem Schwingungs-Verhalten des Fahrzeugs hervorgerufen werden. Die übrigen Merkmale sind in diesem Kontext deutlich robuster und somit nicht von dem Problem der zeitlichen Ausrichtung betroffen.
In diesem Kapitel wurde eine Methode vorgestellt, die, ausgehend von den Wavelet-Tableaus der transformierten Signaldaten, die Ableitung stabiler, beschreibender Merkmale gestattet. Diese Merkmale stellen die letzte Stufe in der Verarbeitungskette der objektiven Messdaten dar. Die vorgestellte Methode ist fest in OfeS-Tool integriert. Für die Anwendung des SQP-Verfahrens im Rahmen der Tableau-Approximation wird dabei auf die *Optimization-Toolbox* von Matlab zurückgegriffen.

Nr.	**Merkmal**	**Beschreibung**
1	a	Amplitude
2	m_t	Zeitlokalisierung
3	m_f	Frequenzlokalisierung
4	r_b	Länge größere Halbachse
5	r_s	Länge kürzere Halbachse
6	φ	Verdrehwinkel
7	Δ_t	projizierte zeitliche Dauer
8	Δ_f	projizierte Bandbreite
9	E_I (E_{II})	Residuen erste (zweite) Approximation
10	$\sin(\varphi)$	Sinus des Verdrehwinkels
11	$\cos(\varphi)$	Cosinus des Verdrehwinkels
12	$\overline{\mathbf{W}}$	Mittelwert der Tableau-Einträge

Tabelle 4.1: Auflistung der extrahierten Merkmale zu einer Approximation und einer Achse

5 Prädiktionsmodelle

Prädiktions- oder Vorhersagemodelle nehmen eine große Stellung in der statistischen Datenanalyse [Ols08], [Bac10], der Parameteridentifikation [Jan04] sowie vielen weiteren Feldern der Natur-, Ingenieurs- und Wirtschaftswissenschaften ein. Sie dienen allgemein der Vorhersage einer oder mehrerer Ausgangsgrößen in Abhängigkeit eines Systemzustands, der durch eine feste Anzahl von Eingangsgrößen beschrieben wird. Das Modell selber verfügt neben den nötigen Schnittstellen nach außen hin über einen Satz von Parametern, die das Modellverhalten bestimmen.
Aus mathematischer Sicht handelt es sich allgemein um eine Abbildung

$$\begin{aligned} M: \quad & X \times P \to Y \\ & (\mathbf{x},\mathbf{p}) \mapsto y = M(\mathbf{x},\mathbf{p}), \end{aligned} \tag{5.1}$$

die für einen jeweils festen Parametersatz $\mathbf{p}$ aus dem Parameterraum P einen durch $\mathbf{x}$ repräsentierten Zustand auf die zugehörige Ausgangsgröße y abbildet. X ist die Menge der möglichen Zustands-Beschreibungen, P die Menge der möglichen Parameter-Konfigurationen und Y die Menge der Zustände, die die Ausgangsgröße annehmen kann.
Die Ermittlung der Parameter erfolgt indirekt. Hierzu werden Daten für Systemzustände, und in der Regel auch für zugehörige Soll-Ausgaben, herangezogen. Man kann diese Daten als Beobachtungen des gewünschten Modellverhaltens ansehen, sofern sie Eingangs- und Ausgangsdaten enthalten.
In der Praxis werden die Mengen X, Y und P als endlich-dimensionale $\mathbb{R}$-Vektorräume vorliegen. Dabei wird Y üblicherweise nur skalare Werte enthalten, auch wenn höherdimensionale Ausgaben im Sinne der Definition (5.1) durchaus möglich wären.
Modelle, die sich auf skalare Ausgabegröße beschränken, werden auch als *MISO*-Systeme bezeichnet (eng.: multiple input single output). Demgegenüber stehen die *MIMO*-Systeme (eng.: multiple input multiple output), die mehrere Ausgabegrößen parallel schätzen. Die Beschränkung auf eine skalare Modellausgabe hat den Hintergrund, dass bereits die Schätzung oder Vorhersage einer einzelnen Ausgangsgröße auf Basis eines durch wenige Vorgaben angeeigneten Wissens über das gewünschte Modellverhalten nicht unproblematisch ist [Cha07]. Muss im konkreten Fall mehr als eine einzige skalare Ausgabegröße vorhergesagt werden, so ist es oftmals praktikabler, für jede dieser Größen ein individuelles Modell zu erstellen. In dieser Arbeit werden nur Modelle mit skalarer Ausgabe betrachtet.
Alle Modelle, die für die Berechnung der Ausgabe $y \in Y$ in Abhängigkeit des Zustands $\mathbf{x} \in X$ auf die Vorschrift $M(\mathbf{x},\mathbf{p})$ zurückgreifen, unterscheiden sich einzig in dem Parame-

tersatz $\mathbf{p} \in P$. Somit induziert umgekehrt ein Parametersatz $\mathbf{p}$ eine Modellausprägung

$$\begin{aligned} M_{\mathbf{p}}: \quad & X \to Y \\ & \mathbf{x} \mapsto y = M_{\mathbf{p}}(\mathbf{x}) = M(\mathbf{x},\mathbf{p}). \end{aligned} \tag{5.2}$$

5.1 Unterteilung und Beispiele

Vorhersagemodelle werden auf der allgemeinsten Ebene in zwei Gruppen unterteilt, die sich in dem Raum Y der Ausgabegröße y unterscheiden [Loh].
Enthält Y nur endlich viele Ausprägungen y, so übernimmt das zugrunde liegende Modell eine Klassifizierungsaufgabe. Hierbei wird einem durch $\mathbf{x}$ aus X beschriebenem Objekt eine Klassenzugehörigkeit y aus Y, auch Klassenlabel genannt, zugeordnet. Bei skalaren Größen wird der Raum der Ausgabegröße auch als Skala bezeichnet. Diese kann metrisch oder nicht-metrisch sein, wobei diese beiden Fälle noch weiter charaktisiert werden können [Bac10]. Häufig ist Y eine zusammenhängende Teilmenge $Y_{\mathbb{Z}} \subset \mathbb{Z}$ der ganzen Zahlen.
Prominente Beispiele für Klassifikations-Modelle sind Naive Bayes Klassifikatoren [Dud73],[Lan92], Support Vector Machines [Vap99],[Sch01] und binäre Entscheidungsdiagramme [Ake78],[hC09]. Auch die selbstorganisierenden Karten nach Kohonen lassen sich als Klassifikationsmodelle einsetzen [Koh82],[Tav90]. Allerdings unterscheiden sie sich hinsichtlich der Ermittlung des Parametersatzes deutlich von den üblichen Verfahren.
Vor dem Hintergrund der Klassifikation werden die in $\mathbf{x}$ hinterlegten Daten ebenfalls als Merkmale (eng.: Features) bezeichnet (vgl. Kapitel 4), da sie die zu klassifizierenden Objekte beschreiben.
Modelle, bei denen die Ausgabe y kontinuierliche Werte annimmt, werden allgemein als Regressionsmodelle bezeichnet. Auch die Bezeichnung eines kontinuierlichen Klassifikators ist zu Weilen anzutreffen [Plu04]. Beispiele für Regressionsmodelle sind lineare Regressionsmodelle (vgl. Abschnitt 5.4) und künstliche neuronale Netze mit Feed-Forward-Struktur (vgl. Abschnitt 5.5).
Analog zu dem vorherigen Klassifikations-Fall lassen sich auch hier die als Eingangsgrößen hinterlegten Daten als Merkmale interpretieren, die Objekte oder Zustände beschreiben. Das Regressionsmodell wiederum kann als Bewertung des umschriebenen Objektes oder Zustandes auf einer kontinuierlichen Skala angesehen werden und eignet sich somit eher zur Aufgabe der Komfortbewertung als (diskrete) Klassifikatoren.

5.2 Grundbegriffe

Im Folgenden werden für den weiteren Verlauf wichtige Begriffe des Konzeptes der Vorhersagemodelle vorgestellt und erläutert.

5.2.1 Training

Wie eingangs beschrieben basiert das in (5.2) durch den Parametersatz $\mathbf{p}$ induzierte Modellverhalten in der Regel auf Daten, die aus endlich vielen Beobachtungen des Modellverhaltens in der Vergangenheit stammen. Durch einen Vorgang, den man Training oder Lernen nennt, soll dieses beobachtete Verhalten in dem Parametersatz $\mathbf{p}$ manifestiert werden [Kot07]. Die hierfür zur Verfügung stehenden Daten werden üblicherweise als Trainingsdaten bezeichnet. Das Anlernen eines Vorhersagemodells wird in der Literatur unterteilt in die Formen des überwachten (eng.: supervised) und des unüberwachten (eng.: unsupervised) Lernens [Gue11],[Sat13].
Vor dem Hintergrund der Vorhersagemodelle kommt dem überwachten Lernen eine übergeordnete Rolle zu [Car06]. Hier bestehen die einzelnen Trainingsdatensätze $\mathbf{d}^j$ jeweils aus einer Zustandsbeschreibung $\mathbf{x}^j \in X$ und dem zugehörigen Sollwert $y^j \in Y$. Das Training kann bei Verwendung von n solchen Trainingsdatensätze als Abbildung

$$\begin{aligned} \tau : \quad & (X \times Y)^n \times \mathcal{M} \times P \to P \\ & (\mathbf{D}, M, \mathbf{p_0}) \mapsto \mathbf{p} = \tau\,(\mathbf{D}, M, \mathbf{p_0}) \end{aligned}$$

aufgefasst werden, die ein Tupel $\mathbf{D}$ von n Trainingsdatensätzen entsprechend einer vorgegebenen Modellvorschrift $M \in \mathcal{M}$ auf den zugehörigen Parametersatz $\mathbf{p} \in P$ abbildet.
Hierbei entspricht

$$\mathcal{M} := \{M : \; X \times P \to Y\}$$

der Menge aller Modellvorschriften bezüglich X, Y und P.
Viele Trainingsverfahren benötigen dabei eine Startkonfiguration $\mathbf{p_0}$ der Modellparameter, die zu Beginn des Trainings vorgegeben werden muss und im Zuge des Trainings verfeinert wird [Sut13],[Ngu90]. In den wenigen Fällen, in denen bereits aus den Trainingsdaten und der Modellvorschrift die geschlossene Berechnung des Parametersatzes $\mathbf{p}$ wohldefiniert ist, kann auf die Vorgabe einer Startkonfiguration verzichtet werden.
Aus (5.2) folgt somit, dass bereits die Wahl einer Modellvorschrift $M \in \mathcal{M}$, einer Trainingsvorschrift τ, zu verwendender Trainingsdaten $\mathbf{D} \in (X \times Y)^n$ sowie einer beliebigen Startkonfiguration $\mathbf{p_0} \in P$ genügt, um eine vollständige Modellausprägung $M_{\mathbf{p}}$ mit $\mathbf{p} = \tau\,(\mathbf{D}, M, \mathbf{p_0})$ zu erzeugen.
Das unüberwachte Lernen unterscheidet sich insofern von dem überwachten Lernen, dass die Trainingsdaten keine Sollwerte enthalten, sondern lediglich eine Reihe von Zustandsbeschreibungen. Es wird bei Modellen angewandt, die eigenständig Muster und Strukturen in Datenmengen aufzudecken versuchen [Par13]. Da nur Eingangsdaten und keine Sollwerte benötigt werden, unterscheidet sich diese Form des Lernens auch in Hinblick auf die Verfügbarkeit von Trainingsdaten grundlegend von dem überwachten Fall. Sie wird daher nur speziell im Zusammenhang mit den selbstorganisierenden Karten nach Kohonen in Abschnitt 5.5.5 näher beleuchtet, spielt aber ansonsten für die hier betrachteten Vorhersagemodelle keine Rolle.

Im englischsprachigen Raum wird das Feld der Vorhersagemodelle mit *machine learning* bezeichnet [Bey08] und die Modelle selber als *learning machines*. Vor dem Hintergrund dieser Bezeichnung kann die Anzahl der freien Parameter eines Modells mit seiner Speicherkapazität und damit der Komplexität des abgebildeten Verhaltens identifiziert werden. In der Regel basiert das nachzubildende Verhalten auf wenigen beobachteten und natürlichen Vorgängen. Die diesen Vorgängen zugrunde liegenden Prinzipien dagegen sind in ihrer Gesamtheit sehr viel komplexer als es jede praktisch verfügbare Kombination aus Modellvorschrift und Zustandsbeschreibung ausdrücken könnte. Es geht also, wie so oft im Bereich der Modellbildung, darum, nur einen beobachteten Teilaspekt des Verhaltens durch ein Modell zu erklären. Das Training zielt daher auch nicht darauf ab, für jeden möglichen Zustand exakt die vorgegebene Sollausgabe zu erzeugen, sondern lässt individuelle Abweichungen zu Gunsten einer möglichst geringen Gesamtabweichung der Ausgaben von den Sollwerten zu. Die Minimierung einer solchen Bewertungszahl kann als Design-Kriterium eines Trainingsverfahrens τ angesehen werden. Aufgabe der Bewertungszahl ist es dabei, die freien Parameter mit den Trainingsdaten auf eine Weise in Beziehung zu setzen, die die qualitativen Ansprüche an die Modellausgaben hinreichend zu quantifizieren vermag. Die in diesem Zusammenhang am häufigsten verwendete Bewertungszahl folgt aus der mittleren Fehlerquadratsumme (eng. mean square error) und wird entsprechend definiert durch:

$$E_{\mathrm{MSE}} = \frac{1}{n} \sum_{j=1}^{n} \left[y^j - M\left(\mathbf{x}^j, \mathbf{p}\right) \right]^2 \tag{5.3}$$

Ein Vorteil dieser Bewertungszahl ist der, dass sie in $\mathbf{p}$ so oft stetig differenzierbar ist wie $M(\mathbf{x},\mathbf{p})$ selber. Bezüglich der Ermittlung geeigneter Parameter-Tupel eröffnet dies den Weg für Verfahren, die auf Gradienteninformationen der zu minimierenden Bewertungszahl aufsetzen. Das Kriterium für einen Parameter-Tupel $\mathbf{p}$ die Bewertungszahl (5.3) zu minimieren, wird im Folgenden auch als MSE-Kriterium bezeichnet.
Das MSE Kriterium (5.3) legt in seiner Darstellung nahe, dass die Anzahl der Trainingsdaten die Anzahl der unbekannten Modellparameter übersschreiten sollte. Um wieviel dabei die Anzahl der Datensätze größer zu sein hat als die Anzahl der freien Parameter lässt sich nicht allgemein festsetzen. Als ein möglicher Richtwert kursiert in der Literatur der Faktor zehn [Bur91].

5.2.2 Überanpassung

Die Idee des überwachten Trainings basiert auf den Annahmen, dass das zu approximierende Modellverhalten aus den vorliegenden Trainingsdaten (beobachtetes Modellverhalten) abgeleitet werden kann, und dass die verwendete Modellvorschrift für die Approximation dieses Verhaltens geeignet ist.
Die Bestimmung der freien Parameter eines Vorhersagemodells findet in der Regel nur bezüglich einer kleinen Auswahl $\mathbf{D} \in (X \times Y)^n$ aller möglichen Eingangs-Ausgangs-Daten-

Paare statt. Die Formulierung des MSE-Kriteriums sieht überdies nur vor, dass die verwendeten Trainingsdaten so gut wie möglich bezüglich der mittleren Fehlerquadratsumme wiedergegeben werden. Eine Bedingung, dass das Modell ebenfalls mit hoher Güte die Daten außerhalb des Trainingsbereiches schätzt, ist hier nicht enthalten. Es wird stattdessen erwartet, dass die Trainingsdaten $\mathbf{D}$ eine geeignete Repräsentation der allgemein mit dem Modell zu beurteilenden Daten darstellen, so dass die Generalisierungsfähigkeit des trainierten Modells hinreichend gut ist.
Dies ist selbstverständlich dann der Fall, wenn der gesamte Eingangsdaten-Raum X endlich ist und vollständig in Form von Trainingsdaten zur Verfügung steht. Eine weitaus gängigere Situation wäre ein unendlicher aber beschränkter Raum, innerhalb dessen die Trainingsdaten beliebig bis auf deren Anzahl gewählt werden können. Die Erhebung repräsentativer Trainingsdatensätze kann dann unter Zuhilfenahme von Methoden der statistischen Versuchsplanung (eng.: design of experiment) erfolgen [Mon12].
Ein weitaus ungünstigerer Fall ist der, dass die Verteilung der Trainingsdaten kaum oder überhaupt nicht beeinflusst werden kann und die Daten als Ergebnis abgeschlossener Messungen zur Verfügung stehen. Hier hängt die bestmögliche Qualität eines zu erstellenden Vorhersagemodells in erster Ordnung davon ab, wie die ermittelten Ein- und Ausgangsdaten verteilt sind.
Gerade bei experimentell zu ermittelnden Daten kann ohne Kenntnis des zu schätzenden funktionalen Zusammenhangs kaum eine gezielte, den Anforderungen genügende, Verteilung der anschließend verfügbaren Daten in die Planung der Versuche eingehen. Häufig ist es auch einfach nicht möglich, die für eine gewünschte Ausgabe erforderliche Eingabe in einem Experiment zu erzwingen.
Ein weiterer großer Nachteil bei der experimentellen Ermittlung der Trainingsdaten ist der, dass aufgrund des hohen Aufwands häufig nur wenige Daten erfasst werden können. Ist die verfügbare Trainingsdatenmenge sehr klein, so muss als erstes die Komplexität des Modells herabgesetzt werden, da weniger freie Parameter fixiert werden können. Ist aber der zu schätzende funktionale Zusammenhang von sehr komplexer Natur, so ist dies bereits eine große Einschränkung für die bestmöglich zu erwartenden Ergebnisse.
Es ist also von Interesse, mit möglichst wenig Daten noch möglichst viele freie Parameter so zu fixieren, dass das resultierende Modell hinreichend generalisierungsfähig ist. Dies ist eine schmale Gratwanderung, da man hierbei sehr leicht eine sogenannte Überanpassung (eng.: overfitting) des Modells an die Trainingsdaten provoziert [Haw04].
Die Überanpassung wurde bereits in Abschnitt 3.2 kurz angerissen. Sie ist ein generelles Problem bei der Parameterschätzung und kann als der Preis angesehen werden, den man dafür zu zahlen hat, dass man dem Modell für die Forderung nach einem Parameter-Tupel $\mathbf{p}$, welches des MSE-Kriterium zu erfüllen hat, nur eine sehr kleine oder schlecht verteilte Trainingsdatenmenge zur Verfügung stellt. Geht man bei der Betrachtung an Stelle des Modells von den erhobenen Daten aus, so handelt es sich bei der Überanpassung analog um die Konsequenz einer unzureichenden Modellwahl.
Das eigentliche Problem der Überanpassung besteht nicht darin, dass das Modell gewisse Eigenschaften des zu Grunde liegenden funktionalen Zusammenhangs nicht lernen kann.

Wenn darüber keine Informationen in den Trainingsdaten enthalten sind, ist dies einfach nicht möglich. Ferner ist die Problematik, dass bei der Verwendung des MSE-Kriteriums die Trainingsdaten eine derart strenge Vorgabe darstellen, dass schlecht verteilte oder fehlende Daten zu einer Anpassung des Modells führen, die für die kleine bekannte Datenauswahl zu einem kleineren Fehler, im Allgemeinen aber zu einem falschen Verhalten führt. Das Modell ist also an die Trainingsdaten übermäßig angepasst und somit weniger generalisierungsfähig.
Als Analogie stelle man sich den Entwurf eines Gesetzes in einem Staat vor. Werden die neuen Bestimmungen so formuliert, dass sie bezüglich einer kleinen, fest gewählten Bevölkerungsgruppe zu einer maximalen Akzeptanz führen, so ist die Gefahr groß, dass das Gesetz bezogen auf die Gesamtbevölkerung auf erhebliche Unzufriedenheit stößt. Diese Gefahr nimmt sowohl für eine kleiner werdende berücksichtigte Bevölkerungsgruppe als auch für eine zunehmende Homogenität innerhalb dieser Gruppe zu. Werden in der Gruppe beispielsweise nur Männer, nur Arbeitgeber oder nur Hochschulmitarbeiter berücksichtigt, so ist diese Gruppe für die Gesamtbevölkerung nicht repräsentativ.
Es wird an diesem Beispiel klar, dass ein fester Modellansatz stets einen Kompromiss darstellt, so wie auch ein Gesetz nie von jedem Bevölkerungsmitglied zur vollsten Zufriedenheit akzeptiert werden wird. Daher besteht bei obiger Veranschaulichung das Problem vor allem in dem Punkt, dass die zu formulierenden Bestimmungen von einer möglicherweise nicht hinreichend repräsentativen Bevölkerungsgruppe bestmöglich akzeptiert werden sollen. Die auf die Gesamtbevölkerung bezogene bestmögliche Formulierung führt möglicherweise bezogen auf die betrachtete Bevölkerungsgruppe zu einer eher geringen Akzeptanz, und könnte somit durch das genannte Entwurfsziel niemals gefunden werden.
Ein großes Problem überangepasster Modelle besteht darin, dass die ihnen innewohnenden falschen Zusammenhänge erst dann festgestellt werden können, wenn weitere Eingangs-Ausgangs-Daten-Paare zur Verfügung stehen. Würden diese unmittelbar zur Verfügung stehen, so hätte man sie auch für das Training verwenden können. Wird ein überangepasstes Modell jenseits der Trainingsdaten auf Eingangsdaten angewendet, so werden Schätzwerte akzeptiert, die im Allgemeinen stark von den unbekannten Sollwerten abweichen können. Da dies ohne unmittelbare Kontrollmöglichkeiten in der Regel erst sehr spät erkannt werden kann, können die Folgen je nach Einsatzgebiet verheerend sein.

5.2.3 Data-Division

Um der Überanpassung entgegen zu wirken, werden verschiedene Maßnahmen getroffen. Die einfachste und am nächsten liegende ist die sogenannte *Data-Division*. Damit ist zunächst eine Aufteilung der für die Parameterfixierung verfügbaren Daten in Trainingsdaten und Testdaten gemeint. Die Testdaten dienen einer stichprobenartigen Überprüfung der Generalisierungsfähigkeit im Anschluss an die Modellerstellung und werden während der Trainingsphase zurückgehalten.
Führt die Anwendung des Modells auf die Testdaten zu schlechten Ergebnissen bezüglich geeigneter Gütekriterien (vgl. Abschnitt 5.3), so kann das schlechte Modell als solches

identifiziert und verworfen werden. Tritt dieses Verhalten bei Variation der Verteilung der Trainings- und Testdaten sehr häufig auf, können weitere Schlussfolgerungen bezüglich der bestmöglichen Modellerstellung mit den verfügbaren Daten gezogen werden.
Dies stellt in einer gewissen Hinsicht eine Verbesserung gegenüber der Situation dar, in der keine Aufteilung der Daten in Trainings- und Testdaten erfolgt. Allerdings kann aus einer erfolgreichen Anwendung des trainierten Modells auf die Testdaten nicht automatisch der Schluss gezogen werden, dass das Modell den zu Grunde liegenden Zusammenhang hinreichend gut erlernt haben muss. Da sowohl die Verteilung der Trainings- als auch die der Testdaten schlecht gewählt sein kann, kann dies lediglich darauf hindeuten.
Durch das Zurückhalten bekannter Daten für eine Überprüfung der Modelle wird das eigentliche Problem der Überanpassung noch nicht gelöst. Es wird hier lediglich eine Möglichkeit geboten, eine Überanpassung bis zu einem gewissen Grad identifizieren zu können. Dass sie tatsächlich ausgeblieben ist, wenn sie sich nicht in der an das Training angrenzenden Probe niederschlägt, kann nicht mit Sicherheit gefolgert werden. Um einer Überanpassung tatsächlich entgegen zu wirken, sind weitere Maßnahmen nötig.
In einigen Trainingsverfahren sind zur internen Steuerung in Zwischenschritten ebenfalls Überprüfungen der Generalisierungsfähigkeit vorgesehen [Ama97]. Die Ergebnisse dieser Überprüfungen können helfen, Tendenzen zur Überanpassung frühzeitig zu erkennen. Der weitere Verlauf des Trainings hängt dann von der Reaktion auf diese Ergebnisse ab. Möglich wäre bei einem iterativen Trainingsverfahren der frühzeitige Abbruch (eng.: early stopping) [Sar95]. Alternativ kann die Überprüfung zur Bestimmung weiterer Steuerparameter wie dem Regularisierungsparameter (vgl. Abschnitt 5.2.4) dienen.
Die für diese internen Validierungen herangezogenen Daten beeinflussen in jedem Fall den Ausgang des Trainings und können somit nicht mehr als vollwertige Testdaten verwendet werden. Da sie aber auch nicht notwendiger Weise direkt in die Anpassung der freien Modellparameter eingehen, fallen sie streng genommen auch aus der Rolle der reinen Trainingsdaten heraus. Sie werden daher als Validierungsdaten bezeichnet.
In vielen Fällen werden diese Daten nicht gesondert bereitgestellt. Stattdessen findet innerhalb des Trainings zwischenzeitlich eine entsprechende vorübergehende Abspaltung einiger Trainingsdaten zum Zwecke der Validierung statt. Die temporären Partitionierungen können durchaus variieren und mehrfach wiederholt werden. Diese Methode kommt beispielsweise bei der K-Kreuzvalidierung (vgl. Abschnitt 5.2.5) zum Einsatz.
Zusammenfassend ist Data-Division eine sinnvolle Maßnahme, die bei jeder Parameterschätzung in Erwägung gezogen werden sollte. Allerdings muss hierbei berücksichtigt werden, dass durch die Reservierung von Teilen der insgesamt verfügbaren bekannten Daten die Anzahl der tatsächlichen Trainingsdaten deutlich verringert wird. Bei entsprechender Knappheit muss damit auch die Anzahl der freien Parameter, und damit die Komplexität des Modells, weiter nach unten korrigiert werden. Dies betrifft, vorweg genommen, in der Regel auch die Anzahl der Eingangsgrößen, die in dem Modell berücksichtigt werden können.

5.2.4 Regularisierung

In Abschnitt 5.2.3 wurde bereits darauf hingewiesen, dass das eigentliche Problem der Überanpassung darin besteht, dass von dem Modell falsche Zusammenhänge erlernt werden, die möglicherweise bezüglich der Trainingsdaten nur eine geringe Verbesserung darstellen. Die Folgen sind große Fehler bei der Eingabe von Daten, die dem Modell aus dem Training nicht bekannt sind. Verringert man nun insgesamt den Einfluss der Eingangsgrößen auf die Ausgangsgröße, so werden entsprechend auch die Fehler für unbekannte Daten im Verhältnis zu den Fehlern für bekannte Daten kleiner werden. Man mache sich dies an dem extremen Fall klar, bei dem die Modellausgabe völlig unabhängig von Eingabegrößen ist. Dann ist das Modell statisch und besteht nur aus einem skalaren Parameter p. Aus (5.3) folgt:

$$\begin{aligned} & -\frac{2}{n}\sum_{j=1}^{n}\left[y^j - p\right] = 0 \\ \Leftrightarrow \quad & \sum_{j=1}^{n} y^j - \sum_{j=1}^{n} p = 0 \\ \Longrightarrow \quad & p = \frac{1}{n}\sum_{j=1}^{n} y^j \end{aligned}$$

Der Parameter p des statischen Modells ist also der Mittelwert der beobachteten Ausgangsgrößen und lässt sich als Schätzung des statischen Anteils der Ausgangsgröße interpretieren. Man kann jede Ausgabe eines Modells als die Summe eines statischen und eines dynamischen Anteils ansehen. Der statische Anteil ist die Modellausgabe im Fall $\mathbf{x} = \mathbf{0} \in \mathbb{R}^m$. Der dynamische Anteil entsteht durch die Verarbeitung der Eingangsgrößen unter Verwendung weiterer Modellparameter.

Die Abweichungen der Modellausgaben von den Sollwerten entstehen zum einen durch schlecht angepasste Parameter, zum anderen durch Einflüsse, die von dem Modell ignoriert werden sollen. Beispiele hierfür sind Messfehler und Rauschen. Die zweite Fehlerart ist im Allgemeinen klein gegen die erste. Wohingegen Fehler von der ersten Art durch das Training minimiert werden sollen, ist die zweite Fehlerart nach Wahl des Modellansatzes nicht mehr sinnvoll beeinflussbar. Sie muss im vollen Umfang akzeptiert werden, möchte man an dem gewählten Modell festhalten.

Im Fall von (5.3) kann zwischen den beiden Fehlerarten nicht unterschieden werden. Ohne weitere einschränkende Forderungen an die freien Parameter, die den dynamischen Anteil des Modells prägen, werden diese dann unter anderem so angepasst, dass sie die Fehler der zweiten Art verkleinern, falls dies zu einer Verkleinerung des gesamten Fehlers im Sinne von (5.3) führt. Dies geschieht aber unter einer für den allgemeinen Fall weniger geeigneten Anpassung der Parameter des dynamischen Anteils. Genau hierin besteht dann die Überanpassung und an dieser Stelle gilt es anzusetzen.

Nach obigen Überlegungen wäre es anstrebsam, nicht auf die kompromisslose Anpassung

der Parameter an die Trainingsdaten bezüglich des MSE-Kriteriums abzuzielen, sondern geringe Abweichungen, die durch Messfehler oder Rauschen entstehen können, zuzulassen. Da der statische Anteil unabhängig von den Eingangsgrößen ist, muss dieser hierfür nicht weiter betrachtet werden. Für die folgenden Überlegungen wird daher das Training auf die Anpassung der Parameter des dynamischen Anteils reduziert.
Ist das Verhalten des dem Modell zugrunde liegenden Zusammenhangs für eine beliebige aber feste Eingangskonstellation unbekannt oder mit großer Unsicherheit verbunden, so wäre die Ausgabe eines betragsmäßig möglichst kleinen Wertes anzustreben, um so den maximal möglichen Fehler zu minimieren. Die Verstärkung der Eingangsgrößen sollte also möglichst klein gehalten werden.
Eine Antwort auf die Frage nach einer entsprechenden Anpassung der Trainingsmethode heißt Regularisierung. Sie besteht in der Abwägung zweier unterschiedlicher Zielkriterien für die Parameterfixierung. Das eine Kriterium ist nach wie vor die Minimierung der mittleren Fehlerquadratsumme. Das zweite, neu hinzukommende, ist die Minimierung der euklidischen Norm des Parametervektors, die man ebenso gut über eine Minimierung des Quadrats der euklidischen Norm des Parametervektors erreichen kann. Dies entspricht gerade dem Wunsch, zusätzlich zu der Anpassung der Modellausgaben an die Sollwerte die Verstärkung der Eingangsgrößen, die nur durch die freien Parameter entstehen kann, so klein wie möglich zu halten. Hierdurch kommt einer unverhältnismäßigen Verstärkung von Eingangsgrößen weniger Potential zur Verkleinerung der Gesamtabweichungen zu.
In Anlehnung an (5.3) lässt sich eine entsprechende Bewertungszahl für Modelle mit freien Parametern p_j, $j = 1,...,m$ angeben durch:

$$E_{\text{Reg}} = \sum_{j=1}^{n} \left[y^j - M\left(\mathbf{x}^j, \mathbf{p}\right) \right]^2 + \mu \sum_{j=1}^{m} p_j^2 \tag{5.4}$$

Der am besten geeignete Regularisierungsparameter $\mu \geq 0$ ist dabei in der Regel für jede neue Wahl von Trainings- und Testdaten zu ermitteln. Für $\mu = 0$ werden die freien Parameter im Sinne des MSE-Kriteriums ermittelt. Für positive Werte werden die ermittelten p_j immer weiter von dieser Lösung weg hin zu der Forderung nach betragsmäßig kleinen Parametern verschoben, so dass für $\mu \to \infty$ die Lösung $p_j = 0$ für alle $j = 1,...,m$ lautet.
In (5.4) wird in Hinblick auf die Parameter nicht zwischen statischem und dynamischem Anteil des Modells unterschieden. Ferner muss sich in diesem Fall eine automatische Verlagerung der Anteile der dynamischen Parameter in Richtung des statischen Parameters einstellen. Durch eine explizite Vorskalierung der Daten für die Eingangsgrößen und die Ausgangsgröße kann die Regularisierung im Allgemeinen effizienter angewendet werden (vgl. Abschnitt 5.4.2).

5.2.5 K-Kreuzvalidierung

Nachdem nun ein geeigneter Ansatz für die Bestimmung einer regularisierten Lösung gegeben ist, stellt sich die Frage, wie man den Regularisierungsparameter μ am geschicktesten wählt. Man führe sich hierbei erneut vor Augen, dass die Regularisierung zum Zweck der Verbesserung der Generalisierbarkeit der ermittelten Modelle herangezogen wird. Ein guter Wert von μ wäre also daran zu messen, wie gut ein dadurch erstelltes Modell die Soll-Ausgaben weiterer bekannter Datensätze erreicht, die nicht für das Training verwendet worden sind.
Um diese Prüfungen innerhalb des Trainings durchführen zu können, werden Validierungsdaten benötigt. Wie in Abschnitt 5.2.3 bereits erwähnt, können diese Daten auch aus den Reihen der Trainingsdaten entnommen werden, was bei der K-Kreuzvalidierung (eng.: K-Fold-Crossvalidation) der Fall ist [Arl08],[Ama97].
Die K-Kreuzvalidierung wurde bereits kurz in Abschnitt 3.2 vorgestellt. Bei ihr wird die Gesamtheit der zur Verfügung stehenden Trainingsdaten in K Partitionen aufgeteilt. Danach wird das Modell mit den Daten aus jeweils $K-1$ Partitionen trainiert und auf die Daten der nicht herangezogenen Partition angewandt. Die Ausgabe des Modells bezüglich der Partition, die nicht am Training teilgenommen hat, kann dann unter Zuhilfenahme von Gütekriterien (vgl. Abschnitt 5.3) zur Bewertung der Generalisierungsfähigkeit des Modells herangezogen werden.
Für jede Aufteilung gibt es somit K Durchläufe von Modellerstellung und -Anwendung. Der Aufteilungsvorgang kann zufällig geschehen und mehrfach wiederholt werden, um die Generalisierbarkeit bezüglich eines K-tels der verfügbaren Trainingsdaten zu maximieren.
Vor dem Hintergrund der Regularisierung eignet sich die K-Kreuzvalidierung sehr gut für die Ermittlung des am besten geeigneten Regularisierungsparameters μ. Benötigt wird dafür eine Auswahl zu testender Kandidaten für μ, für die jeweils die gleiche Anzahl N von K-Kreuzvalidierungen durchlaufen wird. Die Eignung eines Kandidaten für μ wird dann an den jeweils $N \cdot K$ Ergebnissen für die Anwendung auf die Validierungspartitionen gemessen. Hierzu können übliche Gütekriterien für die Modellanwendung herangezogen werden (vgl. Abschnitt 5.3).
Eine Möglichkeit der Charakterisierung des am besten geeigneten Kanditaten μ^* unter allen getesteten μ könnte dann über die Maximierung der jeweils geringsten erreichten Güte $G(\mu,k)$, $k = 1,...,N \cdot K$, geschehen:

$$\min_k \; G(\mu^*,k) \geq \min_k \; G(\mu,k) \quad \forall \mu$$

5.3 Bewertung der Modellqualität

Für die Bewertung der Qualität eines Vorhersagemodells werden üblicherweise die Modellausgaben bezüglich einer Auswahl von Eingangsdaten mit den zugehörigen Sollwerten verglichen. Eine Möglichkeit, diese einzelnen Abweichungen für eine Auswahl von n beliebigen Datensätzen zu bewerten, wurde bereits in Form der mittleren Fehlerquadratsumme

(5.3) vorgestellt:

$$E_{\mathrm{MSE}} = \frac{1}{n} \sum_{j=1}^{n} \left[y^j - M_{\mathbf{p}} \left(\mathbf{x}^j \right) \right]^2$$

Sie besteht darin, die absoluten Abweichungen jeweils quadriert aufzusummieren und diese Summe auf die Anzahl n der Daten zu beziehen.
Eine zweite Möglichkeit der Bewertung ist das sogenannte Bestimmtheitsmaß R^2:

$$\begin{aligned} R^2 =& 1 - \frac{\sum_{j=1}^{n} \left[y^j - M_{\mathbf{p}} \left(\mathbf{x}^j \right) \right]^2}{\sum_{j=1}^{n} \left[y^j - \bar{\mathbf{y}} \right]^2} \\ =& 1 - \frac{E_{\mathrm{MSE}}}{\frac{n-1}{n} \, s_{\mathbf{y}}} \end{aligned}$$

Hierbei ist für den n-Tupel $\mathbf{y} = (y^1, y^2, \ldots, y^n)$ durch

$$\bar{\mathbf{y}} := \frac{1}{n} \sum_{i=1}^{n} y^i \tag{5.5}$$

der emprische Mittelwert von $\mathbf{y}$ und durch

$$s_{\mathbf{y}} := \frac{1}{n-1} \sum_{i=1}^{n} \left(y^i - \bar{\mathbf{y}} \right)^2$$

die empirische Varianz von $\mathbf{y}$ definiert.
Die empirische Standardabweichung $\sigma(\mathbf{y})$ von $\mathbf{y}$ ist entsprechend

$$\sigma(\mathbf{y}) := \sqrt{s_{\mathbf{y}}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(y^i - \bar{\mathbf{y}} \right)^2}. \tag{5.6}$$

Das Bestimmheitsmaß R^2 unterscheidet sich also in zwei wesentlichen Punkten von der mittleren Fehlerquadratsumme E_{MSE}:

1. Wohingegen in E_{MSE} nur die absoluten Abweichungen eingehen, wird in R^2 diese Bewertung der absoluten Abweichungen auf die (mit $\frac{n-1}{n}$ skalierte) Varianz der Sollwerte bezogen

2. Wohingegen E_{MSE} eine Bemessung der Modellfehler darstellt, bemisst R^2 die Güte des Modells, wobei eine Eins die bestmögliche Bewertung darstellt

Da im allgemeinen Fall die Modellausprägung $M_{\mathbf{p}}$ beliebige Ausgaben liefern kann, besteht auch die Möglichkeit, dass R^2 negativ wird. Um derartige Ergebnisse einzuordnen sei als Richtwert bemerkt, dass der Fall $R^2 = 0$ zum Beispiel für eine statische Modellausprägung

$M_{\mathbf{p}}$ mit konstanter Ausgabe $\bar{\mathbf{y}}$ eintritt.
Durch die Änderung des Vorzeichens ließe sich E_{MSE} ebenfalls in ein Gütekriterium umwandeln.
Schließlich sei noch angemerkt, dass die vorgestellten Methoden zur Bewertung der Modellqualität nur in Verbindung mit bekannten Datensätzen Ergebnisse liefern. Die Ergebnisse sind stets an die verwendeten Datensätze gebunden, so dass die Überprüfung der Modellqualität lediglich stichprobenartig und auch nur für Daten mit bekannten Sollwerten erfolgen kann. Daher sind auch die Ergebnisse für Datensätze, die in das Training eingegangen sind, grundsätzlich anders zu interpretieren als Ergebnisse bezüglich Datensätzen, die dem Modell unbekannt sind, wie zum Beispiel die Testdaten aus Abschnitt 5.2.3.

5.4 Lineare Regressionsmodelle

Eine wichtige Klasse der Regressionsmodelle sind die linearen Regressionsmodelle. Für eine abhängige Größe y und zugehörige Regressoren $x_1, x_2, ..., x_{m-1}$ lautet die Modellvorschrift (5.1) im Fall eines linearen Regressionsmodells mit den m freien Parametern $p_0, p_1, ..., p_{m-1}$:

$$y = p_0 + p_1\ x_1 + p_2\ x_2 + ... + p_{m-1}\ x_{m-1} \tag{5.7}$$

Die freien Parameter werden bezüglich des MSE-Kriteriums (5.3) bestimmt. Charakteristisch für lineare Regressionsmodelle ist die lineare Abhängigkeit der Modellausgabe von den Regressoren.
Vorteile dieses einfachen Ansatzes sind die unkomplizierte Bestimmung der freien Parameter und die Möglichkeit der Abschätzung von Art und Schwere des Einflusses einer Eingangsgröße auf die Ausgangsgröße durch den entsprechenden Gewichtungsfaktor.
Für Fälle, in denen nichtlineare Zusammenhänge zwischen den Eingangsgrößen und der Modellausgabe nicht vernachlässigbar sind, ist dieser Modellansatz hingegen nicht direkt geeignet. In diesen Fällen kann aber durch eine geschickte Wahl der Regressoren als explizit nichtlinear transformierte Eingangsdaten immerhin ein gewichteter nichtlinearer Zusammenhang zwischen der Modellausgabe und den Eingangsdaten berücksichtigt werden.
Als Beispiel wird ein Polynom zweiten Grades auf dem $\mathbb{R}^2$ betrachtet:

$$y\,(x_1,x_2) = p_0 + p_1\ x_1 + p_2\ x_2 + p_3\ x_1\ x_2 + p_4\ x_1^2 + p_5\ x_2^2$$

In diesem Fall ist die Modellausgabe y nichtlinear abhängig von den Eingabegrößen x_1 und x_2. Gegeben seien $n \geq 6$ Funktionswerte $y^1,y^2,...,y^n$ und die zugehörigen Stützvektoren $\mathbf{x}^1,\mathbf{x}^2,...,\mathbf{x}^n$ aus $\mathbb{R}^2$. Die Koeffizienten $p_0,p_1,...,p_5$ seien unbekannt. Um sie entsprechend des MSE-Kriteriums zu bestimmen, kann man das Polynom als lineares Regressionsmodell mit fünf Regressoren auffassen. Diese werden zum einen durch die Eingangsgrößen selber, zum Anderen durch den gemischten und die quadratischen Terme gestellt:

$$y = p_0 + p_1\ z_1 + p_2\ z_2 + p_3\ z_3 + p_4\ z_4 + p_5\ z_5$$

mit

$$
\begin{aligned}
z_1 &= x_1, \\
z_2 &= x_2, \\
z_3 &= x_1\ x_2, \\
z_4 &= x_1^2, \\
z_5 &= x_2^2
\end{aligned}
$$

Für die Bestimmung von m freien Parametern unter Verwendung von n Trainingsdatensätzen lässt sich allgemein durch Einführung der Matrix

$$
\mathbf{M} = \begin{bmatrix} 1 & z_1^1 & z_2^1 & \dots & z_{m-1}^1 \\ 1 & z_1^2 & \dots & \dots & z_{m-1}^2 \\ 1 & \vdots & \ddots & \vdots & z_{m-1}^3 \\ \vdots & \vdots & \dots & \ddots & \vdots \\ 1 & z_1^n & z_2^n & \dots & z_{m-1}^n \end{bmatrix} \tag{5.8}
$$

und des Vektors

$$
\mathbf{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}
$$

die mittlere Fehlerquadratsumme (5.3) in Matrixschreibweise formulieren:

$$
E_{\mathrm{MSE}} = \frac{1}{n}\ (\mathbf{M}\ \mathbf{p} - \mathbf{y})^{\mathrm{T}}\,(\mathbf{M}\ \mathbf{p} - \mathbf{y}) \tag{5.9}
$$

Hierbei stehen die zu ermittelnden freien Parameter p_0,$p_1, \dots$,p_{m-1} in dem Vektor $\mathbf{p}$. Leitet man obigen Ausdruck nach den freien Parametern ab und setzt das Resultat zu Null, so erhält man die sogenannte Normalgleichung:

$$
\mathbf{M}^{\mathrm{T}}\mathbf{M}\ \mathbf{p} = \mathbf{M}^{\mathrm{T}}\ \mathbf{y} \tag{5.10}
$$

Die Bestimmung von $\mathbf{p}$ lässt sich also stets auf die Lösung eines linearen Gleichungssystems zurückführen. Dieses ist genau dann lösbar, wenn $\mathbf{M}^{\mathrm{T}}\mathbf{M}$ regulär ist. Dies ist der Fall, wenn der Rang von $\mathbf{M}$ gerade m ist, was bedeuted, dass m linear unabhängige Datensätze für das Training vorliegen müssen.
Die numerische Bestimmung von $\mathbf{p}$ über die Normalgleichung ist im Allgemeinen schlecht konditioniert [Dem97]. In der Praxis wird anstelle von E_{MSE} als äquivalenter Ansatz die euklidische Norm $\|\mathbf{M}\ \mathbf{p} - \mathbf{y}\|_2$ des Vektors der absoluten Abweichungen minimiert.

Da sich durch die Multiplikation eines Vektors mit einer orthogonalen Matrix $\mathbf{Q}$ dessen euklidische Norm nicht ändert, besteht die Idee darin, eine Zerlegung $\mathbf{M} = \mathbf{Q}\ \mathbf{R}$ zu finden, wobei $\mathbf{Q}$ orthogonal ist und

$$\mathbf{R} = \begin{bmatrix} r_1^1 & r_2^1 & r_3^1 & \cdots & r_m^1 \\ 0 & r_2^2 & \cdots & \cdots & r_m^2 \\ 0 & \vdots & \ddots & \vdots & r_m^3 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & r_m^m \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

obere Dreiecksgestalt besitzt.
Dann ist

$$\begin{aligned} \|\mathbf{M}\ \mathbf{p} - \mathbf{y}\|_2 &= \left\|\mathbf{Q}^{\mathrm{T}}\ (\mathbf{M}\ \mathbf{p} - \mathbf{y})\right\|_2 \\ &= \left\|\mathbf{Q}^{\mathrm{T}}\ (\mathbf{Q}\ \mathbf{R}\ \mathbf{p} - \mathbf{y})\right\|_2 \\ &= \left\|\mathbf{R}\ \mathbf{p} - \mathbf{Q}^{\mathrm{T}}\ \mathbf{y}\right\|_2 . \end{aligned}$$

Das optimale $\mathbf{p}$ lässt sich dann durch Rückwärtssubstitution direkt ermitteln.
Die Zerlegung von $\mathbf{M}$ in das Produkt $\mathbf{Q}\ \mathbf{R}$ und die Berechnung von $\mathbf{Q}^{\mathrm{T}}\ \mathbf{y}$ stellen somit die einzigen nennenswerten Operationen bei dieser Berechnung dar. Die Zerlegung kann effizient durch Householder-Reflexionen geschehen [Hou58].
Analog zu (5.9) kann die in (5.4) vorgestellte Bewertungszahl des regularisierten Ausgleichsproblems in Matrixschreibweise formuliert werden:

$$E_{\mathrm{Reg}} = (\mathbf{M}\ \mathbf{p} - \mathbf{y})^{\mathrm{T}}\,(\mathbf{M}\ \mathbf{p} - \mathbf{y}) + \mu\ \mathbf{p}^{\mathrm{T}}\mathbf{p} \tag{5.11}$$

Leitet man diesen Ausdruck nach den freien Parametern in $\mathbf{p}$ ab und setzt den resultierenden Ausdruck zu Null, so erhält man

$$\begin{aligned} & \mathbf{M}^{\mathrm{T}}\mathbf{M}\ \mathbf{p} - \mathbf{M}^{\mathrm{T}}\mathbf{y} + \mu\ \mathbf{p} = \mathbf{0} \\ \Leftrightarrow \quad & \left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\ \mathbf{I}\right)\ \mathbf{p} = \mathbf{M}^{\mathrm{T}}\mathbf{y}. \end{aligned} \tag{5.12}$$

Hierbei ist $\mathbf{I}$ die Einheitsmatrix.
Der Vergleich mit (5.10) zeigt deutlich die Wirkung des Regularisierungsparameters μ als Spektralverschiebung der Systemmatrix $\mathbf{M}^{\mathrm{T}}\mathbf{M}$.

5.4.1 Standardisierung

In der Darstellung (5.7) des linearen Regressionsmodells sind noch keine einschränkenden Forderungen an die Regressoren und die unabhängige Größe gestellt worden. Ohne eine geeignete Skalierung der Regressoren kann aus den sich einstellenden Parametern $p_0,...,p_{m-1}$ alleine noch keine Aussage über den Einfluss der jeweils zugehörigen Eingangsgröße $x_1,...,x_{m-1}$ gezogen werden. Wäre zum Beispiel die Eingangsgröße x_1 eine Kraft, so wäre p_1 ohne explizite Skalierung abhängig davon, ob diese in den Daten in kN oder in mN vorliegt. Desweiteren kann die explizite Trennung der statischen und dynamischen Anteile in (5.7) zu einer effizienteren Regularisierung führen (vgl. Abschnitt 5.2.4).
Als Maßnahme bedient man sich einer Vorskalierung der Regressoren $x_1,...,x_{m-1}$ und der abhängigen Größe y bezüglich einer Datenbasis mit n Ausprägungen von Eingangsdaten und Sollwerten. Die vorskalierten Sollausgaben $\tilde{y}^i$ und Regressoren $\tilde{x}^i_j$, $j = 1,...,m-1$, ergeben sich durch

$$\begin{aligned} \tilde{y}^i &= \frac{y^i - \bar{\mathbf{y}}}{\sigma(\mathbf{y})}, \quad i = 1,...,n \\ \tilde{x}^i_j &= \frac{x^i_j - \bar{\mathbf{x}}_j}{\sigma(\mathbf{x}_j)}, \quad j = 1,...,m-1 \ ; \ i = 1,...,n. \end{aligned} \tag{5.13}$$

Diese spezielle Vorskalierung (vgl. (5.5) und (5.6)) wird als Standardisierung bezeichnet [Bac10].
Da für die n zur Verfügung stehenden Datensätze die abhängige Größe sowie sämtliche Regressoren im standardisierten Fall mittelwertfrei sind, entfällt notwendiger Weise der statische Anteil und das Modell hat die allgemeine Darstellung

$$\tilde{y} = \tilde{p}_1 \ \tilde{x}_1 + ... + \tilde{p}_m \ \tilde{x}_m. \tag{5.14}$$

Mit dieser Skalierung sind die Werte der Regressoren untereinander vergleichbar und der jeweils j-te Koeffizienten $\tilde{p}_j$ spiegelt unmittelbar den durch das lineare Modell geschätzten Einfluss der j-ten Eingangsgröße $\tilde{x}_j$ auf die Ausgabegröße $\tilde{y}$ wider.
Diese Form eignet sich sehr gut für eine regularisierte Ermittlung der Parameter, da diese ansonsten mit ungleichmäßiger Gewichtung geschehen würde und keine klare Trennung bezüglich der statischen und dynamischen Anteile bestünde. Deswegen sollte die Parameterfixierung allgemein in dieser Darstellung geschehen.
Wie man von (5.14) ausgehend wieder zu der Darstellung (5.7) gelangt, die direkt die Werte der Eingangsgrößen x_j akzeptiert und die unverzerrte Ausgabe y erzeugt, folgt

unmittelbar aus (5.13) und (5.14):

$$
\begin{aligned}
y &= \sigma(\mathbf{y})\ \tilde{y} + \bar{\mathbf{y}} \\
&= \sigma(\mathbf{y})\ [\tilde{p}_1\ \tilde{x}_1 + \ldots + \tilde{p}_{m-1}\ \tilde{x}_{m-1}] + \bar{\mathbf{y}} \\
&= \sigma(\mathbf{y})\ \left[\frac{\tilde{p}_1}{\sigma(\mathbf{x}_1)}\ (x_1 - \bar{\mathbf{x}}_1) + \ldots + \frac{\tilde{p}_{m-1}}{\sigma(\mathbf{x}_{m-1})}\ (x_{m-1} - \bar{\mathbf{x}}_{m-1})\right] + \bar{\mathbf{y}} \\
&= \tilde{p}_1\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_1)}\ x_1 + \ldots + \tilde{p}_{m-1}\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_{m-1})}\ x_{m-1} \\
&\quad - \tilde{p}_1\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_1)}\ \bar{\mathbf{x}}_1 - \ldots - \tilde{p}_{m-1}\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_{m-1})}\ \bar{\mathbf{x}}_{m-1} + \bar{\mathbf{y}} \\
&= \sum_{j=1}^{m-1} \tilde{p}_j\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_j)}\ x_j + \bar{\mathbf{y}} - \sum_{j=1}^{m-1} \tilde{p}_j\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_j)}\ \bar{\mathbf{x}}_j
\end{aligned}
$$

Der Koeffizientenvergleich mit (5.7) liefert schließlich

$$
\begin{aligned}
p_0 &= \bar{\mathbf{y}} - \sum_{j=1}^{m-1} \tilde{p}_j\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_j)}\ \bar{\mathbf{x}}_j, \\
p_j &= \tilde{p}_j\ \frac{\sigma(\mathbf{y})}{\sigma(\mathbf{x}_j)}, \quad j = 1,\ldots,m-1,
\end{aligned}
\tag{5.15}
$$

und damit eine Vorschrift zur Bestimmung der Parameter des unskalierten Modells mit Hilfe der Parameter des standardisierten Modells sowie der statistischen Kennzahlen der Regressoren und der abhängigen Größe bezüglich der verwendeten Datenbasis.
Man erkennt in (5.15), dass p_0 nicht nur von dem Mittelwert von $\mathbf{y}$, sondern auch von der Verteilung der Daten der Eingangsgrößen abhängt.
Das Vorgehen zur Erstellung eines linearen Regressionsmodells mit $m-1$ Regressoren und den Datentupeln $(x_1^1,\ldots,x_{m-1}^1,y^1),\ldots,(x_1^n,\ldots,x_{m-1}^n,y^n)$ lässt sich also algorithmisch wie folgt beschreiben:

1. Bestimme für die Standardisierung die Mittelwerte $\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_{m-1},\bar{\mathbf{y}}$ und die Standardabweichungen $\sigma(\mathbf{x}_1),\ldots,\sigma(\mathbf{x}_{m-1}),\sigma(\mathbf{y})$

2. Bestimme mit den im vorherigen Schritt ermittelten statistischen Kennzahlen und (5.13) die standardisierten Datentupel $(\tilde{x}_1^1,\ldots,\tilde{x}_{m-1}^1,\tilde{y}^1),\ldots,(\tilde{x}_1^n,\ldots,\tilde{x}_{m-1}^n,\tilde{y}^n)$

3. Bestimme die Parameter $\tilde{p}_1,\ldots,\tilde{p}_{m-1}$ des standardisierten Modells (5.14) (bezüglich des MSE-Kriteriums oder regularisiert) unter Verwendung der Daten aus dem vorherigen Schritt

4. Bestimme ggf. die Parameter des unskalierten Modells $p_0,\ldots,p_{m-1}$ mit Hilfe von (5.15)

Für den dritten Schritt gilt im standardisierten Modell die Matrixschreibweise für die Bestimmung von E_{MSE} aus (5.9) analog. Allerdings verringert sich die Anzahl der freien Parameter auf $m-1$ und die erste Spalte von $\mathbf{M}$ in (5.8) entfällt.

5.4.2 Regularisierte lineare Regression

Im Folgenden soll eine Methode vorgestellt werden, die die regularisierte Bestimmung der freien Parameter eines linearen Regressionsmodells ermöglicht. Ausgangspunkt ist die Ermittlung der freien Parameter bezüglich des MSE-Kriteriums nach (5.10). Dieser unregularisierte Fall ergibt als Lösung

$$\mathbf{p}^* = \left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right)^{-1} \mathbf{M}^{\mathrm{T}}\mathbf{y}. \tag{5.16}$$

Entsprechend wurde für die Bestimmung der regularisierten Lösung mit dem Regularisierungsparameter μ die Bedingung (5.12) hergeleitet, nach der sich die freien Parameter zu

$$\mathbf{p} = \left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\,\mathbf{I}\right)^{-1} \mathbf{M}^{\mathrm{T}}\mathbf{y}. \tag{5.17}$$

ergeben.

Die Norm des Vektors der regularisierten Parameter nimmt für $\mu \geq 0$ Werte zwischen 0 und $\|\mathbf{p}^*\|_2$ an. Um diesen Bereich vollständig abzudecken, müsste μ allerdings $\mathbb{R}_+$ komplett durchlaufen. Der Fall $\|\mathbf{p}\|_2 = 0$ wird offenbar nur asymptotisch erreicht.
Wie in Abschnitt 5.2.5 dargelegt, erfolgt die Ermittlung des am besten geeigneten Wertes von μ häufig unter der Vorgabe einer festen Anzahl von Kandidaten für μ, aus denen der Beste zu bestimmen ist. Die Kandidaten sollten daher so gewählt werden, dass für die resultierenden freien Parameter $\mathbf{p}$ die zugehörige Norm den Bereich $[0,\ \|\mathbf{p}^*\|_2]$ möglichst gleichmäßig abdeckt.
Die Idee besteht nun darin, nicht direkt den Wert von μ zu wählen, sondern einen Wert $\xi \in [0\ ;\ 1]$, der in die Forderung

$$\|\mathbf{p}\|_2 = \xi\ \|\mathbf{p}^*\|_2 \tag{5.18}$$

eingeht.
Unter Einhaltung dieser Bedingung ist dann nur noch die mittlere Fehlerquadratsumme zu minimieren. Dies lässt sich durch das nichtlineare, restringierte Optimierungsproblem

$$\begin{aligned} \text{minimiere} \qquad & (\mathbf{M}\ \mathbf{p} - \mathbf{y})^{\mathrm{T}}\,(\mathbf{M}\ \mathbf{p} - \mathbf{y}) \\ \text{unter der Nebenbedingung} \qquad & \mathbf{p}^{\mathrm{T}}\mathbf{p} - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^* = 0 \end{aligned} \tag{5.19}$$

ausdrücken. Hierbei wurde für die Formulierung der Nebenbedingung (5.18) quadriert. Die Berücksichtigung der Nebenbedingung bei der Minimierung der Zielfunktion erfolgt unter Einführung einer positiven Schlupfvariable, die wir, zunächst ohne weitere Hintergedanken, erneut mit μ bezeichnen, in der sogenannten Lagrangefunktion

$$L\left(\mathbf{p},\mu\right) = \left(\mathbf{M}\ \mathbf{p} - \mathbf{y}\right)^{\mathrm{T}} \left(\mathbf{M}\ \mathbf{p} - \mathbf{y}\right) + \mu\ \left(\mathbf{p}^{\mathrm{T}}\mathbf{p} - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^*\right).$$

Nach den Bedingungen von Karush [Kar39] sowie Kuhn und Tucker [Kuh51] (KKT-Bedingungen) existiert zu einem zulässigen lokalen Optimum $\mathbf{p}$ von (5.19) ein $\mu \geq 0$, so dass gilt:

$$\nabla_{\mathbf{p},\mu} L\left(\mathbf{p},\mu\right) = \begin{bmatrix} 2\ \left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\ \mathbf{p} - \mathbf{M}^{\mathrm{T}}\mathbf{y} + \mu\ \mathbf{p}\right) \\ \\ \mathbf{p}^{\mathrm{T}}\mathbf{p} - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \\ 0 \end{bmatrix} \tag{5.20}$$

Der Zusammenhang mit (5.12) ist unmittelbar erkennbar. Hier kommt allerdings hinzu, dass μ nicht frei gewählt werden kann sondern über $\mathbf{p}$ an die Nebenbedingung

$$\mathbf{p}^{\mathrm{T}}\mathbf{p} - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^* = 0$$

gekoppelt ist.
Die Suche nach einem geeigneten $\mu \geq 0$, für das (5.20) erfüllt ist, lässt sich nun formulieren als Suche nach einer Nullstelle der skalaren Funktion

$$l\left(\mu\right) := \left[\left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\mathbf{I}\right)^{-1}\ \mathbf{M}^{\mathrm{T}}\mathbf{y}\right]^{\mathrm{T}} \left[\left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\mathbf{I}\right)^{-1}\ \mathbf{M}^{\mathrm{T}}\mathbf{y}\right] - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^*. \tag{5.21}$$

Es sei an dieser Stelle darauf hingewiesen, dass $\mathbf{p}^*$ nach (5.16) als Lösung des nicht regularisierten Falles bekannt und ξ als Parameter vorgegeben ist. Somit liegen alle Informationen vor, um eine Nullstelle von (5.21) iterativ zu bestimmen.
Hierfür wird ein einfacher Bisektionsalgorithmus verwendet:

1. Initialisierung: $\mu_- = \mu_{lo}$, $\mu_+ = \mu_{hi}$
2. Berechne $\mu = \frac{\mu_+ - \mu_-}{2}$, $l^* = l\left(\mu\right)$
3. Abbruch, falls $|l^*|$ klein genug
4. Falls $l^* < 0$: $\mu_- = \mu$
   sonst: $\mu_+ = \mu$
5. gehe zu 2.

Im ersten Schritt des Algorithmus (Initialisierung) werden möglichst enge Schranken μ_{lo} und μ_{hi} für die gesuchte Nullstelle benötigt mit

$$l\left(\mu_{lo}\right) \leq 0 \leq l\left(\mu_{hi}\right).$$

Für deren Bestimmung werden zunächst die Hilfsgrößen

$$\mathbf{b} = \mathbf{M}^{\mathrm{T}}\mathbf{y}$$

und

$$\mathbf{A} = \left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\mathbf{I}\right)^{-1}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu\mathbf{I}\right)^{-1} \tag{5.22}$$

eingeführt, um eine bessere Übersicht zu gewährleisten. Dann ist

$$l\left(\mu\right) = \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{b} - \xi^2\ \mathbf{p}^{*\mathrm{T}}\mathbf{p}^{*}.$$

Ist λ_{min} der kleinste und λ_{max} der größte Eigenwert von $\mathbf{A}$, so folgt aus der Definition des Rayleigh-Quotienten [Par80]

$$\lambda_{min}\ \mathbf{b}^{\mathrm{T}}\mathbf{b} \leq \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{b} \leq \lambda_{max}\ \mathbf{b}^{\mathrm{T}}\mathbf{b}.$$

Dies führt zu den Implikationen

$$\lambda_{min} \geq \xi^2\ \frac{\mathbf{p}^{*\mathrm{T}}\mathbf{p}^{*}}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \Rightarrow l \geq 0 \tag{5.23}$$

und

$$\lambda_{max} \leq \xi^2\ \frac{\mathbf{p}^{*\mathrm{T}}\mathbf{p}^{*}}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \Rightarrow l \leq 0 \tag{5.24}$$

Es wird also für die Bestimmung von μ_{hi} eine von μ abhängige untere Schranke von λ_{min} benötigt, und analog eine von μ abhängige obere Schranke von λ_{max} für die Bestimmung von μ_{lo}.
Ist σ_{max} der größte Eigenwert der Matrix $\mathbf{M}^{\mathrm{T}}\mathbf{M}$ und σ_{min} entsprechend der kleinste, so gilt aufgrund der Definition von $\mathbf{A}$ in (5.22)

$$\lambda_{min} = \left(\frac{1}{\sigma_{max} + \mu}\right)^2 \geq \left(\frac{1}{\mathrm{tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right) + \mu}\right)^2 \tag{5.25}$$

und

$$\lambda_{max} = \left(\frac{1}{\sigma_{min} + \mu}\right)^2 \leq \left(\frac{1}{\mu}\right)^2. \tag{5.26}$$

Durch Anwendung der Schranke aus (5.25) in (5.23) erhält man

$$\begin{aligned}
\left(\frac{1}{\mu}\right)^2 &= \xi^2 \, \frac{\mathbf{p}^{*\mathrm{T}}\mathbf{p}^*}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \\
\Leftrightarrow \qquad \mu &= \frac{\|\mathbf{b}\|_2}{\xi \; \|\mathbf{p}^*\|_2} \\
\Rightarrow \qquad \mu_{lo} &= \frac{\left\|\mathbf{M}^{\mathrm{T}}\mathbf{y}\right\|_2}{\xi \; \|\mathbf{p}^*\|_2}
\end{aligned}$$

und analog aus (5.26) und (5.24)

$$\begin{aligned}
\left(\frac{1}{\mathrm{tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right) + \mu}\right)^2 &= \xi^2 \, \frac{\mathbf{p}^{*\mathrm{T}}\mathbf{p}^*}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \\
\Leftrightarrow \qquad \mu &= \frac{\|\mathbf{b}\|_2}{\xi \; \|\mathbf{p}^*\|_2} - \mathrm{tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right) \\
\Rightarrow \qquad \mu_{hi} &= \frac{\left\|\mathbf{M}^{\mathrm{T}}\mathbf{y}\right\|_2}{\xi \; \|\mathbf{p}^*\|_2} - \mathrm{tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right).
\end{aligned}$$

Offenbar kann in dieser Darstellung μ_{hi} negativ werden. Das darf allerdings nicht geschehen, da dies eine Verletzung von (5.20) bedeuten würde. Zusätzlich könnte dies unmittelbar zu numerischen Problemen führen. Die Matrix $\mathbf{M}^{\mathrm{T}}\mathbf{M} + \mu \, \mathbf{I}$ könnte singulär werden, da $\mathbf{M}^{\mathrm{T}}\mathbf{M}$ positiv semi-definit ist und daher nur nicht negative Eigenwerte besitzen kann. Das hieße dann aber, dass das Gleichungssystem (5.12) in diesem Fall nicht lösbar und $\mathbf{p}$ nach (5.17) somit nicht definiert ist.
Um dem entgegenzuwirken wird μ_{hi} zu

$$\mu_{hi} = \max\left\{0, \; \frac{\|\mathbf{M}^{\mathrm{T}}\mathbf{y}\|_2}{\xi \; \|\mathbf{p}^*\|_2} - \mathrm{tr}\left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right)\right\}$$

gesetzt. Im Falle $\mu_{hi} = 0$ ist der Wert der Hilfsfunktion $l\left(\mu_{hi}\right)$ für $\xi < 1$ stets positiv und diese Maßnahme somit zulässig.
Bei den Auswertungen von $l\left(\mu\right)$ nach (5.21) im Verlauf des Bisektionsalgorithmus wird in der Praxis nicht die inverse Matrix berechnet. Stattdessen wird mit dem übergebenen μ das zugehörige $\mathbf{p}$ über das lineare Gleichungssystem (5.12) bestimmt. Dann reduziert sich die Auswertung von l zu

$$l\left(\mu\right) := \mathbf{p}^{\mathrm{T}}\mathbf{p} - \xi^2 \; \mathbf{p}^{*\mathrm{T}}\mathbf{p}^*,$$

wobei der Term $\xi^2 \; \mathbf{p}^{*\mathrm{T}}\mathbf{p}^*$ sogar nur ein einziges Mal zu Beginn bestimmt werden muss.

5.4.3 Kombination linearer Regressionsmodelle

In den Abschnitten 5.2.2 und 5.2.3 wurde der Bedarf an möglichst vielen bekannten Daten für das Training eines Modells erläutert. Die Anzahl dieser verfügbaren Daten stellt eine Einschränkung der Anzahl der freien Modellparameter dar. Im Allgemeinen ist dies insbesondere auch eine direkte Einschränkung der Regressorenzahl. Bei den linearen Regressionsmodellen ist letztere im standardisierten Fall sogar identisch mit der Anzahl der freien Parameter (vgl. Abschnitt 5.4.1).
Im Klartext bedeutet dies, dass die maximale Zahl der miteinzubeziehenden Einflussgrößen des zu erstellenden linearen Regressionsmodells von der Anzahl der verfügbaren Trainingsdaten abhängt. Um bei der Modellerstellung gute Ergebnisse zu erzielen, sollte die Komplexität, und damit auch die Zahl der Regressoren, klein gehalten werden. Auf der anderen Seite ist die Berücksichtigung einer größeren Anzahl an Regressoren unabdingbar, sofern diese als Merkmale benötigt werden, um den Systemzustand hinreichend genau zu beschreiben.
Eine Idee, diesem Konflikt zu begegnen, liegt darin, mehrere (in Hinblick auf die Anzahl der Regressoren) kleinere lineare Regressionsmodelle mit unterschiedlichen Regressorkombinationen zu erstellen und diese durch (nach Modellgüte) gewichtete Mittelung zu einem Gesamtmodell zusammenzufassen. Die so erstellten Gesamtmodelle weisen eine Reihe anzustrebender Eigenschaften auf:

- Die gewichtete Summe linearer Modelle ist wieder ein lineares Modell, in dem die Parameter sich aus der gewichteten Summe der Parameter der einzelnen Modelle zu dem entsprechenden Regressor ergeben.
- Die einzelnen Modelle neigen bei gleicher Konstellation der Trainingsdaten weniger zur Überanpassung der Modellparameter als ein komplexeres Modell, dass alle insgesamt eingeflossenen Regressoren berücksichtigt.
- Durch die Zusammenfassung der Modelle als gewichtete Summe wird im Gesamtmodell jeder Regressor berücksichtigt, der in mindestens eines der einzelnen Modelle eingeht.
- Regressoren, die einen signifikanten Einfluss auf die Ausgabegröße haben, werden in mehreren kleineren Modellen auf ähnliche Weise auftreten und dadurch im Gesamtmodell eine entsprechend bestärkte Ausprägung aufweisen.
- Die Gewichtung von Regressoren, die in verschiedene kleinere Modelle mit wechselndem Vorzeichen eingehen (und somit als unsicher einzustufen sind), wird durch die Summation im Gesamtmodell geringer sein als in den einzelnen Modellen.

Die Ausgabe des Gesamtmodells ist identisch mit der gewichteten Summe der verschiedenen Ausgaben der einzelnen Modelle.
Im Folgenden wird eine Abschätzung der mittleren Fehlerquadratsumme des Gesamtmodells in Abhängigkeit der gewichteten Summe der mittleren Fehlerquadratsummen der

einzelnen Modelle für eine beliebige Datenbasis vorgenommen.
Ausgangspunkt ist eine Gesamtzahl von q standardisierten Einzelmodellen mit jeweils m Regressoren und den Koeffizienten p_k^i, $k = 1,...,m$, $i = 1,...,q$.
Die Mittelung der einzelnen Modelle entspricht einer gewichteten Summe mit den Koeffizienten α_i, $i = 1,...,q$ und

$$\sum_{i=1}^{q} \alpha_i = 1.$$

Für die Ermittlung der mittleren Fehlerquadratsumme seien insgesamt n Datensätze

$$\mathbf{d}^j = \left(x_1^j,...,x_m^j,y^j\right), \quad j = 1,...,n$$

vorhanden.
Die Koeffizienten des gemittelten Modells ergeben sich jeweils zu

$$\sum_{i=1}^{q} \alpha_i \, p_k^i, \quad k = 1,...,m.$$

Damit berechnet sich E_{MSE} gemäß (5.3) bezüglich des Gesamtmodells in der Form

$$\begin{aligned} &\frac{1}{n}\sum_{j=1}^{n}\left(y^j - \sum_{k=1}^{m} x_k^j \sum_{i=1}^{q} \alpha_i \, p_k^i\right)^2 \\ =&\frac{1}{n}\sum_{j=1}^{n}\left[\left(y^j - \sum_{i=1}^{q} \alpha_i \sum_{k=1}^{m} x_k^j \, p_k^i\right)^2\right]. \end{aligned} \tag{5.27}$$

Dem gegenüber lässt sich die gewichtete Summe der E_{MSE}-Werte der q einzelnen Modelle durch

$$\begin{aligned} &\sum_{i=1}^{q} \alpha_i \left[\frac{1}{n}\sum_{j=1}^{n}\left(y^j - \sum_{k=1}^{m} x_k^j \, p_k^i\right)^2\right] \\ =&\frac{1}{n}\sum_{j=1}^{n}\left[\sum_{i=1}^{q} \alpha_i \left(y^j - \sum_{k=1}^{m} x_k^j \, p_k^i\right)^2\right] \end{aligned} \tag{5.28}$$

darstellen.
In den folgenden Betrachtungen werden jeweils die Eingabedaten des j-ten Datensatzes in dem Vektor

$$\mathbf{x}^j = \left(x_1^j,...,x_m^j\right)^{\mathrm{T}} \in \mathbb{R}^m$$

und die Koeffizienten des i-ten Modells in dem Vektor

$$\mathbf{p}^i := \left(p_1^i,\ldots,p_m^i\right)^{\mathrm{T}} \in \mathbb{R}^m$$

zusammengefasst.
Dann ist für $j = 1,\ldots,n$ beliebig aber fest die quadratierte Abweichung der Modellausgabe vom Sollwert

$$\begin{aligned} Q_j : \quad \mathbb{R}^m &\to \mathbb{R} \\ \mathbf{p} &\mapsto \left(y^j - {\mathbf{x}^j}^{\mathrm{T}}\mathbf{p}\right)^2 \end{aligned}$$

ein konvexes Funktional auf $\mathbb{R}^m$ und es folgt:

1. Die mittlere Fehlerquadratsumme des Gesamtmodells (5.27) lässt sich darstellen zu

$$\frac{1}{n}\sum_{j=1}^{n}\left[Q_j\left(\sum_{i=1}^{q}\alpha_i\ \mathbf{p}^i\right)\right]$$

2. Die gewichtete Summe der E_{MSE}-Werte der q einzelnen Modelle (5.28) lässt sich darstellen zu

$$\frac{1}{n}\sum_{j=1}^{n}\left[\sum_{i=1}^{q}\alpha_i\ Q_j\left(\mathbf{p}^i\right)\right]$$

Nach der Ungleichung von Jensen [Jen06] gilt für jeden Datensatz $\mathbf{d}^j$ aufgrund der Konvexität von Q_j die Abschätzung

$$Q_j\left(\sum_{i=1}^{q}\alpha_i\ \mathbf{p}^i\right) \leq \sum_{i=1}^{q}\alpha_i\ Q_j\left(\mathbf{p}^i\right) \tag{5.29}$$

und somit insbesondere

$$\frac{1}{n}\sum_{j=1}^{n}\left[\left(y^j - \sum_{i=1}^{q}\alpha_i\ {\mathbf{x}^j}^{\mathrm{T}}\mathbf{p}^i\right)^2\right] \leq \frac{1}{n}\sum_{j=1}^{n}\left[\sum_{i=1}^{q}\alpha_i\ \left(y^j - {\mathbf{x}^j}^{\mathrm{T}}\mathbf{p}^i\right)^2\right].$$

Die mittlere Fehlerquadratsumme des Gesamtmodells ist also höchstens so groß wie die gewichtete Summe der mittleren Fehlerquadratsummen der einzelnen Modelle.
Dabei ist die Aussage von (5.29) sogar schärfer, da für jedes $j = 1,\ldots,n$ die quadrierte Abweichung der Ausgabe des Gesamtmodells vom Sollwert höchstens so groß ist wie die gewichtete Summe der quadrierten Abweichungen der Ausgaben aller q Einzelmodelle vom Sollwert.
Die bislang in Kapitel 5 beschriebenen Ansätze, Techniken und Konzepte bilden die

Grundlage für das OTFS-Verfahren, das in Kapitel 7 vorgestellt wird. Dies betrifft sowohl die in Abschnitt 5.4 vorgestellte regularisierte lineare Regression mit Standardisierung und die Kombination linearer Regressionsmodelle als auch die Data-Division (vgl. Abschnitt 5.2.3) und die K-Kreuzvalidierung (vgl. Abschnitt 5.2.5).

5.5 Künstliche Neuronale Netze

Seit es die Verfügbarkeit schneller Rechner mit großer Speicherkapazität und geeigneten Software-Umgebungen gestattet, halten zunehmend Methoden und Konzepte Einzug in die numerische Behandlung mathematischer und technischer Probleme, die durch Vorbilder aus der Natur inspiriert sind. Charakteristisch für diese Art von Konzepten ist ihr relativ einfaches Grundprinzip, welches erst durch einen sehr hohen Grad an paralleler Verarbeitung und Interaktion zu überzeugenden Ergebnisse führt.
Etablierte Beispiele aus dem Gebiet der globalen Optimierung [Wei09] sind die evolutionären und genetischen Algorithmen [Bäc96],[Sim13] sowie Verfahren, die Schwarmintelligenz nutzen. Hierzu gehören unter Anderem die Ameisenalgorithmen (eng.: Ant Colony Optimization) [Col91],[Hu08] und die Partikelschwarmoptimierung [Ken95][Par02].
Die in diesem Abschnitt vorgestellten künstlichen neuronalen Netze [Bis96],[Hay98] stellen ein durch die Natur inspiriertes Konzept für die Realisierung künstlicher Intelligenz dar, das in seiner ursprünglichen Form angelehnt ist an die Vernetzung von Nervenzellen im Gehirn und im Rückenmark. Auf der höchsten Abstraktionsebene bestehen die künstlichen neuronalen Netze aus durch Kanten miteinander verbundenen Knoten, den (künstlichen) Neuronen. Die Neuronen sind dabei auf Schichten verteilt, die entweder komplett innerhalb des Netzes liegen oder eine Schnittstelle nach außen hin darstellen. Letztere Schichten werden entsprechend als Ein- oder Ausgabeschicht, eine komplett im Innern liegende hingegen als verdeckte Schicht bezeichnet.
Wie zuvor allgemein erwähnt entsteht das große Potential zur Verarbeitung verschiedenster Problemstellungen auch bei den künstlichen neuronalen Netzen vor allem durch den hohen Grad an Parallelität und Interaktion. Die Grundregeln, auf die sämtliche Berechnungen letzendlich zurückgeführt werden können, sind hingegen sehr einfach und werden in Abschnitt 5.5.1 vorgestellt.
Ausgehend von dem abstrakten Grundkonzept sind viele Typen künstlicher neuronaler Netze entstanden, die sich in ihrer Topologie unterscheiden und dadurch auch in dem Einsatzgebiet, für das sie prädestiniert sind. Die Topologie kapselt dabei Anzahl und Arten der Neuronenschichten sowie die Regeln, nach denen diese miteinander Informationen austauschen können.
Neben dem in Abschnitt 5.5.1 vorgestellten Typ der Feed-Forward-Netze, in dem jede Schicht die Informationen nur in Richtung der Ausgabeschicht weiterleiten kann, und der in Abschnitt 5.5.5 beschriebenen selbstorganisierenden Karten nach Kohonen, existieren noch weitere Typen wie Hopfield-Netze [Hop82] als Beispiel einer rekurrenten Netztopologie und viele andere [Par91],[Tan13],[Sin00].

Gemeinsam ist den verschiedenen Typen ihre grundsätzliche Einsatzmotivation. Diese kann zum einen aus dem Ziel entstehen, spezielle Funktionen oder potentielle Befähigungen des natürlichen Pendants durch das numerische Modell zu simulieren und zu untersuchen. Am häufigsten entsteht sie allerdings aus der Notwendigkeit ein Problem zu bearbeiten, bei dem grundlegendes Wissen über interne Funktionsweisen nicht vorhanden ist. Hier wird also an die Fähigkeit neuronaler Netze appelliert, ihre interne Struktur alleine durch die Vorgabe von Daten an den Schnittstellen nach außen an eine gegebene Problemstellung anzupassen. Dabei interessiert in der Regel auch nicht die sich einstellende innere Struktur, sondern nur das von außen zu beobachtende Verhalten für gewisse Zustände in der Eingabeschicht. Man bezeichnet ein derartiges System deshalb auch als eine Black-Box [Day01].
Über die Betrachtung künstlicher neuronaler Netze als Black-Box-Systeme wird ein direkter Bezug zu den Vorhersagemodellen hergestellt. Beispiele für die technische Anwendung künstlicher neuronaler Netze aus dem Umfeld der Fahrzeugtechnik sind in [Pra09],[Wan07] und [Wan14] gegeben. Die vor dem Hintergrund der Vorhersagemodelle am besten geeignete Netz-Topologie sowie die Ansatzfunktion des verwendeten Neuronenmodells werden in Abschnitt 5.5.1 vorgestellt. Ein für diese Topologie grundlegendes Konzept, auf dem fast alle gängigen Trainingsverfahren für Netze dieser Art basieren, ist Thema von Abschnitt 5.5.2. In Abschnitt 5.5.3 wird ein Trainingsverfahren beschrieben, welches, analog zu Abschnitt 5.4.2, eine regularisierte Anpassung entsprechender Netze an die Trainingsdaten ermöglicht. Schließlich hat Abschnitt 5.5.5 mit den selbstorganisierenden Karten nach Kohonen ein sich in Topologie und Einsatzgebiet grundlegend von den zuvor gezeigten künstlichen neuronalen Netzen unterscheidendes Konzept zum Inhalt. Die Anwendung dieser speziellen Netze zur Untersuchung und Bewertung von Merkmalsauswahlen wird im späteren Verlauf in den Kapiteln 6 und 7 eine Rolle spielen.

5.5.1 Feed-Forward-Struktur

Die am häufigsten anzutreffende Topologie künstlicher neuronaler Netze ist die sogenannte Feed-Forward-Struktur. Sie ist ausgezeichnet durch eine klare Richtung der Datenverarbeitung ausgehend von der Eingabeschicht bis hin zu der Ausgabeschicht. Zwischen diesen beiden Schichten liegen weitere, sogenannte verdeckte Schichten. Der Name rührt daher, dass die Neuronen dieser verdeckten Schichten keine Verbindung nach außen besitzen und somit für die Schnittstellenbeschreibung eines Netzes nicht benötigt werden. Auch wenn sie von außen betrachtet keinerlei Bedeutung für die Anwenderseite haben, so sind sie doch in ihrer Funktion als datenverarbeitende Einheiten des Netzes gerade maßgeblich für die Flexibilität des Netzes verantwortlich.
Die Anzahl der verdeckten Schichten ist theoretisch nicht nach oben beschränkt. Allerdings existieren auch keine Richtlinien dafür, wann und bei welcher Aufteilung genau die Verwendung mehrerer verdeckter Schichten bei gleicher Anzahl verdeckter Neuronen insgesamt besser geeignet ist als eine einzige verdeckte Schicht. Da aber bewiesen ist, dass theoretisch eine einzige verdeckte Schicht reicht, um beliebige funktionale Zusammenhänge zwischen

den beteiligten Ein- und Ausgabedaten beliebig gut zu approximieren [Cyb89], wird in vielen Fällen auf die Verwendung mehrerer verdeckter Schichten verzichtet [Huy07],[Hua00]. In Hinblick auf das Design der Netze vereinfacht dies den Entwurf und damit die Untersuchungen dahingehend, dass lediglich die Anzahl der Neuronen in der verdeckten Schicht die Flexibilität des gesamten Netztes entscheidet.
Die Anzahl der Neuronen in der Eingabeschicht ist üblicherweise eine Randbedingung und folgt aus der Struktur der zu verarbeitenden Daten. Die Anzahl der Neuronen in der Ausgabe-Schicht ist im Rahmen dieser Arbeit stets eins, da hierin, wie zu Beginn von Kapitel 5 erwähnt, nur Vorhersagemodelle mit skalaren Ausgabegrößen betrachtet werden. Abbildung (5.1) zeigt schematisch die Struktur eines allgemeinen Feed-Forward-Netzes mit L Neuronen-Schichten.

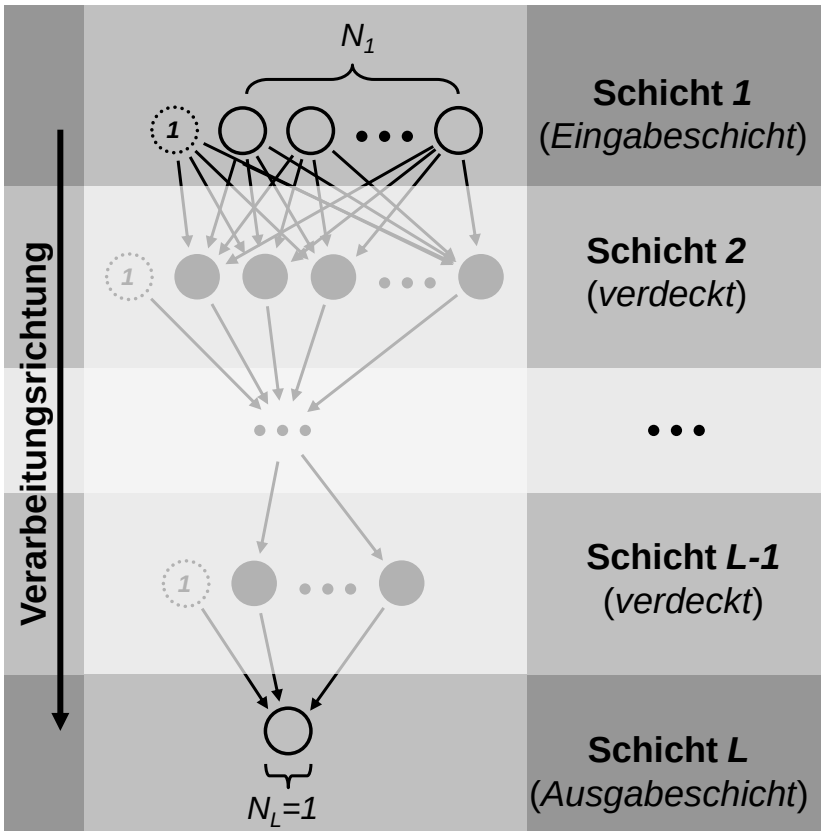

Abbildung 5.1: Allgemeine Struktur eines L-schichtigen Feed-Forward-Netzes mit einem Ausgabe-Neuron

Die fortlaufende Nummerierung der Schichten beginnt bei der Eingabeschicht 1, geht über die verdeckten Schichten $2, \ldots, L-1$ und endet bei der Ausgabeschicht L.
Die Anzahl der Neuronen in der j-ten Schicht ist jeweils mit N_j bezeichnet. Wie bereits

erwähnt ist bei den hier betrachteten Netzen stets $N_L = 1$ und N_1 die Anzahl der Elemente eines von dem Netz zu verarbeitenden Datensatzes $\mathbf{x} \in X$ aus der Menge X der möglichen Zustands-Beschreibungen des zugrunde liegenden Systems.
Die Neuronen selber werden nach ihrer Schicht j und ihrer Position i innerhalb der Schicht (von links nach rechts bei 1 beginnend) mit $n_{i,j}$ bezeichnet.
Die Ausgabe, die das Neuron $n_{i,j}$ über die abgehenden Verbindungen weiter durch das Netz in Richtung Ausgabeschicht propagiert, ist $o_{i,j}$.
Wie eingangs erwähnt sind Neuronen-Verbindungen bei der Feed-Forward-Struktur stets gerichtet. Erlaubt sind nur Verbindungen ausgehend von einem beliebigen Neuron $n_{i_1,j}$ der j-ten Schicht hin zu einem anderen Neuron $n_{i_2,j+1}$ der $j+1$-ten Schicht. Im Rahmen des Erlaubten sind Netze mit Feed-Forward-Struktur außerdem vollständig verbunden, so dass alle nach diesem Schema möglichen Verbindungen auch tatsächlich vorhanden sind.
Neuronen-Verbindungen sind außerdem gewichtet. Das Gewicht der Verbindung von dem i_1-ten Neuron $n_{i_1,j}$ der j-ten Schicht zu dem i_2-ten Neuron $n_{i_2,j+1}$ der $j+1$-ten Schicht wird mit $w^j_{i_1,i_2}$ bezeichnet. Auch Gewichte mit dem Wert Null sind erlaubt, so dass über diese Maßnahme unerwünschte Neuronen-Verbindungen stillgelegt werden könnten.
Das Neuron $n_{i,j}$ erhält durch die eingehenden Verbindungen die sogenannte Netzeingabe

$$\mathrm{net}_{i,j} = \sum_{k=1}^{N_{j-1}} w^{j-1}_{k,i} \; o_{k,j-1}. \tag{5.30}$$

Die Aktivierung $z_{i,j}$ des Neurons $n_{i,j}$ ist eine Bewertung der Netzeingabe mit einer sogenannten Aktivierungsfunktion. Bei den hier betrachteten künstlichen neuronalen Netzen besitzen in jeder Schicht j die enthaltenen Neuronen $n_{i,j}$ die gleiche Aktivierungsfunktion f_j.
Die Aktivierung ergibt sich durch die Anwendung von f_j auf die Netzeingabe abzüglich eines individuellen Bias-Wertes $b_{i,j}$:

$$z_{i,j} = f_j\left(\mathrm{net}_{i,j} - b_{i,j}\right) \tag{5.31}$$

Die Aktivierungsfunktion f_j muss zur Wahrung der Approximations-Eigenschaften in hinreichend vielen verdeckten Schichten eine sigmoidale Gestalt haben [Cyb89]. In der Regel wird hierfür der Tangens-Hyperbolicus verwendet:

$$\begin{aligned}\tanh(x) =& \frac{\exp[x] - \exp[-x]}{\exp[x] + \exp[-x]} \\ =& \frac{2}{1 + \exp[-2\ x]} - 1\end{aligned}$$

In obiger Darstellung lässt er sich als lineare Transformation der logistischen Funktion

$$L(x) = \frac{1}{1 + \exp[-x]}$$

interpretieren, wobei beim Tangens-Hyperbolicus das Argument gegenüber L zusätzlich skaliert ist. Hierdurch hat tanh für $x = 0$ die Steigung eins, L hingegen $\frac{1}{4}$.
Auch die logistische Funktion kann als Aktivierungsfunktion in künstlichen neuronalen Netzen verwendet werden. Allerdings hat der Tangens-Hyperbolicus den Vorteil, den Wertebereich $]-1{,}1[$ abzudecken, wohingegen die logistische Funktion lediglich Werte aus $]0{,}1[$ annehmen kann. So kann die Ausgabe der logistischen Funktion lediglich zwischen den Zuständen *aktiv* und *nicht aktiv* liegen, wohingegen die Ausgabe des Tangens-Hyperbolicus außerdem noch eine *hemmende* Aktivität der Neuronen abbilden kann. Die logistische Funktion und der Tangens-Hyperbolicus sind in den Abbildungen 5.2a und 5.2b dargestellt.

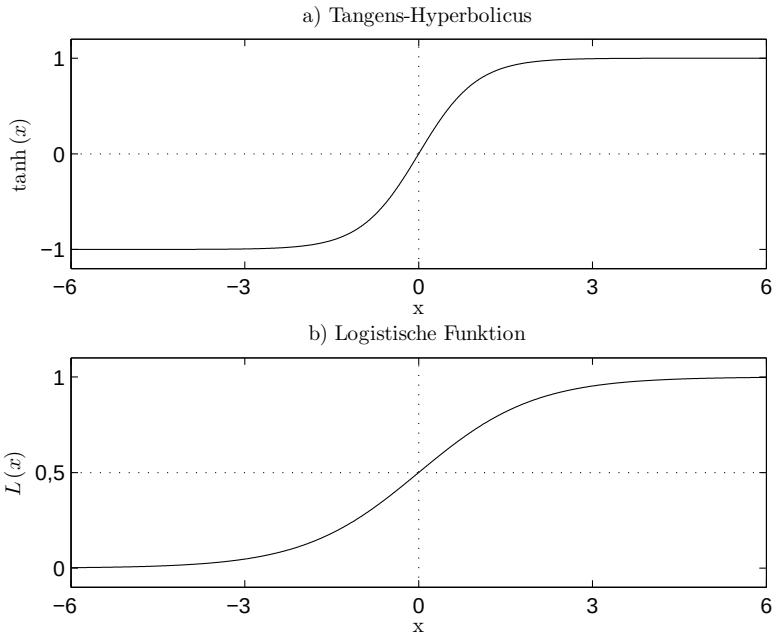

Abbildung 5.2: Sigmoidale Aktivierungsfunktionen

Der Bias b in (5.31) hat die Funktion eines Schwellwertes, den der Netz-Input überwinden muss, um das Neuron in einen höheren Aktivierungszustand zu versetzen.
Allgemein wird für die Generierung des Ausgabewertes $o_{i,j}$ eine Ausgabefunktion $\Phi_{i,j}$ mit

$$o_{i,j} = \Phi_{i,j}(z_{i,j}) \tag{5.32}$$

vorgesehen. Diese wird allerdings in aller Regel als die *Identität* gewählt, da im Allgemeinen davon ausgegangen werden kann, dass ihre Funktionalität bereits in der Aktivierungsfunktion berücksichtigt ist [Zel97]. So gilt auch in dieser Arbeit im Folgenden

$$\begin{aligned} o_{i,j} =& z_{i,j} \\ =& f_j \left(\sum_{k=1}^{N_{j-1}} w_{k,i}^{j-1} \; o_{k,j-1} \; - b_{i,j} \right). \end{aligned}$$

Häufig wird bei dem Neuron in der Ausgabeschicht nicht nur auf eine spezielle Ausgabefunktion verzichtet, sondern sogar an Stelle einer sigmoidalen Funktionen für die Aktivierungsfunktion ebenfalls die Identität verwendet, was einen größeren Wertebereich für die Netzausgabe bei weniger Rechenoperationen ermöglicht.
In diesem Spezialfall erhält man die Netzausgabe direkt zu

$$o_{1,L} = \sum_{k=1}^{N_{L-1}} w_{k,i}^{L-1} \; o_{k,L-1} \; - b_{i,L}.$$

Da zu den Neuronen der ersten Schicht keine Verbindungen hin gerichtet sind, gilt dort der Spezialfall, dass die Ausgabe $o_{i,1}$ dem i-ten Wert des Eingabe-Datensatzes entspricht ($i = 1, \ldots, N_1$). Dies geschieht, trotz fehlender Netzeingabe, im Einklang mit dem Neuronen-Modell, wenn man für die einzelnen Eingabe-Neuronen jeweils eine Ausgabefunktion zugrunde legt, die sämtliche übergebenen Werte stets auf den entsprechenden konstanten Wert des Datensatzes abbildet.
Die Unterteilung der freien Parameter des Netzes in Verbindungsgewichte und Bias-Werte erschwert deren Bestimmung durch etablierte Trainingsverfahren. Mit dem Ziel der Vereinheitlichung werden daher auch die Bias-Werte als Verbindungsgewichte formuliert. Möglich wird dies durch die rein formelle Einführung sogenannter On-Neuronen $n_{0,j}$ in den Schichten $j = 1, \ldots, L-1$ mit der festen Ausgabe $o_{0,j} \equiv 1$ [Zel97] (vgl. Abbildung 5.1).
Die von dem On-Neuron $n_{0,j}$ der j-ten Schicht abgehenden Verbindungen erhalten die zusätzlichen Gewichte $w_{0,i}^{j}$ mit $i = 1, \ldots, N_{j+1}$. Der Netzinput des Neurons $n_{i,j+1}$ lautet dann

$$\text{net}_{i,j+1} = w_{0,i}^{j} + \sum_{k=1}^{N_j} w_{k,i}^{j} \; o_{k,j}. \tag{5.33}$$

Der Vergleich von (5.33) mit (5.31) und (5.30) zeigt, dass durch die zusätzlichen Gewichte $w_{0,i}^{j}$ formell für die Berechnung der Aktivierung auf die Bias-Werte verzichtet werden kann.
Auf den ersten Blick stellt das Konzept der On-Neuronen mit dem Wegfall der Bias-Werte einen Eingriff in das Neuronen-Modell dar, dessen Netzeingabe und Aktivität

gemäß (5.30) und (5.31) definiert sind. Mathematisch sind die beiden Netz-Varianten aber absolut gleichwertig mit dem Vorteil bei der zweiten Variante, dass es nur einen Typ freier Parameter gibt.
Da außerdem letzten Endes für die Erstellung und die Anwendung der Feed-Forward-Netze nur die Ausgabewerte der Neuronen von Relevanz sind und die Netzeingaben vor diesem Hintergrund lediglich Zwischenergebnisse für die Berechnung der zugehörigen Ausgaben verkörpern, stellt die Verwendung der zweiten Variante kein nennenswertes Problem dar und ist die gängige Praxis.
Abschließend kann mit dieser Einführung der allgemeinen Feed-Forward-Struktur die Anzahl n_{DOF} der freien Parameter der darauf aufgebauenden Netze entsprechend der Anzahl der Schichten L und der Anzahlen $N_1, N_2, \ldots, N_{L-1}$ von Neuronen innerhalb der einzelnen Schichten angegeben werden zu

$$n_{\mathrm{DOF},L} = 1 + N_{L-1}\ \left(2 + N_{L-2}\right) + \sum_{j=1}^{L-3} \left(N_j + 1\right)\ N_{j+1},$$

wobei $N_L = 1$ bereits berücksichtigt wurde.
Für den häufig anzutreffenden Fall einer einzigen verdeckten Schicht fällt der hintere Term weg und es ergibt sich der Spezialfall

$$n_{\mathrm{DOF},3} = 1 + N_2\ \left(2 + N_1\right).$$

Hier ist speziell N_1 die durch die Eingabedaten-Struktur festgelegte Anzahl der Neuronen der Eingabeschicht und N_2 die Anzahl der Neuronen in der verdeckten Schicht. In der Praxis ist N_1 also eine Randbedingung und N_2 der wesentliche Design-Freiheitsgrad des Netzes.
Weitere Freiheiten zu dem Design sind die zu verwendenden Aktivierungsfunktionen sowie die Methode, mit der das Netz trainiert werden soll.

5.5.2 Backpropagation und gradientenbasierte Trainingsverfahren

Das überwachte Training eines künstlichen neuronalen Netzes mit Feed-Forward-Struktur geschieht bezüglich einer festen Menge $\mathbf{D}$ von m Datensätzen $\mathbf{d}^l$, die jeweils aus Eingabewerten $\mathbf{x}^l \in X$ und dem zugehörigen Sollwert $y^l \in Y$ bestehen. Für die Menge der möglichen Zustands-Beschreibungen kann hier $X \subset \mathbb{R}^{N_1}$ und für die Menge der möglichen Ausgabewerte $Y \subset \mathbb{R}$ angenommen werden.
Die Beschreibung des Netzfehlers bezüglich der Trainingsdaten $\mathbf{D}$ geschieht in der Regel über den Mittelwert der quadrierten Abweichungen

$$E_{\mathrm{MSE},\mathbf{D}} = \frac{1}{m} \sum_{l=1}^{m} \left(o_{1,L}^{(l)} - y^l\right)^2. \tag{5.34}$$

Hierbei bedeutet der obere Index l in $o_{1,L}^{(l)}$, dass es sich jeweils um die Netzausgabe zu dem Eingabezustand $o_{1,1}^{(l)} = x_1^l, \ldots, o_{N_1,1}^{(l)} = x_{N_1}^l$ handelt (vgl. Abschnitt 5.5.1).
Das Ziel des Trainings eines künstlichen neuronalen Netzes besteht darin, die unbekannten Gewichte so zu wählen, dass für eine gegebene Menge $\mathbf{D}$ von Trainingsdaten der Netzfehler $E_{\mathrm{MSE},\mathbf{D}}$ so klein wie möglich wird.
Die gleichen Gewichte erhält man durch die Minimierung des Ausdrucks

$$F_{\mathrm{Ziel}}(\mathbf{w}) = \frac{1}{2} \sum_{l=1}^{m} \left(o_{1,L}^{(l)} - y^l \right)^2, \tag{5.35}$$

der als Funktion des Tupels

$$\mathbf{w} = \left(w_{i_1,i_2}^j \right)_{j \in \{1,\ldots,L-1\}, i_1 \in \{1,\ldots,N_j\}, i_2 \in \{1,\ldots,N_{j+1}\}}$$

formuliert werden kann, und dessen Verwendung verglichen mit (5.34) im Folgenden zu einer besseren Lesbarkeit beiträgt.
In dieser Formulierung beschreibt das Training ein unrestringiertes Optimierungsproblem mit der zu minimierenden Zielfunktion F_{Ziel} und den freien Parameter w_{i_1,i_2}^j, $j = 1, \ldots, L-1$, $i_1 = 1, \ldots, N_j$, $i_2 = 1, \ldots, N_{j+1}$, zusammengefasst in dem Tupel $\mathbf{w}$.
Nach der Kettenregel für die Differentiation ergeben sich die partiellen Ableitungen der Zielfunktion nach den Gewichten jeweils zu

$$\frac{\partial}{\partial w_{i_1,i_2}^j} F_{\mathrm{Ziel}}(\mathbf{w}) = \sum_{l=1}^{m} \left(o_{1,L}^{(l)} - y^l \right) \frac{\partial o_{1,L}^{(l)}}{\partial w_{i_1,i_2}^j}.$$

Allgmein lässt sich ebenfalls unter Verwendung der Kettenregel zeigen:

$$\frac{\partial o_{i_o,j_o}^{(l)}}{\partial w_{i_1,i_2}^j} = f'_{j_o}\left(\sum_{k=0}^{N_{j_o-1}} w_{k,i_o}^{j_o-1} \; o_{k,j_o-1} \right) \cdot \begin{cases} 0 & \text{falls } (j_o \leq j) \vee (j_o = j+1 \wedge i_o \neq i_o) \\ o_{i_1,j}^{(l)} & \text{falls } j_o = j+1 \wedge i_o = i_2 \\ \sum_{k=0}^{N_{j_o-1}} w_{k,i_o}^{j_o-1} \frac{\partial o_{k,j_o-1}^{(l)}}{\partial w_{i_1,i_2}^j} & \text{sonst} \end{cases}$$

Da bei dieser Vorgehensweise die Ausgabefehler entgegen der eigentlichen Verarbeitungsrichtung durch das Netz propagiert werden, wird der zugehörige Algorithmus als (Error-) Backpropagation-Algorithmus bezeichnet. Es ist hierdurch (für einen Computer) unmittelbar möglich, mit wenig Aufwand und hoher Genauigkeit zu einer gegebenen Netzausprägung den Gradienten $\nabla F_{\mathrm{Ziel}}(\mathbf{w})$ der Zielfunktion (5.35) zu bestimmen.
Diese Tatsache eröffnet Wege, die Netzgewichte durch Methoden der unrestringierten, nichtlinearen Optimierung zu bestimmen. Allen voran die Gradienten-Abstiegsverfahren erlangten in der Vergangenheit hierbei eine große Bedeutung [Sei00]. Sie basieren auf der

Tatsache, dass der Wert der Zielfunktion, ausgehend von einem Gewichts-Tupel $\mathbf{w}$ in einer hinreichend kleinen Umgebung von $\mathbf{w}$ am stärksten in Richtung des negativen Gradienten $-\nabla F_{\text{Ziel}}(\mathbf{w})$ der Zielfunktion abnimmt. Es wird daher auch als das Verfahren des steilsten Abstiegs bezeichnet.
Bei diesen iterativen Verfahren wird im Wesentlichen in jedem Schritt k eine Verbesserung der Gewichte durch die Vorschrift

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \, \nabla F_{\text{Ziel}}(\mathbf{w}_k) \tag{5.36}$$

erreicht. Dabei ist $(\alpha_k)_{k\in\mathbb{N}}$ eine Folge positiver (skalarer) Schrittweiten.
Vor dem Hintergrund des Trainings künstlicher neuronaler Netze werden die α_k auch als Lernraten bezeichnet.
Die gängigen Implementierungen derartiger Trainingsmethoden verwenden sehr unterschiedliche Arten der Lernratenanpassung [Pla98]. Weitere Methoden erweitern die Iterationsvorschrift (5.36) um einen sogenannten Trägheitsterm (eng.: momentum), der gewissermaßen die Gewichtsänderung in zweiter Ordnung miteinbezieht [Rie92],[Ist02].
Eine weitere, von der grundsätzlichen Struktur ähnliche Gruppe von Verfahren sind die sogenannten Newton-Verfahren. Diese werden mit dem konkreten Ziel eingesetzt, iterativ ein Parametertupel $\mathbf{w}^*$ zu ermitteln, für das der Gradient der Zielfunktion verschwindet.
Das exakte Newton-Verfahren ist allgemein für Abbildungen $g : \mathbb{R}^n \to \mathbb{R}^n$ definiert und erzeugt dabei, ausgehend von einer Näherung $\mathbf{x}_0 \in \mathbb{R}^n$, eine Folge reeller n-Tupel, die gegen ein $\mathbf{x}^* \in \ker(g)$ konvergiert, d.h $g(\mathbf{x}^*) = \mathbf{0} \in \mathbb{R}^n$.
Die Konvergenzgeschwindigkeit ist dabei lokal quadratisch [Gei99].
Die hierfür nötige Iterationsvorschrift ergibt sich allgemein zu

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}g(\mathbf{x}_k)^{-1} \, g(\mathbf{x}_k),$$

wobei mit $\mathbf{J}g$ die Jakobi-Matrix der Abbildung g bezeichnet ist.
Fasst man die Zielfunktion F_{Ziel} als ein Potential auf, so lässt sich das Newton-Verfahren auf das induzierte Gradientenfeld ∇F_{Ziel} anwenden, um somit ein Tupel $\mathbf{w}^*$ von Netzgewichten zu ermitteln, für das $\nabla F_{\text{Ziel}}(\mathbf{w}^*) = \mathbf{0}$ gilt.
Da für diesen Spezialfall in jedem Iterationsschritt die Jakobi-Matrix des Gradienten eines höherdimensionalen Funktionals benötigt wird, lässt sich die zugehörige Vorschrift mit der Hesse-Matrix $\mathbf{H}F_{\text{Ziel}}$ der Zielfunktion formulieren:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}F_{\text{Ziel}}^{-1}(\mathbf{w}_k) \;\; \nabla F_{\text{Ziel}}(\mathbf{w}_k)$$

In der Praxis wird dieser Schritt nicht durch eine Invertierung der Hesse-Matrix mit anschließender Matrix-Vektor-Multiplikation, sondern durch die Schritte

$$\begin{aligned} \mathbf{H}F_{\text{Ziel}}(\mathbf{w}_k) \;\; \Delta &= -\nabla F_{\text{Ziel}}(\mathbf{w}_k), \\ \mathbf{w}_{k+1} &= \Delta + \mathbf{w}_k \end{aligned}$$

über die Lösung eines linearen Gleichungssystems realisiert.
Das exakte Newton-Verfahren überzeugt durch seine guten lokalen Konvergenzeigenschaften. Dagegen birgt es aber in seiner allgemeinsten Form auch zahlreiche Probleme, denen in sogenannten inexakten Newton-Verfahren [Dem82] begegnet wird. So existieren beispielsweise regularisierte Newton-Verfahren [Mey07], die in den Fällen Abhilfe schaffen, in denen $\mathbf{H}F_{\text{Ziel}}(\mathbf{w}_k)$ tendenziell singulär wird. Stellt die exakte Bestimmung der Hesse-Matrix ein Problem dar, so kommen Quasi-Newton-Verfahren zum Einsatz, bei denen diese durch geeignete Approximationen ersetzt wird [Bro70],[Fle70],[Gol70],[Sha70].
Im Allgemeinen ist das Newton-Verfahren mit seinen Varianten in Hinblick auf die Startwerte nicht sehr robust. Liegen diese nicht bereits nahe genug an einem stationären Punkt, so ist das Konvergenzverhalten des Verfahrens nicht vorhersagbar. Möglich sind Oszillation und sogar Divergenz.
Da aber gerade vor dem Hintergrund des Trainings künstlicher neuronaler Netze keine praktisch relevanten Möglichkeiten existieren, die Netze zuverlässig mit hinreichend guten Startlösungen zu initialisieren, bieten sich die Newton-Verfahren für diesen Einsatz nicht direkt an. An dieser Stelle knüpft das Verfahren nach Levenberg und Marquardt an, das die Robustheit der Gradientenverfahren mit den sehr guten lokalen Konvergenzeigenschaften der Newton-Verfahren eindrucksvoll kombiniert [Lev44],[Mar63]. Es ist eine der am häufigsten verwendeten Trainingsmethoden für künstliche neuronale Feed-Forward-Netze.

5.5.3 Bayessche Regularisierung

Aus Gründen, die bereits in Abschnitt (5.2.2) genannt wurden, ist insbesondere das Training künstlicher neuronaler Netze mit begrenzter Anzahl an Trainingsdaten anfällig für das Problem der Überanpassung, das wiederum ein schlechtes Generalisierungsverhalten der erstellten Netze zur Folge hat.
In der Regel wird daher bei dem Trainieren künstlicher neuronaler Netze von der in (5.2.3) erläuterten Möglichkeit der Data-Division Gebrauch gemacht. Der Nachteil ist dabei die weitere Reduzierung der für das eigentliche Training verfügbaren Daten sowie die Tatsache, das sich durch die Data-Division nichts an der grundlegenden Art und Weise ändert, wie genau die Netzgewichte an die Trainingsdaten angepasst werden.
Da in der Zielfunktion (5.35) analog zu der Situation der einfachen linearen Regression keinerlei Restriktionen an die freien Parameter enthalten sind, die einer Überanpassung entgegen wirken könnten, ist diese Trainingsmethode prinzipiell sehr empfindlich bezüglich der jeweiligen Startkonfiguration der Netzgewichte, die üblicherweise zufällig ermittelt wird.
Bei künstlichen neuronalen Netzen besitzt die Fehlerfunktion bezüglich der Trainingsdaten im Allgemeinen zahlreiche lokale Minima, so dass gradientenbasierte Verfahren in der Regel eine Parameterkonfiguration als Ergebnis liefern, die einem lokalen Minimum der Fehlerfunktion in der Nähe der jeweiligen Startkonfiguration entspricht. In vielen Fällen wird tatsächlich für nahezu jeden neuen Trainingsversuch mit individueller Startkonfiguration auch ein neues Optimum ermittelt, da der Parameterraum in der Regel sehr groß ist.

40 bis 60 freie Netzgewichte sind bereits bei den kleineren künstlichen neuronalen Netzen durchaus übliche Werte.
Diesem Problem wird häufig so begegnet, dass das Training tatsächlich für viele verschiedene Startkonfigurationen durchgeführt wird, wobei am Ende das austrainierte Netz mit den besten Ergebnissen bezüglich aller Daten-Partitionen gewählt wird. Dies ist in vielerlei Hinsicht keine optimale Vorgehensweise.
Erstens ist auf Grund der angesprochenen Größe des Parameterraums die benötigte Anzahl von Trainingsdurchläufen zur Abdeckung aller relevanten Startkonfigurationen viel zu groß. Tatsächlich werden die auf diese Weise in endlicher Zeit ermittelbaren lokalen Minima der Fehlerfunktion nicht annähernd repräsentativ sein.
Zum Zweiten ist dieser Ansatz *greedy* in der Hinsicht, dass aus einer Reihe ermittelter trainierter Netze einfach das Beste gewählt wird. Für die Beurteilung, welches Netz das Beste ist, werden aber wieder nur die aktuell verfügbaren Daten verwendet, was insgesamt auf natürliche Weise wieder zu einer schlechten Generalisierungfähigkeit führen wird [Guy06]. Wenn bei der Netzauswahl für die Beurteilung außerdem die Anwendung auf die Testdaten berücksichtigt wird, so stellen diese insgesamt keine reine Testpartition mehr dar, da sie für die Ermittlung des zu verwendenden trainierten Netzes benutzt worden sind.
Unter Anerkennung dieser Probleme wurde entschieden, eine Trainingsmethode zu verwenden, die nicht ausschließlich auf die Minimierung der Fehlerfunktion abzielt, sondern außerdem in Anlehnung an Abschnitt (5.2.4) die Anpassung der Netzgewichte auf natürlich gerechtfertigte Weise restringiert.
Die Trainingsmethode der Bayesschen Regularisierung erfüllt diese Anforderung auf beeindruckend effektive Weise. Das Verfahren wird in [DF97] beschrieben und geht zu wesentlichen Anteilen auf die Arbeit von McKay [Mac91] zurück, in der er Regularisierungsmethoden vor dem Hintergrund der Interpolation verrauschter Daten über den Bayes-Ansatz [Dee81] aufzeigt.
Das Trainingsziel bei der Bayes'schen Regularisierung ist analog zu (5.4) die Minimierung einer Linearkombination aus Fehlerquadratsumme E_D und Summe der quadrierten Netzgewichte E_W:

$$F = \beta\ E_D + \alpha\ E_W$$

Die Steuerparameter α und β regulieren, in welchem Maße E_W gegenüber E_D verringert werden muss, um letzlich F zu verringern. Sie werden mit $\alpha = 0$ und $\beta = 1$ zu Beginn des Trainings initialisiert und im Zuge des Verfahrens fortlaufend angepasst.
Das Verfahren selber beruht zu wesentlichen Teilen auf dem Levenberg-Marquardt-Verfahren, mit dem Unterschied, dass nach jedem Levenberg-Marquardt-Schritt zunächst durch

$$\begin{aligned}\mathbf{H} =& 2\beta\,(\mathbf{J}F)^{\mathrm{T}}\,(\mathbf{J}F) + 2\alpha\mathbf{I}\\ \approx& \mathbf{H}F\end{aligned}$$

eine Gauß-Newton-Approximation der Hesse-Matrix von F bestimmt wird, wobei $\mathbf{J}F$ die Jakobi-Matrix von F ist.
Im Anschluss an die Bestimmung von $\mathbf{H}$ wird durch

$$\gamma = N - 2\alpha \ \mathrm{tr}\,(\mathbf{H})^{-1}$$

die sogenannte Anzahl der effektiv genutzten Netzgewichte ermittelt. N ist dabei die gesamte Anzahl der Netzgewichte und $0 \leq \gamma \leq N$.
Schließlich werden die Steuerparameter gemäß

$$\alpha = \frac{\gamma}{2\ E_W}$$

und

$$\beta = \frac{n - \gamma}{2\ E_D},$$

wobei n die Anzahl der verwendeten Trainingsdaten bezeichnet, aktualisiert und der nächste Levenberg-Marquard-Schritt durchgeführt.
Auf diese Weise wird bis zur Konvergenz von α und β, und damit des Verfahrens, fortgefahren. Angemerkt sei hierbei, dass sich dadurch in jedem Schritt auch die Zielfunktion F ändert und somit auch das gesuchte Minimum.

5.5.4 Paralleles und sequentielles Lernen

In Abschnitt 5.5.2 wurde für allgemeine Trainingsverfahren die Zielfunktion (5.35) vorgestellt. Sie entspricht der allgemeinsten Form des überwachten Lernens. Hier geschieht die Gewichtsanpassung in den einzelnen Iterations-Schritten jeweils bezüglich eines fest gewählten Satzes von Trainingsdaten. Diese Art des Trainings kann als paralleles Lernen (eng.: batch training) bezeichnet werden.
Die beim parallelen Lernen verwendeten Daten können alle zu diesem Zeitpunkt bekannten Paare von Ein- und Ausgabe-Daten ganz oder teilweise enthalten. Insbesondere kann auch ein einzelner Datensatz (m=1) trainiert werden.
Ein anderer Fall des Lernens verläuft derart, dass das Netz der Reihe nach mit jeweils einem Datensatz geschlossen trainiert wird. Dabei wird anfänglich ein Netz unabhängig von den Trainingsdaten initialisiert und gemäß F_{Ziel} mit nur einem der Trainings-Datensätze trainiert. Das so neu entstandene Netz wird sukzessiv mit den weiteren Datensätzen auf die gleiche Weise weiter angepasst. Diese Vorgehensweise lässt sich entsprechend als sequentielles Lernen bezeichnen. Im Englischen wird für diese Vorgehensweise die Bezeichnung *online training* verwendet.
Allgemein spricht man bei dem weiteren Anlernen eines bereits trainierten Netzes mit neuen Daten auch von einer Adaptierung des Netzes an die entsprechenden Daten.
Der wesentliche Unterschied zwischen parallelem und sequentiellem Lernen besteht darin,

dass bei der parallelen Variante gesichert ist, dass das Netz an keinen der Datensätze bevorzugt angepasst ist. Das Ergebnis stellt stets einen Kompromiss im Sinne der kleinsten Fehlerquadrate bezüglich aller Daten dar.
Beim sequentiellen Lernen und beim Adaptieren einzelner Datensätze ist dies nicht notwendiger Weise der Fall. Hier wird die Änderung der Verbindungsgewichte, die durch den Netzfehler bezüglich des ersten trainierten Datensatzes hervorgerufen wurde, durch die sukzessiven Trainings-Durchläufe immer wieder überschrieben. Demnach besitzt der zuletzt trainierte Datensatz potentiell den größten Gesamteinfluss und der erste Datensatz den geringsten.
Insbesondere hat beim sequentiellen Lernen die Reihenfolge der verwendeten Datensätze wesentlichen Einfluss auf das Ergebnis. Es wird daher empfohlen, beim sequentiellen Lernen mehrere Runden mit jeweils allen Datensätzen durchzuführen. Die Reihenfolge der Datensätze sollte dabei jeweils zufällig erfolgen [Wil03], damit die Inhalte der Daten und nicht etwa deren Reihenfolge von dem Netz erlernt werden. Die hierdurch ausgedrückte Irrelevanz der Abfolge der Trainingsdaten liegt prinzipiell auch dem parallelen Lernen zu Grunde.
Bei dem Adaptieren des Netzes an einzelne neue Datensätze besteht diese Option nicht. Hier kann allerdings eine entsprechende zusätzliche Gewichtung für die Anpassung der Netzgewichte dafür sorgen, dass die Erkenntnisse aus den neuen Daten sich nur in beschränktem Maße auf die Gewichtungen der Neuronen-Verbindungen auswirken, und den Stand des Netzes seit dem ursprünglichen Training dadurch nicht beliebig umstoßen können.
Fasst man das Adaptieren als einen der Natur nachempfundenen Prozess des täglichen Lebens bezogen auf den Menschen auf, so kann die Tatsache, dass dessen Lernfähigkeit mit zunehmendem Alter abnimmt, analog durch eine streng monoton fallende Folge positiver Gewichtungen modelliert werden, mit der die durch die Adaptierung ermittelten Änderungen der Verbindungsgewichte weiter abgeschwächt werden können. Hierdurch würde zusätzlich eine Konvergenz der Adaptierung erzwungen.
Das Konzept der Adaptierung trainierter Netze an neue Daten oder Erkenntnisse spielt eine wichtige Rolle auf dem Gebiet der künstlichen Intelligenz und findet beispielsweise in der Robotik Anwendung [Con00],[Li03].

5.5.5 Selbstorganisierende Karten nach Kohonen

Für die Beschreibung der selbstorganisierenden Karten nach Kohonen sei für den Moment der Blick wieder auf die Problematik gerichtet, bei der eine für Vorhersagemodelle geeignete Daten-Repräsentation in Form kleinerer Kombinationen beschreibender Merkmale gefunden werden muss.
Eine zentrale und dabei noch recht schwache Forderung, die ein entsprechendes Verfahren erfüllen sollte, ist die, dass Datensätze, die sich bezüglich dieser Repräsentation auf eine noch festzulegende Weise ähneln, auch in einer entsprechenden Ähnlichkeitsbeziehung bezüglich der Komfortbewertung zueinander stehen. Weiter müssten bei starken Abwei-

chungen in der subjektiven Bewertung auch deutliche Unterschiede in der Repräsentation der objektiven Daten vorliegen.
Die Umkehrung dieser Implikation ist hingegen nicht zu fordern, da die Komfortbewertung nicht als injektive Abbildung angesehen werden kann. So sind Situationen leicht vorstellbar, in denen objektive Repräsentationen zu Datensätze mit ähnlichen subjektiven Bewertungen stark variieren.
Diese abstrakten Zusammenhänge der Änhlichkeitsbeziehungen auf ojektiver und subjektiver Seite sind dabei grundlegend und somit unabhängig von der Wahl des Vorhersagemodells. Es handelt sich um strukturelle Eigenschaften, die bei Verwendung der jeweiligen Repräsentation der Menge der objektiven Daten innewohnen muss.
Eine Möglichkeit, solche Strukturen sichtbar zu machen, sind die selbstorganisierenden Karten nach Kohonen (eng.: self-organising maps, kurz som) [Koh82],[Koh90]. Häufig findet man auch die Bezeichnung Kohonen-Netze, die der Einfachheit halber im Folgenden verwendet wird.
Bei Kohonen-Netzen handelt es sich um künstliche neuronale Netze mit einer internen Struktur, die sich stark von der bekannten, in Abschnitt 5.5.1 vorgestellten, Feed-Forward-Struktur unterscheidet. Die Kohonen-Netze besitzen bei Annahme einer n-dimensionalen Eingabedaten-Menge $X \subset \mathbb{R}^n$ eine Eingabe-Schicht mit n Neuronen und eine verdeckte Schicht, auf der die Neuronen in einem festen, regulären Gitter angeordnet sind.
In den meisten Fällen kommt in der verdeckten Schicht ein zweidimensionales Gitter zum Einsatz [Koh00]. Eine Erweiterung auf die dritte Dimension (und höher) ist zwar prinzipiell möglich, aber kaum (bzw. gar nicht) von praktischer Relevanz. Vorwegnehmend sei hier angemerkt, dass die Anordnung der Neuronen auf diesem Gitter im Wesentlichen eine übersichtliche Visualisierung der gefundenen Ähnlichkeitsbeziehungen ermöglichen soll. Somit birgt ein dreidimensionales Gitter zwar vergleichsweise mehr Möglichkeiten der Abbildung von Ähnlichkeitsbeziehungen, vermag es aber bereits nicht mehr, diese übersichtlich darzustellen.
Schließlich bietet sich in speziellen Fällen sogar die Verwendung eines eindimensionalen Gitters, also einer Kette, an, wenn der Anwendung als Ziel eine entsprechend einfache Anordnung der Datensätze zugrunde liegt. Als Beispiel kann durch ein Kohonen-Netz die Lösung des aus der kombinatorischen Optimierung bekannten und vielen technischen Problemen zugrunde liegenden Travelling-Salesman-Problems [Gut07] näherungsweise ermittelt werden. Hierfür werden die zu besuchenden Positionen als zweidimensionale Datensätze ($n = 2$) entsprechend ihrer Ähnlichkeit in Hinblick auf eine geringe euklidische Distanz in der Daten-Repräsentation auf einem eindimensionalen Ring angeordnet. Schabauer hat vor diesem Hintergrund ein Verfahren vorgestellt, mit dem TSP-Probleme mit tausenden von Städten heuristisch mit selbstorganisierenden Karten bearbeitet werden können [Sch05].
Dieses Beispiel zeigt unter Anderem bereits, dass es durchaus Sinn machen kann, den Raum G für die Visualisierung der ermittelten strukturellen Eigenschaften bezüglich der Abstandsmessung mit periodischen Randbedingungen zu versehen. So kann eine eindimensionale Anordnung nicht nur als offene Kette sondern auch, wie im obigen Fall, als

ein geschlossener Ring geschehen. In der zweiten Dimension ist entsprechend neben der einfachen Anordnung auf einem rechteckigen Gitter die Oberfläche eines Torus nicht nur möglich, sondern außerdem auch noch vergleichsweise übersichtlich darstellbar.
Der Gewinn der Verwendung periodischer Randbedingung ist in beiden Fällen eine Erweiterung der Möglichkeit des Kohonen-Netzes, vorhandene strukturelle Ähnlichkeitsbeziehungen darzustellen. Hierzu sind keinerlei Änderungen an der Netzstruktur nötig, sondern lediglich eine Anpassung der Metrik d_G für die geeignete rechnerische Bestimmung von Abständen in G. Die Metrik des Eingaberaumes sei im Folgenden entsprechend mit d_X bezeichnet.
Jedes verdeckte Neuron besitzt neben seiner festen Position $\mathbf{p}$ in dem Neuronengitter außerdem einen n-dimensionalen Gewichtsvektor $\mathbf{w}$. Durch die Kapselung beider Konstrukte in einem Neuron bildet letzteres eine Schnittstelle zwischen dem n-dimensionalen Eingabedaten-Raum X und dem Raum G für die Visualisierung, der die Dimension n_G des Neuronen-Gitters der verdeckten Schicht besitzt.
Die Aufgabe des Kohonen-Netzes besteht in einer Anordnung der übergebenen Datensätze aus X in G. Diese Abbildung von einem n-dimensionalen Raum in einen Raum der Dimension n_G soll dabei topologie-erhaltend sein dahingehend, dass die sich jeweils einstellenden sichtbaren Abstände der Datensätze in G die entsprechenden Abstände bezüglich der Metrik d_X in X widerzuspiegeln vermögen.
Die Zuordnung eines Gitter-Neurons zu einem Datensatz geschieht nach dem winner-take-all-Prinzip [Laz89]. Dabei werden der Reihe nach die Gewichtsvektoren aller Gitter-Neuronen mit dem Vektor der Eingabedaten des Datensatzes bezüglich d_X verglichen. Das Neuron, dessen Gewichtsvektor $\mathbf{w}$ bezüglich d_X am geringsten von dem Eingabevektor $\mathbf{x}$ abweicht, wird, bezüglich des aktuellen Datensatzes, als Champion-Neuron bezeichnet.
Die Zuordnung von Datensätzen zu Gitter-Neuronen, und damit zu Gitterpositionen, wird in zwei Situationen benötigt. Zum Ersten im Zuge des unüberwachten Lernens, in dessen Rahmen die Gewichtsvektoren der verdeckten Neuronen angepasst werden. Zum Zweiten bei der Visualisierung der Daten-Anordnung mit dem ausgelernten Kohonen-Netz. Vor dem Hintergrund von Kohonen-Netzen wird der Trainingsvorgang auch als Ausbreiten der Kohonen-Karte und das ausgelernte Kohonen-Netz entsprechend als ausgebreitete Karte bezeichnet.
Die Anpassung der Gewichtsvektoren der Gitter-Neuronen wird auch bei Kohonen-Netzen als Training oder Lernen bezeichnet. Allerdings werden in diesem Fall keine Sollwerte für das Training benötigt, sondern nur die objektiven Daten selber.
Die Sollwerte sind dennoch vor dem Hintergrund der Verwendung der Kohonen-Netze im vorliegenden Fall von großer Bedeutung, da durch sie die ermittelte Anordnung der Datensätze erst auf sinnvolle Weise bewertet werden kann.
Das unüberwachte Training der Kohonen-Netze besteht aus einer Sequenz von $i = 1, \ldots, N$ Trainings-Epochen. Innerhalb einer jeden Epoche i wird jeder Eingabevektor dem Netz genau einmal präsentiert, wobei die Reihenfolge jeweils zufällig festgelegt wird. Diese Präsentation umfasst die folgenden Schritte:

1. Ermittlung des Champion-Neurons bezüglich des aktuellen Eingabevektors $\mathbf{x}$ sowie dessen Gitterposition $\mathbf{p}_C$
2. Für den Gewichtsvektor $\mathbf{w}^j$ und die Gitterposition $\mathbf{p}^j$ eines jeden verdeckten Neurons:
   a) Berechnung des Abstandes $d_j = d_G\left(\mathbf{p}^j,\mathbf{p}_C\right)$ der Gitterpositionen
   b) Ermittlung der Nachbarschaftsbeziehung $h_{j,i} = h\left(d_j,r_i\right)$ zu dem Champion-Neuron in Abhängigkeit der Nachbarschaftsfunktion h, des Abstandes d_j und des aktuellen Nachbarschaftsradius r_i
   c) Anpassung des Gewichtsvektors gemäß $\mathbf{w}^j \rightarrow \mathbf{w}^j + h_{j,i}\ l_i\ \left(\mathbf{x} - \mathbf{w}^j\right)$

In 2c) ist die Lernrate l_i das i-te Glied der Lernratenfolge l und somit innerhalb der Epoche i ein konstanter Wert. Während des Trainings fällt l streng monoton von dem definierten Wert des Anfangsgliedes l_1 in der ersten Epoche auf den definierten Wert des letzten Gliedes l_N in der letzten Epoche ab.
Die Ermittlung der Nachbarschaftsbeziehung eines Gitter-Neurons zu dem jeweiligen Champion-Neuron geschieht unter Verwendung einer festzulegenden Nachbarschaftsfunktion h und dem Nachbarschaftsradius r_i der Epoche i. Die Folge r der Nachbarschaftsradien verhält sich während des Trainings qualitativ analog zu l.
Die Nachbarschaftsfunktion ist eine symmetrische Gewichtungsfunktion, die über der Position des Champion-Neurons zentriert ist und dort den Wert eins besitzt. Mit zunehmender Entfernung von dem Champion-Neuron bezüglich d_G nimmt der Wert der Nachbarschaftsfunktion in Abhängigkeit des aktuellen Nachbarschaftsradius' r_i ab.
Praktisch relevante Beispiele für Nachbarschaftsfunktionen sind die Kegel-Form

$$h_{\text{cone}}\left(d,r_i\right) = \begin{cases} 1 - \frac{d}{r_i}, & \text{falls } d < r_i \\ 0 & \text{sonst} \end{cases}, \tag{5.37}$$

die Gaußsche Glockenkurve

$$h_{\text{gaussian}}\left(d,r_i\right) = \exp\left[-\left(\frac{d}{r_i}\right)^2\right], \tag{5.38}$$

die Cosinus-Form

$$h_{\text{cosine}}\left(d,r_i\right) = \begin{cases} cos\left(\frac{\pi}{2}\frac{d}{r_i}\right), & \text{falls } d < r_i \\ 0 & \text{sonst} \end{cases}, \tag{5.39}$$

oder der unstetige Verlauf

$$h_{\text{step}}(d,r_i) = \begin{cases} 1, & \text{falls } d < r_i \\ 0 & \text{sonst} \end{cases} . \tag{5.40}$$

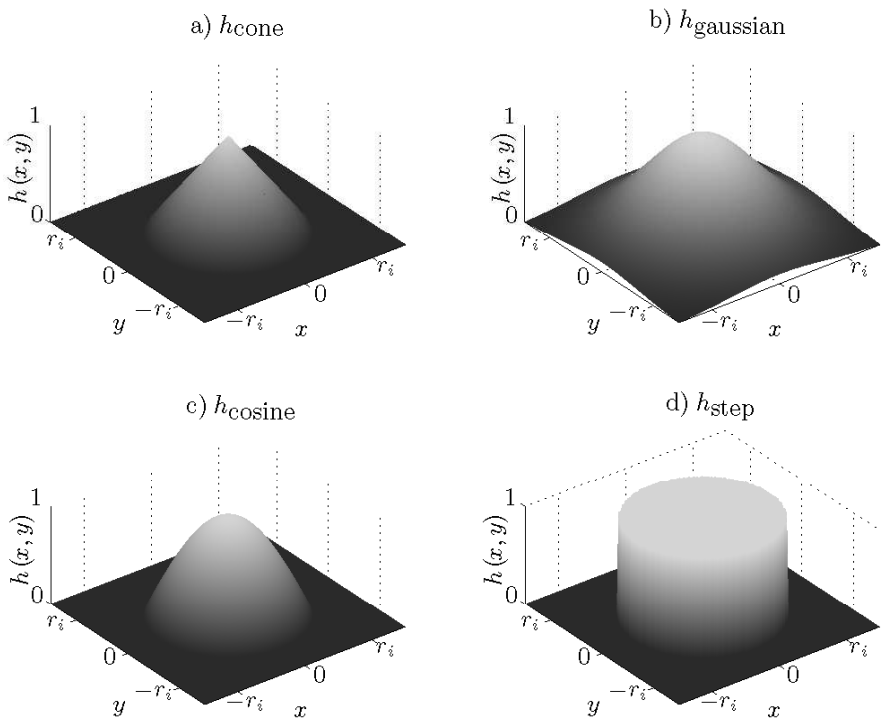

Abbildung 5.3: Die Nachbarschafts-Funktionen (5.37) bis (5.40)

In Abbildung 5.3a bis 5.3d sind die Nachbarschafts-Funktionen (5.37) bis (5.40) dargestellt. Die Iterationsvorschrift in 2c) bewirkt eine gewichtete Verschiebung des jeweiligen Gewichtsvektors $\mathbf{w}$ in Richtung des Eingabevektors $\mathbf{x}$. Diese Gewichtung ist umso stärker, je größer die aktuelle Lernrate l_i ist und je größer die Bewertung der Nachbarschaftsbeziehung von $\mathbf{p}^j$ und $\mathbf{p}_C$ ausfällt. Große Änderungen treten somit vor allem zu Beginn des Trainings (große Lernrate und große Nachbarschaftsradien) und in der unmittelbaren Umgebung der Position $\mathbf{p}_C$ des Champion-Neurons auf.
Der Abfall der Lernrate im Zuge des Trainings über mehrere Epochen bewirkt eine erzwungene Konvergenz der Gewichtsanpassungen. Der Abfall des Nachbarschaftsradius bewirkt entsprechend eine stete Verkleinerung der Einzugsgebiete der Nachbarschaftsbeziehung. Hierdurch wird möglich, dass sich die Gewichtsvektoren der verdeckten Neuronen im Idealfall jeweils auf einen einzigen Datensatz spezialisieren. Durch die bedingte Änderung der Gewichtsvektoren unter Berücksichtigung der Nachbarschaftsbeziehung geschieht diese

Anpassung dahingehend, dass die Gewichtsvektoren bezüglich d_G benachbarter Gitter-Neuronen sich auch bezüglich d_X ähneln.
Die Freiheitsgrade für die Erstellung einer Kohonen-Karte lassen sich in folgender Liste zusammenfassen:

1. Die Dimension n des Eingabedatenraums
   Dies ist im Allgemeinen eine Randbedingung und ergibt sich aus der zugrunde liegenden Problemstellung

2. Die Dimension n_G des Gitters in der verdeckten Neuronenschicht
   Wie bereits angemerkt kommen hier praktisch nur $n_G = 1$ oder $n_G = 2$ in Frage.

3. Die Bemaßung $m_1 \times m_2$ des Neuronen-Gitters
   Von ihr hängt unter anderem ab, inwiefern die gewünschte Verteilung der Datensätze auf die einzelnen Gitterpositionen erreicht werden kann. Daher hängt die Wahl der gesamten Anzahl $m_1 \cdot m_2$ an Neuronen wiederum wesentlich von der Anzahl der Trainings-Datensätze ab, die durch das Kohonen-Netz anzuordnen sind.

4. Die Anzahl der Epochen N
   Sie hat großen Einfluss auf das Ergebnis in Form der ausgebreiteten Karte und sollte keinesfalls zu klein gewählt werden. Größenordnungen von 10^4 bis 10^6 werden empfohlen [Zel97].

5. Die Lernraten-Folge l
   Nach Vorgabe muss l eine streng monoton fallende Folge sein. Obwohl nicht explizit gefordert wird in den meisten Fällen l als Nullfolge mit $l_1 = 1$ und $l_{N+1} = 0$ formuliert, so dass l_N das kleinste noch von Null verschiedene Folgenglied ist. So besteht der Freiheitsgrad hier in der Regel in der qualitativen Beschreibung des Verlaufs für l

6. Die Folge r der Nachbarschaftsradien
   Für sie gelten prinzipiell die gleichen Richtlinien wie für l. Zu beachten ist hier besonders, dass die Folgenglieder r_i stets positiv sein müssen, da sie bei der Berechnung der Nachbarschaftsfunktion häufig im Nenner eines Bruchs stehen (vgl. (5.37),(5.38) und (5.39)).

7. Die verwendete Nachbarschaftsfunktion h
   Auch wenn in der Literatur nicht aussagekräftig bewertet, wird sehr häufig die Kegelform erfolgreich eingesetzt. Ihr linearer Charakter kann bei entsprechenden Optimierungen des Trainingsalgorithmus den Berechnungsaufwand und damit die benötigte Rechenzeit verringern. Dies kann bei Berücksichtigung der empfohlenen großen Anzahl an Epochen ein entscheidender Faktor für die Zweckmäßigkeit des ganzen Verfahrens sein.

8. Die Metrik d_X des Eingaberaumes
   Ihre Wahl ist wesentlich Verantwortlich für die durch das Kohonen-Netz aufgedeckten (und insbesondere die nicht aufgedeckten) Strukturen der übergebenen Daten.

9. Die Metrik d_G des Raumes für die Visualisierung
   Je nach Design der Anordnung der verdeckten Neuronen auf dem Gitter kann d_G im einfachsten Fall als euklidischer Abstand in einer Teilmenge des $\mathbb{R}^2$ gewählt werden, oder, bei komplizierteren Nachbarschaftsbeziehungen auch deutlich komplexer sein. Beispielsweise werden in Software-Umgebungen für die Berechnung von Kohonen-Karten teilweise hexagonale Anordnungen verwendet [Gor10]. Soll, wie zuvor bemerkt, der Abstandsbegriff in G periodische Randbedingungen berücksichtigen, so ist dieses Verhalten ebenfalls in die Metrik d_G einzuarbeiten.

Im Rahmen dieser Arbeit wurde in Matlab eine Klasse für die Erstellung von Kohonen-Netzen von Grund auf erstellt. Es können ein- bis zweidimensionale Gitter beliebiger Größe sowie beliebige Eingaberaumdimensionen verwendet werden. Als Metriken stehen die von den gängigen Vektornormen induzierten Metriken zur Verfügung. Die Folgen der Lernrate und der Nachbarschaftsradien können jeweils linear, exponentiell oder als Cosinus-Verlauf zwischen 0 und $\frac{\pi}{2}$ gewählt werden. Für den qualitativen Verlauf der Nachbarschaftsfunktion stehen die zuvor genannten Beispiele in (5.37) bis (5.40) zur Verfügung.

5.6 Zusammenfassung

Der große Umfang dieses Kapitels rechtfertigt an dieser Stelle eine Zusammenfassung der vorgestellten Konzepte und Verfahren.
Mit der großen Klasse der Vorhersagemodelle stehen an diesem Punkt abstrakte Konstrukte zur Verfügung, die, ausgehend von entsprechend vorbereiteten Eingabe-Daten, Ausgaben berechnen. Strukturell sind sie somit vor dem ausgemachten Ziel der Erstellung von Prädiktionsmodellen für die Vorhersage der Komfortnoten grundsätzlich als geeignet einzustufen.
Ein wesentlicher Aspekt bei den vorgestellten Modellen ist deren Eigenschaft, die eigentliche Funktionalität ausschließlich an Hand gegebener Paare von Eingabe-Daten und zugehörigen Sollausgaben zu erlernen, was sie somit noch attraktiver für die Anwendung in dem Gesamtkontext der Problemstellung erscheinen lässt. Die Nachteile, die diese Art der Modellerstellung mit sich bringt, wurden ebenso vorgestellt wie Möglichkeiten, diesen zu begegnen.
Im Einzelnen wurde aus der großen Klasse der Vorhersagemodelle mit den linearen Regressionsmodellen und den künstlichen neuronalen Netzen mit Feed-Forward-Struktur eine geeignete Auswahl zu verwendender Modelle getroffen. Für deren Erstellung wurde vor dem Hintergrund der Gefahr einer möglichen Überanpassung jeweils ein Verfahren vorgestellt, das eine regularisierte Bestimmung der freien Parameter ermöglicht. Zuvor wurden mit Konzepten wie der Data-Division zum Einen und der K-Kreuzvalidierung zum

Anderen wichtige Grundsteine für Möglichkeiten der Beurteilung der Qualität und der Generalisierungsfähigkeit von Vorhersagemodellen gelegt. Die Vorteile der Kombination mehrerer linearer Regressionsmodelle zu einem Gesamtmodell wurden gesondert herausgearbeitet und werden im Zusammenhang mit der Vorstellung des OTFS-Verfahrens in Kapitel 7 ausgeschöpft.
Die für die Erstellung der Prädiktionsmodelle nötige Verarbeitung der objektiven Daten von der ursprünglichen Form als Zeitschriebe über die Wavelet-Transformation hin zu der Überführung in komfortrelevante Merkmale wurde in den Kapiteln 3 und 4 ausführlich beschrieben. Als nächster wesentlicher Schritt folgt deshalb in den Kapiteln 6 und 7 jeweils die Vervollständigung der Prädiktionsmodelle in zwei getrennten Zweigen, wobei, wie bereits in Kapitel 4 kurz erwähnt, zunächst eine geeignete Auswahl der zu verwendenden Merkmale zu ermitteln ist.
Für die Beurteilung der Eignung einer solchen Merkmalskombination ohne direkte Bezugnahme auf eine spezielle Struktur eines Vorhersagemodells wurde in Abschnitt 5.5.5 mit den selbstorganisierenden Karten nach Kohonen vorab ein geeignetes Instrument vorgestellt.

6 Auswahl von Merkmalen

Wie bereits in Kapitel 4 erwähnt umfasst der Prozess der Abbildung der transformierten Messdaten auf geeignete Eingangsgrößen des Vorhersagemodells neben der zuvor beschriebenen Merkmalsextrahierung auch den Schritt der Merkmalsauswahl. Verglichen mit der Ermittlung von Merkmalen ist deren Auswahl auf einer höheren Abstraktionsebene anzusiedeln. Wohingegen der Extrahierungs-Schritt eng an die zugrunde liegenden Daten gebunden und damit stark problemspezifisch ist, ist der Auswahl-Schritt enger an das Vorhersagemodell gebunden und damit leichter auf allgemeiner Ebene zu behandeln.
Allgemein geht es bei der Merkmalsauswahl darum, aus der Fülle aller im Rahmen der Merkmalsextrahierung ermittelten Kennzahlen eine Auswahl zu treffen, die die folgenden Eigenschaften besitzt:

- Die Anzahl an Merkmalen in der Auswahl sollte möglichst klein sein, da komplexere Vorhersagemodelle stärker zur Überanpassung tendieren
- Das Vorhersagepotential der Auswahl sollte möglichst groß sein

Diese beiden Eigenschaften stellen vor dem Hintergrund, dass zur Erstellung der Vorhersagemodelle nur eine beschränkte Anzahl von Trainingsdaten zur Verfügung steht, bereits die gesamte Problemstellung der Merkmalsauswahl dar.

6.1 Begriffe und Einordnung der Methoden

Die Methoden der Merkmalsauswahl werden in der Fachliteratur zunächst grob klassifiziert in univariate und multivariate Methoden [Guy06].
Die univariaten Methoden stellen eine sehr einfache Form der Merkmalsauswahl dar, die in vielen Fällen allerdings bereits zu guten Ergebnissen führt. Hierbei werden Merkmale der Auswahl hinzugefügt, die für sich gesehen ein geeignetes Vorhersagepotential besitzen. Üblicherweise werden hierzu die verfügbaren Merkmale einem sogenannten Ranking unterzogen, das ihre jeweilige Eignung für die Vorhersage bewertet, und das ausschlaggebend dafür ist, ob das Merkmal verwendet wird oder nicht. Bei dieser Betrachtung wird das Vorhersagepotential der gemeinsamen Verwendung verschiedener Merkmale vernachlässigt, was den Aufwand der Methoden vergleichsweise gering hält. Entsprechend findet aber auch das Vorhersagepotential der Kombination mehrerer Merkmale in der betrachteten Auswahl keine Beachtung.
Die Eignung univariater Methoden für ein spezifisches Problem hängt wesentlich von den verwendeten Merkmalen und der Art der Vorhersagemodelle ab. Im Rahmen der

Untersuchungen in dieser Arbeit wurde unter anderem eine univariate Methode entwickelt und angewandt. Sie basiert auf der Erstellung linearer Regressionsmodelle, die jeweils nur das zu untersuchende Merkmal als Regressor besitzen. Die Güte dieses Modells in Form des Determinationskoeffizienten R^2 (vgl. Abschnitt 5.3) wurde direkt als Ranking für das Merkmal verwendet. Die Ergebnisse der Erstellung von Vorhersagemodellen unter Verwendung dieser Art der Merkmalsauswahl führten allerdings nicht zu befriedigenden Ergebnissen.
Im Vergleich zu den univariaten Methoden zeichnen sich die multivariaten Methoden dadurch aus, dass sie nicht auf dem Vorhersagepotential einzelner, isolierter Merkmale, sondern direkt auf dem einer ganzen Auswahl von Merkmalen aufsetzen. Daher sind multivariate Methoden allgemein deutlich komplexer als ihre univariaten Pendants. Dies hat zur Folge, dass nicht nur die Entwicklung einer derartigen Methode ein größeres Problem darstellt und die Berechnungszeit bei der Anwendung der Methode ansteigt, sondern auch, dass eine derartige Methode eher zu einer Überanpassung der Beobachtungsdaten in Folge der Merkmalsauswahl führt. Dies trifft vor allem auf derartige Methoden zu, die auf einem intensiven Durchsuchen des Raumes (eng.: exhaustive search) der möglichen Kombinationen beruhen [Guy06],[Liu07].
Die multivariaten Methoden lassen sich in drei Klassen weiter unterteilen:

- Filter-Methoden ([SM07])
- Wrapper-Methoden ([Koh97])
- Eingebettete Methoden ([Lal06])

Sie alle haben zum Ziel, eine Auswahl von Merkmalen mit möglichst großer Vorhersagequalität zu ermitteln. Die Abgrenzung der Methoden voneinander besteht in der Interpretation des Begriffs der Vorhersagequalität, die auf irgendeine Weise ermittelt werden muss.
Bei den Filter-Methoden beruht die Ermittlung dieser Qualität auf einfacheren Korrelationsuntersuchungen oder statistischen Tests. Bei den Wrappern und den eingebetteten Methoden wird hingegen die Vorhersagequalität einer Merkmalskombination mit der Güte eines Vorhersagemodells identifiziert, das mit dieser Kombination erstellt und trainiert wird. Dies geschieht bei den Wrapper-Methoden im Rahmen des Absuchens des Raumes der möglichen Merkmale. Die eingebetteten Methoden sind aus dieser Sichtweise spezielle Wrapper-Methoden, bei denen die zur Bewertung der Merkmalskombinationen herangezogenen Vorhersagemodelle mit dem letztendlich zu erstellenden Vorhersagemodell gleichzusetzen sind und somit die Merkmalsauswahl in das Trainieren des eigentlichen Vorhersagemodells einbezogen werden kann.
Die Funktionsweise der multivariaten Methoden kann auf einer höheren Abstraktionsebene für alle drei Klassen gemeinsam zusammengefasst werden zu einem Algorithmus, der Merkmalskombinationen sucht und bewertet. Im folgenden Abschnitt 6.2 wird der Aspekt des Suchens potentieller Merkmalsauswahlen näher erörtert. Auf Methoden der Bewertung von Merkmalskombinationen wird in Abschnitt 6.3 näher eingegangen. Auf

diesen Grundlagen aufbauend wird in Abschnitt 6.4 die entwickelte Wrapper-Methode des Feature-Browsers vorgestellt und, gemeinsam mit den Erkenntnissen der vorherigen Kapitel, auf die verfügbaren Daten angewandt.

6.2 Suchmethoden der Merkmalsauswahl

Die für die Merkmalsauswahl verwendete Such-Methode hängt von mehreren Rahmenbedingungen ab. Die wichtigsten sind dabei die maximale Zeit, die dem Suchvorgang zugesprochen wird, die Genauigkeit, mit der der Raum der möglichen Merkmalskombinationen zu traversieren ist und die Art der Bewertung der zu überprüfenden Auswahlen.
Eine hohe Genauigkeit führt im Allgemeinen zu einer großen Dauer des gesamten Vorgangs. Welche Genauigkeiten überhaupt wirtschaftlich sind hängt auch wesentlich von der Art der Bewertung ab, die, vor allem bei den Wrapper-Methoden, alleine bereits einen hohen Berechnungsaufwand mit sich bringen kann.
Die höchste Genauigkeit wird bei der Bewertung aller möglichen Merkmalskombinationen erreicht, was einer vollständigen Suche entspricht. Zumindest von der theoretischen Seite her kann diese in endlicher Zeit durchgeführt werden, auch wenn letztere in der Regel nicht praxisrelevant sein wird. Trotzdem wirft die theoretische Möglichkeit einer vollständigen Suche die Frage auf, ob sie auch wirklich das beste Ergebnis liefert. Offenbar kann damit in jedem Fall eine Merkmalskombination ermittelt werden, die bezüglich des Bewertungsverfahrens und der Beobachtungsdaten die beste Vorhersagequalität erreicht. Wie allerdings bereits in Abschnitt 6.1 erwähnt wurde, kann dieses Vorgehen verstärkt zu Auswahlen führen, die die Neigung des zugrunde liegenden Vorhersagemodells zur Überanpassung bestärken und damit dessen Generalisierbarkeit verschlechtern. Hingegen sind Merkmalskombinationen, die durch heuristische Suchmethoden ermittelt werden, hiervon weniger betroffen [Liu07].
In diesem Kontext zeichnen sich gute Merkmalskombinationen durch gewisse Stetigkeitseigenschaften bezüglich der Bewertung aus. Führen geringfügige Änderungen an der Merkmalskombination zu geringfügigen Änderungen in der Bewertung, so kann durch sukzessive Abwägung zwischen guter Bewertung und guter Verallgemeinerbarkeit eine geeignete Merkmalskombination ermittelt werden. Die Suchmethode prüft also die Merkmalskombinationen entlang einer An- oder Abstiegsrichtung der Bewertungsfunktion. Durch weitere Maßnahmen wie z.B. K-Kreuzvalidierung (vgl. 5.2.5) können geeignete Abbruchkriterien definiert werden.
Beispiele für derartige systematische Suchstrategien sind sogenannte *hill-climbing*-Techniken wie Vorwärtsauswahl (eng.: forward-selection) und Rückwärts-Elimination (eng.: backward-elimination) [Liu98],[Guy03]. Bei der Vorwärtsauswahl wird in jedem Schritt zu der bestehenden Merkmalsauswahl das Merkmal hinzugefügt, dass die größte Verbesserung bezüglich der Bewertung bewirkt. Es wird also zu Beginn von leeren oder kleinen Merkmalsauswahlen ausgegangen und durch die Hinzunahme von Merkmalen eine Verbesserung erzielt. Umgekehrt ist die Vorgehensweise bei der Rückwärts-Elimination. Hier wird von

einer großen (ggf. der größtmöglichen) Merkmalsauswahl ausgegangen und durch sukzessives Entfernen von Merkmalen die Auswahl verkleinert. Dabei wird stets das Merkmal entfernt, dass die geringste Verschlechterung der aktuellen Merkmalsauswahl bezüglich der Bewertungsfunktion hervorruft.
Es sei darauf hingewiesen, dass an dieser Stelle nur die Grundidee der beiden Techniken vorgestellt wurde. Der Umfang an darauf aufsetzenden Varianten und Implementierungen dieser Techniken ist immens. So kann die Anzahl der in einem Schritt hinzuzufügenden oder zu entfernenden Merkmale gesteuert, die Bewertungsfunktion variiert, die Anzahl der parallel zu verfolgenden Pfade vergrößert und die Wahl der Initial-Kombinationen unterschiedlich festgelegt werden. Schließlich können durch Kombination der beiden Techniken noch Verfahren entwickelt werden, die durch Austauschen von Merkmalen sukzessive zu besser bewerteten Merkmalskombinationen führen ohne dabei die Anzahl der betrachteten Merkmale zu verändern.
Im Rahmen dieser Arbeit wurden zunächst sehr einfache multivariate Methoden getestet, die einer vollständigen Suche unter vorangehenden massiven Einschränkungen der heranzuziehenden Merkmale entsprachen. Der Ergebnisse waren in Hinblick auf die Erstellung gut generalisierbarer Vorhersagemodelle nicht befriedigend, weswegen mit dem Feature-Browser eine Wrapper-Methode entworfen wurde, die auf der Vorwärtsauswahl basiert.

6.3 Bewertung von Merkmalsauswahlen

Wie bereits in Abschnitt 6.1 angesprochen wurde, gehört zu dem Vorgang der Merkmalsauswahl stets ein Instrument für die Bewertung der Merkmalsauswahlen. Da die Suche nach geeigneten Merkmalskombinationen mit einer größeren Anzahl verfügbarer Merkmale bereits zu einem sehr zeitintensiven Unterfangen ausarten kann, ist ein wesentlicher Aspekt bei der Wahl der Bewertungsmethode dessen Komplexität. So ist es beispielsweise bei realistischen Größenordnungen für die Anzahl der bekannten Merkmale und die Größe der zu ermittelnden Merkmalsauswahl nicht auf sinnvolle Weise möglich, eine Bewertung der Auswahl an die Bewertung eines künstlichen neuronalen Netzes zu koppeln, welches die jeweilige Merkmalskombination als Eingänge verwendet und für eine Bewertung der Auswahl mehrfach initialisiert und trainiert werden muss.
Anders sieht es bei der Verwendung einfacher linearer Regressionsmodelle aus. Sie bedürfen keiner ein- oder gar mehrfachen Initialisierung und werden insbesondere nicht iterativ trainiert, da sich die Koeffizienten mit quasi konstantem Zeitaufwand durch Lösen des Ausgleichsproblems (vgl. Abschnitt 5.4) ergeben. Selbst im Falle der regularisierten linearen Regression mit festem Regularisierungsparameter μ ist der zusätzliche Aufwand überschaubar.
Die zuvor beschriebenen Szenarien betreffen die Bewertung von Merkmalskombinationen im Kontext der Wrapper-Methoden. Auf die Klasse der Filter-Methoden wird im weiteren Verlauf nicht eingegangen.

Ein weiterer Aspekt bei der Bewertung von Merkmalskombinationen über lineare Regressionsmodelle ist der, dass die Übertragung der Regressoren auf die Modellausgabe sehr viel direkter geschehen kann als bei einem künstlichen neuronalen Netz, bei dem diverse Netzgewichte und Kopplungen der Eingänge untereinander über die verdeckten Neuronen zwischen der Ein- und der Ausgabeschicht liegen. Bei einem linearen Regressionsmodell ist daher ebenfalls die Übertragung der Qualität eines mit einer gewissen Merkmalskombination erstellten Modells auf die Qualität der Merkmalskombination selber in höherem Maße gerechtfertigt als im Fall der künstlichen neuronalen Netze.
Mit dieser Abwägung konnte sich die lineare Regression in dieser Arbeit als das wesentliche Beurteilungsinstrument für die Bewertung von Merkmalskombinationen durchsetzen. Als Gütekriterien dienen dabei das Bestimmtheitsmaß R^2 sowie die Varianten der mittleren Fehlerquadratsumme (vgl. Abschnitt 5.3).
Eine gänzlich andere Methode zur Untersuchung der Qualität von Merkmalsauswahlen eröffnen die in Abschnitt 5.5.5 vorgestellten selbstorganisierenden Karten nach Kohonen. Sie ermöglichen Untersuchungen unter der Fragestellung, ob durch eine gegebene Kombination beschreibender Merkmale durch ein Kohonen-Netz strukturelle Beziehungen unter den einzelnen Datensätzen detektiert werden können. In welcher Beziehung die Datensätze bezüglich ihrer (diskreten oder kontinuierlichen) Klassifizierung tatsächlich zueinander stehen, wird in dieser Phase nicht bekannt gegeben. Diese Information wird so lange vorenthalten, bis die Erstellung der Kohonen-Karte abgeschlossen ist und kann im Anschluss zur Kontrolle verwendet werden.
Das Ergebnis einer derartigen Untersuchung ist eine ausgebreite Kohonen-Karte und die geometrische Anordnung der Datensätze auf dieser Karte. Dabei sind die relativen Ausrichtungen von großem Interesse, die absoluten hingegen völlig bedeutungslos. Desweiteren sei daran erinnert, dass eine Untersuchung der Merkmalsauswahl über selbstorganisierende Karten nach Kohonen durch die große Anzahl benötigter Iterationen (vgl. Abschnitt 5.5.5) sehr aufwendig ist. Sie ist daher als in die Merkmalssuche eingebettete Bewertungsmethode auszuschließen, drängt sich dafür allerdings als an eine scheinbar gelungene Merkmalsauswahl anschließende Überprüfungs-Methode auf.

6.4 Der Feature-Browser von OfeS-Tool

Der Feature-Browser von OfeS-Tool (vgl. Abschnitt 1.4) ist im weitesten Sinne eine auf den Konzepten der regularisierten linearen Regression, der Data-Division und der Vorwärtsauswahl basierende Wrapper-Methode. Der Regularisierungsparameter μ wird dabei jeweils vorab fest gewählt und kann insbesondere null sein.
Ein wesentlicher Aspekt des Feature-Browsers ist die Tatsache, dass die eigentliche Merkmalsauswahl weiterhin manuell geschieht und der Anwender somit sein Wissen über mögliche Kausalitäten jeder Zeit einbringen kann.

Ausgangspunkt der Merkmalsauswahl über den Feature-Browser ist das statische Modell

$$y(\mathbf{x}) = p_0.$$

Durch die Hinzunahme von Regressoren wird das Modell in jedem Schritt erweitert. Von dem jeweils aktuellen Modell sind als neue Folge-Modelle gerade die linearen Regressionsmodelle möglich, die jeweils genau einen neuen Regressor beinhalten. Somit besitzt das erste Modell genau so viele Folge-Modelle wie Merkmale insgesamt zur Verfügung stehen. Mit jeder Erweiterungs-Stufe des Modells verringert sich die Anzahl der möglichen Folge-Modelle jeweils um eins.
Sowohl das aktuelle Modell als auch alle möglichen Folge-Modelle lassen sich über die Norm des Residuum-Vektors $\mathbf{r}$ bewerten. Er beinhaltet für jeden Datensatz die absolute Abweichung der Modellausgabe von dem entsprechenden Sollwert. Als Norm können sowohl die euklidische ($\|\cdot\|_2$) als auch die Maximums-Norm ($\|\cdot\|_\infty$) gewählt werden. Über diese Bewertung der Folge-Modelle kann implizit der Regressor ermittelt werden, durch dessen Hinzunahme im aktuellen Schritt die größte Verbesserung des linearen Modells erreicht werden kann.
Entsprechend dem Konzept der Data-Division (vgl. Abschnitt 5.2.3) werden für die Bestimmung der jeweiligen Modellparameter $p_1, \ldots, p_n$ nur die zuvor festgelegten Trainingsdaten verwendet.
Der Feature-Browser ermittelt in jedem Schritt die Bewertungen aller möglichen Folge-Modelle und sortiert die unberücksichtigten Merkmale entsprechend ihres Potentials, das lineare Modell zu verbessern, absteigend in einer Liste. An dieser Stelle ist der Anwender gefordert, eine Auswahl für den hinzuzufügenden Regressor zu treffen. Als Grundlage dafür dienen ihm neben dem berechneten Ranking sein Hintergrundwissen über die beteiligten Ein- und Ausgabegrößen und deren eventuell a priori bekannten Zusammenhänge. Zusätzlich wird in jedem Schritt ein Streudiagramm angezeigt, dass die Anwendung des aktuellen Modells auf die Trainingsdaten visuell darstellt. Dies ermöglicht eine qualitative Einschätzung des aktuellen Modells und unterstützt den Anwender somit bei der Entscheidung, wann die Merkmalssuche zu beenden ist.
Die Auswahl des neuen Regressors erfolgt über einen einfachen Mausklick auf den jeweiligen Eintrag. Hierauf aktualisiert der Feature-Browser binnen kurzer Zeit den Stand des Modells, analysiert die neuen Folge-Modelle und stellt den Anwender erneut vor die Wahl eines weiteren Regressors.
Das Verfahren ist vollständig transparent und der Anwender hat volle Kontrolle über die Auswahl der Regressoren sowie den Abbruch des Verfahrens. Der Feature-Browser stellt daher eine im Sinne der Wrapper-Methoden fundierte Unterstützung bei der Ermittlung geeigneter Merkmalskombinationen dar. Entscheidungen hingegen trifft nur der Anwender selbst.
Während der Konstruktion einer Merkmalskombination über den Feature-Browser kann der Anwender jeder Zeit den zuletzt hinzugefügten Regressor wieder entfernen und somit zu einem vorherigen Modell-Zustand zurückkehren, um einen anderen Pfad im Raum der

insgesamt möglichen Kombinationen einzuschlagen. Diese Möglichkeit der Löschung von Regressoren aus dem Modell betrifft jedoch nur den jeweils letzten Schritt, weswegen der Feature-Browser allgemein nicht das Konzept der Rückwärts-Elimination unterstützt. Zwar ist es grundsätzlich möglich, gezielt einen Regressor aus einer beliebigen Modell-Stufe zu entfernen; die Umsetzung ist allerdings sehr umständlich, da die Regressoren, die hierzu zwischenzeitlich entfernt werden müssten, obwohl sie eigentlich weiter berücksichtigt werden sollen, auf irgendeine Weise gespeichert werden müssten.
Abgesehen davon war die Unterstützung allgemeiner Rückwärts-Elimination zu keiner Zeit ein Entwurfskriterium des Feature-Browsers. Letztere bestanden im Wesentlichen in der fundierten mathematischen Basis des Verfahrens sowie in einer intuitiven und schnell ansprechenden Bedienbarkeit, die dem Anwender eine möglichst große Unterstützung im Rahmen seines eigenverantwortlichen Handelns bietet.
So einfach die grundlegenden Prinzipien des Feature-Browsers sind, so vielfältig sind auch die Möglichkeiten, die er dem Anwender eröffnet. Neben dem konstruktiven Aspekt der Erstellung geeigneter Merkmalskombinationen können auch ganz grundlegende Erkenntnisse über die Bedeutung einzelner Merkmale in Kombination mit anderen sowie über den Einfluss durch den Regularisierungsparameter gewonnen werden.

6.4.1 Merkmalsauswahl mit dem Feature-Browser

Im Folgenden werden das Vorgehen und die Ergebnisse der Ableitung geeigneter Merkmalskombinationen für die Prädiktionsmodelle über den Feature-Browser beschrieben.
Es wurden Kombinationen für die sechs zu erstellenden Vorhersagemodelle (vgl. Tabelle 6.1) ermittelt und im Vorfeld der Modellerstellung mit Kohonen-Netzen analysiert.

Nr.	Bewertung	Hindernis	Kriterium
1	Gesamteindruck	mnegK	VA
2	Gesamteindruck	mnegK	HA
3	Gesamteindruck	mposK	VA
4	Gesamteindruck	mposK	HA
5	Gesamteindruck	UR	VA
6	Gesamteindruck	UR	HA

Tabelle 6.1: Zu erstellende Vorhersagemodelle

Für die Bestimmung der Merkmalskombinationen über den Feature-Browser wurden vorab die verfügbaren Datensätze aufgeteilt in eine Trainings- und eine Testpartition. Die Aufteilung der Daten in die beiden annähernd gleich großen Partitionen (21 Trainingsdatensätze, 20 Testdatensätze) geschah durch Zufalls-Auswahl, wobei zusätzlich gesichert wurde, dass

1. zu der höchsten und der niedrigsten subjektiven Bewertung aus der Gesamtheit der Daten jeweils mindestens ein Datensatz in der Trainingspartition liegt,

2. die relative Abweichung der Mittelwerte der subjektiven Bewertungen beider Parti-

tionen unter 5% liegt und

3. die relative Abweichung der Standardabweichungen der subjektiven Bewertungen beider Partitionen unter 5% liegt.

Im Feature-Browser wurden nur die Trainingsdaten berücksichtigt. Als Bewertungskriterium wurde die 2-Norm verwendet.
Für die regularisierte lineare Regression minimiert der Feature-Browser standardmäßig die Zielfunktion (5.11), wobei speziell der Regularisierungsparameter $\mu \in \mathbb{R}_+$ über $\alpha \in [0{,}1)$ durch

$$\mu = \frac{\alpha}{1-\alpha}$$

festgelegt wird. Für den hier beschriebenen Anwendungsfall wurde mit $\alpha = 0{,}9$ ein relativ großer Regularisierungs-Anteil gewählt. Die Anzahl der Regressoren für die Modelle wurde vorab auf zwölf gesetzt.
Als weitere Maßnahme für die Ermittlung möglichst plausibler Eingangskombinationen wurden im Vorfeld der Merkmalsauswahl die zulässigen Merkmale pro Achse jeweils entsprechend des Bewertungskriteriums weiter eingeschränkt. Für die Vorhersagemodelle eins, drei und fünf wurden daher jeweils nur die Merkmale bezüglich der Vorderachse und entsprechend für die Modelle zwei, vier und sechs nur die Merkmale bezüglich der Hinterachse zugelassen und in der Auswahl berücksichtigt. Dies geschah im Einklang mit den Ausführungen in Abschnitt 2.1.3, nach denen der Testfahrer für die Bewertung der Hinterachse die Phänomene in Zusammenhang mit der Vorderachsüberfahrt sowie für die Bewertung der Vorderachse die Phänomene in Zusammenhang mit der Hinterachsüberfahrt stets ausblendet.
Um in den Modellen später eine möglichst umfassende Beschreibung des Schwingungszustandes zu erfassen wurde explizit dafür gesorgt, dass jeder der sechs Kanäle aus Tabelle 2.6 mit wenigstens einem Merkmal in der Auswahl vertreten ist. Daher wurde für die ersten sechs Regressoren die von dem Feature-Browser vorgeschlagene Reihenfolge nur mit zweiter Priorität bedacht. In den ersten sechs Schritten wurde jeweils das Merkmal mit dem höchsten Feature-Browser-Ranking gewählt, dass zu einem Kanal aus Tabelle 2.6 gehört, zu dem bis dahin noch kein Merkmal in der Auswahl vorhanden war. Für die Merkmale sieben bis zwölf wurde dann ausschließlich nach der durch den Feature-Browser vorgeschlagenen Reihenfolge weiterverfahren. Hierbei sei angemerkt, dass durch die vorangegangene erzwungene Teilauswahl nicht nur gesichert wurde, dass jeder Kanal zumindest einfach vertreten war. Insbesondere hatte diese Wahl der ersten sechs Regressoren auch einen wesentlichen Einfluss auf das Ranking der weiteren Merkmale durch den Feature-Browser.
Im Rahmen der Untersuchungen wurden über den Feature-Browser für die sechs Vorhersagemodelle (vgl. Tabelle 6.1) jeweils drei Merkmalskombinationen ermittelt und zur Überprüfung mit Kohonen-Netzen analysiert (vgl. Abschnitt 6.4.2). Die jeweils beste Auswahl wurde behalten, die übrigen wurden verworfen.
Die auf diese Weise ermittelten Merkmalskombinationen sind in den Tabellen 6.2 bis 6.7

dargestellt (vgl. Tabelle 4.1). Dabei entspricht die Nummerierung gerade der Reihenfolge, in der die Merkmale mit dem Feature-Browser bestimmt wurden. Zusätzlich zu der Kanal-Nummer (vgl. Tabelle 2.6) ist die Einleitungsrichtung der Beschleunigung (x für Längsrichtung, z für Vertikalrichtung) dargestellt.
Der Reihenfolge der Merkmale im Feature-Browser gegenüber steht das Bestimmtheitsmaß R^2 (vgl. Abschnitt 5.3) des linearen Einzel-Modells bezüglich des jeweiligen Merkmals, das dem Ranking für den Fall einer univariaten Merkmalsauswahl (vgl. Abschnitt 6.1) zu Grunde liegt.
Da der Feature-Browser nach dem ersten Schritt ein lineares Modell mit einem einzigen Regressor erstellt, stimmt in diesem Fall die Auswahl mit der der univariaten Methode überein. Für die folgenden Schritte allerdings weichen die ausgewählten Merkmale zum Teil gravierend voneinander ab.
Die Beschleunigungseinleitung in Längsrichtung ist auffällig häufig in den Tabellen vertreten, was zunächst verwunderlich sein mag, sich aber tatsächlich mit den Ergebnissen von Stammen ([Sta09]) deckt. Ebenfalls erwähnenswert ist, dass vor allem bei den Modellen für die Bewertung der Vorderachse häufige Auftreten des Schalldrucks als Kanal. Das sich einstellende Verhältnis der Merkmale der ersten Approximation zu denen der zweiten Approximation ist sehr ausgeglichen. Diese Verteilung stellte sich unabhängig von dem strikteren Vorgehen bei der Auswahl der ersten sechs Merkmale ein, da dieses lediglich Einfluss auf die Kanal-Auswahl hatte. Diese Tatsache unterstreicht, dass die Berücksichtigung beider Merkmalsgruppen durchaus berechtigt ist und diese sich gut ergänzen.

Nr.	**Kanal**	**Einleitung**	**Merkmal**	**Approximation**	**Achse**	$\mathbf{R^2}$
1	5	z	m_f	I	VA	0,55
2	4	z	r_b	I	VA	0,00
3	6	–	a	I	VA	0,64
4	2	z	$\cos(\varphi)$	II	VA	0,01
5	1	x	$\overline{\mathbf{W}}$	I	VA	0,14
6	3	x	r_s	I	VA	0,02
7	6	–	$\overline{\mathbf{W}}$	II	VA	0,12
8	4	z	$\sin(\varphi)$	II	VA	0,00
9	1	x	m_f	I	VA	0,11
10	6	–	$\sin(\varphi)$	II	VA	0,04
11	2	z	r_s	II	VA	0,12
12	3	x	$\sin(\varphi)$	I	VA	0,01

Tabelle 6.2: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das erste Vorhersagemodell (mittlere negative Kante Vorderachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	4	z	r_b	I	HA	0,62
2	2	z	r_s	II	HA	0,17
3	5	z	m_f	I	HA	0,49
4	3	x	r_s	I	HA	0,01
5	6	–	m_f	I	HA	0,46
6	1	x	r_s	II	HA	0,11
7	3	x	φ	II	HA	0,03
8	5	z	r_b	II	HA	0,15
9	4	z	$\sin(\varphi)$	I	HA	0,05
10	3	x	r_s	II	HA	0,11
11	3	x	$\cos(\varphi)$	I	HA	0,00
12	2	z	m_f	I	HA	0,18

Tabelle 6.3: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das zweite Vorhersagemodell (mittlere negative Kante Hinterachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	6	–	$\overline{\mathbf{W}}$	I	VA	0,74
2	1	x	$\cos(\varphi)$	I	VA	0,00
3	5	z	a	II	VA	0,02
4	3	x	m_f	II	VA	0,02
5	4	z	r_b	I	VA	0,46
6	2	z	a	II	VA	0,07
7	1	x	r_b	I	VA	0,01
8	6	–	a	I	VA	0,40
9	4	z	$\cos(\varphi)$	I	VA	0,10
10	5	z	$\sin(\varphi)$	II	VA	0,02
11	2	z	Δ_f	II	VA	0,12
12	5	z	Δ_t	II	VA	0,01

Tabelle 6.4: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das dritte Vorhersagemodell (mittlere positive Kante Vorderachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	1	x	$\overline{\mathbf{W}}$	I	HA	0,47
2	2	z	r_s	I	HA	0,13
3	5	z	r_b	II	HA	0,02
4	6	–	$\overline{\mathbf{W}}$	I	HA	0,73
5	3	x	Δ_t	II	HA	0,01
6	4	z	Δ_f	II	HA	0,00
7	2	z	φ	II	HA	0,17
8	1	x	$\cos(\varphi)$	II	HA	0,05
9	1	x	$\cos(\varphi)$	I	HA	0,09
10	1	x	a	I	HA	0,19
11	4	z	m_f	II	HA	0,01
12	4	z	r_s	I	HA	0,04

Tabelle 6.5: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das vierte Vorhersagemodell (mittlere positive Kante Hinterachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	6	–	$\overline{\mathbf{W}}$	I	VA	0,83
2	3	x	r_b	II	VA	0,02
3	4	z	Δ_t	II	VA	0,38
4	2	z	a	II	VA	0,40
5	1	x	Δ_f	I	VA	0,65
6	5	z	$\overline{\mathbf{W}}$	II	VA	0,08
7	3	x	Δ_f	II	VA	0,01
8	3	x	Δ_f	I	VA	0,40
9	6	–	r_b	II	VA	0,02
10	1	x	$\overline{\mathbf{W}}$	I	VA	0,50
11	4	z	r_s	II	VA	0,13
12	3	x	$\overline{\mathbf{W}}$	II	VA	0,08

Tabelle 6.6: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das fünfte Vorhersagemodell (U-Rille Vorderachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	6	–	$\overline{\mathbf{W}}$	I	HA	0,83
2	1	x	$\overline{\mathbf{W}}$	I	HA	0,64
3	3	x	φ	I	HA	0,15
4	5	z	m_f	II	HA	0,14
5	4	z	r_b	II	HA	0,01
6	2	z	a	I	HA	0,20
7	3	x	r_s	I	HA	0,32
8	5	z	r_b	II	HA	0,12
9	3	x	Δ_f	II	HA	0,13
10	6	–	a	I	HA	0,76
11	3	x	r_s	II	HA	0,00
12	1	x	a	I	HA	0,45

Tabelle 6.7: Ergebnisse der Merkmalsauswahl über den Feature-Browser für das sechste Vorhersagemodell (U-Rille Hinterachse)

6.4.2 Bewertung der Merkmalskombinationen mit Kohonen-Netzen

Die in Abschnitt 6.4.1 ermittelten Merkmalskombinationen wurden vor der Erstellung der Vorhersagemodelle einer Analyse mit selbstorganisierenden Karten nach Kohonen unterzogen (vgl. Abschnitte 5.5.5 und 6.3). Hierzu wurde für die 41 Datensätze ein Gitter mit 15×14 verdeckten Neuronen verwendet. Die Abstände wurden sowohl im Eingaberaum als auch im Gitter mit der euklidischen Länge gemessen, wobei keine periodischen Ränder berücksichtigt wurden. Die Lernraten- sowie die Nachbarschafts-Radien-Anpassung wurden jeweils linear (von eins bis null) gewählt. Als Nachbarschaftsfunktion kam die Gaußsche Glockenkurve (5.38) zum Einsatz.
Die Kohonen-Karten wurden jeweils mit 10^5 Epochen trainiert. Die Visualisierung der Ergebnisse erfolgt in zwei Schritten:

1. **Anordnung der Datensätze:** Bei dieser Darstellung werden die einzelnen Datensätze entsprechend der Gitterposition des zugehörigen Champion-Neurons (im ausgebreiteten Zustand der Karte) in der zweidimensionalen Ebene als kleine Rechtecke angezeigt. Zur Kontrolle wird über die Farbe die tatsächliche Komfortnote wiedergegeben.

2. **Notenverteilung auf der Karte:** Hier erfolgt die Darstellung in der anderen Richtung. Während zuvor den Datensätzen jeweils eine Gitterposition zugeordnet wurde, wird in diesem Fall jeder Gitterposition der Datensatz zugeordnet, der bezüglich der Metrik des Eingaberaums (hier die euklidische Länge) am geringsten von dem zugehörigen Gewichtsvektor abweicht (winner-take-all). Entsprechend der subjektiven Bewertung dieses Datensatzes wird der jeweiligen Gitterposition eine

Farbe zugeordnet, in der symbolisch der Datensatz als kleines Rechteck an der entsprechenden Position in der zweidimensionalen Ebene angezeigt wird.

Die Ergebnisse der Untersuchungen sind in Abschnitt A.1.1 dargestellt. Die Bewertung eines solchen Ergebnisses geschieht im Wesentlichen nach zwei Fragestellungen:

1. **Anordnung** Gelingt die Aufteilung der Daten in möglichst wenige Cluster ähnlich bewerteter Datensätze?

2. **Stetigkeit** Sind die Verläufe der Bewertungen ineinander übergehender Cluster glatt?

Das Stetigkeitskriterium hat bei dem vorliegenden Problem eine größere Relevanz als das Anordnungs-Kriterium, da, wie bereits ausgeführt, die Komfortbewertung unterschiedlicher Fahrzeuge nicht als injektive Abbildung angesehen werden kann (vgl. Abschnitt 5.5.5). Dies trifft vor allem für niedrige Komfortbewertungen zu, da die jeweilige Ursache für den geringen Komfort stark variieren kann.
Bezüglich des Stetigkeitskriteriums sind die Ergebnisse aus Abschnitt A.1.1 als gut bis sehr gut zu bezeichnen. So ergeben sich Situationen größerer Notensprünge nur in sehr wenigen Fällen (siehe zum Beispiel Abbildung A.4). Demgegenüber stehen zahlreiche weiche Abstufungen wie zum Beispiel in den Abbildungen A.2 und A.5.
Bezüglich des Anordnungskriteriums ergeben sich gerade für die Hinterachsmodelle gute Ergebnisse, zu sehen beispielsweise in Abbildung A.2 oder Abbildung A.6. Bei den Vorderachsmodellen fällt die Anordnung nicht ganz so gut aus, wenn auch, wie beispielsweise in Abbildung A.5, zum Teil gute Abschnitte dabei sind.
Insgesamt zeigen die Ergebnisse der Bewertungen der Merkmalsauswahlen mit Kohonen-Netzen jeweils eine Struktur der objektiven Daten auf, die sich unter Berücksichtigung der zugehörigen subjektiven Bewertungen überwiegend als plausibel beschreiben lässt. Auffällig sind dagegen bereits zu diesem Zeitpunkt die zum Teil sehr geringen Streuungen der subjektiven Bewertungen. Die durch die verschiedenen Fahrzeugvarianten hervorgerufenen Änderungen des subjektiven Komfortempfindens sind zum Teil sehr gering, wobei sich erneut eine deutliche Diskrepanz zwischen Vorderachs- und Hinterachsmodellen zeigt. Die mittlere Spreizung der abgegebenen Komfortnoten ist bei den Modellen zu der Hinterachse mit $\frac{29}{12}$ um ca. 50% größer als bei den Modellen zu der Vorderachse mit $\frac{19}{12}$. Größere Spreizungen der subjektiven Bewertungen zu erzielen war im Rahmen der Experimente nicht möglich.
Das Problem der zum Teil sehr geringen Notenspreizungen wird auch bei der späteren Interpretation der Ergebnisse der Modellerstellung eine große Rolle spielen.

6.4.3 Modellerstellung mit künstlichen neuronalen Netzen

Im Folgenden werden, ausgehend von den Merkmalskombinationen aus den Tabellen 6.2 bis 6.7, Vorhersagemodelle auf Basis künstlicher neuronaler Netze erstellt. Verwendet wird die in Abschnitt 5.5.1 vorgestellte Feed-Forward-Struktur mit einer verdeckten Schicht.

Entsprechend der ermittelten Eingangskombinationen liegen ferner jeweils zwölf Neuronen in der Eingabeschicht. Die Ausgabe der approximierten Komfortnote erfolgt über ein einzelnes Neuron in der Ausgabeschicht. Als Aktivierungsfunktion wurde in der verdeckten Schicht der Tangens-Hyperbolicus und in der Ausgabeschicht die Identität verwendet. Dadurch müssen die Netzausgaben nicht weiter auf den Bereich der Komfortnoten umskaliert werden.
Die Anzahl der verdeckten Neuronen wurde, ausgehend von großen Werten, sukzessive verringert, bis eine weitere Reduzierung eine Verschlechterung der Ergebnisse bewirkte. So ergab sich jeweils eine Anzahl von vier verdeckten Neuronen. Für die Erstellung und das Training der Feed-Forward-Netze wurde die Neural-Network-Toolbox von Matlab verwendet.
Von den 41 verfügbaren Datensätzen wurden jeweils 21 für das Training genutzt. Die übrigen 20 wurden als Testdaten zurückgehalten und nach dem Training zur Beurteilung der zu erwartenden Generalisierbarkeit des Netzes genutzt. Die Verteilung der Datensätze auf die beiden Partitionen geschah erneut unter Beachtung der drei Regeln aus Abschnitt 6.4.1.
Durch die verwendete Netztopologie ergeben sich insgesamt 57 Netzgewichte, für deren Ermittlung somit nur 21 (maximal 41) Datensätze zur Verfügung stehen. Da für das Training das Verfahren der Bayes'schen Regularisierung (vgl. Abschnitt 5.5.3) verwendet wird, werden nicht alle 57 Freiheitsgrade voll ausgenutzt. Die Anzahl γ der effektiv genutzten Netzgewichte liegt deutlich unter dieser oberen Grenze, wie in Tabelle 6.8 erkennbar ist.
Die Abbildungen 6.1 bis 6.6 zeigen die Streudiagramme der trainierten Vorhersagemodelle jeweils bei Anwendung auf die Trainingsdaten (a), die Testdaten (b) und alle Daten gemeinsam (c). Dabei sind die Komfortnoten jeweils abzüglich der Komfortnote des Basisfahrzeugs dargestellt, so dass in den Abbildungen Datensätze mit der subjektiven Bewertung null dem Basisfahrzeug oder einer identisch bewerteten Variante entsprechen. Die Teilabbildung (d) zeigt jeweils die relative Verteilung der absoluten Ausgabefehler für die Trainings- und die Testdaten. Hierzu wurden die Modellausgaben zunächst auf die Achtel-Quantisierung der verwendeten Notenskala gerundet und dann mit den Sollwerten verglichen.
Die dargestellten Ergebnisse zeigen, dass alle sechs Modelle hohe Güten sowohl bezüglich der Trainings- als auch bezüglich der Testdaten erreichen. Die Vorhersagequalität ist durchgehend mit sehr gut zu bezeichnen und die qualitative Genauigkeit der Vorhersage bezüglich beider Daten-Partitionen liegt jeweils in der gleichen Größenordnung. Insbesondere erreicht das sechste Modell hervorragende Werte sowohl bei den Determinationskoeffizienten als auch bei den Modell-Fehlern. Es zeigt sich aber auch eine Parallele zu den in Abschnitt 6.4.2 angemerkten Diskrepanzen zwischen Vorderachs- und Hinterachsmodellen bezüglich der Notenspreizungen. So sind jeweils die Modell-Fehler und die Abweichungen der Steigung der Ausgleichsgeraden von dem Wert eins bei den Modellen zu der Vorderachse tendenziell größer als bei den Modellen zu der Hinterachse. Insgesamt ist dies zusätzlich zu der sehr hohen erreichten Modellqualität eine Bestätigung der Erkenntnisse aus der Analyse der Merkmalskombinationen über die Kohonen-Netze. Die sehr erfolgreiche Anwendung auf die Testdaten spricht insbesondere für eine gute Generalisierbarkeit der Modelle.

Als Zusammenfassung sind in Tabelle 6.8 für die einzelnen Modelle die Bestimmheitsmaße sowie die Zahl γ der effektiv genutzten Netzgewichte zusammengestellt. Diese Ergebnisse spiegeln den sehr guten Eindruck aus den Streudiagrammen wider. Auffällig ist hierbei die Anzahl der effektiv genutzen Netzgewichte, die deutlich unterhalb der Anzahl der verwendeten Trainingsdaten und insbesondere der oberen Grenze von 57 liegt. Dies spricht zusätzlich für gut ausbalancierte Verbindungsgewichte und gegen eine Überanpassung des Modells an die Trainingsdaten.
Insgesamt bestätigen somit die sehr guten Ergebnisse der Modellerstellung die Verwendung der Bayes'schen Regularisierung als Trainingsmethode (vgl. Abschnitt 5.5.3) und des Feature-Browsers als Instrument für die statistisch unterstützte Ermittlung plausibler Merkmalskombinationen.

Modell	**R^2 (Training)**	**R^2 (Test)**	**R^2 (Gesamt)**	**eff. # Netzgewichte**
1	0,96	0,89	0,91	13,40
2	0,92	0,91	0,91	9,75
3	0,91	0,89	0,89	10,30
4	0,91	0,94	0,92	10,20
5	0,93	0,89	0,91	11,00
6	0,97	0,97	0,97	12,30

Tabelle 6.8: Bestimmheitsmaße und effektive Anzahl verwendeter Netzgewichte zu den sechs verschiedenen Vorhersagemodellen aus Kapitel 6

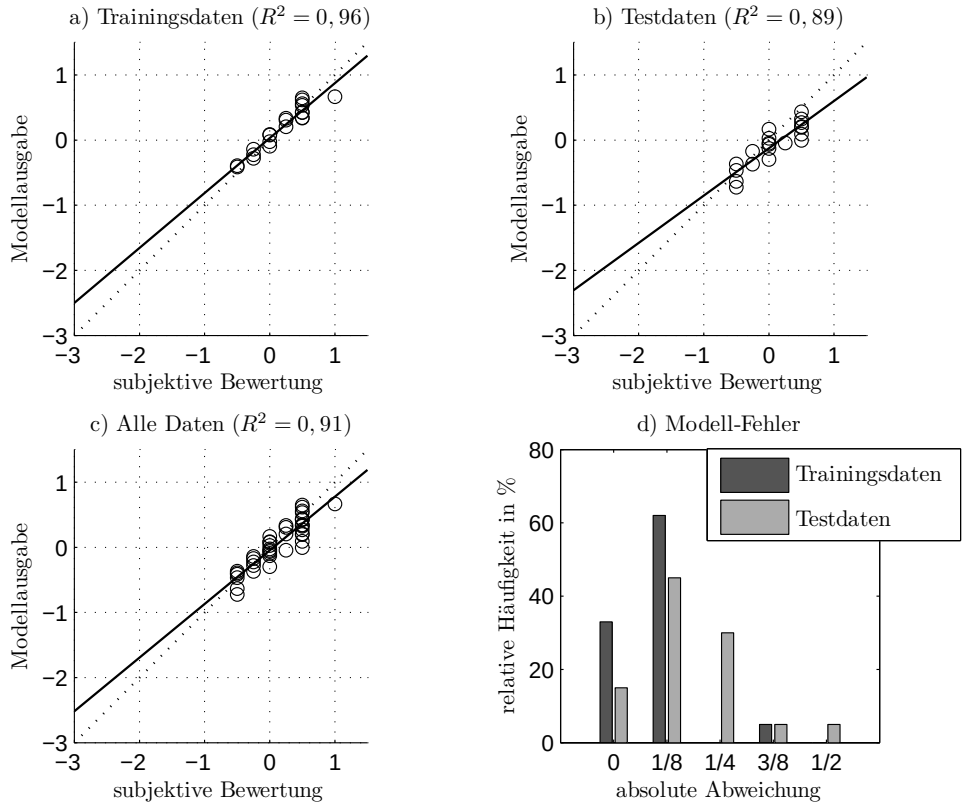

Abbildung 6.1: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das erste Modell (Feature-Browser-Merkmalsauswahl)

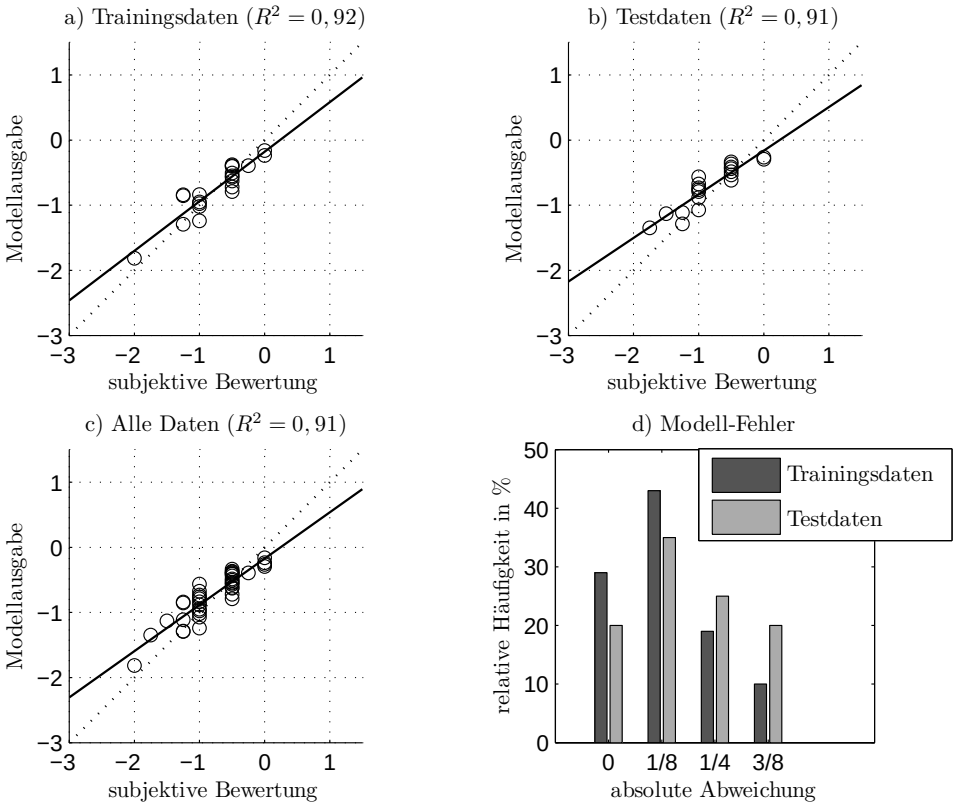

Abbildung 6.2: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das zweite Modell (Feature-Browser-Merkmalsauswahl)

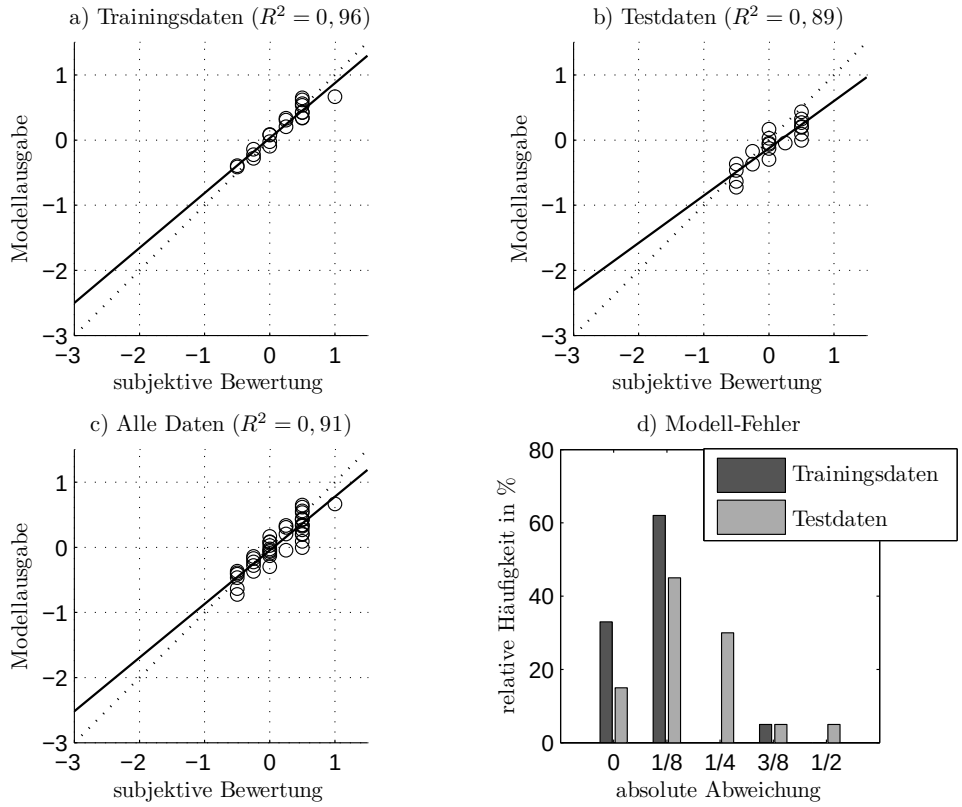

Abbildung 6.3: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das dritte Modell (Feature-Browser-Merkmalsauswahl)

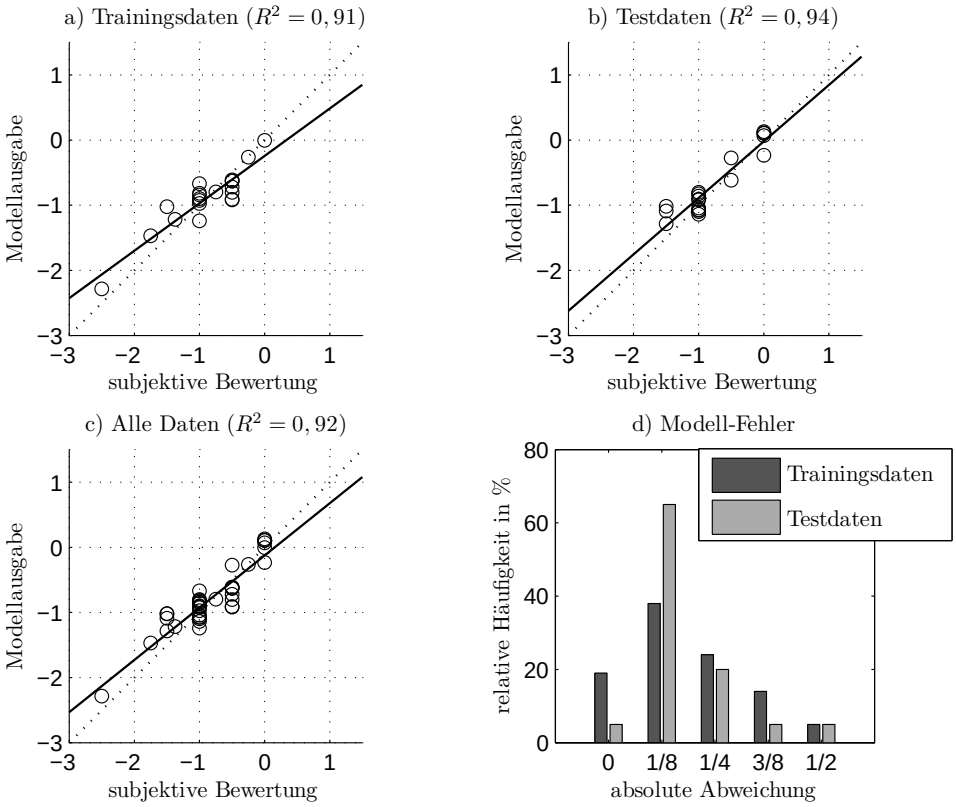

Abbildung 6.4: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das vierte Modell (Feature-Browser-Merkmalsauswahl)

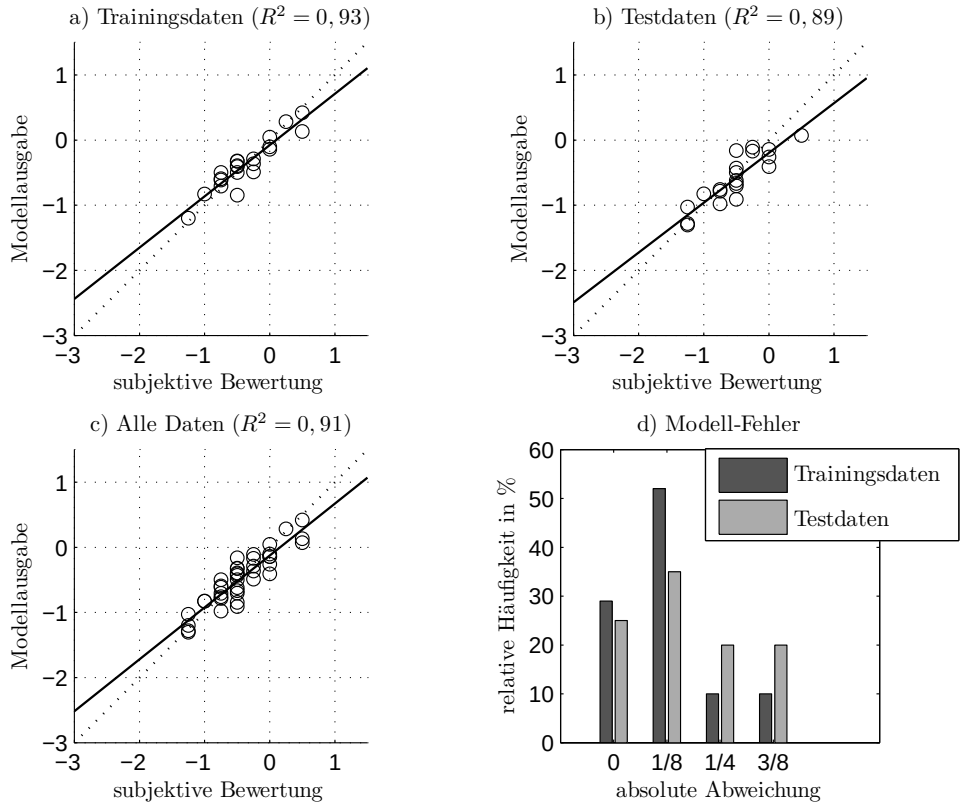

Abbildung 6.5: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das fünfte Modell (Feature-Browser-Merkmalsauswahl)

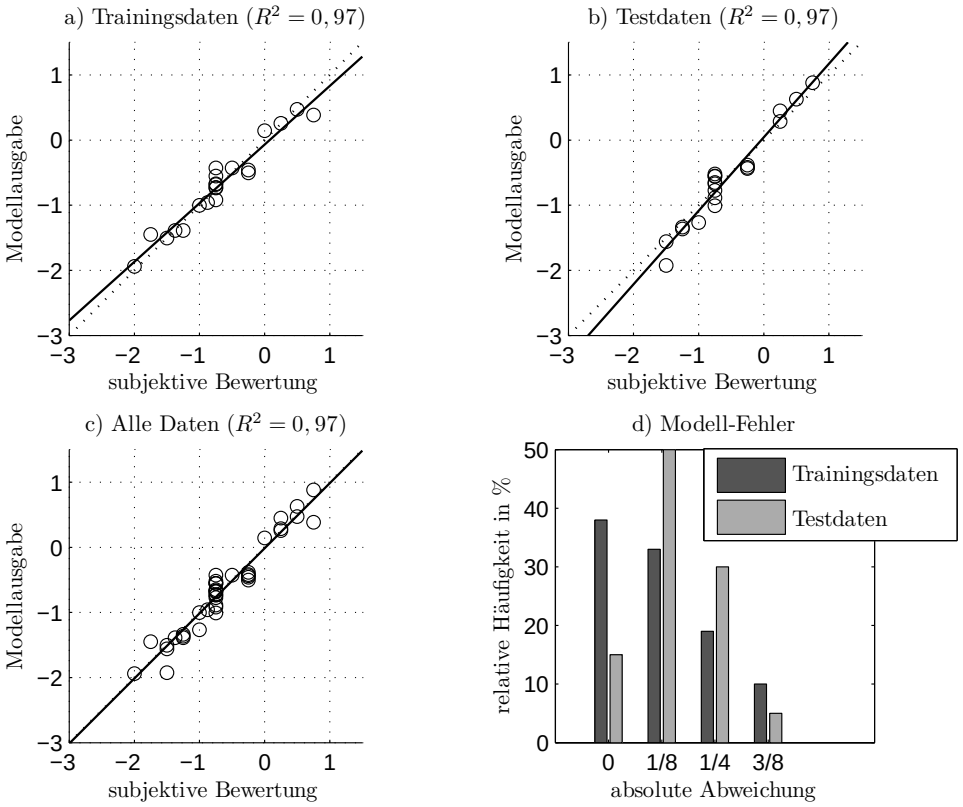

Abbildung 6.6: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das sechste Modell (Feature-Browser-Merkmalsauswahl)

7 Merkmalsauswahl und Modellerstellung mit OTFS

Als Alternative zu dem in Kapitel 6 vorgestellen Feature-Browser wurde im Rahmen dieser Arbeit mit dem OTFS-Verfahren (OfeS-Tool Feature-Selection) eine weitere Methode für die Merkmalsauswahl entwickelt. Sie basiert auf einer Hill-Climbing Such-Strategie, kombiniert mit einer Bewertung über lineare Regressionsmodelle. In sie fließen die gewonnenen Erkenntnisse aus den Abschnitten 5.2.5 und 5.4 ein.
Eine Vorstellung des OTFS-Verfahrens findet in Abschnitt 7.1 statt. Hierbei wurde der Fokus auf eine grundlegende Darstellung der prinzipiellen Funktionsweise beschränkt, da der Matlab-Code der Implementierung des Verfahrens rund 2000 Zeilen umfasst.
In Abschnitt 7.2 wird anschließend die Ermittlung von Merkmalsauswahlen mit dem OTFS-Verfahren sowie deren Bewertung über selbstorganisierende Karten nach Kohonen beschrieben. Der dann folgende Abschnitt 7.2.1 behandelt die Erstellung von Vorhersagemodellen in Form von künstlichen neuronalen Netzen, die auf den zuvor ermittelten Eingangskombinationen aufsetzen. Bei der Bewertung der Merkmalsauswahlen und der Erstellung der künstlichen neuronalen Netze wurde exakt so wie in den Abschnitten 6.4.2 und 6.4.3 vorgegangen, so dass die jeweiligen Ergebnisse einen direkten Vergleich des OTFS-Verfahrens mit der Ermittlung von Merkmalskombinationen über den Feature-Browser erlauben.
Im Zuge der Ermittlung einer Merkmalsauswahl erstellt das OTFS-Verfahren jedes Mal auch direkt ein allgemeines lineares Modell, das als Vorhersagemodell für das zugrunde liegende Bewertungskriterium herangezogen werden kann. Somit stellt die OTFS-Methode ein eingebettetes Verfahren dar (vgl. Abschnitt 6.1). Die im Rahmen der Ermittlung der Merkmalskombinationen erstellten linearen Modelle werden in Abschnitt 7.3 analysiert und mit den weiteren Vorhersagemodellen verglichen.

7.1 Das OTFS-Verfahren

Bei dem OTFS-Verfahren geschieht die Auswahl der Merkmale durch die fortlaufende Entwicklung eines linearen Regressionsmodells für die Komfortvorhersage, weswegen es den eingebetteten Methoden zugeordnet werden kann. Es besteht im Wesentlichen aus drei Komponenten:

1. Such-Strategie
2. Bewertung ermittelter Kombinationen
3. Kombination einzelner Modelle

Zu Beginn wird erneut eine Aufteilung der vorhandenen Daten in eine Trainings- und eine Test-Partition vorgenommen. Die Testdaten bleiben zu jeder Zeit von dem Verfahren unberührt.

7.1.1 Kombination der einzelnen Modelle

Das OTFS-Verfahren durchläuft eine vorgegebene Zahl n_{epoch} an Epochen. Am Ende jeder dieser Epochen stehen eine aktuelle (und unabhängige) Kombination von Merkmalen sowie zugehörige Koeffizienten, die, gemeinsam mit den Merkmalen, ein lineares Vorhersagemodell in standardisierter Form darstellen. Zusätzlich zu der Auswahl und den Koeffizienten liegt nach jeder Epoche eine Bewertung des Modells vor, die ein (stets positives) Maß für seine Qualität ist.
Über alle Epochen hinweg wird von dem Verfahren ein allgemeines lineares Modell erstellt, das sich aus der gewichteten Mittelung der speziellen Modelle aus den einzelnen Epochen ergibt (vgl. Abschnitt 5.4.3). Die Gewichtung geschieht an Hand der Modellqualität. Der Gewichtungsfaktor a_i des i-ten Modells bei der Mittelung von $n \leq n_{\text{epoch}}$ Modellen mit den Qualitäts-Bewertungen $q_1, \ldots, q_n$ ergibt sich durch

$$a_i = \frac{q_i}{\sum_{j=1}^{n} q_j}.$$

Somit ist stets sichergestellt, dass die gewichtete Mittelung der Modellkoeffizienten eine Konvexkombination darstellt.

7.1.2 Such-Strategie

Die Such-Strategie des OTFS-Verfahrens ist eine Kombination aus Vorwärts-Selektion und Rückwärts-Elimination. Ausgegangen wird von einer festen Auswahl von n_{initial} Merkmalen, die zufällig ermittelt wird, und insbesondere leer sein kann. Diese Auswahl stellt den Ist-Zustand dar. Als Folge-Zustände ergeben sich alle möglichen Merkmalskombinationen, die jeweils ein zusätzliches, nicht bereits berücksichtigtes Merkmal beinhalten (Vorwärts-Selektion), oder denen ein Merkmal aus dem Ist-Zustand fehlt (Rückwärts-Elimination). In diesem Zusammenhang werden Schranken b_{lo} für die minimale und b_{hi} für die maximale Anzahl von Merkmalen in einer Auswahl festgelegt, so dass die Anzahl der Folge-Zustände auch von der Größe der aktuellen Auswahl abhängt.
Die Schranken haben nur Einfluss auf die Bestimmung der Folge-Zustände. Beispielsweise ist eine Auswahl mit vier Merkmalen auch bei einer unteren Schranke von sechs Merkmalen zulässig. Die Beschränkung besteht nur darin, dass keine Folgezustände mit drei Merkmalen berücksichtigt werden, sondern nur solche mit fünf. So ist es insbesondere möglich, die minimale Anzahl von Merkmalen auf einen positiven Wert zu setzen, und trotzdem bei der Suche jeweils mit einer leeren Auswahl zu beginnen.
In jedem Schritt der aktuellen Epoche werden sowohl der Ist-Zustand als auch sämtliche Folge-Zustände bewertet (vgl. Abschnitt 7.1.3). Erreicht dabei mindestens einer der Folge-

Zustände eine höhere Bewertung als der Ist-Zustand, so wird der am besten bewertete Folge-Zustand zum neuen Ist-Zustand. Andernfalls wird die laufende Epoche abgeschlossen und liefert neben der aktuellen Auswahl in Form des Ist-Zustands auch das lineare Modell samt Bewertung an das Verfahren zurück.

7.1.3 Bewertung von Auswahlen

Die Bewertung der Auswahl des jeweiligen Ist-Zustandes vereint die Konzepte der K-Kreuzvalidierung (vgl. Abschnitt 5.2.5), der regularisierten linearen Regression (vgl. Abschnitt 5.4.2) und der gewichteten Mittelung linearer Regressionsmodelle (vgl. Abschnitt 5.4.3).
Innerhalb der Bewertung wird eine vorgegebene Anzahl n_{cross} von Kreuzvalidierungs-Schritten durchgeführt, die jeweils ein lineares Modell in standardisierter Form und eine Qualitäts-Bewertung liefern. Die n_{cross} Modelle werden, analog zu der Vorgehensweise in Abschnitt 7.1.1, entsprechend ihrer Qualität zusammengefasst, mit dem Unterschied, dass die verwendeten Regressoren stets die gleichen sind. Die resultierende Gesamtqualität ergibt sich entsprechend als gewichtete Mittelung aus den einzelnen Qualitäts-Bewertungen.
In jedem der n_{cross} Kreuzvalidierungs-Schritte werden die Trainingsdaten in K Partitionen aufgeteilt. Anschließend wird in K Schritten jeweils mit den Daten aus $K-1$ Partitionen ein lineares Modell erstellt und sowohl auf die Erstellungs-Daten der verwendeten $K-1$ Partitionen als auch auf die Validierungs-Daten der vebleibenden Partition angewandt. Diese Modellanwendungen führen zu zwei *MSE*-Werten, von denen der größere die Grundlage für die Bewertung der Qualität des Modells ist. Ausgewählt wird das Modell mit dem kleinsten (über die K Modellerstellungen) maximalen (über die jeweils zwei Anwendungen) *MSE*.
Bei der Erstellung der einzelnen Modelle für jeweils $K-1$ Partitionen werden n_α vorgebbare Regularisierungsparamter berücksichtigt. Dabei wird für jeden einzelnen Regularisierungsparameter ein eigenes Modell erstellt (vgl. Abschnitt 5.4.2) und, zur Validierung der Wahl des Regularisierungsparameters, wie zuvor, getrennt auf die Erstellungs- und die Validierungsdaten angewandt. Verwendet wird letztendlich der Regularisierungsparameter, für den der maximale *MSE*-Wert bezüglich der Erstellungs-Daten zum Einen und der Validierungs-Daten zum Anderen am geringsten ausfällt.

7.2 Ermittlung und Bewertung von Merkmalskombinationen

Wie in Abschnitt 7.1.3 beschrieben ist der Umfang der Rechenschritte für die Bewertung einer Auswahl von Merkmalen bereits für überschaubare Werte für n_α, K, n_{cross} und n_{epoch} recht hoch. Da diese Bewertung in jedem Suchschritt jeder Epoche sowohl auf den Ist-Zustand als auch auf alle Folge-Zustände anzuwenden ist, ergeben sich rasch sehr große Rechenzeiten. Aus diesem Grunde wurden für die folgenden Untersuchungen die Parameter entsprechend Tabelle 7.1 gewählt. Die n_α Regularisierungsparameter werden äquidistant aus dem Intervall [0,1) gewählt.

Parameter	Wert
n_α	16
K	6
n_{cross}	5
n_{epoch}	100
n_{initial}	2
b_{lo}	1
b_{hi}	4

Tabelle 7.1: Wahl der OTFS-Parameter für die Untersuchungen in Abschnitt 7.2

Die Wahl von n_{initial} zu zwei bewirkt, dass in jeder Epoche zunächst zwei Merkmale zufällig gewählt werden. Diese Start-Merkmale können nach und nach durch Rückwärts-Elimination mit darauf folgender Vorwärts-Selektion entfernt werden, beeinflussen aber wesentlich die Such-Region und die weiteren herangezogenen Merkmale. Insgesamt erhöht diese Maßnahme die Diversität der Merkmalsauswahl.
Durch die Werte von b_{hi} und n_{epoch} ergibt sich jeweils eine sehr große Anzahl berücksichtigter Merkmale, deren Einfluss auf die Komfortbewertung jeweils mit dem Betrag des zugehörigen Koeffizienten des standardisierten Modells identifiziert werden kann. Hierdurch wird implizit eine Anordnung der Merkmale nach deren Relevanz definiert. Für die Vergleichbarkeit der Ergebnisse der Merkmalsauswahl des OTFS-Verfahrens mit denen des Feature-Browsers werden für die folgenden Betrachtungen jeweils nur die ersten zwölf Merkmale in dieser Anordnung verwendet.
Die qualitativen Verläufe der Anteile der Regressoren für die einzelnen Modelle sind in den Abbildungen 7.1 bis 7.3 dargestellt. Die gepunktete Linie deutet jeweils die Grenze an, bis zu der die Regressoren für die Merkmalsauswahl herangezogen wurden. Die gestrichelte Linie stellt den Verlauf der kumulierten Regressoranteile dar. Die Y-Koordinate des Schnittpunkts dieser Kurve mit der Grenze für die Merkmalsauswahl entspricht jeweils dem relativen Anteil der ausgewählten Regressoren an dem linearen Modell. Das Komplement bezüglich 100% ergibt entsprechend den Regressoranteil des Modells, der der Merkmalsauswahl fehlt. Allgemein bleibt durch die Reduzierung der Regressorwahl auf die ersten zwölf der im Wesentlichen für die Erklärung der Änderungen in den subjektiven Daten durch die objektiven Daten verantwortliche Modellanteil erhalten. Die Werte schwanken zwischen 71% im Fall von Modell drei und 97% im Fall von Modell vier (vgl. jeweils Abbildung 7.2).

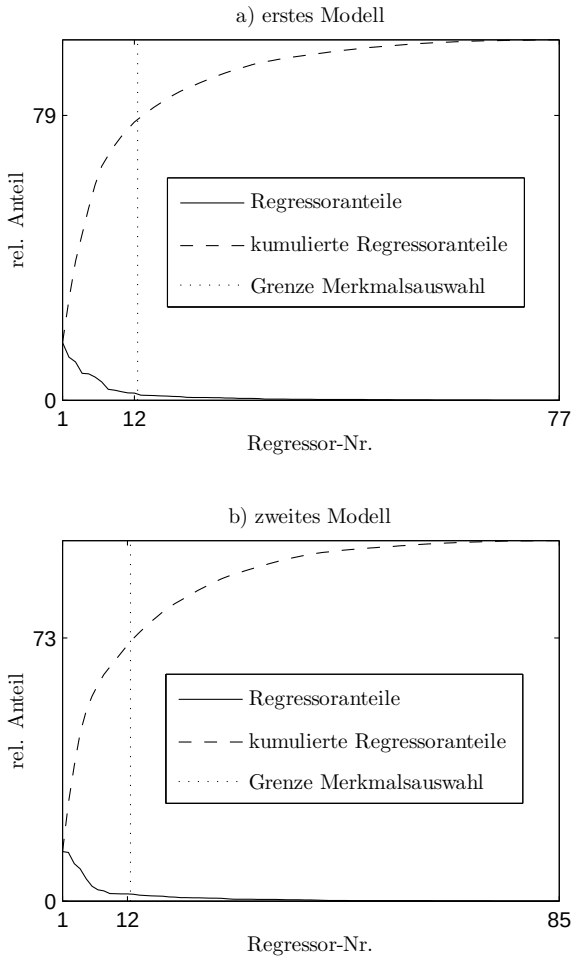

Abbildung 7.1: Verteilung der Regressoranteile der linearen OTFS-Modelle eins und zwei

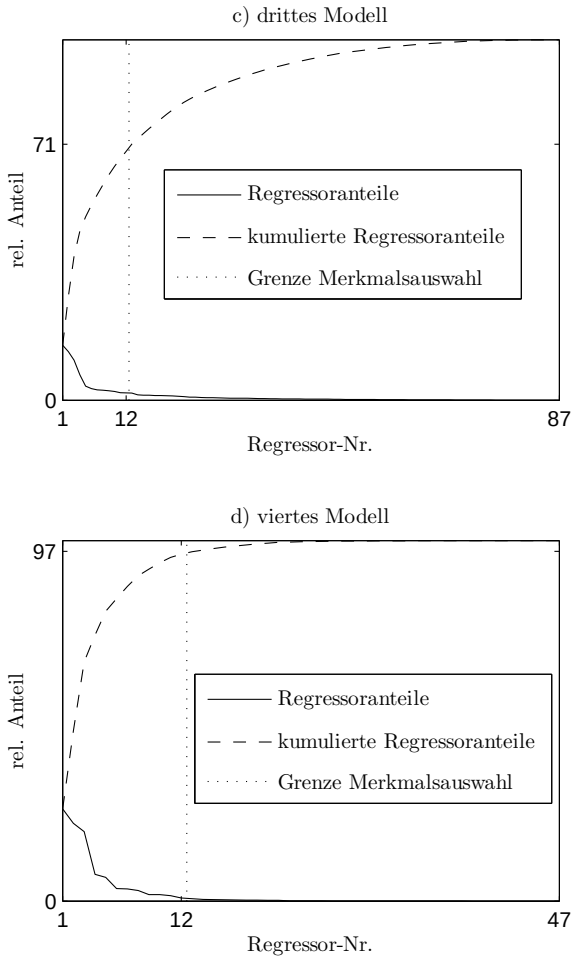

Abbildung 7.2: Verteilung der Regressoranteile der linearen OTFS-Modelle drei und vier

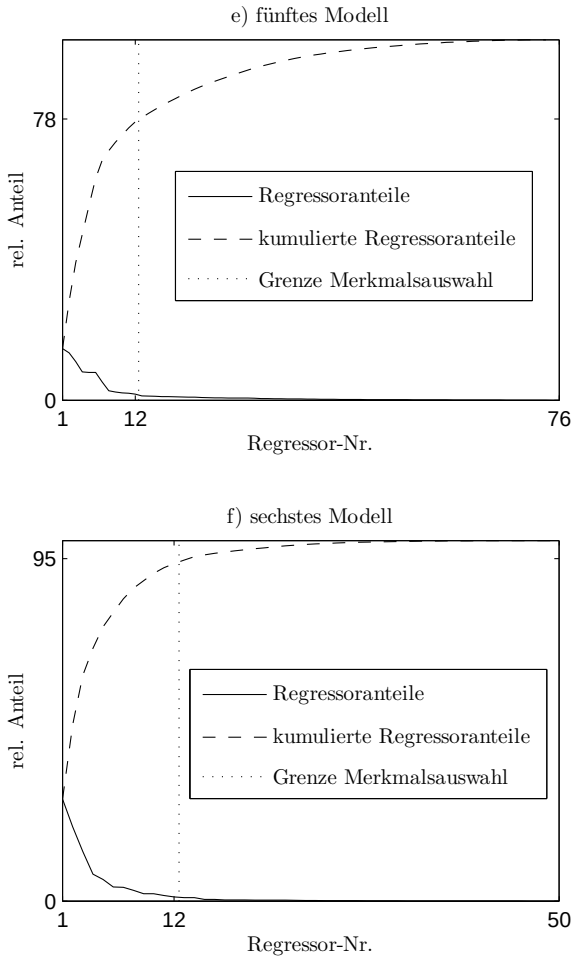

Abbildung 7.3: Verteilung der Regressoranteile der linearen OTFS-Modelle fünf und sechs

Die gewählten Merkmale der einzelnen Modelle (vgl. Tabelle 6.1) sind in den Tabellen 7.2 bis 7.7 dargestellt. Wie auf Grund der unterschiedlichen methodischen Ansätze und der Vielzahl an Kombinationsmöglichkeiten zu erwarten war, unterscheiden sich die Eingangskombinationen stark von denen aus Abschnitt 6.4. So verlieren allgemein bei den Merkmalsauswahlen nach dem OTFS-Verfahren die Merkmale zu der zweiten Approximation deutlich an Relevanz. Desweiteren rückt der Tableau-Mittelwert als Merkmal (vgl. Tabelle 4.1) bei dieser Methode deutlich mehr in den Hintergrund, wohingegen er noch bei den Tabellen 6.2 bis 6.7 überdurchschnittlich häufig vertreten war.

Nr.	**Kanal**	**Einleitung**	**Merkmal**	**Approximation**	**Achse**	$\mathbf{R^2}$
1	2	z	m_f	I	VA	0,69
2	1	x	a	I	VA	0,61
3	2	z	r_s	II	VA	0,12
4	6	–	a	I	VA	0,64
5	5	z	m_f	I	VA	0,55
6	4	z	φ	I	VA	0,10
7	6	–	a	II	VA	0,04
8	5	z	$\cos(\varphi)$	I	VA	0,26
9	4	z	r_s	I	VA	0,04
10	2	z	$\cos(\varphi)$	II	VA	0,01
11	2	z	a	II	VA	0,00
12	6	–	$\cos(\varphi)$	II	VA	0,00

Tabelle 7.2: Ergebnisse der Merkmalsauswahl über OTFS für das erste Vorhersagemodell (mittlere negative Kante Vorderachse)

Nr.	**Kanal**	**Einleitung**	**Merkmal**	**Approximation**	**Achse**	$\mathbf{R^2}$
1	4	z	r_b	I	HA	0,61
2	6	–	Δ_t	I	HA	0,59
3	1	x	m_f	I	HA	0,34
4	6	–	$\sin(\varphi)$	I	HA	0,08
5	1	x	$\overline{\mathbf{W}}$	II	HA	0,19
6	6	–	φ	I	HA	0,12
7	5	z	$\cos(\varphi)$	II	HA	0,05
8	3	x	a	II	HA	0,20
9	4	z	r_s	II	HA	0,01
10	5	z	m_f	I	HA	0,49
11	3	x	φ	II	HA	0,03
12	1	x	r_s	II	HA	0,12

Tabelle 7.3: Ergebnisse der Merkmalsauswahl über OTFS für das zweite Vorhersagemodell (mittlere negative Kante Hinterachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	6	–	a	I	VA	0,61
2	1	x	$\sin(\varphi)$	I	VA	0,59
3	1	x	φ	I	VA	0,34
4	1	x	Δ_t	I	VA	0,08
5	5	z	r_b	I	VA	0,19
6	2	z	r_b	I	VA	0,12
7	2	z	Δ_t	I	VA	0,05
8	4	z	$\sin(\varphi)$	II	VA	0,20
9	4	z	φ	II	VA	0,01
10	3	x	r_b	I	VA	0,49
11	2	z	φ	I	VA	0,03
12	2	z	r_s	I	VA	0,12

Tabelle 7.4: Ergebnisse der Merkmalsauswahl über OTFS für das dritte Vorhersagemodell (mittlere positive Kante Vorderachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	1	x	a	I	HA	0,19
2	1	x	Δ_t	I	HA	0,00
3	6	–	a	I	HA	0,56
4	1	x	$\cos(\varphi)$	I	HA	0,08
5	6	–	a	II	HA	0,27
6	3	x	Δ_f	II	HA	0,03
7	6	–	Δ_f	I	HA	0,48
8	4	z	m_f	I	HA	0,01
9	1	x	r_s	I	HA	0,00
10	2	z	r_s	II	HA	0,02
11	4	z	φ	II	HA	0,22
12	1	x	$\cos(\varphi)$	II	HA	0,04

Tabelle 7.5: Ergebnisse der Merkmalsauswahl über OTFS für das vierte Vorhersagemodell (mittlere positive Kante Hinterachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	3	x	r_b	I	VA	0,59
2	1	x	Δ_f	I	VA	0,66
3	6	–	a	I	VA	0,79
4	4	z	Δ_t	II	VA	0,38
5	3	x	Δ_f	I	VA	0,40
6	1	x	r_s	I	VA	0,66
7	2	z	Δ_t	II	VA	0,20
8	4	z	a	I	VA	0,13
9	4	z	Δ_f	I	VA	0,53
10	3	x	Δ_t	I	VA	0,04
11	1	x	a	I	VA	0,62
12	4	z	r_s	I	VA	0,32

Tabelle 7.6: Ergebnisse der Merkmalsauswahl über OTFS für das fünfte Vorhersagemodell (U-Rille Vorderachse)

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	R^2
1	6	–	a	I	HA	0,76
2	1	x	$\cos(\varphi)$	I	HA	0,14
3	1	x	Δ_f	I	HA	0,11
4	3	x	r_b	I	HA	0,06
5	3	x	Δ_f	I	HA	0,02
6	3	x	r_s	I	HA	0,32
7	1	x	r_s	II	HA	0,01
8	1	x	a	I	HA	0,45
9	3	x	a	II	HA	0,19
10	4	z	r_s	II	HA	0,04
11	5	z	m_f	II	HA	0,14
12	1	x	Δ_f	II	HA	0,00

Tabelle 7.7: Ergebnisse der Merkmalsauswahl über OTFS für das sechste Vorhersagemodell (U-Rille Hinterachse)

Analog zu den Untersuchungen in Abschnitt 6.4.2 wurden auch für die nach dem OTFS-Verfahren ermittelten Merkmalskombinationen Kohonen-Netze erstellt. Dabei wurden für die Vergleichbarkeit der Methoden die gleichen Einstellungen verwendet wie zuvor. Die Ergebnisse können den Abbildungen A.7 bis A.12 in Abschnitt A.1.2 entnommen werden. Verglichen mit den Untersuchungen aus Abschnitt 6.4.2 sind die Ergebnisse zu den Merkmalskombinationen nach dem OTFS-Verfahren zwar auf dem gleichen Niveau anzusiedeln, liefern aber auch leichte Verbesserungen. Bezüglich des Stetigkeitskriteriums sind diese in A.10 und A.11 festzustellen. Dagegen treten allerdings bei A.9 etwas größere Abstufungen auf als bei A.3. Die Anordnung hingegen konnte durchweg bei allen Modellen leicht (vgl. A.8 und A.9) bis etwas stärker (vgl. A.10 und A.11) verbessert werden.
Diese kleinen bis größeren Verbesserungen gegenüber der Untersuchungen aus Abschnitt 6.4.2 müssen außerdem vor dem Hintergrund betrachtet werden, dass aus der über das jeweilige lineare Modell gewählten Regressorkombination nur zwölf Einträge in die Merkmalsauswahl übernommen wurden. Somit bildet die jeweilige Auswahl das entsprechende lineare OTFS-Modell nicht vollständig sondern nur annähernd ab. Es bleibt zu zeigen, ob die Erstellung der Vorhersagemodelle auf Basis der neuen Merkmalskombinationen die positiven Erkenntnisse dieser Ergebnisse zu bestätigen vermag.

7.2.1 Erstellung von Vorhersagemodellen

Mit den in Abschnitt 7.2 ermittelten Merkmalskombinationen wurden analog zu der Vorgehensweise in Abschnitt 6.4.3 Vorhersagemodelle auf Basis künstlicher neuronaler Netze mit Feed-Forward-Struktur erstellt. Wie zuvor entspricht die Wahl der Modellparameter denen aus den vorherigen Untersuchungen in Kapitel 6.
Die Ergebnisse sind in der gewohnten Form den Abbildungen 7.4 bis 7.9 zu entnehmen. Tabelle 7.8 zeigt zusätzlich eine Zusammenfassung der erreichten Bestimmtheitsmaße sowie der Anzahlen der jeweils effektiv verwendeten Netzgewichte.
Im Vergleich zu den Modellen aus Abschnitt 6.4.3, bei denen die Merkmalsauswahl über den Feature-Browser vorgenommen wurde, fallen die Ergebnisse der Modellerstellung mit den OTFS-Auswahlen nicht ganz so gut aus. Am deutlichsten fällt dabei das Hinterachsmodell für die mittlere positive Kante ab, wobei hier auch wieder auf die extrem kleine Notenspreizung von 1,5 hingewiesen sei. Die übrigen Modelle sind dagegen auch bei der Anwendung auf die Testdaten wieder gut bis sehr gut, reichen aber nicht ganz an die Qualität der Modelle aus Abschnitt 6.4.3 heran - weder bezüglich der Streudiagramme noch bezüglich der Modell-Fehler.
Die mit den OTFS-Auswahlen erstellten Modelle bestätigen somit nicht die Ergebnisse der Bewertung der Merkmalsauswahlen aus Abschnitt 7.2. Eine Möglichkeit dafür könnte die zu starke Anpassung der Merkmalsauswahl an die Struktur der linearen Regressionsmodelle sein, bezüglich der sie ermittelt wurden. Ein weiterer Hinweis hierauf sind die Anzahlen der effektiv genutzten Netzgewichte, die in Tabelle 7.8 durchweg kleiner sind als in Tabelle 6.8. Das Potential der künstlichen neuronalen Netze wird also in diesem Fall nicht so intensiv genutzt wie bei den Vorhersagemodellen aus Abschnitt 6.4.3. Als

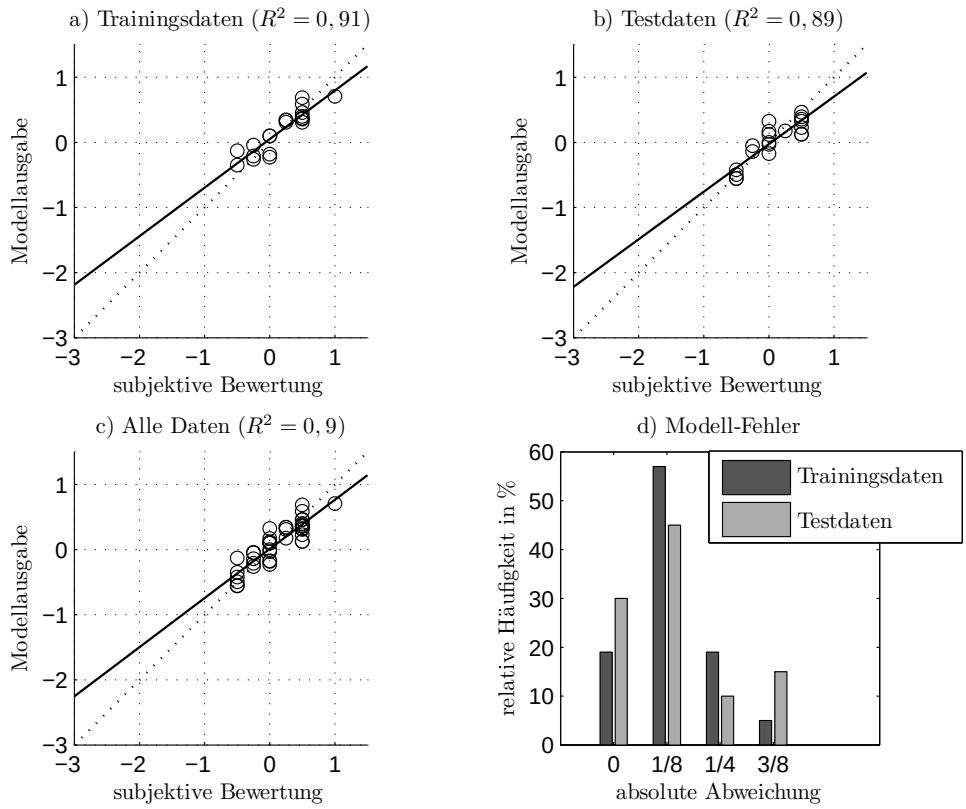

Abbildung 7.4: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das erste Modell (OTFS-Merkmalsauswahl)

Konsequenz muss daher vorläufig festgestellt werden, dass das OTFS-Verfahren nicht die optimale Wahl für die Ermittlung von Eingangskombinationen für künstliche neuronale Netze darstellt. Daher werden in Abschnitt 7.3 als letzte Variante von Vorhersagemodellen die von dem OTFS-Verfahren erstellten linearen Modelle genauer betrachtet, um den Nutzen der Methode im Gesamtkontext bewerten zu können.
Eine weitere wichtige Erkenntnis der deutlichen qualitativen Unterschiede bei den Ergebnissen der Modellerstellung ist, dass der Merkmalsauswahl in diesem Zusammenhang tatsächlich eine gewichtige Rolle zukommt und dass sie einen großen Einfluss auf die Qualität der Vohersagemodelle hat.

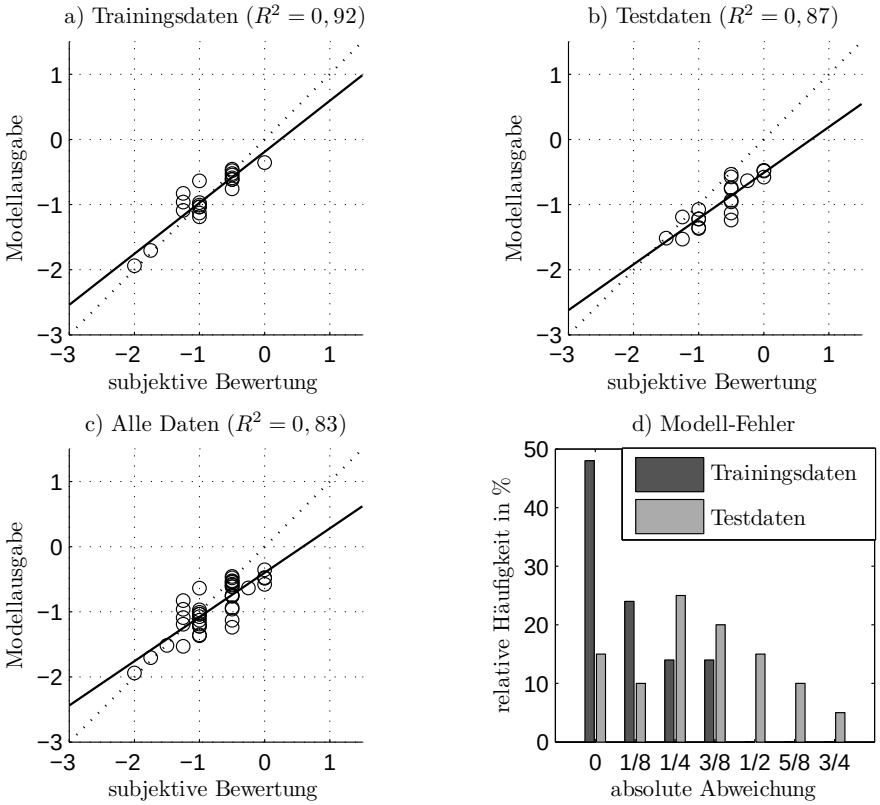

Abbildung 7.5: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das zweite Modell (OTFS-Merkmalsauswahl)

Modell	R^2 (Training)	R^2 (Test)	R^2 (Gesamt)	eff. # Netzgewichte
1	0,91	0,89	0,90	9,34
2	0,92	0,87	0,83	8,06
3	0,68	0,81	0,73	5,13
4	0,91	0,88	0,89	10,0
5	0,87	0,89	0,89	7,27
6	0,95	0,93	0,94	11,2

Tabelle 7.8: Bestimmheitsmaße und effektive Anzahl verwendeter Netzgewichte zu den sechs verschiedenen Vorhersagemodellen aus Kapitel 7

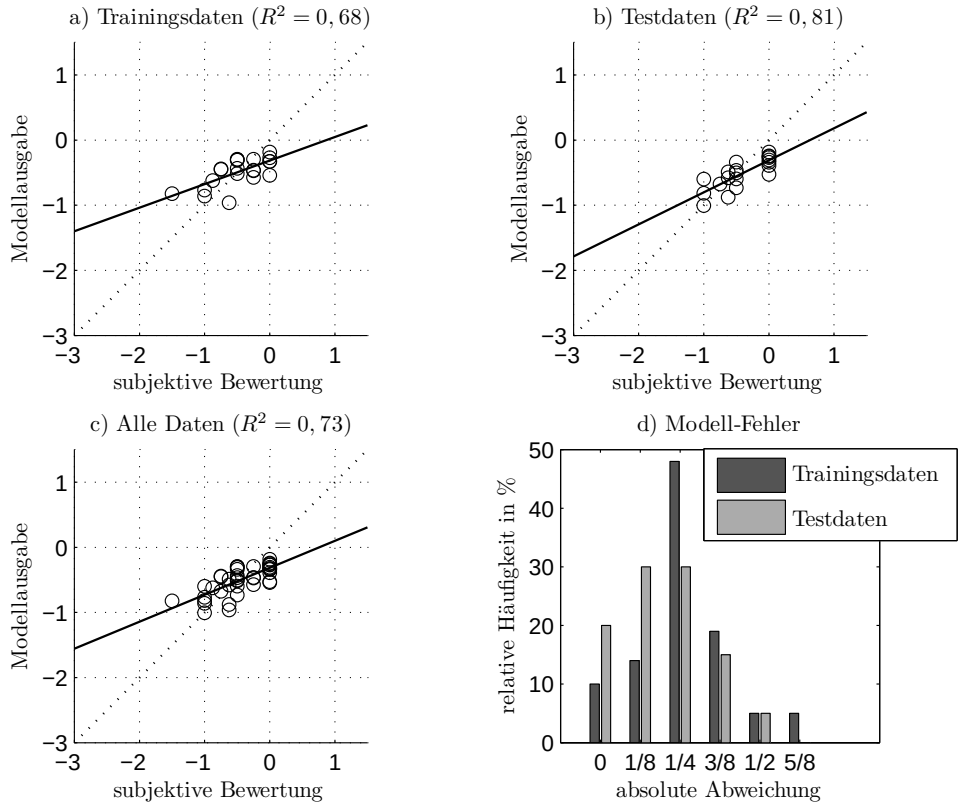

Abbildung 7.6: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das dritte Modell (OTFS-Merkmalsauswahl)

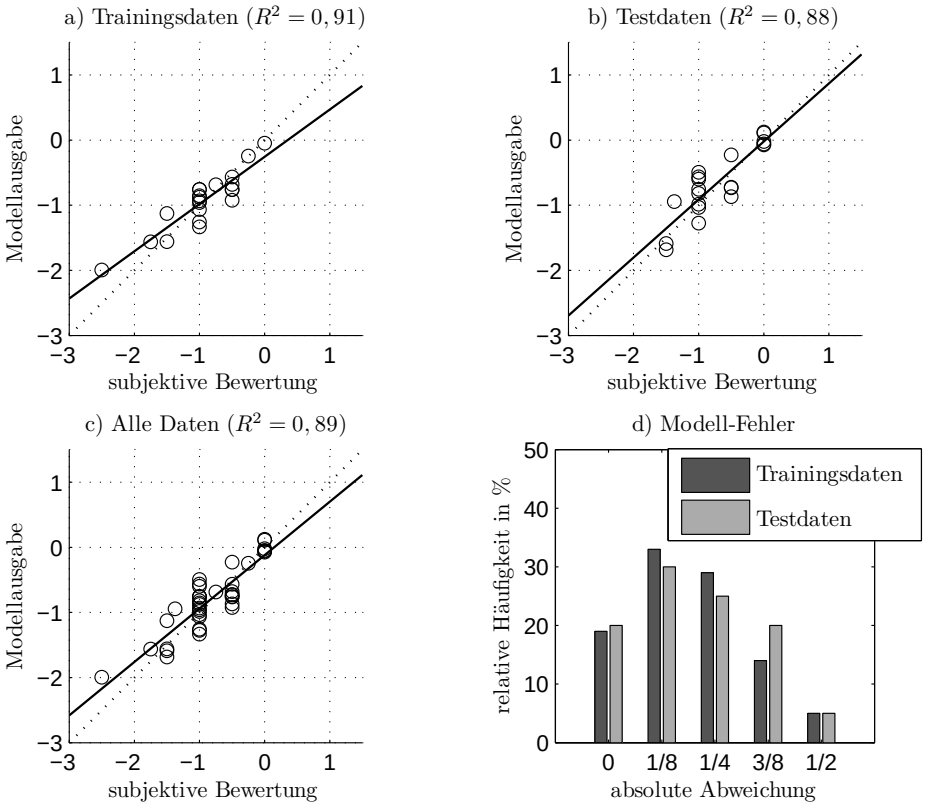

Abbildung 7.7: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das vierte Modell (OTFS-Merkmalsauswahl)

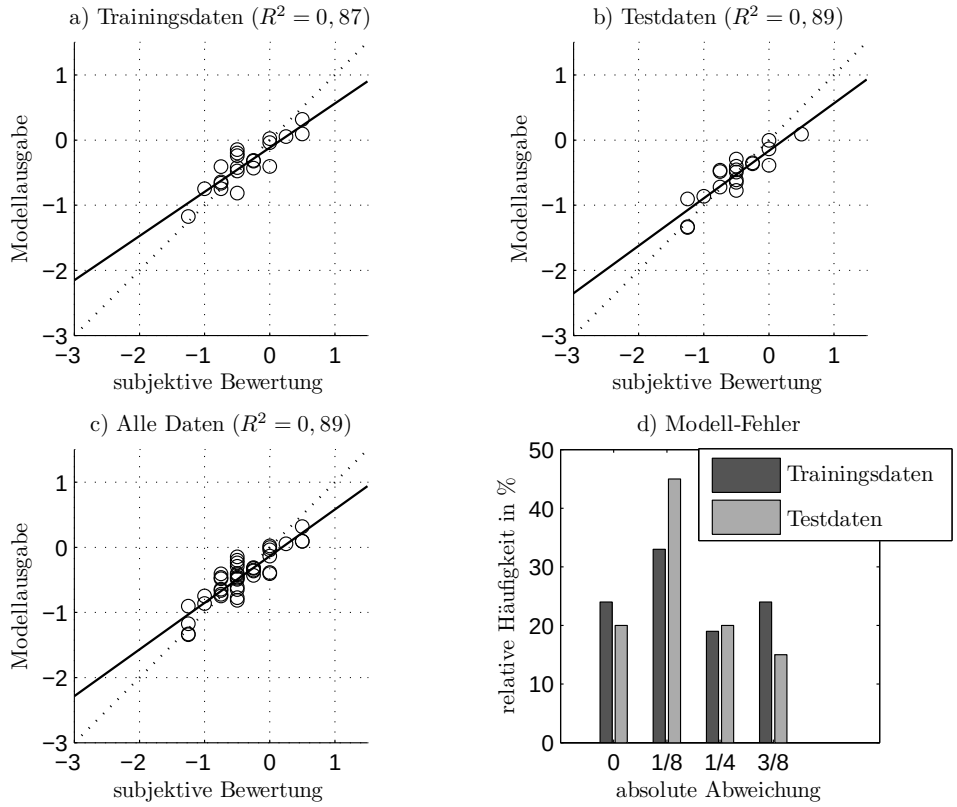

Abbildung 7.8: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das fünfte Modell (OTFS-Merkmalsauswahl)

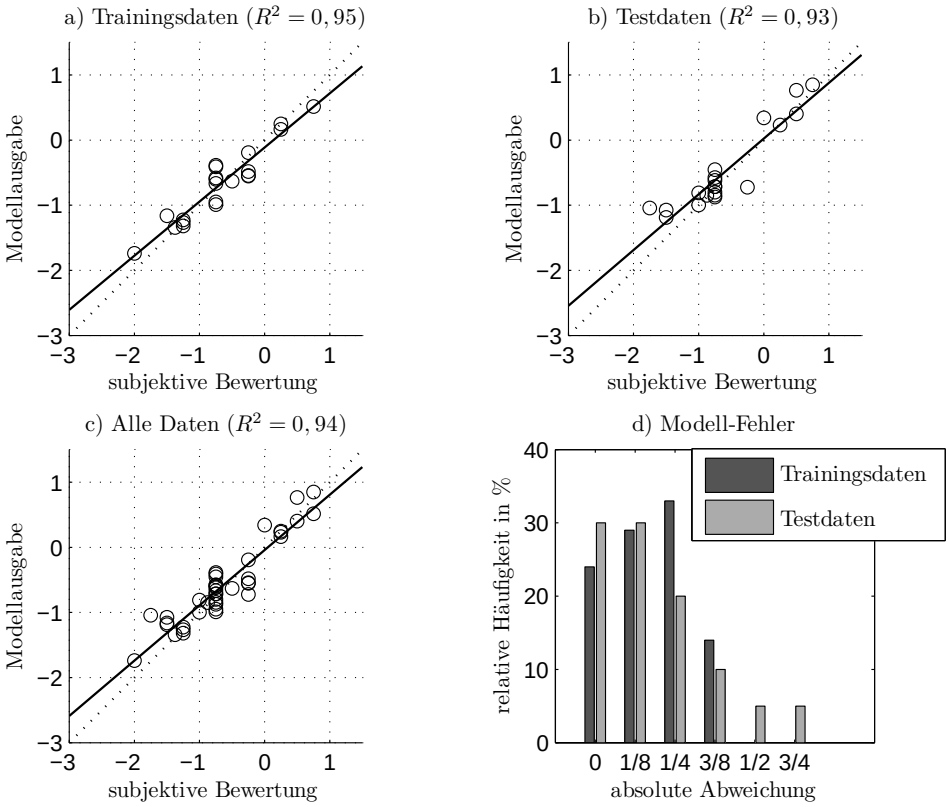

Abbildung 7.9: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das sechste Modell (OTFS-Merkmalsauswahl)

7.3 Analyse der linearen OTFS-Modelle

Als letzte Variante von Vorhersagemodellen werden in diesem Abschnitt nach den künstlichen neuronalen Netzen für die zwei verschiedenen Methoden der Merkmalsauswahl die linearen Modelle betrachtet, die jeweils im Rahmen der Merkmalsauswahl über das OTFS-Verfahren ermittelt wurden. Die vollständigen Regressorlisten dieser Modelle können Abschnitt A.2 entnommen werden. Man überprüft leicht, dass jeweils die ersten zwölf Einträge mit den Tabellen 7.2 bis 7.7 übereinstimmen.
Die Streudiagramme und die Verteilungen der Modell-Fehler für die linearen Modelle sind in den Abbildungen 7.10 bis 7.15 dargestellt. Die erreichten Bestimmtheitsmaße sowie die jeweiligen Regressoranzahlen sind in Tabelle 7.9 zusammengefasst.
Die Qualität dieser letzten Ergebnisse ist nicht wesentlich geringer einzustufen als die bezüglich der künstlichen neuronalen Netze in Abschnitt 7.2.1. Die Anwendung der Modelle auf die Testpartition führt sogar überwiegend zu besseren Ergebnissen. Da es sich dabei um lineare Modelle und keine komplexen Feed-Forward-Strukturen künstlicher neuronaler Netze handelt, ist dies eine Besonderheit. Die großen Regressorzahlen in Tabelle 7.9 und die Regressorlisten in Abschnitt A.2 zeigen, dass die Modelle in der Lage sind, den durch die komfortrelevanten Merkmale beschriebenen Schwingungszustand äußerst detailliert zu erfassen. Trotz der großen Anzahl an Regressoren und der damit verbundenen großen Zahl an fixierten Modellparametern sind die Modelle selber keineswegs überangepasst. Diese Vielfalt entsteht lediglich durch die Zusammenfassung vieler kleinerer Regressionsmodelle (vgl. Abschnitt 5.4.3), die jeweils über maximal vier Regressoren verfügen (vgl. Tabelle 7.1).
Neben der Tatsache, dass durch die große Anzahl herangezogener Merkmale eine genauere Einschätzung des Schwingungszustands ermöglicht wird, besteht ein weiterer Vorteil der linearen Modelle in ihrer Transparenz. Der jeweilige Einfluss eines Regressors auf die Komfortbewertung wird nachvollziehbar und vergleichbar mit den anderen Einflüssen. Man sieht den Modellen unmittelbar an, welche Zustände welche Komfortbewertung hervorrufen, wodurch es eher möglich wird, konstruktive Maßnahmen zur Erhöhung des Schwingungskomforts der untersuchten Fahrzeuge (oder Fahrzeugmodelle) zu ergreifen, als es bei den künstlichen neuronale Netzen der Fall ist.
Die hohe Qualität der linearen Modelle bestätigt die in Abschnitt 7.2.1 aufgestellte Vermutung, dass die Merkmalsauswahl über die OTFS-Methode sehr angepasst ist an die Struktur der linearen Regressionsmodelle, was möglicherweise ein Grund dafür sein könnte, dass die Anwendung der Merkmalskombinationen auf künstliche neuronale Netze keine noch deutlicheren Verbesserungen liefert.

Modell	R^2 (Training)	R^2 (Test)	R^2 (Gesamt)	# Regressoren
1	0,83	0,86	0,84	77
2	0,84	0,87	0,84	85
3	0,58	0,88	0,72	87
4	0,82	0,90	0,86	47
5	0,85	0,93	0,89	76
6	0,95	0,92	0,94	50

Tabelle 7.9: Bestimmheitsmaße und Regressoranzahlen der sechs linearen OTFS-Modelle

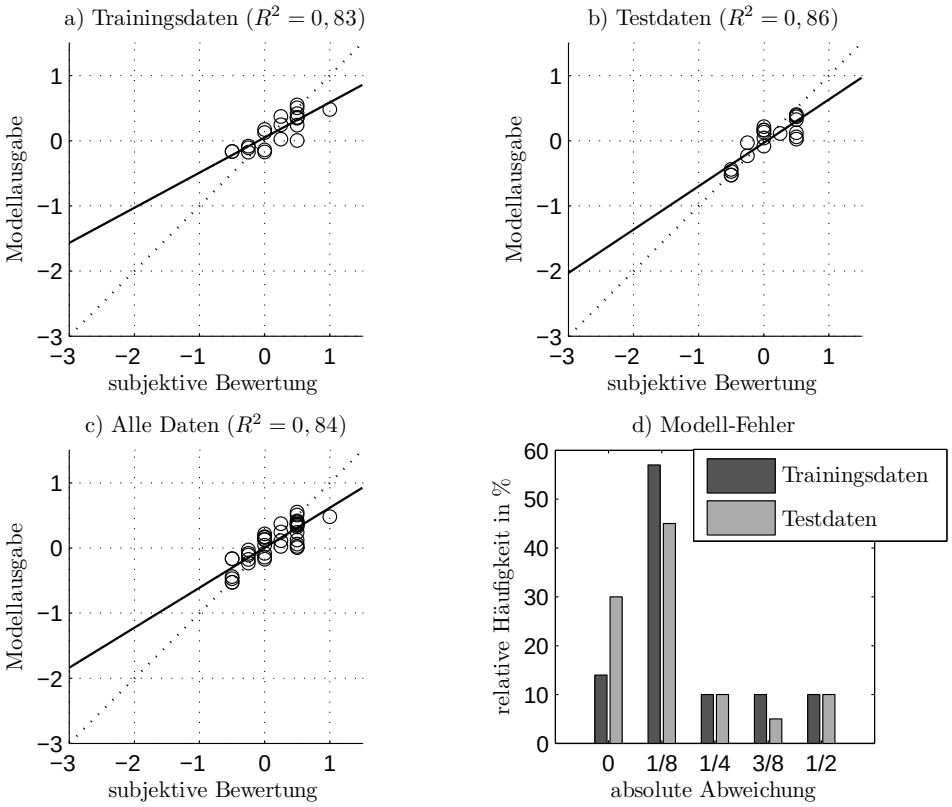

Abbildung 7.10: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das erste lineare OTFS-Modell

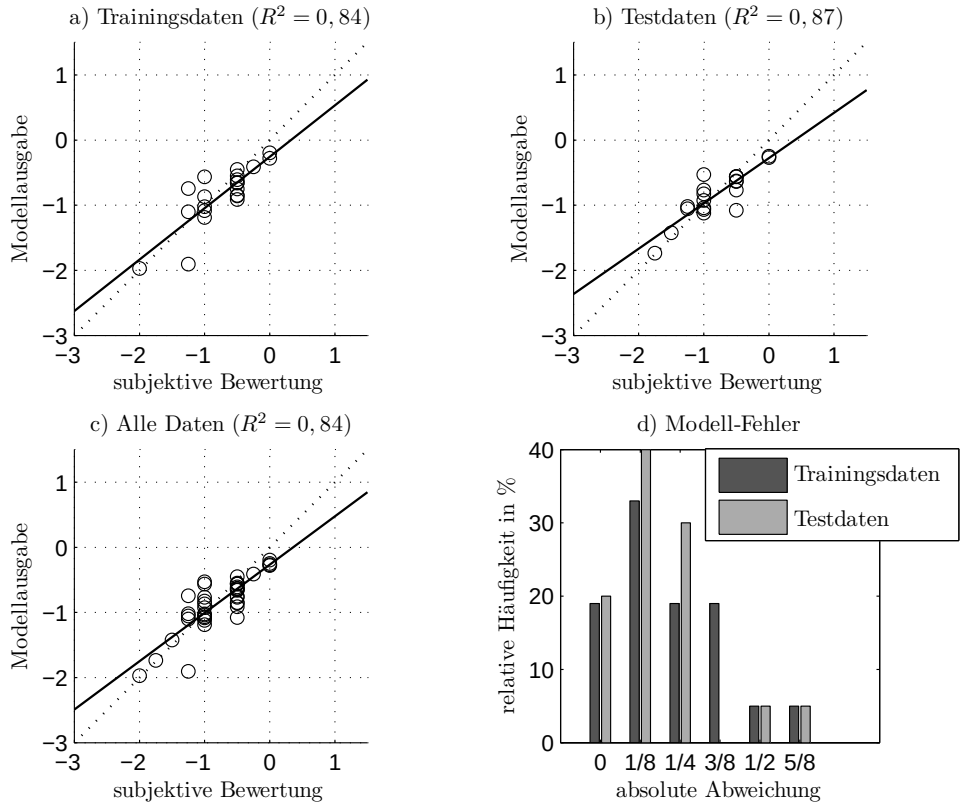

Abbildung 7.11: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das zweite lineare OTFS-Modell

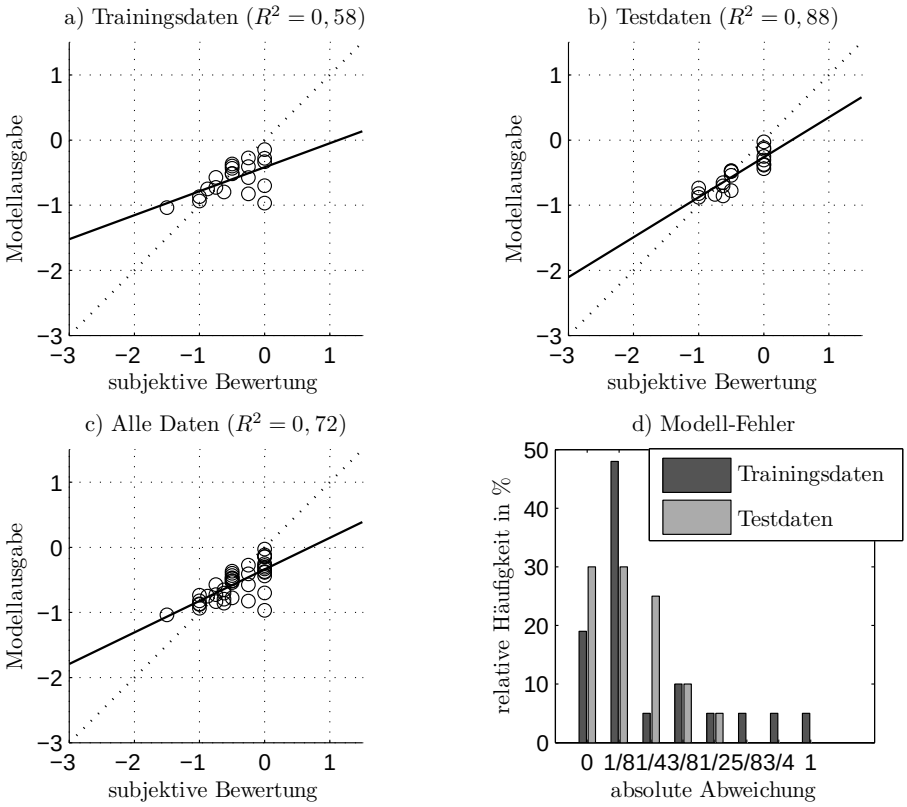

Abbildung 7.12: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das dritte lineare OTFS-Modell

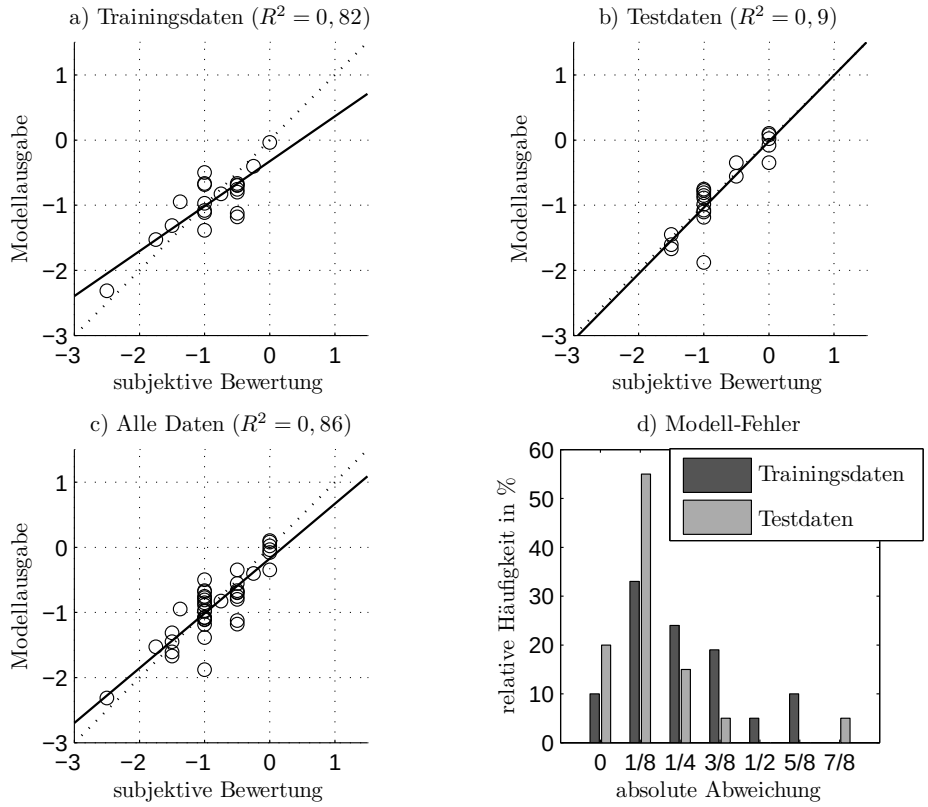

Abbildung 7.13: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das vierte lineare OTFS-Modell

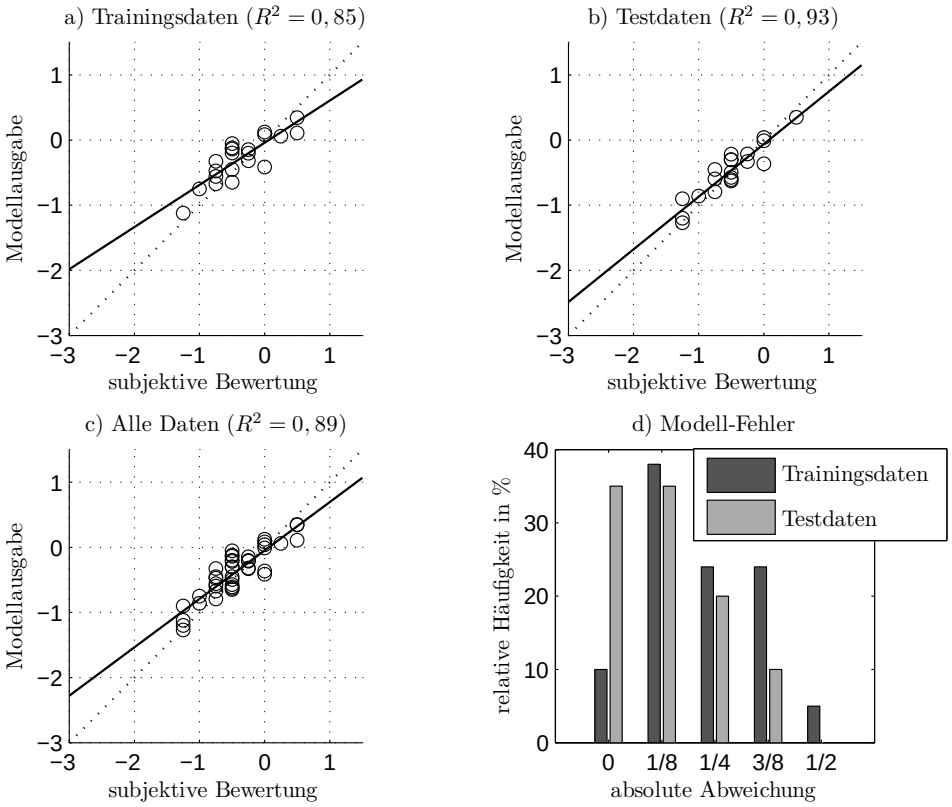

Abbildung 7.14: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das fünfte lineare OTFS-Modell

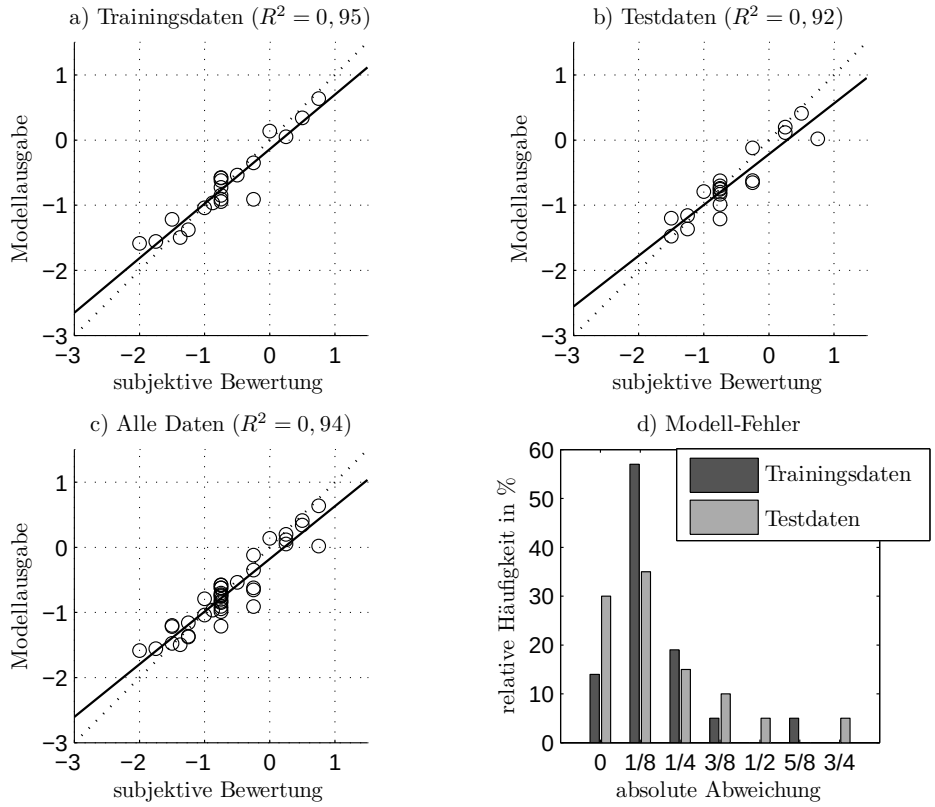

Abbildung 7.15: Streudiagramme der Modellausgaben und Verteilung der Ausgabe-Fehler für das sechste lineare OTFS-Modell

8 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine Methode entwickelt, die es ermöglicht, den Fahrkomfort von PKW basierend auf Messungen von Beschleunigungen und Luftschall in der Fahrerumgebung vorherzusagen. Die dabei verwendete, auf Gabor-Wavelets basierende Datentransformation konnte ihrer Aufgabe, die komfortrelevanten Signalanteile in eine geeignetere Darstellung zu überführen, gerecht werden. Für die Extrahierung aussagekräftiger Merkmale aus den transformierten Daten wurde eine eigene Methode entwickelt, die auf der Approximation der Wavelet-Tableaus durch ein parametriertes Muster beruht. Die extrahierten Merkmale stellten sich in den Untersuchungen sowohl als beschreibend in Hinblick auf die zugrunde liegenden Daten als auch als robust vor dem Hintergrund kleinerer Störungen in der Tableauausrichtung heraus. Dies wird zum einen durch teilweise relativ hohe Bestimmtheismaße bei der Erstellung linearer Modelle auf Basis einzelner Merkmale, vor allem aber durch die insgesamt erreichte Qualität der Vorhersagemodelle bei Verwendung spezieller Kombinationen von Merkmalen bestätigt.
In Hinblick auf die Auswahl von Merkmalskombinationen für die Vorhersagemodelle wurden zwei multivariate Methoden entwickelt und erprobt. Die Methode des Feature-Browsers liefert statistisch ermittelte Empfehlungen für den Anwender, lässt ihm aber die freie Entscheidung bei der Merkmalsauswahl und passt sich flexibel an dessen Eingaben an. So kann der Anwender durch seine Erfahrung, seinen Wissensschatz und unterstützt durch die aktuellen Empfehlungen des Feature-Browsers rasch zu gut geeigneten Merkmalskombinationen gelangen, was dadurch gestützt wird, dass in dieser Arbeit die Verwendung des Feature-Browsers langfristig zu den Vorhersagemodellen mit der höchsten Qualität führte.
Die OTFS-Methode konnte vor allem durch die damit erstellten linearen Modelle überzeugen, die jeweils hervorragende Ergebnisse bei der Anwendung auf die Testdaten erzielten und damit vielversprechende Generalisierungseigenschaften besitzen. Die mit der Methode ermittelten Merkmalskombinationen zeigten gute Ergebnisse bei der Analyse mit selbstorganisierenden Karten nach Kohohnen. Die damit erstellten Modelle auf Basis künstlicher neuronaler Netze wiesen leichte Verbesserungen gegenüber den linearen Modellen auf, konnten aber nicht an die Qualität der Vorhersagemodelle heranreichen, deren Merkmalskombinationen über den Feature-Browser ermittelt wurden. Als unmittelbare Konsequenz der unterschiedlichen Modellqualitäten kann somit festgestellt werden, dass die Art der Auswahl der Eingangskombinationen einen wesentlichen Einfluss auf die Ergebnisse der Modellerstellung und -anwendung hat.
Für die Erstellung der künstlichen neuronalen Netze zur Komfortvorhersage wurde jeweils die Trainingsmethode der Bayesschen Regularisierung verwendet. Hierdurch konnten Netze mit ausbalancierten Konstellationen der Verbindungsgewichte erstellt werden, von denen

im Schnitt nur knapp zehn (von 57 möglichen) effektiv genutzt wurden. Für die Erstellung der linearen Regressionsmodelle konnte die Kombination von Regularisierungs-Maßnahmen, K-Kreuzvalidierung, Data-Division und des Konzepts der gewichteten Mittelung mehrerer Modellinstanzen durch die erzielten Ergebnisse überzeugen. Ihre Qualität reicht zwar nicht an die der ersten Variante der künstlichen neuronalen Netze heran, dafür bieten sie einen höhren Grad an Transparenz und, durch ihre großen Regressoranzahlen, jeweils eine umfassendere Analyse des Schwingungszustandes.

Mit den insgesamt drei Varianten liegen Vorhersagemodelle für die drei Hindernistypen und die jeweils zwei Kriterien vor, die durch ihre Genauigkeit und die zu erwartende Generalisierungfähigkeit überzeugen. Das Ziel der Erstellung von Prädiktionsmodellen für die Vorhersage des Fahrkomforts von PKW bei Einzelhindernisüberfahrt konnte also für den durch die Versuche abgedeckten Rahmen erfüllt werden.

Die Anwendung der Modelle auf die Testdaten und die getroffenen Maßnahmen vor dem Hintergrund der Gefahr einer Überanpassung der Modelle an die Trainingsdaten lassen zudem eine gute Generalisierungsfähigkeit der Vorhersagemodelle erwarten. Es muss geschlossen werden, dass die trainierten künstlichen neuronalen Netze auf Daten weiterer Fahrzeuge, die vom Schwingungsverhalten nicht stark von den untersuchten Varianten abweichen, mit einer vergleichbaren Qualität angewendet werden können. Entsprechend ist nicht damit zu rechnen, dass die Modelle für die Komfortvorhersage auf Fahrzeuge mit deutlichen Abweichungen in Hinblick auf Karosserie, Abmessungen und Trägheitseigenschaften sinnvoll angewendet werden können.

Als Konsequenz der geringen Notenspreizungen, die den erstellten Modellen zugrunde liegen, kann ebenso nicht erwartet werden, dass andere Fahrzeuge, deren Komfortwerte in anderen Bereichen anzusiedeln sind, von den Modellen korrekt bewertet werden. Hierzu fehlt den künstlichen neuronalen Netzen die Fähigkeit der Extrapolation. Außerdem muss davon ausgegangen werden, dass die objektiven Daten entsprechender Fahrzeuge sich zu stark von denen unterscheiden würden, die nach dem Training der Modelle in diese eingeprägt wurden.

Die Ergebnisse stellen daher keine vollständig geschlossene Lösung für die universelle Komfortbewertung in dem genannten Rahmen dar. Sie bestärken und stützen vielmehr den eingeschlagenen Weg, der Verfahren und Ansätze bereitstellt, entsprechende Modelle selber zu entwickeln und zu validieren. Für diesen Schritt sollte zukünftig in Hinblick auf die Versuchsplanung eine deutliche Vergrößerung der zu erwartenden Notenspreizungen anvisiert werden, da gerade die Modelle zu entsprechenden Daten (Hinterachsbewertung der U-Rille und Hinterachsbewertung der mittleren positiven Kante) mit die höchste Qualität erreichen konnten. Zudem würde dies die Anwendbarkeit der erstellten Modelle auf ein größeres Feld erweitern.

Eine Erweiterung der Verfahren und Ansätze auf alternative Formen der subjektiven Bewertung ist ebenfalls möglich. So wurden bereits während der Entwicklung Versuche durchgeführt, an Stelle der absoluten Bewertungen relative vorherzusagen. Der wesentliche Aufwand einer entsprechenden Anpassung ist die Ableitung von Kennzahlen, die auf den vorhandenen Merkmalen aufbauen, aber den differenzierten Schwingungszustand

zweier an Stelle des absoluten Schwingungszustandes einer einzelnen Fahrzeugvariante beschreiben. Außerdem könnten an Stelle der Bewertungen eines einzigen Testfahrers die ganzer Gruppen herangezogen werden, um die Optimierung des Schwingungskomforts mehr in die Richtung der Sicht des Kunden zu verlagern. Die reproduzierbare Durchführung entsprechender Probandenversuche am Institut für Fahrzeug- und Antriebssystemtechnik der Helmut-Schmidt-Universität im Sommer 2011 stellte sich allerdings als schwierig heraus. Für diesen Zweck sind spezielle Ride-Simulatoren [Ada07] besser geeignet, auch wenn sie nicht alle Aspekte einer Realfahrt abbilden können.

A Anhang

A.1 Ergebnisse der SOM-Bewertung

Im Folgenden sind die Ergebnisse der Bewertung der Merkmalsauswahlen mit der Feature-Browser-Methode (Abschnitt A.1.1) und der OTFS-Methode (Abschnitt A.1.2) dargestellt. Für die Erstellung der Kohonen-Netze wurden jeweils die in Abschnitt 6.4.2 ausgeführten Einstellungen verwendet.

A.1.1 Feature-Browser-Auswahl

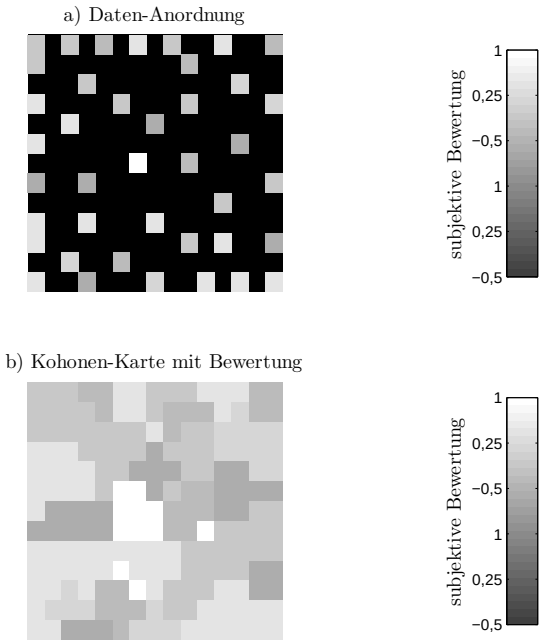

Abbildung A.1: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem ersten Vorhersagemodell (vgl. Tabelle 6.2)

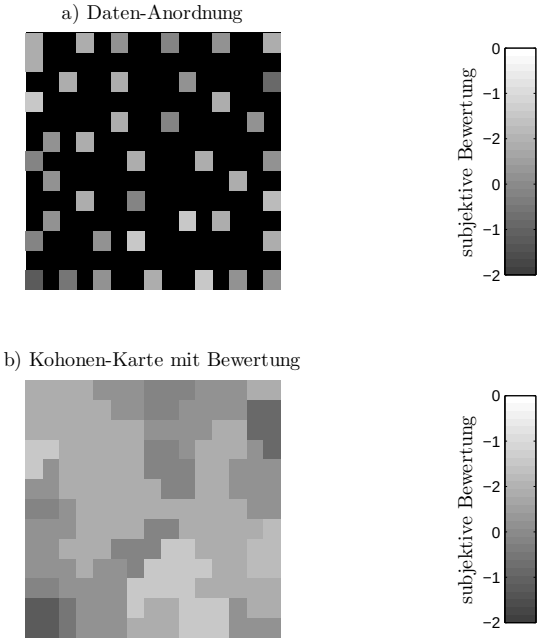

Abbildung A.2: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem zweiten Vorhersagemodell (vgl. Tabelle 6.3)

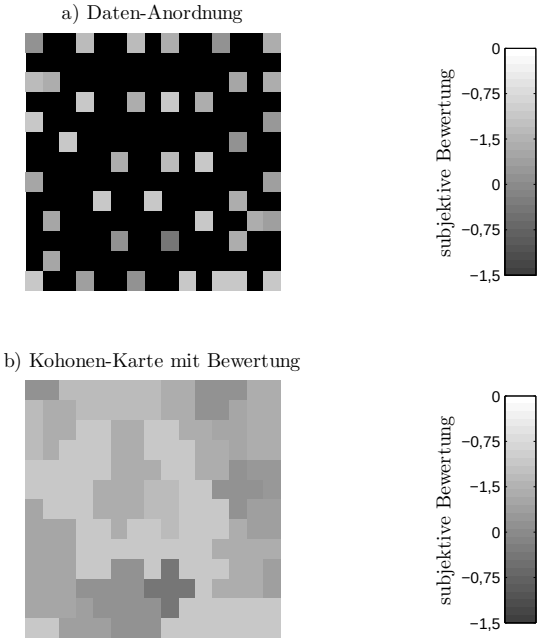

Abbildung A.3: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem dritten Vorhersagemodell (vgl. Tabelle 6.4)

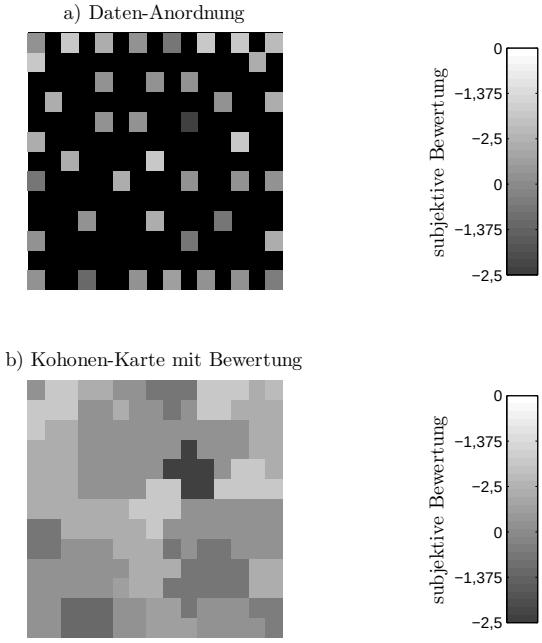

Abbildung A.4: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem vierten Vorhersagemodell (vgl. Tabelle 6.5)

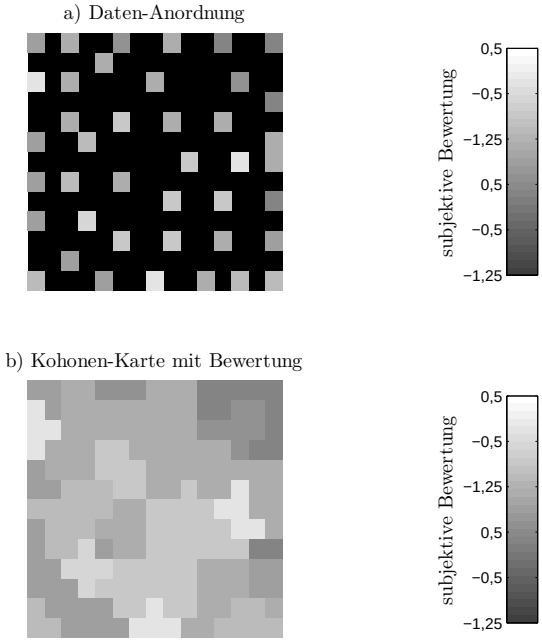

Abbildung A.5: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem fünften Vorhersagemodell (vgl. Tabelle 6.6)

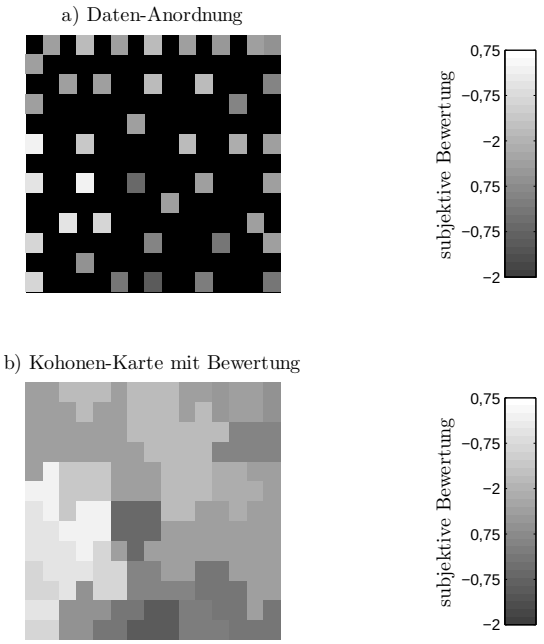

Abbildung A.6: Ergebnisse der SOM-Analyse der Eingangsauswahl über den Feature-Browser zu dem sechsten Vorhersagemodell (vgl. Tabelle 6.7)

A.1.2 OTFS-Auswahl

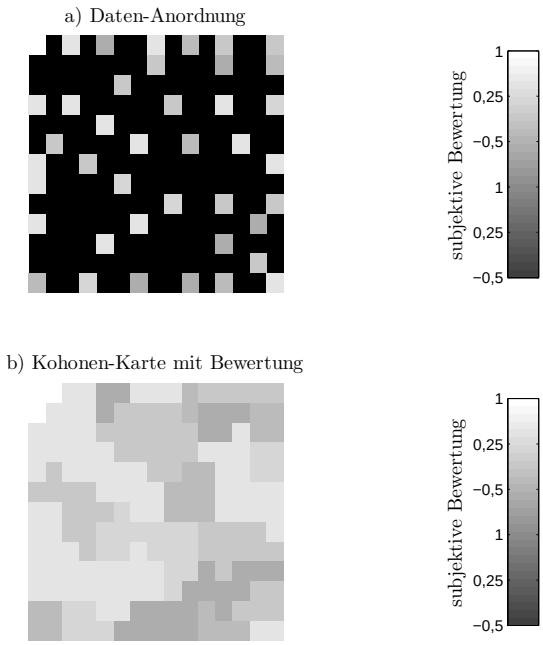

Abbildung A.7: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem ersten Vorhersagemodell (vgl. Tabelle 7.2)

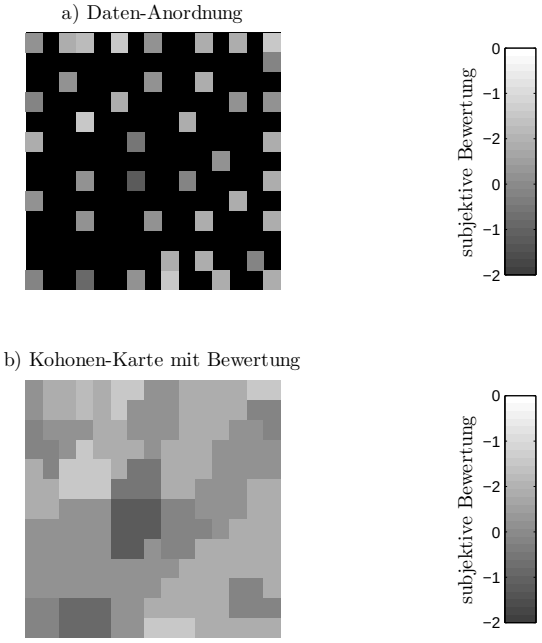

Abbildung A.8: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem zweiten Vorhersagemodell (vgl. Tabelle 7.3)

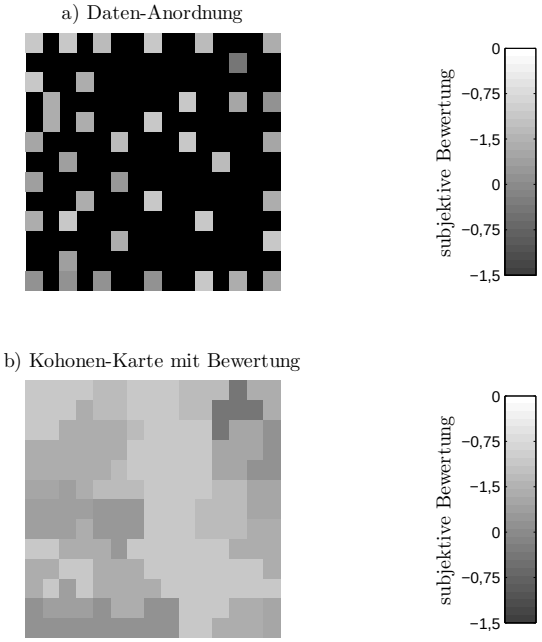

Abbildung A.9: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem dritten Vorhersagemodell (vgl. Tabelle 7.4)

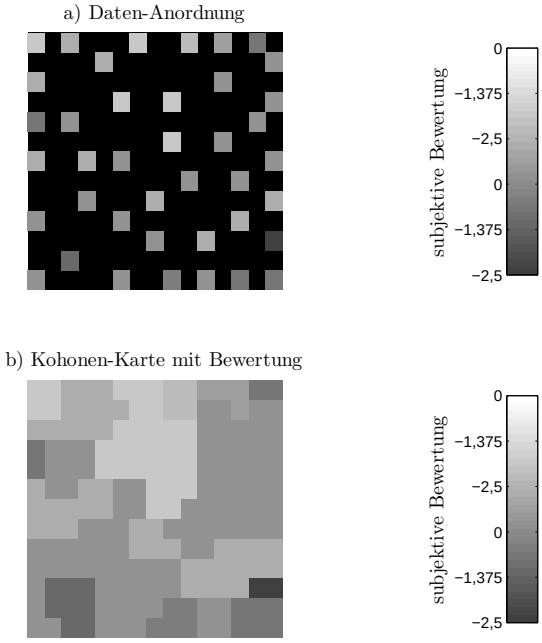

Abbildung A.10: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem vierten Vorhersagemodell (vgl. Tabelle 7.5)

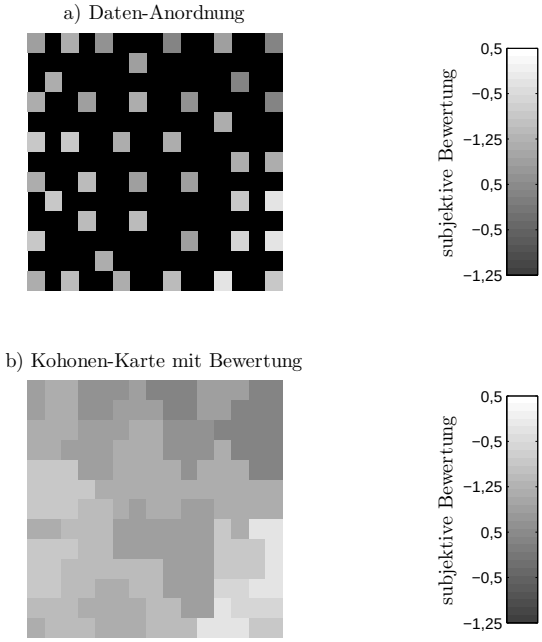

Abbildung A.11: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem fünften Vorhersagemodell (vgl. Tabelle 7.6)

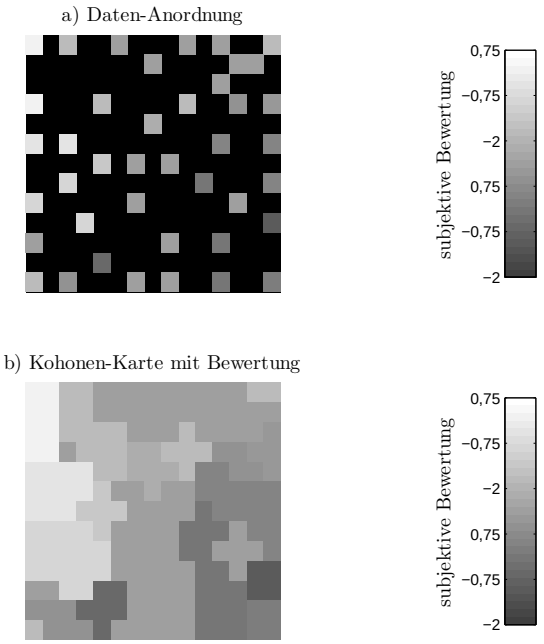

Abbildung A.12: Ergebnisse der SOM-Analyse der Eingangsauswahl über OTFS zu dem sechsten Vorhersagemodell (vgl. Tabelle 7.7)

A.2 Regressorlisten der linearen OTFS-Modelle

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	2	z	m_f	I	VA	16,01%
2	1	x	a	I	VA	11,95%
3	2	z	r_s	II	VA	10,62%
4	6	–	a	I	VA	7,48%
5	5	z	m_f	I	VA	7,31%
6	4	z	φ	I	VA	6,43%
7	6	–	a	II	VA	5,13%
8	5	z	$\cos(\varphi)$	I	VA	3,03%
9	4	z	r_s	I	VA	2,81%
10	2	z	$\cos(\varphi)$	II	VA	2,4%
11	2	z	a	II	VA	2,04%
12	6	–	$\cos(\varphi)$	II	VA	2%
13	6	–	Δ_t	I	VA	1,41%
14	6	–	m_f	I	VA	1,34%
15	2	z	m_f	II	VA	1,32%
16	2	z	a	I	VA	1,21%
17	2	z	$\sin(\varphi)$	II	VA	1,18%
18	4	z	$\sin(\varphi)$	I	VA	1,04%
19	2	z	Δ_f	II	VA	1%
20	6	–	m_f	II	VA	0,79%
21	4	z	$\overline{\mathbf{W}}$	II	VA	0,77%
22	4	z	a	II	VA	0,77%
23	6	–	Δ_f	II	VA	0,77%
24	4	z	Δ_f	I	VA	0,72%
25	1	x	r_s	I	VA	0,71%
26	2	z	φ	I	VA	0,63%
27	4	z	r_b	I	VA	0,61%
28	5	z	Δ_t	I	VA	0,54%
29	5	z	Δ_t	II	VA	0,53%
30	3	x	$\cos(\varphi)$	I	VA	0,53%
31	1	x	$\cos(\varphi)$	I	VA	0,49%
32	6	–	r_b	I	VA	0,34%
33	5	z	$\cos(\varphi)$	II	VA	0,33%
34	2	z	φ	II	VA	0,32%
35	3	x	r_s	I	VA	0,31%
36	6	–	r_b	II	VA	0,28%
37	6	–	r_s	I	VA	0,27%

38	2	z	$\sin(\varphi)$	I	VA	0,25%
39	4	z	r_s	II	VA	0,25%
40	5	z	Δ_f	II	VA	0,24%
41	4	z	φ	II	VA	0,24%
42	2	z	r_b	I	VA	0,24%
43	5	z	$\sin(\varphi)$	II	VA	0,22%
44	6	–	Δ_t	II	VA	0,22%
45	6	–	r_s	II	VA	0,21%
46	1	x	Δ_f	I	VA	0,2%
47	5	z	r_b	I	VA	0,2%
48	2	z	r_b	II	VA	0,17%
49	4	z	$\sin(\varphi)$	II	VA	0,17%
50	4	z	Δ_f	II	VA	0,16%
51	2	z	Δ_f	I	VA	0,15%
52	6	–	φ	I	VA	0,14%
53	3	x	φ	I	VA	0,14%
54	3	x	m_f	II	VA	0,14%
55	3	x	φ	II	VA	0,13%
56	4	z	a	I	VA	0,12%
57	6	–	$\sin(\varphi)$	I	VA	0,1%
58	3	x	r_b	II	VA	0,09%
59	3	x	Δ_t	I	VA	0,09%
60	5	z	a	I	VA	0,08%
61	2	z	Δ_t	II	VA	0,08%
62	1	x	$\cos(\varphi)$	II	VA	0,07%
63	5	z	r_s	I	VA	0,06%
64	4	z	Δ_t	II	VA	0,05%
65	6	–	Δ_f	I	VA	0,05%
66	1	x	m_f	II	VA	0,05%
67	1	x	φ	I	VA	0,04%
68	1	x	r_b	I	VA	0,03%
69	3	x	$\sin(\varphi)$	I	VA	0,03%
70	3	x	m_f	I	VA	0,03%
71	3	x	a	I	VA	0,03%
72	6	–	φ	II	VA	0,03%
73	4	z	$\cos(\varphi)$	I	VA	0,03%
74	3	x	a	II	VA	0,03%
75	1	x	$\sin(\varphi)$	I	VA	0,02%
76	3	x	Δ_f	II	VA	0,01%

77	5	z	r_s	II	VA	0,01%

Tabelle A.1: Regressorliste für das erste lineare OTFS-Modell

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	4	z	r_b	I	HA	13,73%
2	6	–	Δ_t	I	HA	13,49%
3	1	x	m_f	I	HA	10,39%
4	6	–	$\sin(\varphi)$	I	HA	8,95%
5	1	x	$\overline{\mathbf{W}}$	II	HA	6,19%
6	6	–	φ	I	HA	4,1%
7	5	z	$\cos(\varphi)$	II	HA	3,14%
8	3	x	a	II	HA	2,83%
9	4	z	r_s	II	HA	2,09%
10	5	z	m_f	I	HA	2,05%
11	3	x	φ	II	HA	2,02%
12	1	x	r_s	II	HA	2%
13	2	z	$\overline{\mathbf{W}}$	II	HA	1,88%
14	6	–	a	I	HA	1,66%
15	2	z	$\cos(\varphi)$	I	HA	1,53%
16	6	–	m_f	II	HA	1,44%
17	3	x	$\sin(\varphi)$	II	HA	1,42%
18	1	x	a	II	HA	1,34%
19	1	x	a	I	HA	1,14%
20	4	z	Δ_f	II	HA	1,09%
21	3	x	m_f	I	HA	0,95%
22	5	z	Δ_t	II	HA	0,94%
23	4	z	m_f	I	HA	0,93%
24	3	x	Δ_f	I	HA	0,92%
25	3	x	$\overline{\mathbf{W}}$	I	HA	0,85%
26	5	z	$\overline{\mathbf{W}}$	II	HA	0,82%
27	3	x	Δ_t	II	HA	0,8%
28	3	x	$\cos(\varphi)$	I	HA	0,77%
29	6	–	r_b	I	HA	0,61%
30	6	–	$\overline{\mathbf{W}}$	II	HA	0,54%
31	5	z	r_b	II	HA	0,53%
32	3	x	r_s	II	HA	0,52%
33	5	z	a	I	HA	0,5%
34	6	–	$\cos(\varphi)$	I	HA	0,48%
35	5	z	Δ_t	I	HA	0,47%

36	4	z	$\overline{\mathbf{W}}$	II	HA	0,47%
37	5	z	r_s	I	HA	0,47%
38	1	x	$\sin(\varphi)$	I	HA	0,42%
39	2	z	m_f	II	HA	0,38%
40	1	x	r_b	I	HA	0,37%
41	4	z	$\sin(\varphi)$	I	HA	0,35%
42	2	z	m_f	I	HA	0,31%
43	6	–	φ	II	HA	0,3%
44	3	x	a	I	HA	0,28%
45	2	z	$\overline{\mathbf{W}}$	I	HA	0,21%
46	5	z	Δ_f	II	HA	0,19%
47	3	x	$\cos(\varphi)$	II	HA	0,18%
48	1	x	m_f	II	HA	0,18%
49	3	x	m_f	II	HA	0,16%
50	5	z	r_b	I	HA	0,13%
51	6	–	$\overline{\mathbf{W}}$	I	HA	0,13%
52	3	x	Δ_t	I	HA	0,13%
53	1	x	Δ_f	II	HA	0,13%
54	1	x	$\cos(\varphi)$	II	HA	0,13%
55	2	z	a	II	HA	0,12%
56	5	z	r_s	II	HA	0,12%
57	2	z	r_b	I	HA	0,12%
58	4	z	$\sin(\varphi)$	II	HA	0,12%
59	6	–	m_f	I	HA	0,11%
60	3	x	r_b	II	HA	0,11%
61	4	z	Δ_t	II	HA	0,11%
62	3	x	r_b	I	HA	0,1%
63	2	z	Δ_t	II	HA	0,1%
64	6	–	a	II	HA	0,09%
65	4	z	φ	I	HA	0,08%
66	2	z	$\cos(\varphi)$	II	HA	0,08%
67	4	z	Δ_f	I	HA	0,08%
68	6	–	r_b	II	HA	0,07%
69	2	z	$\sin(\varphi)$	I	HA	0,06%
70	1	x	$\overline{\mathbf{W}}$	I	HA	0,06%
71	2	z	Δ_t	I	HA	0,06%
72	2	z	a	I	HA	0,05%
73	4	z	$\cos(\varphi)$	I	HA	0,05%
74	5	z	Δ_f	I	HA	0,04%

75	2	z	φ	II	HA	0,04%
76	4	z	r_b	II	HA	0,03%
77	5	z	$\sin(\varphi)$	II	HA	0,03%
78	5	z	$\overline{\mathbf{W}}$	I	HA	0,02%
79	4	z	a	II	HA	0,02%
80	4	z	a	I	HA	0,02%
81	2	z	r_s	I	HA	0,02%
82	6	–	r_s	I	HA	0,01%
83	5	z	$\sin(\varphi)$	I	HA	0,01%
84	6	–	Δ_t	II	HA	0,01%
85	6	–	r_s	II	HA	0%

Tabelle A.2: Regressorliste für das zweite lineare OTFS-Modell

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	6	–	a	I	VA	15,27%
2	1	x	$\sin(\varphi)$	I	VA	13,52%
3	1	x	φ	I	VA	11,14%
4	1	x	Δ_t	I	VA	7,05%
5	5	z	r_b	I	VA	3,94%
6	2	z	r_b	I	VA	3,25%
7	2	z	Δ_t	I	VA	2,87%
8	4	z	$\sin(\varphi)$	II	VA	2,8%
9	4	z	φ	II	VA	2,64%
10	3	x	r_b	I	VA	2,42%
11	2	z	φ	I	VA	2,11%
12	2	z	r_s	I	VA	2,04%
13	4	z	r_s	I	VA	2,01%
14	6	–	r_s	I	VA	1,5%
15	4	z	Δ_f	II	VA	1,43%
16	2	z	$\sin(\varphi)$	I	VA	1,41%
17	4	z	r_s	II	VA	1,32%
18	1	x	Δ_t	II	VA	1,32%
19	2	z	a	II	VA	1,28%
20	4	z	Δ_t	II	VA	1,22%
21	2	z	a	I	VA	1,11%
22	6	–	$\sin(\varphi)$	I	VA	1,03%
23	4	z	r_b	I	VA	0,88%
24	1	x	$\cos(\varphi)$	I	VA	0,84%
25	3	x	$\sin(\varphi)$	I	VA	0,75%

26	1	x	$\cos(\varphi)$	II	VA	0,74%
27	3	x	Δ_f	II	VA	0,7%
28	2	z	$\cos(\varphi)$	I	VA	0,66%
29	2	z	$\sin(\varphi)$	II	VA	0,61%
30	2	z	m_f	I	VA	0,59%
31	6	–	r_b	I	VA	0,58%
32	3	x	a	II	VA	0,57%
33	1	x	a	I	VA	0,57%
34	6	–	φ	I	VA	0,5%
35	2	z	r_s	II	VA	0,49%
36	6	–	Δ_t	II	VA	0,48%
37	5	z	$\cos(\varphi)$	II	VA	0,44%
38	4	z	Δ_t	I	VA	0,42%
39	4	z	$\sin(\varphi)$	I	VA	0,39%
40	1	x	r_b	II	VA	0,39%
41	2	z	Δ_t	II	VA	0,39%
42	2	z	r_b	II	VA	0,37%
43	4	z	φ	I	VA	0,33%
44	3	x	φ	I	VA	0,33%
45	5	z	a	I	VA	0,33%
46	5	z	m_f	I	VA	0,32%
47	1	x	r_s	I	VA	0,31%
48	5	z	r_s	I	VA	0,29%
49	2	z	Δ_f	I	VA	0,29%
50	3	x	r_s	II	VA	0,23%
51	2	z	m_f	II	VA	0,21%
52	5	z	Δ_f	I	VA	0,21%
53	6	–	$\cos(\varphi)$	II	VA	0,21%
54	6	–	a	II	VA	0,2%
55	4	z	Δ_f	I	VA	0,2%
56	6	–	m_f	I	VA	0,18%
57	2	z	φ	II	VA	0,18%
58	4	z	a	II	VA	0,18%
59	1	x	Δ_f	I	VA	0,18%
60	6	–	m_f	II	VA	0,15%
61	6	–	φ	II	VA	0,14%
62	3	x	m_f	I	VA	0,14%
63	1	x	m_f	II	VA	0,12%
64	3	x	Δ_f	I	VA	0,12%

65	3	x	r_b	II	VA	0,12%
66	3	x	$\cos(\varphi)$	I	VA	0,11%
67	3	x	φ	II	VA	0,11%
68	4	z	m_f	II	VA	0,1%
69	6	–	Δ_f	I	VA	0,09%
70	3	x	$\overline{\mathbf{W}}$	I	VA	0,09%
71	6	–	r_b	II	VA	0,09%
72	5	z	$\sin(\varphi)$	I	VA	0,05%
73	5	z	r_b	II	VA	0,05%
74	1	x	r_s	II	VA	0,05%
75	6	–	$\cos(\varphi)$	I	VA	0,05%
76	5	z	Δ_t	II	VA	0,04%
77	5	z	Δ_t	I	VA	0,03%
78	4	z	m_f	I	VA	0,03%
79	6	–	r_s	II	VA	0,03%
80	1	x	m_f	I	VA	0,02%
81	5	z	$\cos(\varphi)$	I	VA	0,02%
82	4	z	$\cos(\varphi)$	II	VA	0,01%
83	4	z	$\overline{\mathbf{W}}$	II	VA	0,01%
84	5	z	φ	I	VA	0,01%
85	6	–	Δ_t	I	VA	0,01%
86	6	–	Δ_f	II	VA	0,01%
87	5	z	φ	II	VA	0%

Tabelle A.3: Regressorliste für das dritte lineare OTFS-Modell

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	1	x	a	I	HA	25,56%
2	1	x	Δ_t	I	HA	21,56%
3	6	–	a	I	HA	19,25%
4	1	x	$\cos(\varphi)$	I	HA	7,42%
5	6	–	a	II	HA	6,61%
6	3	x	Δ_f	II	HA	3,5%
7	6	–	Δ_f	I	HA	3,39%
8	4	z	m_f	I	HA	2,95%
9	1	x	r_s	I	HA	1,81%
10	2	z	r_s	II	HA	1,79%
11	4	z	φ	II	HA	1,5%
12	1	x	$\cos(\varphi)$	II	HA	0,87%
13	4	z	$\sin(\varphi)$	II	HA	0,69%

14	6	–	$\sin(\varphi)$	II	HA	0,47%
15	2	z	r_b	I	HA	0,44%
16	6	–	φ	II	HA	0,36%
17	3	x	m_f	II	HA	0,31%
18	6	–	Δ_f	II	HA	0,29%
19	3	x	Δ_f	I	HA	0,29%
20	3	x	r_s	II	HA	0,23%
21	2	z	Δ_f	II	HA	0,21%
22	1	x	φ	II	HA	0,09%
23	4	z	m_f	II	HA	0,08%
24	2	z	$\cos(\varphi)$	I	HA	0,07%
25	6	–	φ	I	HA	0,05%
26	3	x	φ	II	HA	0,03%
27	6	–	r_s	I	HA	0,03%
28	4	z	Δ_t	II	HA	0,02%
29	1	x	$\sin(\varphi)$	II	HA	0,02%
30	6	–	$\cos(\varphi)$	I	HA	0,02%
31	5	z	m_f	II	HA	0,01%
32	1	x	r_s	II	HA	0,01%
33	2	z	Δ_f	I	HA	0,01%
34	4	z	Δ_f	I	HA	0,01%
35	3	x	r_b	I	HA	0,01%
36	5	z	$\cos(\varphi)$	II	HA	0,01%
37	2	z	Δ_t	II	HA	0,01%
38	6	–	Δ_t	I	HA	0,01%
39	6	–	Δ_t	II	HA	0%
40	4	z	a	I	HA	0%
41	1	x	Δ_f	I	HA	0%
42	5	z	Δ_t	I	HA	0%
43	5	z	r_b	I	HA	0%
44	5	z	φ	II	HA	0%
45	6	–	r_b	II	HA	0%
46	6	–	r_s	II	HA	0%
47	3	x	$\sin(\varphi)$	I	HA	0%

Tabelle A.4: Regressorliste für das vierte lineare OTFS-Modell

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	3	x	r_b	I	VA	14,33%
2	1	x	Δ_f	I	VA	13,15%

3	6	–	a	I	VA	10,71%
4	4	z	Δ_t	II	VA	7,89%
5	3	x	Δ_f	I	VA	7,74%
6	1	x	r_s	I	VA	7,72%
7	2	z	Δ_t	II	VA	5,07%
8	4	z	a	I	VA	2,65%
9	4	z	Δ_f	I	VA	2,31%
10	3	x	Δ_t	I	VA	2,05%
11	1	x	a	I	VA	1,94%
12	4	z	r_s	I	VA	1,7%
13	1	x	$\sin(\varphi)$	I	VA	1,19%
14	1	x	φ	I	VA	1,18%
15	5	z	a	I	VA	1,09%
16	1	x	r_b	I	VA	1,02%
17	6	–	Δ_f	II	VA	1%
18	4	z	Δ_t	I	VA	0,95%
19	6	–	Δ_f	I	VA	0,93%
20	5	z	r_b	I	VA	0,88%
21	6	–	$\sin(\varphi)$	II	VA	0,85%
22	4	z	m_f	I	VA	0,76%
23	3	x	φ	I	VA	0,73%
24	6	–	φ	II	VA	0,66%
25	2	z	r_b	I	VA	0,65%
26	6	–	a	II	VA	0,63%
27	6	–	r_b	I	VA	0,63%
28	2	z	a	II	VA	0,61%
29	6	–	Δ_t	II	VA	0,6%
30	5	z	m_f	I	VA	0,59%
31	2	z	m_f	II	VA	0,48%
32	3	x	r_s	I	VA	0,48%
33	3	x	$\overline{\mathbf{W}}$	II	VA	0,44%
34	1	x	$\cos(\varphi)$	I	VA	0,43%
35	4	z	$\overline{\mathbf{W}}$	II	VA	0,39%
36	4	z	$\cos(\varphi)$	II	VA	0,37%
37	4	z	$\cos(\varphi)$	I	VA	0,35%
38	4	z	$\sin(\varphi)$	I	VA	0,33%
39	3	x	r_b	II	VA	0,31%
40	1	x	r_b	II	VA	0,28%
41	6	–	r_s	I	VA	0,27%

42	3	x	a	II	VA	0,25%
43	4	z	r_b	II	VA	0,23%
44	5	z	Δ_t	I	VA	0,22%
45	5	z	$\sin(\varphi)$	II	VA	0,21%
46	1	x	m_f	II	VA	0,21%
47	3	x	a	I	VA	0,18%
48	5	z	a	II	VA	0,17%
49	5	z	Δ_f	I	VA	0,17%
50	6	–	m_f	II	VA	0,16%
51	2	z	a	I	VA	0,16%
52	4	z	$\overline{\mathbf{W}}$	I	VA	0,15%
53	4	z	r_b	I	VA	0,15%
54	3	x	$\cos(\varphi)$	I	VA	0,14%
55	5	z	φ	I	VA	0,12%
56	5	z	$\sin(\varphi)$	I	VA	0,12%
57	1	x	r_s	II	VA	0,11%
58	6	–	r_b	II	VA	0,1%
59	1	x	$\cos(\varphi)$	II	VA	0,1%
60	6	–	φ	I	VA	0,08%
61	1	x	Δ_f	II	VA	0,08%
62	3	x	m_f	II	VA	0,07%
63	6	–	$\sin(\varphi)$	I	VA	0,07%
64	5	z	$\overline{\mathbf{W}}$	II	VA	0,07%
65	2	z	$\cos(\varphi)$	II	VA	0,07%
66	3	x	Δ_f	II	VA	0,07%
67	4	z	$\sin(\varphi)$	II	VA	0,06%
68	2	z	Δ_f	I	VA	0,04%
69	1	x	m_f	I	VA	0,04%
70	3	x	φ	II	VA	0,03%
71	5	z	Δ_t	II	VA	0,03%
72	4	z	m_f	II	VA	0,01%
73	6	–	$\cos(\varphi)$	II	VA	0,01%
74	6	–	r_s	II	VA	0,01%
75	3	x	r_s	II	VA	0%
76	5	z	r_b	II	VA	0%

Tabelle A.5: Regressorliste für das fünfte lineare OTFS-Modell

Nr.	Kanal	Einleitung	Merkmal	Approximation	Achse	rel. Anteil
1	6	–	a	I	HA	28,3%

2	1	x	$\cos(\varphi)$	I	HA	20,61%
3	1	x	Δ_f	I	HA	13,74%
4	3	x	r_b	I	HA	7,46%
5	3	x	Δ_f	I	HA	6,01%
6	3	x	r_s	I	HA	3,93%
7	1	x	r_s	II	HA	3,8%
8	1	x	a	I	HA	2,98%
9	3	x	a	II	HA	2,05%
10	4	z	r_s	II	HA	2,04%
11	5	z	m_f	II	HA	1,57%
12	1	x	Δ_f	II	HA	1,14%
13	5	z	r_s	II	HA	0,96%
14	3	x	Δ_f	II	HA	0,95%
15	4	z	$\overline{\mathbf{W}}$	II	HA	0,48%
16	1	x	r_s	I	HA	0,47%
17	2	z	a	I	HA	0,33%
18	4	z	$\overline{\mathbf{W}}$	I	HA	0,33%
19	3	x	Δ_t	I	HA	0,31%
20	4	z	a	I	HA	0,29%
21	3	x	r_b	II	HA	0,27%
22	3	x	$\cos(\varphi)$	II	HA	0,27%
23	1	x	Δ_t	II	HA	0,25%
24	1	x	m_f	II	HA	0,2%
25	6	–	Δ_f	I	HA	0,19%
26	2	z	Δ_f	II	HA	0,15%
27	6	–	a	II	HA	0,12%
28	5	z	Δ_t	I	HA	0,12%
29	4	z	r_s	I	HA	0,1%
30	4	z	$\cos(\varphi)$	I	HA	0,1%
31	2	z	r_s	I	HA	0,07%
32	4	z	$\sin(\varphi)$	I	HA	0,07%
33	3	x	a	I	HA	0,05%
34	5	z	m_f	I	HA	0,05%
35	1	x	Δ_t	I	HA	0,04%
36	1	x	$\sin(\varphi)$	II	HA	0,04%
37	1	x	$\sin(\varphi)$	I	HA	0,03%
38	6	–	$\cos(\varphi)$	I	HA	0,03%
39	4	z	Δ_t	I	HA	0,02%
40	6	–	Δ_t	I	HA	0,02%

41	1	x	φ	II	HA	0,02%
42	3	x	φ	I	HA	0,01%
43	2	z	$\cos(\varphi)$	II	HA	0,01%
44	4	z	a	II	HA	0,01%
45	2	z	Δ_f	I	HA	0,01%
46	3	x	$\sin(\varphi)$	I	HA	0,01%
47	5	z	r_b	II	HA	0%
48	5	z	φ	I	HA	0%
49	5	z	$\sin(\varphi)$	I	HA	0%
50	5	z	Δ_f	II	HA	0%

Tabelle A.6: Regressorliste für das sechste lineare OTFS-Modell

Literaturverzeichnis

[Ada07] Adamski, D.; Just, W. und Dragon, L.: Subjektive Bewertung des Ride Komforts von Digitalen Prototypen mit Hilfe eines Ride Simulators, in: *Erprobung und Simulation in der Fahrzeugentwicklung - Mess- und Versuchstechnik -*, Bd. 1990 von *VDI-Berichte*, VDI, VDI Verlag GmbH, Düsseldorf, S. 289–302

[Aig82] Aigner, J.: Zur zuverlässigen Beurteilung von Fahrzeugen. *ATZ Automobiltechnische Zeitschrift* (1982), Bd. 84(9): S. 447–450

[Ake78] Akers, S. B.: Binary Decision Diagrams. *IEEE Trans. Comput.* (1978), Bd. 27(6): S. 509–516

[Alb05] Albrecht, M.: *Modellierung der Komfortbeurteilung aus Kundensicht am Beispiel des automatisierten Anfahrens*, Forschungsberichte, Institut für Produktentwicklung Universität Karlsruhe (TH) (2005)

[All77] Allen, J.B.: Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on* (1977), Bd. 25(3): S. 235–238

[Ama97] Amari, S.-I.; Murata, N.; Muller, K.-R.; Finke, M. und Yang, H.H.: Asymptotic statistical theory of overtraining and cross-validation. *Neural Networks, IEEE Transactions on* (1997), Bd. 8(5): S. 985–996

[Ari95] Arino, Miguel A.; No, Miguel A. Ari; Morettin, Pedro und Vidakovic, Brani: Wavelet Scalograms And Their Applications In Economic Time Series (1995)

[Ari06] Arivazhagan, S.; Ganesan, L. und Priyal, S. Padam: Texture classification using Gabor wavelets based rotation invariant features. *Pattern Recogn. Lett.* (2006), Bd. 27(16): S. 1976–1982

[Arl08] Arlot, S.: V-fold cross-validation improved: V-fold penalization. *ArXiv e-prints* (2008)

[Bac10] Backhaus, Klaus; Erichson, Bernd; Plinke, Wulff und Weiber, Rolf: *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung (Springer-Lehrbuch) (German Edition)*, Springer (2010)

[Bäc96] Bäck, Thomas: *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, Oxford, UK (1996)

[Ber05] Bernardino, Alexandre und Santos-Victor, Jose: A Real-Time Gabor Primal Sketch for Visual Attention, in: *Pattern Recognition and Image Analysis*, Bd.

3522 von *Lecture Notes in Computer Science*, Springer Berlin Heidelberg (2005), S. 335–342

[Bey08] Beygelzimer, Alina; Langford, John und Zadrozny, Bianca: Machine Learning Techniques - Reductions Between Prediction Quality Metrics, in: Zhen Liu und CathyH. Xia (Herausgeber) *Performance Modeling and Engineering*, Springer US (2008), S. 3–28

[Bis96] Bishop, Christopher M.: *Neural Networks for Pattern Recognition*, Oxford University Press, USA (1996)

[Bog95] Boggs, Paul T. und Tolle, Jon W.: Sequential Quadratic Programming. *Acta Numerica* (1995), Bd. 4: S. 1–51

[Bro70] Broyden, C. G.: The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics* (1970), Bd. 6(1): S. 76–90

[Bub00] Bubb, Heiner: *Ergonomie und Verkehrssicherheit - GfA-Konferenzbeiträge der Herbstkonferenz 2000, 12. - 13. Oktober 2000 an der Technischen Universität München*, Herbert Utz Verlag, München (2000)

[Bur91] Burkitt, Anthony N.: Optimization of the Architecture of Feed-forward Neural Networks with Hidden Layers by Unit Elimination. *Complex Systems* (1991), Bd. 5

[Car06] Caruana, Rich und Niculescu-Mizil, Alexandru: An Empirical Comparison of Supervised Learning Algorithms, in: *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, ACM, New York, NY, USA, S. 161–168

[Cha07] Chang, Fi-John; Chiang, Yen-Ming und Chang, Li-Chiu: Multi-step-ahead neural networks for flood forecasting. *Hydrological Sciences Journal* (2007), Bd. 52(1): S. 114–130

[Col91] Colorni, Alberto; Dorigo, Marco und Maniezzo, Vittorio: Distributed Optimization by Ant Colonies, in: *European Conference on Artificial Life*, S. 134–142

[Con00] Conradt, Jörg; Tevatia, Gaurav; Vijayakumar, Sethu und Schaal, Stefan: On-line Learning for Humanoid Robot Systems, in: *in Proceedings of the Seventeenth International Conference on Machine Learning*, S. 191–198

[Cuc93] Cucuz, S.: *Schwingempfindung von Pkw-Insassen: Auswirkung von stochastischen Unebenheiten und Einzelhindernissen der realen Fahrbahn*, TU-Braunschweig (1993)

[Cyb89] Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)* (1989), Bd. 2(4): S. 303–314

[Dah95] Dahlke, S. und Maass, P.: The Affine uncertainty principle in one and two dimensions. *Computers & Mathematics with Applications* (1995), Bd. 30(3–6): S. 293–305

[Dau92] Daubechies, Ingrid: *Ten Lectures on Wavelets (CBMS-NSF Regional Conference Series in Applied Mathema*, SIAM: Society for Industrial and Applied Mathematics (1992)

[Day01] Dayhoff, JE und DeLeo, JM: Artificial neural networks: opening the black box (2001)

[Dee81] Deely, J.J. und Lindley, D.V.: Bayes empirical Bayes. *JASA. Journal of the American Statistical Association* (1981), Bd. 76: S. 833–841

[Dem82] Dembo, Ron S.; Eisenstat, Stanley C. und Steihaug, Trond: Inexact Newton Methods. *SIAM Journal on Numerical Analysis* (1982), Bd. 19(2): S. pp. 400–408

[Dem97] Demmel, James W.: *Applied Numerical Linear Algebra*, SIAM, 1 Aufl. (1997)

[DF97] Dan Foresee, F. und Hagan, M.T.: Gauss-Newton approximation to Bayesian learning, in: *Neural Networks,1997., International Conference on*, Bd. 3, S. 1930–1935

[Dud73] Duda, Richard O. und Hart, Peter E.: *Pattern Classification and Scene Analysis*, Wiley, 1 Aufl. (1973)

[Fis10] Fischer, N.; Meywerk, M.; Karrer, H. und Reinalter, W.: Komfortbeurteilung von PKW bei Einzelhindernisüberfahrten durch Wavelet-Transformierte, in: *4. VDI-Tagung Humanschwingungen Auswirkungen auf Gesundheit - Leistung - Komfort*, Bd. 2097 von *VDI-Berichte*, VDI, VDI Verlag GmbH, Düsseldorf, S. 261–272

[Fle70] Fletcher, R.: A new approach to variable metric algorithms. *The Computer Journal* (1970), Bd. 13(3): S. 317–322

[Fuh12] Fuhr, B.: *Methode zur Klassifizierung von Unfallszenarien mithilfe von FE-Gesamtfahrzeugsimulationen, Wavelettransformierten und künstlichen neuronalen Netzen*, disserta Verlag (2012)

[FvR13] Fischer-von Rönn, Niklas und Meywerk, Martin: Predicting the ride comfort of passenger cars for single-obstacle crossings through an innovative feature extraction method consisting of non-linear geometric approximations of wavelet-transformed acceleration data and airborne sound. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* (2013)

[Gab46] Gabor, Dennis: Theory of Communication. *J. Inst. Elect. Eng.* (1946), Bd. 93: S. 429–457

[Gei99] Geiger, Carl und Kanzow, Christian: *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben (Springer-Lehrbuch) (German Edition)*, Springer, 1999 Aufl. (1999)

[Gei02] Geiger, Carl und Kanzow, Christian: *Theorie und Numerik restringierter Optimierungsaufgaben (Springer-Lehrbuch Masterclass) (German Edition)*, Springer, 2002 Aufl. (2002)

[Gol70] Goldfarb, Donald: A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation* (1970), Bd. 24(109): S. pp. 23–26

[Gor10] Gorgonio, F.L. und Costa, J.A.F: PartSOM: A Framework for Distributed Data Clustering Using SOM and K-Means, in: George K Matsopoulos (Herausgeber) *Self-Organizing Maps*, Kap. 4, InTech (2010)

[Gou84] Goupillaud, P.; Grossmann, A. und Morlet, J.: Cycle-octave and related transforms in seismic signal analysis. *Geoexploration* (1984), Bd. 23(1): S. 85–102

[Gri96] Griffin, Michael J: *Handbook of Human Vibration*, Academic Press (1996)

[Gue11] Guerra, L.; Robles, V.; Bielza, C.; Larranaga, P. und Yuste, R.: Comparison between Supervised and Unsupervised Classification of Neuronal Cell Types: A Case Study. *Developmental Neurobiology* (2011), Bd. 71(1): S. 71–82

[Gut07] Gutin, G. und Punnen, A.P.: *The Traveling Salesman Problem and Its Variations*, Springer, 2002 Aufl. (2007)

[Guy03] Guyon, Isabelle und Elisseeff, André: An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* (2003), Bd. 3: S. 1157–1182

[Guy06] Guyon, I.; Gunn, S.; Nikravesh, M. und Zadeh, L.A.: *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*, Springer-Verlag New York, Inc. Secaucus, NJ, USA (2006)

[GW09] Grzegozek W., Szczygiel J., Krol S.: An attempt of an employment of a continuous wavelet transform for evaluation of temporary comfort disturbances. *Journal of KONES Powertrain and transport* (2009), Bd. 16(1): S. 157–165

[Haw04] Hawkins, D. M.: The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences* (2004), Bd. 44(1): S. 1–12

[Hay98] Haykin, Simon: *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 2 Aufl. (1998)

[Haz94] Hazelaar, M.: *Fahrwerkschwingungen und Komfortbeurteilung bei kurzwelliger Anregung (Achsrauhigkeit)*, Fortschrittberichte VDI / 12: Verkehrstechnik, Fahrzeugtechnik, VDI-Verlag (1994)

[hC09] hyuk Cha, Sung: A Genetic Algorithm for Constructing Compact Binary Decision Trees (2009)

[Hei02] Heissing, Bernd: *Subjektive Beurteilung des Fahrverhaltens (German Edition)*, Vogel Business Media (2002)

[Hop82] Hopfield, J J: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* (1982), Bd. 79(8): S. 2554–2558

[Hor19] Horner, William George: *A new method of solving numerical equations of all orders, by continuous approximation* (1819), S. 308–335

[Hor90] Horn, Roger A. und Johnson, Charles R.: *Matrix Analysis*, Cambridge University Press, reprint Aufl. (1990)

[Hou58] Householder, Alston S.: Unitary Triangularization of a Nonsymmetric Matrix. *J. ACM* (1958), Bd. 5(4): S. 339–342

[Hu08] Hu, X.; Zhang, J. und Li, Y.: Orthogonal methods based ant colony search for solving continuous optimization problems. *Journal of Computer Science and Technology* (2008), Bd. 23(1): S. 2–18

[Hua00] Huang, Guang-Bin; Chen, Yan-Qiu und Babri, H.A.: Classification ability of single hidden layer feedforward neural networks. *Neural Networks, IEEE Transactions on* (2000), Bd. 11(3): S. 799–801

[Huy07] Huynh, Hieu Trung; Kim, Jung-Ja und Won, Yonggwan: DNA Microarray Classification with Compact Single Hidden-Layer FeedForward Neural Networks, in: *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, S. 193–198

[ISO95] ISO/TC: ISO 8608:1995, Mechanical vibration - Road surface profiles - Reporting of measured data (1995)

[ISO97] ISO: ISO 2631-1. Mechanical vibration and shock-evaluation of human exposure to whole body vibration-part 1: general requirements, Techn. Ber., International Organization for Standardization (1997)

[Ist02] Istook, Ernest und Martinez, Tony: Improved Backpropagation Learning in Neural Networks with windowed Momentum. *International Journal of Neural Systems* (2002), Bd. 12(03n04): S. 303–318

[JÖ5] Jönsson, P. und Johansson, Ö.: Prediction of vehicle discomfort from transient vibrations. *Journal of Sound and Vibration* (2005), Bd. 282: S. 1043–1064

[Jan04] Janczak, Andrzej: *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach (Lecture Notes in Control and Information Sciences)*, Springer, 2005 Aufl. (2004)

[Jen06] Jensen, J.L.W.V.: Sur les fonctions convexes et les inÃ©galitÃ©s entre les valeurs moyennes. *Acta Mathematica* (1906), Bd. 30(1): S. 175–193

[Jür96] Jürgensohn, T.; Müller, W. und Scheffer, T.: *Verbesserte Methoden zur Objektivierung von subjektiven Bewertungen des Fahrverhaltens*, Forschungsbericht, ZMMS (1996)

[Kai80] Kaiser, J. und Schafer, R.W.: On the use of the I0-sinh window for spectrum analysis. *Acoustics, Speech and Signal Processing, IEEE Transactions on* (1980), Bd. 28(1): S. 105–107

[Kar39] Karush, William: *Minima of Functions of Several Variables with Inequalities as Side Conditions*, Diplomarbeit, Department of Mathematics, University of Chicago, Chicago, IL, USA (1939)

[Ken95] Kennedy, J. und Eberhart, R.: Particle swarm optimization, in: *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Bd. 4, S. 1942–1948 vol.4

[Koh82] Kohonen, Teuvo: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* (1982), Bd. 43(1): S. 59–69

[Koh90] Kohonen, T.: The self-organizing map. *Proceedings of the IEEE* (1990), Bd. 78(9): S. 1464–1480

[Koh97] Kohavi, Ron und John, George H.: Wrappers for Feature Subset Selection. *Artificial Intelligence* (1997), Bd. 97(1): S. 273–324

[Koh00] Kohonen, Teuvo: *Self-Organizing Maps*, Springer, 3rd Aufl. (2000)

[Kos02] Kosfelder, M. und Periard, F.: *Ein modalanalytischer Ansatz zur Bestimmung der globalen Bewegungsformen von Kraftfahrzeugen (Teil II: Verifikation und Anwendungen)*, DAGA (2002)

[Kot07] Kotsiantis, S. B.: Supervised Machine Learning: A Review of Classification Techniques, in: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, IOS Press, Amsterdam, The Netherlands, The Netherlands, S. 3–24

[Kuh51] Kuhn, H. W. und Tucker, A. W.: Nonlinear Programming (1951)

[Lal06] Lal, ThomasNavin; Chapelle, Olivier; Weston, Jason und Elisseeff, Andre: Embedded Methods, in: Isabelle Guyon; Masoud Nikravesh; Steve Gunn und LotfiA. Zadeh (Herausgeber) *Feature Extraction*, Bd. 207 von *Studies in Fuzziness and Soft Computing*, Springer Berlin Heidelberg (2006), S. 137–165

[Lan92] Langley, Pat; Iba, Wayne und Thompson, Kevin: An analysis of Bayesian classifiers, in: *Proceedings of the tenth national conference on artificial intelligence*, MIT Press, S. 223–228

[Laz89] Lazzaro, J.; Ryckebusch, S.; Mahowald, M. A. und Mead, C. A.: Winner-take-all networks of O(n) complexity, in: *Advances in Neural Information Processing Systems*, Morgan Kaufmann, S. 703–711

[Lee00] Lee, S.-K. und White, P.R.: Application of wavelet analysis to the impact harshness of a vehicle. *Proceedings of the Institute of Mechanical Engineers Part C, Journal of Mechanical Engineering Science* (2000), Bd. 214(11): S. 1331–1338

[Lee01] Lee, Sang-Kwon und Son, Choong-Yul: Ride comfort analysis of a vehicle based on continuous wavelet transform. *KSME International Journal* (2001), Bd. 15(5): S. 535–543

[Lev44] Levenberg, Kenneth: A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics* (1944), Bd. II(2): S. 164–168

[Li03] Li, J. und Duckett, T.: Robot behaviour learning with a dynamically adaptive rbf network: experiments in offline and online learning, in: *Second International Conference on Computational Intelligence, Robotics and Autonomous Systems*

[Liu98] Liu, Huan und Motoda, Hiroshi: *Feature Selection for Knowledge Discovery and Data Mining*, Springer, 1998 Aufl. (1998)

[Liu07] Liu, Huan und Motoda, Hiroshi: *Computational Methods of Feature Selection*, Chapman and Hall/CRC (2007)

[Loh] Loh, Wei-Yin: *Classification and Regression Tree Methods*

[Lou97] Louis, Alfred K.; Maass, Peter und Rieder, Andreas: *Wavelets - theory and applications*, Pure and applied mathematics, Wiley (1997)

[Mac91] MacKay, David J.C.: Bayesian Interpolation. *Neural Computation* (1991), Bd. 4: S. 415–447

[Man92] Manjunath, B.S.: Gabor wavelet transform and application to problems in early vision, in: *Signals, Systems and Computers, 1992. 1992 Conference Record of The Twenty-Sixth Asilomar Conference on*, Bd. 2, S. 796–800

[Man02] Mannor, Shie; Meir, Ron und Zhang, Tong: The Consistency of Greedy Algorithms for Classification, in: *In Proceedings of the 15th Annual Conference on Computational Learning Theory*, Springer, S. 319–333

[Mar63] Marquardt, Donald W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics* (1963), Bd. 11(2): S. pp. 431–441

[Mey07] Meywerk, Martin: *CAE-Methoden in der Fahrzeugtechnik (German Edition)*, Springer, 2007 Aufl. (2007)

[Mey10] Meywerk, M.; Fischer, N.; Karrer, H.; Reinalter, W. und Wimmer, P.: Vorhersage von Komfortnoten fur PKW bei Einzelhindernisuberfahrten durch Wavelet-Transformierte und neuronale Netze, in: *15. Kongress SIMVEC Berechnung und Simulation im Fahrzeugbau 2010 : Baden-Baden, 16. und 17. November 2010*, Bd. 2107 von *VDI-Berichte*, VDI-Verlag, Baden-Baden, S. 653–664

[Mis09] Mishra, Sudib K. und Mishra, Samir R.: Wavelet based identification of dominant scales for self-affine road roughness in context of riding comfort in vehicles. *Geotechnical and Geological Engineering* (2009), Bd. 27: S. 473–484

[Mit04] Mitschke, Manfred und Wallentowitz, Henning: *Dynamik der Kraftfahrzeuge (VDI-Buch) (German Edition)*, Springer, 4. neu bearb. aufl. 2004 Aufl. (2004)

[Mon12] Montgomery, Douglas C.: *Design and Analysis of Experiments*, Wiley, 8 Aufl. (2012)

[Naj] Najmi, Amir-Homayoon und Sadowsky, John: The Continuous Wavelet Transform and Variable Resolution Time-Frequency Analysis

[Ngu90] Nguyen, D. und Widrow, B.: Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in: *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, S. 21–26 vol.3

[Nix08] Nixon, M.S. und Aguado, A.S.: *Feature Extraction & Image Processing*, Academic Press (2008)

[Ols08] Olson, David L. und Delen, Dursun: *Advanced Data Mining Techniques*, Springer, 2008 Aufl. (2008)

[Par80] Parlett, Beresford N.: *Symmetric Eigenvalue Problem*, Prentice Hall, 1st edition Aufl. (1980)

[Par91] Park, J. und Sandberg, I. W.: Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation* (1991), Bd. 3(2): S. 246–257

[Par02] Parsopoulos, K.E. und Vrahatis, M.N.: Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing* (2002), Bd. 1(2-3): S. 235–306

[Par13] Parameswari, S.Uma und Jayanthi, B.: Survey on Unsupervised Learning in Datamining. *OSR Journal of Computer Engineering* (2013), Bd. 14: S. 46–53

[Pla98] Plagianakos, V. P.; Sotiropoulos, D. G. und Vrahatis, M. N.: An Improved Backpropagation Method with Adaptive Learning Rate, in: *Procceeding of the 2nd Intern. Confer. on: Circuits, Systems and Computers*

[Plu04] Plumb, A. W.; Chan, A. D C und Goge, A. R.: Continuous classifier training for myoelectrically controlled prostheses, in: *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, Bd. 1, S. 474–477

[Pra09] Pracny, Vladislav: *Neural network-based shock absorber model with a thermodynamical coupling*, Fahrzeugtechnik, Shaker Verlag (2009)

[Rad10] Radhakrishnan, C. und Jenkins, W.K.: Modified Discrete Fourier Transforms for fast convolution and adaptive filtering, in: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, S. 1611–1614

[Rer86] Rericha, I.: *Methoden zur objektiven Bewertung des Fahrkomforts*, Automobil-Industrie 2/86, Vogel-Verlag Würzburg (1986)

[Rie92] Riedmiller, Martin und Braun, Heinrich: RPROP - A Fast Adaptive Learning Algorithm, Techn. Ber., Proc. of ISCIS VII), Universitat (1992)

[Rie00] Riedel, A. und Becker, K.: *Subjektive und objektive Beurteilung des Handling von PKW*, Expert-Verlag (2000)

[Sar95] Sarle, Warren S.: Stopped Training and Other Remedies for Overfitting, in: *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, S. 352–360

[Sat13] Sathya, R. und Abraham, A.: Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence* (2013): S. 34–38

[Sch00] Schoeggl, P. und Ramschak, E.: Vehicle Driveability Assessment using Neural Networks for Development, Calibration and Quality Tests. *SAE Technical Paper* (2000), (2000-01-0702)

[Sch01] Schölkopf, Bernhard und Smola, Alexander J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*, The MIT Press, 1st Aufl. (2001)

[Sch05] Schabauer, H.; Schikuta, E. und Weishaupl, T.: Solving Very Large Traveling Salesman Problems by SOM Parallelization on Cluster Architectures, in: *Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on*, S. 954–958

[Sch11] Schwarz, Hans-Rudolf und Köckler, Norbert: *Numerische Mathematik*, Vieweg+Teubner Verlag, 8., akt. aufl. 2011 Aufl. (2011)

[Sei00] Seiffert, Udo; Michaelis, Bernd und Words, Key: On The Gradient Descent In Backpropagation And Its Substitution By A Genetic Algorithm (2000)

[Sha70] Shanno, D. F.: Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation* (1970), Bd. 24(111): S. pp. 647–656

[Sim13] Simon, Dan: *Evolutionary Optimization Algorithms*, Wiley, 1 Aufl. (2013)

[Sin00] Sinha, N.K.; Gupta, M.M. und Rao, D. H.: Dynamic neural networks: an overview, in: *Industrial Technology 2000. Proceedings of IEEE International Conference on*, Bd. 1, S. 491–496 vol.2

[SM07] Sanchez-Marono, Noelia; Alonso-Betanzos, Amparo und Tombilla-Sanroman, Maria: Filter Methods for Feature Selection - A Comparative Study, in: Hujun Yin; Peter Tino; Emilio Corchado; Will Byrne und Xin Yao (Herausgeber) *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, Bd. 4881 von *Lecture Notes in Computer Science*, Springer Berlin Heidelberg (2007), S. 178–187

[Smy97] Smyth, Gordon K.: Numerical Integration, in: *In Encyclopedia of Biostatistics*, S. 3088–3095

[Sta] Stammen, K. und Meywerk, M.: Künstliche neuronale Netze als Hilfsmittel zur Abschätzung des Komfortempfindens. *VDI-Berichte Nr. 2014*: S. 323–339, Düsseldorf, VDI–Verlag 2007

[Sta09] Stammen, K.: *Bewertung des Schwingungskomforts in PKW mit Hilfe von Regressionsmodellen und künstlichen neuronalen Netzen*, Fortschritt-Berichte VDI: Reihe 12, Verkehrstechnik/Fahrzeugtechnik, VDI-Verlag (2009)

[Sut13] Sutskever, Ilya; Martens, James; Dahl, George und Hinton, Geoffrey: On the importance of initialization and momentum in deep learning, in: Sanjoy Dasgupta und David Mcallester (Herausgeber) *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Bd. 28, JMLR Workshop and Conference Proceedings, S. 1139–1147

[Tan13] Tang, Yichuan und Salakhutdinov, Ruslan: Learning Stochastic Feedforward Neural Networks, in: C.J.C. Burges; L. Bottou; M. Welling; Z. Ghahramani und K.Q. Weinberger (Herausgeber) *Advances in Neural Information Processing Systems 26* (2013), S. 530–538

[Tav90] Tavan, P.; Grubmüller, H. und Kühnel, H.: Self-organization of associative memory and pattern classification: recurrent signal processing on topological feature maps. *Biological Cybernetics* (1990), Bd. 64(2): S. 95–105

[Tei94] Teixeira, M. und Rodriguez, D.: A new method mathematically links fast Fourier transform algorithms with fast cyclic convolution algorithms, in: *Circuits and Systems, 1994., Proceedings of the 37th Midwest Symposium on*, Bd. 2, S. 829–833 vol.2

[Vap99] Vapnik, Vladimir: *The Nature of Statistical Learning Theory*, Springer, 2nd Aufl. (1999)

[Wan07] Wang, Y.S.; Lee, C.-M.; Kim, D.-G. und Xu, Y.: Sound-quality prediction for nonstationary vehicle interior noise based on wavelet pre-processing neural network model. *Journal of Sound and Vibration* (2007), Bd. 299(4-5): S. 933 – 947

[Wan14] Wang, Y.S.; Shen, G.Q. und Xing, Y.F.: A sound quality model for objective synthesis evaluation of vehicle interior noise based on artificial neural network. *Mechanical Systems and Signal Processing* (2014), Bd. 45(1): S. 255 – 266

[Wei09] Weise, Thomas: *Global Optimization Algorithms – Theory and Application*, it-weise.de (self-published): Germany (2009)

[Wil03] Wilson, D. Randall und Martinez, Tony R.: The general inefficiency of batch training for gradient descent learning. *Neural Networks* (2003), Bd. 16(10): S. 1429–1451

[Zel97] Zell, Andreas: *Simulation neuronaler Netze*, Oldenbourg Wissenschaftsverlag (1997)

[Zha09] Zhang, D. und Sui, W.: The Optimal Morlet Wavelet and Its Application on Mechanical Fault Detection, in: *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, S. 1–4

Lebenslauf

Persönliche Angaben

Name	**Niklas Fischer-von Rönn, geb. Fischer**.
Geburt	**2. September 1981 in Hamburg**.
Familienstand	**verheiratet, ein Kind**.

Ausbildung

Schule

1988–1992	**Grundschule Klein-Flottbeker Weg, Hamburg**.
1992–2001	**Gymnasium Hochrad, Hamburg** **Abschluss: Allgemeine Hochschulreife**.

Zivildienst

2001–2002	**Zivildienst an der Ernst & Cläre Jung Stiftung in Hamburg**.

Akademische Ausbildung und beruflicher Werdegang

2002–2008	**Studium der Technomathematik an der Universität Hamburg und der Technischen Universität Hamburg Harburg** **Abschluss: Diplom**.
2008–2014	**Wissenschaftlicher Mitarbeiter am Institut für Fahrzeug- und Antriebssystemtechnik der Helmut-Schmidt-Universität, Universität der Bundeswehr Hamburg**.
seit 05/2014	**Softwareentwickler bei der GDV Dienstleistungs-GmbH & Co. KG in Hamburg**.