Universität Rostock

Institut für Automatisierungstechnik

Interpretierbare datenbasierte Modellierung für die Anwendung in Motorsteuergeräten

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

vorgelegt von:

Björn Kolewe

Rostock, 2. Februar 2018

urn:nbn:de:gbv:28-diss2018-0083-8

eingereicht: 02.02.2018
verteidigt: 28.05.2018

Gutachter: Prof. Dr.-Ing. Torsten Jeinsch
Universität Rostock

Prof. Dr.-Ing. Bert Buchholz
Universität Rostock

Prof. Dr. Ping Zhang
Technische Universität Kaiserslautern

Kurzfassung

Die Komplexität neuer energie- und emmissionseffizienter Verbrennungsmotoren und Hybridantriebe sowie deren umfangreiche und hochgenaue Steuerungen und Regelungen der Motorfunktionen erfordern immer komplexere mathematische Modelle der zu kontrollierenden Prozesse. Diese müssen zudem in Motorsteuergeräten in Echtzeit berechenbar sein.

Datenbasierte Verfahren bieten die Chance, den damit verbundenen hohen Entwicklungs- und Applikationsaufwand zu reduzieren und die komplexen Zielgrößenverläufe auf Grundlage der am Prüfstand gewonnenen Messdaten zu approximieren. Dem entgegen steht der exponentiell wachsende Datenbedarf, der bei einer großen Anzahl von Eingangsgrößen die klassische Modellvalidierung über gesonderte Datensätze beschränkt und Optimierungsalgorithmen für eine tendenziell dünn besetzte Datenbasis erfordert.

In dieser Arbeit wird eine datenbasierte Modellstruktur zur Approximation komplexer, hochdimensionaler Prozesse für die Anwendung in Motorsteuergeräten vorgestellt, die als lokal unabhängiges, paralleles Modellnetz auf Grundlage der Basisfunktionsdefinition entworfen wurde sowie gut über Prozesswissen validierbar ist. Es wird eine spezielle Basisfunktion vorgeschlagen, welche die Forderung nach einer intuitiven und wissensbasierten Validierung unterstützt sowie lokale Änderungen der Modellparameter zulässt. Die resultierende Modellstruktur weist ein einfaches Interpolationsverhalten auf und kann in ihrem Extrapolationsverhalten flexibel an die konkreten Anforderungen angepasst werden. Ergänzend wird ein iterativer Optimierungsalgorithmus vorgeschlagen, welcher das Modell über eine hierarchische achsenorthogonale Partitionierung unter Vorgabe eines frei definierbaren Gütekriteriums approximiert. Die Teilungen erfolgen in jedem Iterationsschritt optimal und garantieren so eine minimale Anzahl an Teilmodellen. Die Ausgangsgleichung des Modells ermöglicht eine ressourcenschonende Berechnung auf den typischerweise auf Motorsteuergeräten eingesetzten Prozessoren.

Die Güte datenbasierter Modellierungen hängt in hohem Maße von der Qualität des zur Optimierung eingesetzten Datensatzes ab. In der vorliegenden Arbeit wird eine iterative Versuchsplanung vorgestellt, deren gruppierte sequentielle Ermittlung der Versuchspunkte den Vorteil bietet, das mit der Modellierung wachsende Prozesswissen zur optimalen Platzierung der nächsten Versuchspunkte zu nutzen. Die Versuchsplanung ist speziell für die Anforderungen des vorgestellten Modellierungsalgorithmus entworfen und in dessen Ablauf eingebunden. Sie garantiert eine optimale Verteilung der Versuchspunkte im Hinblick auf die Struktur- und Parameteroptimierung des vorgestellten Modellansatzes und vermeidet Überanpassungen an die Messdaten. Damit ermöglicht sie eine zuverlässige Bewertung der Modellgüte auf Grundlage der Trainingsdaten. Diese Eigenschaften tragen zu einer Minimierung des Versuchsaufwandes bei und ermöglichen eine effektive Vermessung von Prozessen, in denen keine Kenntnisse über den Verlauf der Zielgröße vorliegen. Die Versuchsplanung gestattet die Berücksichtigung praktischer Anforderungen, wie die freie Definition der Werte diskreter Stellgrößen sowie von Begrenzungen und Ausschlüssen im Versuchsraum. Geplante, jedoch nicht ausführbare Messungen werden online in die weitere Planung einbezogen.

Zur Umsetzung der Modellierung und Versuchsplanung wurde eine Toolkette unter MATLAB/Simulink entwickelt. Die Eigenschaften der vorgestellten Algorithmen und Methoden werden an Hand der Modellierung der Füllungserfassung eines Motors mit variablen Ventiltrieb aufgezeigt. Neben dieser praktischen Umsetzung wurden Vergleiche anhand synthetischer Testfunktionen mit GMR- und LOLIMOT-Modellierungen durchgeführt.

Abstract

The complexity of new, energy- and emission-efficient internal combustion engines and hybrid drives and their extensive and highly accurate control of engine functions require increasingly complex mathematical models of the processes to be controlled. In addition, these must be computable in real-time in engine control units. Data-based methods offer the opportunity to reduce the high development and application effort and to approximate the complex process behavior on the basis of the measurement data obtained at the test bench. On the other hand, there is the exponentially growing demand for data in processes with a large number of input variables. This restricts the possibility of model validation via test data sets and requires optimization algorithms for a sparse database.

In this thesis, a data-based model structure for the approximation of complex, high-dimensional processes for the application in ECUs is presented. It was designed as a locally independent model network (ILMON) according to the basic function definition and can easily be validated using process knowledge. A special basic function will be proposed, which supports intuitive and knowledge-based validation and allows local changes of the model parameters. The resulting model structure features a simple interpolation behavior and can be flexibly adapted to specific requirements in its extrapolation behavior. In addition, an iterative optimization algorithm will be proposed, which adapts the model to the process by means of a hierarchical, axis-orthogonal partitioning algorithm with specification of a user-defined quality criterion. Partitioning is carried out optimally in each iteration step and guarantees a minimum number of submodels. The model equation allows a resource-efficient calculation on the processors typically used on ECUs.

This thesis presents an iterative experimental design method. Its grouped sequential determination of the measuring points offers the advantage that with the modeling growing process knowledge can be used for an optimal placement of the next measuring points. The experimental design is specially developed to meet the requirements of the presented modeling algorithm and is integrated into its implementation. It guarantees an optimal distribution of the measuring points with regard to structure and parameter optimization of the ILMON model and avoids overfitting to the measured data. This enables a reliable evaluation of the model quality based on the training data. These properties contribute to a minimization of the testing effort and enable an effective measurement of processes without a-priori knowledge. The experimental design allows the consideration of practical requirements, such as the free definition of the values of discrete control variables as well as limitations and exclusions in the experimental space. Planned but non-executable measurements are taken into account and excluded in the further planning process.

A tool chain has been developed in MATLAB/Simulink for the implementation of the ILMON algorithm and the design of experiments method. The properties of the presented algorithms and methods are demonstrated by means of modeling the charge determination of an internal combustion engine with variable valve train. In addition to these practical implementations, comparisons were made using synthetic test functions with GMR and LOLIMOT modeling.

Inhaltsverzeichnis

Nomenklatur

Die Nomenklatur in dieser Arbeit entspricht den gebräuchlichen Symbolen und Schreibweisen der Ingenieurwissenschaften. Vektorielle Größen werden zur Unterscheidung gegenüber skalarer Größen in fetter Schriftstärke geschrieben. Matrizen werden in fetten Großbuchstaben bezeichnet.

Symbole

α ..... lokale Komponente
ANW .... Winkel der Auslassnockenwelle
AO ..... Winkel des Auslassöffnens
AS ..... Winkel des Auslassschließens
c ..... Koeffizient, Zentren
C ..... Komplexität
d ..... Durchmesser
D ... Datensatz, Menge von Messpunkten
e ..... Messfehler
ϵ ..... Modellfehler
ENW .... Winkel der Einlassnockenwelle
EO ..... Winkel des Einlassöffnens
ES ..... Winkel des Einlassschließens
φ ... Basisfunktion, Aktivierungsfunktion
Φ ..... Validierungsfunktion
γ ..... linearer Koeffizient
K ..... Komponentenanzahl, Parameteranzahl, Teilungszahl
$\mathcal{L}$ ..... Likelihoodfunktion
$\mathcal{M}$ ..... höherdimensionaler Datenraum
n ..... Motorendrehzahl
N ..... Messpunkteanzahl
ν ..... Parameteranzahl
$\boldsymbol{\nu}$ ..... Parametervektor
p ..... Wahrscheinlichkeitsdichte, Druck
q ..... Anzahl der Eingangsdimensionen
$\boldsymbol{Q}$ ..... Wichtungsmatrix
σ ..... Standardabweichung
σ^2 ..... Varianz
$\boldsymbol{S}$ ..... Glättungsmatrix
$\boldsymbol{\Sigma}$ ..... Normierungsmatrix
$\boldsymbol{\theta}$ ..... Parametersatz
T ..... Temperatur
u ..... Eingangsgröße
$\tilde{\boldsymbol{u}}$ ..... erweiterter Eingangsgrößenvektor
w, v ..... Wichtung, Koeffizient
$\boldsymbol{w}$ ..... Normalenvektor
x ..... Funktionsargument
$\boldsymbol{X}$ ..... Designmatrix
y ..... Ausgangsgröße, Funktionswert
$\hat{y}$ ..... geschätzte Ausgangsgröße

Abkürzungen

AIC ..... Akaike Information Criterion
BIC ..... Bayesian Information Criterion
CFD ..... Computational Fluid Dynamics
EM ..... Expectation-Maximization
FPU ..... Floating Point Unit
GMR ..... Gaussian-Mixture-Regression
ILMON ..... Interpretierbares lokales Modellnetz
KNN ..... Künstliches neuronales Netz

LOLIMOT	Lokal-Linear-Model-Tree
MLP	Multilayer Perzeptron
RBF	Radiale Basis Funktion
SVR	Support-Vektor-Regression

1. Einleitung

1.1. Motivation

Trotz des Booms der Elektromobilität ist der Verbrennungsmotor weiterhin die wichtigste Antriebsquelle in allen Arten von Fahrzeugen. Mit seinen Vorteilen in Reichweite und Geschwindigkeit des Auftankens wird er auch in den nächsten Jahren nicht durch alternative Antriebstechnologien zu ersetzen sein [1], [2], [3]. Um so dringlicher ist die weitere Optimierung des Verbrauchs und die Reduktion der Emissionen, was den Einsatz immer komplexerer Technik sowie eine umfangreiche und hochgenaue Steuerung und Regelung der Motorfunktionen erfordert. Erweiterte Stellmöglichkeiten, welche durch moderne Funktionen wie den variablen Ventiltrieb oder der bedarfsgesteuerten Zylinderabschaltung einen weitgehenden Betrieb in optimalen Wirkungsgradbereichen ermöglichen, erfordern für die Motorsteuerung immer komplexere mathematische Prozessmodelle. Diese müssen zudem in Echtzeit in Motorsteuergeräten berechenbar sein. Hybridantriebe, welche die günstigsten Eigenschaften von Verbrennungs- und Elektromotoren kombinieren, erweitern die Optionen zum energie- und emmissionseffizienten Betrieb nochmals und erhöhen damit ebenfalls die Komplexität der Motorsteuerungen.

Dies alles führt zu einem immer höheren Entwicklungs- und Applikationsaufwand der auf physikalischen Prozessmodellen beruhenden Softwarefunktionen. Als einen möglichen Ausweg bieten sich datenbasierte Modellierungsverfahren an, welche auf Grundlage der Vermessung an Motorversuchsständen auch komplexe Verläufe der Zielgröße approximieren können und damit einen Großteil des Applikationsaufwandes zu reduzieren helfen. Voraussetzung für deren praktischen Einsatz in Motorsteuerungen ist jedoch die Einhaltung der hohen Zuverlässigkeits- und Sicherheitsvorgaben der Automobilindustrie, welche eine alle Einsatzbereiche umfassende Validierung der identifizierten Modellparameter erfordert. Eine hohe Modellkomplexität gepaart mit der großen Anzahl an Stellmöglichkeiten und Einflussfaktoren führt jedoch häufig dazu, dass klassische Validierungsmethoden datenbasierter Modelle nicht mehr angewendet werden können. Diese arbeiten über Validierungsdatensätze, die durch den exponentiell wachsenden Datenbedarf nur noch mit einem extrem großen Messaufwand zu ermitteln sind.

Einen Ausweg bieten hier datenbasierte Entwurfsverfahren, die nach der Identifizierung der Parameter die Möglichkeit eröffnen, alle Bereiche per Expertenwissen zu validieren. Dafür geeignete Modellstrukturen sollten die unabhängige Untersuchung einzelner Teilbereiche zulassen sowie nachträgliche lokale Anpassungen zur Applikation des Modells ermöglichen. Neben diesen Eigenschaften sind eine geringe Neigung zur Überanpassung an die Trainingsdaten, ein einfaches Interpolationsverhalten sowie ein geringer Messdatenbedarf wünschenswerte Merkmale.

Die Forderungen nach langen Produktzyklen, hoher Zuverlässigkeit und einer langjährigen Ersatzteilversorgung sind die Gründe für den bevorzugten Einsatz von erprobten und robusten Mikrocontrollern in den aktuellen Motorsteuergeräten, die weit weniger Rechenleistung zur Verfügung stellen als allseits präsente kurzlebige Kommunikations- und Unterhaltungs-

elektronik. Für einen Einsatz der datenbasierten Verfahren im Fahrzeug müssen diese den eingeschränkten Ressourcen gerecht werden sowie eine schnelle und effektive Berechnung der Ausgangsgrößen des vollständig applizierten Modells erlauben.

Die resultierende Modellgüte datenbasierter Modellierungsverfahren hängt neben der Quantität auch im hohen Maße von der Verteilung der zur Identifikation und Validierung verwendeten Messdaten ab. Eine optimale Versuchsplanung, welche sowohl den Verlauf der zu approximierenden Ausgangsgröße als auch den verwendeten Entwurfsalgorithmus berücksichtigt, ist somit die Voraussetzung für eine realistische Abschätzung des Modellfehlers. Ist bedingt durch das exponentielle Wachstum der benötigten Datenbasis eine vollständige und engmaschige Vermessung in hochkomplexen Prozessen nicht mehr möglich, steigt die Bedeutung der optimalen Verteilung der aufgenommenen Messdaten noch weiter. Versuchsplanungsverfahren, die diesen Gegebenheiten gerecht werden und durch die Verringerung der notwendigen Messdatenmenge den Versuchsaufwand minimieren, sind somit für einen effektiven Einsatz datenbasierter Verfahren unabdingbar. Dabei müssen sie den praktischen Anforderungen der Motorvermessung wie die Berücksichtigung nicht anfahrbare Arbeitsbereiche oder konstruktions- oder prozessbedingten Rasterungen der Stell- und Eingangsgrößen gerecht werden.

1.2. Problemstellung

In dieser Arbeit soll ein datenbasiertes Modellierungsverfahren vorgestellt werden, welches geeignet ist, komplexe nichtlineare Funktionen mit mehreren Eingangsgrößen zu approximieren und sich speziell für den Einsatz in Motorsteuergeräten anbietet. Eine wichtige Forderung an das Verfahren ist die Zugehörigkeit zur Klasse der universellen Approximatoren, womit die Genauigkeit der Approximation über die Erhöhung der Modellkomplexität bis zu einem gewünschten Maß gesteigert werden kann. Mit dieser Forderung ist die erreichbare Genauigkeit unabhängig vom zu modellierenden Prozess und ausschließlich von der Quantität und Qualität der Messdaten begrenzt. Neben dieser grundsätzlichen Eigenschaft sollen weitere Anforderungen erfüllt werden:

1. *Frei definierbares Gütekriterium:* Die Optimierung der Modellstruktur und der Modellparameter auf die Messdaten des Prozesses erfolgt über die Definition eines Gütekriteriums zur Beurteilung der Modellgenauigkeit. Die in der Motorenentwicklung traditionell in verschiedenen Bereichen eingesetzten Modellgütekriterien sollen direkt für die Optimierung der Modellparameter genutzt werden können, was zur Forderung eines frei definierbaren Modellgütekriteriums führt.

2. *Berücksichtigung von Prozesswissen:* Mehrere Jahrzehnte Forschungs- und Entwicklungsarbeit in der Automobilindustrie haben zu einem großen Wissens- und Erfahrungsschatz geführt, der gegebenenfalls in eine datenbasierte Modellierung einfließen sollte. Für ein Modellierungsverfahren in der Motorenentwicklung besteht daher die Anforderung neben der Optimierung ohne spezielle Kenntnisse des Prozesses auch Vorwissen integrieren zu können und dieses vorteilhaft bezüglich der Genauigkeit, der Validierung und des Datenbedarfs zu nutzen.

3. *Geringer Bedatungsaufwand:* Das Ziel, die enormen Entwicklungskosten neuer Verbrennungsmotoren zu senken, spiegelt sich auch in der Forderung nach einem geringen

Aufwand zur Gewinnung der für die datenbasierte Modellierung notwendigen Messdaten wider. Verfahren, die den Vermessungsaufwand auf den Motorenprüfständen gering halten, haben hier deutliche Vorteile und sollen gegenüber anderen Verfahren bevorzugt werden.

4. *Validierung und Verifikation:* Den hohen Sicherheits- und Zuverlässigkeitsanforderungen der Automobilindustrie entsprechend muss für neue Funktionen in Motorsteuergeräten der Nachweis eines fehlerfreien und zuverlässigen Betriebs in allen praktisch relevanten Betriebssituationen erfolgen. Klassische, auf punktuelle Überprüfungen mit Validierungsdaten ausgelegte Testszenarien können dies bei komplexen hochdimensionalen datenbasierten Modellen nicht mehr gewährleisten. Hier bedarf es Validierungsstrategien, die per Expertenwissen große Modellbereiche auch ohne zusätzliche Messdaten überprüfen können. Es wurden daher Modellstrukturen gesucht, die solche Untersuchungen ermöglichen und unterstützen.

5. *Geringer Ressourcenbedarf:* Trotz der rasant fortschreitenden Entwicklung der Leistungsfähigkeit moderner Mikrocontroller ist weiterhin eine starke Ressourcenbeschränkung der auf den Fahrzeugen zur Motorsteuerung eingesetzten Steuergeräte gegeben. Dies wird durch die Vielzahl neuer und komplexer Funktionen auch auf zukünftigen Generationen von Motorsteuergeräten ein Punkt bleiben, dem eine große Aufmerksamkeit zukommen muss. Eingesetzte datenbasierte Modelle müssen daher eine ressourcenschonende Berechnung auf der typischerweise mikrocontrollerbasierten Hardware gewährleisten und je nach Einsatzzweck auch eine Echtzeitberechnung der Ausgangsgröße ermöglichen.

Die erreichbare Modellgüte hängt in datenbasierten Verfahren im hohen Maße von der für die Optimierung verwendeten Datenbasis ab. Anzahl und Verteilung der Messdaten müssen im Einklang mit dem zu modellierenden Prozess sein, um die gewünschten Optimierungsziele erreichen zu können. Ist der zu modellierende Prozess gut bekannt, können die Versuchspunkte auf Grundlage des Prozesswissens festgelegt werden. Ohne ausreichende Kenntnisse über den Verlauf der Zielgröße ist die Anwendung gleichverteilter Versuchspläne die typische Vorgehensweise, welche in komplexen, hochdimensionalen Prozessen jedoch zu einem sehr großen Messaufwand führt. Zur Reduzierung des Versuchsaufwandes und zur Erhöhung der Modellgüte soll begleitend eine speziell auf die Eigenschaften des Modellierungsalgorithmus angepasste Versuchsplanung entwickelt werden, die zum einen den Messaufwand reduziert und zum anderen eine optimale Abschätzung des Modellfehlers auf Grund der zum Training des Modells verwendeten Datenbasis erlaubt.

1.3. Aufbau der Arbeit

Die Arbeit gliedert sich in folgende Kapitel:

Kapitel 2 In diesem Kapitel werden die wesentliche Grundlagen zur datenbasierten Modellierung, die verschiedenen Arten der Modellfehler und Verfahren zur Bewertung der optimierten Modelle dargestellt. Insbesondere wird auf die Möglichkeiten der Modellvalidierung über Testdaten und Prozesswissen eingegangen sowie die Vor- und Nachteile verschiedener

Vorgehensweisen diskutiert. Es werden Kriterien für den Einsatz von datenbasierten Modellen in Motorsteuergeräten erarbeitet und die Eignung populärer Modellierungsverfahren hinsichtlich dieser Kriterien untersucht.

Kapitel 3 Hier wird systematisch die Struktur eines gut per Prozesswissen validierbaren Modellansatzes erarbeitet, der den im Kapitel 2 aufgestellten Kriterien genügt. Im Fokus stehen dabei der Entwurf einer geeigneten, multivariaten Basisfunktion und die Entwicklung eines iterativen Algorithmus' zur Strukturoptimierung. Weiterhin wird auf die lokale Parameterschätzung der linearen Komponenten eingegangen und der Effekt der damit verknüpften impliziten Regularisierung näher beleuchtet. Einen großen Raum nehmen die Erläuterungen zu Realisierungsaspekten des Modellierungsalgorithmus ein. Hier wird auf die konkreten Fragestellungen der Initialisierung, Optimierung und Bewertung der Modellgüte eingegangen sowie die Möglichkeit der Integration von Prozesswissen aufgezeigt. Ebenso wird das Inter- und Extrapolationsverhalten sowie der Ressourcenbedarf näher untersucht. Mit der Erweiterung der lokalen Komponenten um quadratische Regressoren wird eine Möglichkeit erörtert, die Modellstruktur der ILMON-Modellierung flexibler zu gestalten und so den Anwendungsbereich des Verfahrens zu vergrößern.

Kapitel 4 In diesem Kapitel wird einleitend auf die verschiedenen Verfahren der statistischen Versuchsplanung eingegangen. Es werden die Eigenschaften sowie die Vor- und Nachteile im Hinblick auf die Nutzung im Rahmen einer ILMON-Modellierung dargestellt. Im Anschluss werden die Kriterien einer optimalen Versuchsplanung für eine iterative Modellierung erarbeitet sowie auf die Besonderheiten der Parameter- und Strukturoptimierung eingegangen. Im letzten Abschnitt erfolgt auf Grundlage dieser Kriterien der Entwurf einer Versuchsplanung für eine ILMON-Struktur und es wird die Integration der Abläufe in den iterativen Modellierungsprozess erläutert. In dieser Umsetzung wird ein Planungsalgorithmus vorgeschlagen, welcher die Versuchspunkte optimal in den Gültigkeitsbereichen der Teilmodelle setzt und eine ausgeglichene Verteilung der Messpunkte in den einzelnen Eingangsgrößen erlaubt. Dabei werden vorhandene Messungen, Rasterungen der Stellgrößen sowie nicht anfahrbare Bereiche im Eingangsraum berücksichtigt und Korrelationen vermieden.

Kapitel 5 Mit dem vorgestellten Modellierungsalgorithmus werden in diesem Kapitel synthetische Testfunktionen approximiert und die Eigenschaften der entstehenden ILMON-Modellstruktur in Abhängigkeit der Parameter des Algorithmus' dargestellt. Weiterhin werden beispielhaft die Integration von Prozesswissen sowie die Interpretierbarkeit der Modellstruktur veranschaulicht. Der Vergleich mit den Ergebnissen einer LOLIMOT- sowie einer GMR-Modellierung dient zur Einordnung des Anwendungsspektrums und der Leistungsfähigkeit des Verfahrens. Neben der Modellierung mit gleichverteilten Datensätzen erfolgt die Auswahl der Datenpunkte auch über die iterative Versuchsplanung unter verschiedenen Parametern, deren Ergebnisse gegenübergestellt werden.

Als praxisnahes Beispiel wird die Füllungserfassung eines 3,2l-Saugmotors mit variablem Ventiltrieb mit 6 Eingangsgrößen modelliert, welcher als Simulation unter der Software GT-ISE von Gamma Technology zur Verfügung stand. In den mit verschiedenen Parametern durchgeführten Modellierungsdurchläufen kommt durchgehend die iterative Versuchsplanung zur Festlegung der zu messenden Arbeitspunkte zum Einsatz.

2. Interpretierbare datenbasierte Modellierung

Datenbasierte Modellierungsverfahren können sich ihrer Struktur entsprechend optimal an die vorgegebenen Trainingsdaten anpassen. Eventuelle Messfehler fließen dabei mehr oder weniger direkt in die Parameter des Modells mit ein und zu hoch gewählte Modellkomplexitäten führen zu Überanpassungen an die Trainingsdaten. Die Optimierung anhand der Messdaten bedingt prinzipiell, dass Modellbereiche ohne Messdaten nicht approximiert werden, was je nach Interpolationsverhalten des gewählten Verfahrens zu problematischen Verläufen in diesen Bereichen des Modells führen kann. Im Allgemeinen werden datenbasierte Modelle anhand von Validierungsdaten überprüft und getestet. In der Praxis stehen aus verschiedenen Gründen oft zu wenige Daten für diesen Zweck zur Verfügung, sodass als Möglichkeit der Validierung nur eine heuristische Überprüfung des Modells unter Verwendung von Prozesswissen in Frage kommt. Einleitend wird in diesem Kapitel auf die Problematik der oft ungenügenden Datenbasis sowie der verschiedenen Ursachen der Modellfehler eingegangen und weiterführend die Möglichkeiten der Validierung der datenbasierten Modelle erörtert. Zusammenfassend wird eine Modellstruktur vorgestellt, die eine einfache heuristische Überprüfung unterstützt.

2.1. Fluch der Dimensionalität

Die Optimierung datenbasierter Modelle hängt im starken Maße von der Anzahl und der Verteilung der Daten im Eingangsraum ab. Ohne eine spezielle Annahme der Modellstruktur bzw. ohne Kenntnisse des zu modellierenden Prozesses sind gleichmäßig über den gesamten Eingangsraum verteilte Messdaten als optimal anzusehen (siehe dazu auch Kapitel 4.1). Ein gleichmäßiges, dichtes Versuchsraster in jeder Eingangsdimension gewährleistet somit meist eine gute Approximation des Prozesses. Jedoch steigt das Volumen und mit ihm die Messpunkteanzahl des durch die Eingangsgrößen aufgespannten mathematischen Raumes bei einer linearen Erhöhung der Eingangsdimensionen exponentiell an. Das Problem des mit der Anzahl der Eingangsdimensionen exponentiell steigenden Volumens im mathematischen Raum wurde erstmals von R. E. Bellman, als „Fluch der Dimensionalität“ bezeichnet [4].

An einem einfachen Beispiel soll die Problematik verdeutlicht werden: Gegeben sei ein eindimensionaler Prozess

$$y = f(u) + e \tag{2.1}$$

mit y als skalare Ausgangsgröße, u als skalare Eingangsgröße und e als unabhängiger Messfehler. Ist der Zusammenhang zwischen y und u hinreichend glatt und der Betrag des Messfehlers sehr klein, so genügen bei einer vorgegebenen Approximationsgenauigkeit wenige Messungen zur Bestimmung der Parameter der Gleichung. Im Fall eines linearen Zusammenhanges wären dies mindestens 2 Messwerte. Mit einem in der Praxis vorhandenem

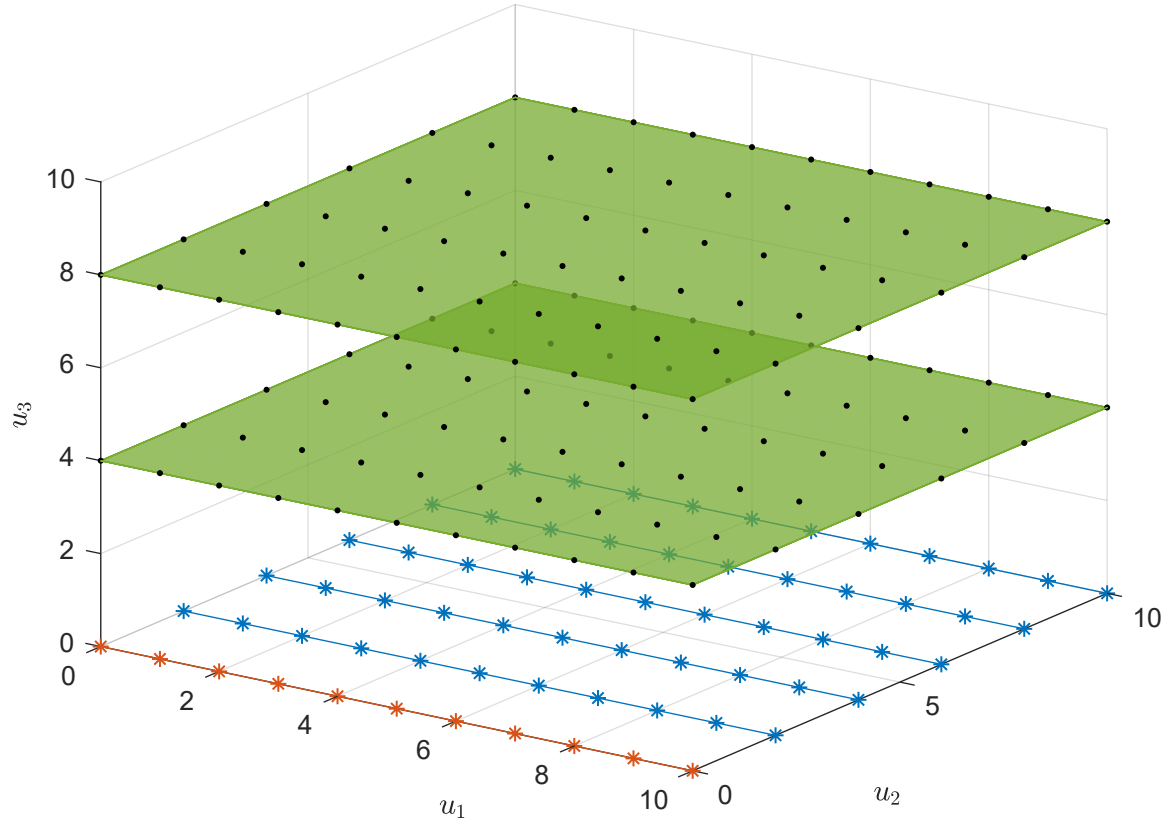

Abbildung 2.1.: Entwicklung der Messpunkteanzahl bei Erhöhung der Eingangsdimensionen mit $N_{S1} = 11$ (rot); $N_{S2} = 6$ (blau); $N_{S3} = 3$ (grün)

Messrauschen wird man selbst bei relativ glatten Zusammenhängen mindestens 10 Messwerte aufnehmen. Erweitert man die Prozessbeschreibung auf q Eingangsdimensionen

$$y = f(u_1, u_2, \ldots, u_q) + e \tag{2.2}$$

so vervielfältigen sich die Stützstellen der ersten Eingangsdimension in jeder weiteren Eingangsdimension entsprechend der dortigen Anzahl an Stützstellen. Abbildung 2.1 veranschaulicht dies.

Die Anzahl der Messpunkte N_M erhöht sich somit multiplikativ um die Anzahl der Stützstellen N_S in jeder weiteren Eingangsdimension.

$$N_M = \prod_{i=1}^{q} N_{Si} \tag{2.3}$$

Damit wächst die Anzahl der Messpunkte bei einer vorgegebenen Stützstellenanzahl pro Eingangsdimension expotentiell und erreicht selbst mit einer moderaten Anzahl von 10 Stützstellen pro Eingangsdimension ab 4 bis 5 Eingangsgrößen unpraktikable Versuchsanzahlen. Ein vollständiges Versuchsraster ist damit in der Regel zu zeit- und kostenintensiv.

In der praktischen Umsetzung einer datenbasierten Modellierung gibt es allerdings zwei Faktoren, die sich reduzierend auf die benötigte Anzahl an Daten auswirken und somit

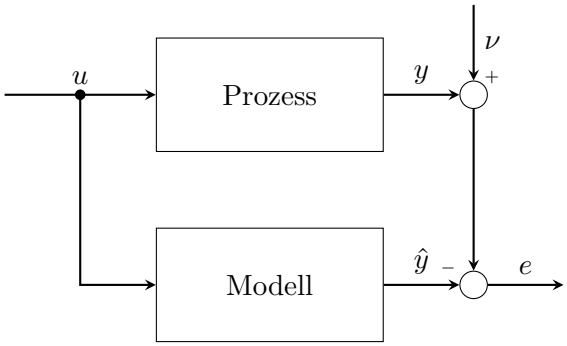

Abbildung 2.2.: Zusammenhang von Prozess, Modell, Messrauschen und Modellfehler

eine höherdimensionale Modellierung im Regelfall möglich machen. Zum einen wird der mathematische Eingangsraum oft durch Bereiche eingeschränkt, die im Betrieb nicht erreicht werden können bzw. keinen sinnvollen Betriebszustand darstellen oder aus Gründen des Bauteilschutzes nicht durchfahren werden dürfen. Des Weiteren sind die Zusammenhänge zwischen Eingangs- und Ausgangsgröße bei vielen realen Prozessen in weiten Bereichen des Eingangsraumes annähernd linearer Natur, sodass für diese Bereiche wenige Messpunkte zur Optimierung der Parameter ausreichen.

2.2. Validierung datenbasierter Modelle

Zur Beurteilung des Fehlers eines datenbasierten Modells gibt es verschiedene etablierte Verfahren und Methoden. In diesem Kapitel sollen die wichtigsten Vorgehensweisen zur Validierung eines datenbasierten Modells diskutiert sowie die konkreten Anforderungen erarbeitet werden, welche aus dem physikalischen Prozess und den praktischen Gegebenheiten resultieren.

Die Abbildung eines komplexen realen Prozesses auf ein mathematisches Modell führt in der Regel zu Modellfehlern, deren Ursache grob in 3 Kategorien eingeteilt werden kann:

- Es wurde ein ungeeigneter Modellansatz gewählt.
- Die gewählte Modellkomplexität ist nicht optimal.
- Die Schätzung der Modellparameter ist auf Grund der Menge und Qualität der Daten ungenau.

Während die Wahl des Modellansatzes in der Regel auf A-priori-Wissen beruht, sich nach dem zu modellierenden Prozess richtet und nicht Teil des Optimierungsprozesses ist, sollten die Modellkomplexität und die Schätzung der Modellparameter auf Grund geeigneter Kriterien bewertet werden. Um dies genauer zu betrachten, sollen im Folgenden die Ursachen für den Modellierungsfehler untersucht werden.

Es sei $y(\boldsymbol{u})$ die Ausgangsgröße des Prozesses, ν ein von den Eingangsgrößen unkorreliertes, normalveteiltes Rauschen mit dem Erwartungswert Null und $\hat{y}(\boldsymbol{u})$ die vom Modell geschätzte Ausgangsgröße (Bild 2.2), dann kann der Erwartungswert des quadratischen Fehlers des Modells wie folgt zerlegt werden [5],[6],[7],[8]:

$$E\{(y(\boldsymbol{u}) + \nu - \hat{y}(\boldsymbol{u}))^2\} = E\{(y(\boldsymbol{u}) + \nu)^2\} + E\{(\hat{y}(\boldsymbol{u}))^2 - 2E\{(y(u) + \nu)\hat{y}\}\} \tag{2.4}$$

Mit dem Verschiebungssatz gilt:

$$E\{(y(\boldsymbol{u}) + \nu)^2\} = Var\{y(\boldsymbol{u}) + \nu)\} + E\{y(\boldsymbol{u}) + \nu)\}^2$$
$$E\{(\hat{y}(\boldsymbol{u}))^2\} = Var\{\hat{y}(\boldsymbol{u})\} + E\{\hat{y}(\boldsymbol{u})\}$$

Weiterhin gilt mit den oben genannten Voraussetzungen für ν:

$$E\{\nu\} = 0 \quad \text{und} \quad Var\{\nu\} = \sigma^2.$$

Da die Ausgangsgröße $y(\boldsymbol{u})$ des Prozesses deterministisch ist, ist der Erwartungswert $E\{y(\boldsymbol{u})\} = y(\boldsymbol{u})$ und es ergibt sich:

$$\begin{aligned} E\{(y(\boldsymbol{u}) + \nu - \hat{y}(\boldsymbol{u}))^2\} &= Var\{\hat{y}(\boldsymbol{u})\} + (y(\boldsymbol{u}) - E\{\hat{y}(\boldsymbol{u})\})^2 + \sigma^2 \\ &= Var\{\hat{y}(\boldsymbol{u})\} + Bias\{\hat{y}(\boldsymbol{u})\}^2 + \sigma^2 \end{aligned} \tag{2.5}$$

Die in Gleichung 2.5 enthaltenen Terme stellen die drei grundsätzlichen Fehler dar, aus denen sich der Modellfehler bei einer datenbasierten Modellierung zusammensetzt.

Der *Bias* kann als der Fehler interpretiert werden, der durch eine zu niedrige Modellkomplexität und damit einer Unteranpassung des Modells an den realen Prozess entsteht. Er ergibt sich auch bei optimal bestimmten Modellparametern und lässt sich nur durch die Erhöhung der Modellkomplexität verringern. Gehört das Modellierungsverfahren zur Klasse der universellen Approximatoren, so geht der Bias mit der Erhöhung der Modellkomplexität gegen Null. Durch die Begrenzung der Ressourcen im Motorsteuergerät und den sehr komplexen Prozessen im automotiven Sektor ist es in der Regel nicht möglich, den Prozess durch das Modell exakt abzubilden. Dementsprechend muss ein Kompromiss zwischen der Komplexität des Modells und der Größe des tolerierbaren Bias gefunden werden.

Werden für die Optimierung des Modells zwei verschiedene Datensätze verwendet, bei ansonsten gleicher Modellstruktur und -komplexität, so wird man voneinander abweichende Modellparameter für beide Optimierungsdurchläufe erhalten. Der *Varianz*-Term in Gleichung 2.5 beschreibt diese Unsicherheit der Parameter des Modells, welche durch einen endlichen, messfehlerbehafteten Trainingsdatensatz verursacht wird. Mit einem unendlich großen Datensatz zur Optimierung des Modells würde der Varianzfehler verschwinden. Ist hingegen die Anzahl der Trainingsdaten und der Parameter des Modells gleich, so wird der Varianzfehler maximiert. Das Modell wäre in diesem Fall optimal an den Trainingsdatensatz angepasst, inklusive des enthaltenen Messrauschens, was wiederum bedeutet, dass die abweichende Rauschverteilung im Validierungsdatensatz zu größeren Fehlern führt. Als Konsequenz sollte zur Minimierung des Varianzfehlers der Trainingsdatensatz sehr viel größer sein, als die Anzahl der Parameter des Modells.

In Bild 2.3 ist der prinzipielle Verlauf des Bias und Varianzfehlers einer datenbasierten Modellierung für einen Trainingsdatensatz fester Größe dargestellt. Der Bias des Modells verringert sich wie oben erläutert mit der Erhöhung der Modellkomplexität. Die damit einhergehende Steigerung der Parameteranzahl vergrößert aber den Varianzfehler des Modells. Diese gegenläufige Entwicklung wird als Bias-Varianz-Dilemma bezeichnet [9],[6],[7]. Der An-

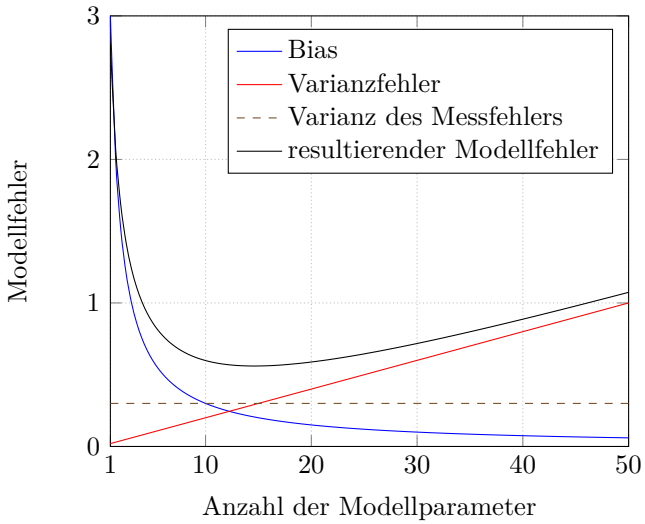

Abbildung 2.3.: Verlauf des Bias, des Varianzfehlers und der Varianz des Messfehlers bei steigender Modellkomplexität

stieg des Varianzfehlers ist dabei umso flacher, je mehr Messpunkte im Trainingsdatensatz zur Verfügung stehen.

Der dritte Term, welcher in Gleichung 2.5 den Modellfehler beeinflusst, ist die Varianz σ^2 des Messrauschens ν. Da ν über die Parameter der Modellierung nicht beeinflusst werden kann, stellt die Varianz des Messrauschens eine untere Schranke des Modellfehlers dar. Es sei darauf hingewiesen, dass die Abschätzung des Messfehlers in der praktischen Umsetzung ein wichtiger Punkt vor dem Festlegen der geforderten Modellgüte ist, da ein Modellfehler kleiner als der Messfehler nicht erreicht werden kann.

Zur Berechnung des Modellfehlers können sowohl der Trainingsdatensatz als auch der Validierungsdatensatz herangezogen werden. Im Folgenden wird der mit dem Trainingsdatensatz berechnete Fehler als Trainingsfehler und der sich aus dem Validierungsdatensatz ergebene Wert als Validierungsfehler bezeichnet. Beide Fehlerwerte zeigen bei steigender Modellkomplexität ein unterschiedliches Verhalten, welches in Bild 2.4 dargestellt ist. Der Trainingsfehler verringert sich kontinuierlich mit der Erhöhung der Modellkomplexität und es erfolgt eine Überanpassung an die Trainingsdaten inklusive der in ihnen enthaltenen Messfehler. Dagegen weißt der Validierungsfehler ein Minimum an der Stelle auf, an der die Summe aus Bias und Varianzfehler am kleinsten ist. Die Varianz des Messrauschens ist, wie oben beschrieben, eine für den Modellfehler prinzipiell vorhandene untere Schranke, die jedoch nur bei der Berechnung des Validierungsfehlers zum Tragen kommt.

In den folgenden Kapiteln soll auf die geeignete Auswahl der Trainings- und Validierungsdaten sowie auf alternative Möglichkeiten der Modellvalidierung eingegangen werden.

2.2.1. Modellvalidierung über Testdaten

Die einfachste und weit verbreitete Form der Modellvalidierung über Testdaten ist das Aufteilen der zur Verfügung stehenden Daten auf zwei Datensätze, von dem einer als Trainingsdatensatz und der andere als Validierungsdatensatz verwendet wird. Stehen ausreichend Daten zur Verfügung, erwächst aus dieser Variante auch kein Nachteil. Üblich sind Teilungen

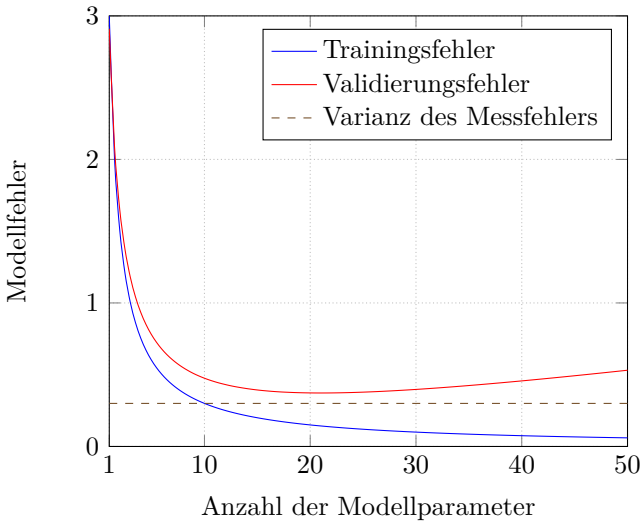

Abbildung 2.4.: Prinzipieller Verlauf des Trainings- und Validierungsfehlers bei steigender Modellkomplexität

von 70% Trainingsdaten zu 30% Validierungsdaten. Bei der Aufteilung ist zu beachten, dass beide Datensätze den Eingangsraum sowohl ausreichend als auch gleichmäßig abdecken. Um Korrelationen mit Messfehlern zu vermeiden, die aus der zeitlichen Variation der Messungen resultieren, sollten die Datensätze Messpunkte aus dem gesamten Zeitraum der Messkampagne enthalten. Beide Kriterien müssen daher schon beim Aufstellen des Versuchsplanes berücksichtigt werden. Die einfache Teilung und die damit verbundene erhebliche Reduzierung der zum Training des Modells nutzbaren Datenmenge findet ihre Grenzen, wenn gemessen an der Komplexität des Modells, nur wenig Messdaten zur Verfügung stehen. Insbesondere ist dies in der Praxis häufig bei höherdimensionalen Prozessen, durch die im Kapitel 2.1 aufgezeigten Problematik, der Fall. Die Konsequenz eines zu kleinen Trainingsdatensatzes wäre eine Erhöhung des Bias des Modells und damit eine Unteranpassung an den Prozess. Mit einem zu kleinen Validierungsdatensatz besteht die Gefahr einer zu optimistischen Beurteilung des Modellfehlers, da Modellbereiche mit starken Abweichungen vom Prozessverhalten nicht oder nicht ausreichend abgedeckt werden könnten.

Kreuzvalidierung

Die Modellierung komplexer Prozesse mit einer hohen Anzahl an Eingangsgrößen und der damit verbundenen hohen Eingangsdimensionalität rückt die Problematik eines für diesen Zweck nicht ausreichenden Datensatzes in den Vordergrund. Mit der klassischen Aufteilung in Trainings- und Validierungsdaten wird die Anzahl der Datenpunkte für beide Bereiche weiter reduziert. Eine Möglichkeit diese Problematik zu entschärfen, ist die sogenannte K-fache Kreuzvalidierung [5],[10]. Bei dieser wird der Datensatz D mit N Datenpunkten in K gleichgroße Teildatensätze D_k mit N_k Messpunkten aufgeteilt. $(K-1)$ Teile werden nun als Trainingsdatensatz verwendet und die Validierung wird mit dem verbleibenen Datensatz D_{-k} durchgeführt. Der Fehler ϵ_k des optimierten Modells $f_k(\boldsymbol{u})$aus dem k-ten Durchlauf

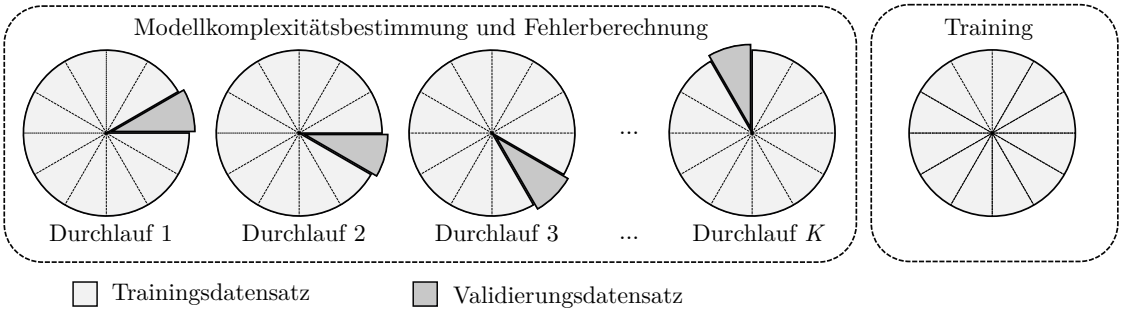

Abbildung 2.5.: Aufteilung des Datensatzes in Trainings- und Validierungsdaten bei der Kreuzvalidierung, Aufteilung des Datensatz in $K = 12$ Teile, K Durchläufe zur Fehler- und Modellkomplexitätsbestimmung, Abschließende Optimierung des Modells mit dem kompletten Datensatz

ergibt sich mit der Verlustfunktion $L(y, f_k(\boldsymbol{u}))$ zu:

$$\epsilon_k = \frac{1}{N_{-k}} \sum_{i=1}^{N_{-k}} L(y_i, f_k(\boldsymbol{u}_i)) \tag{2.6}$$

Dies wird insgesamt K mal durchgeführt, wobei bei jedem Durchlauf k ein anderer Teildatensatz zur Validierung verwendet wird. Bild 2.5 zeigt das prinzipielle Vorgehen. Mit der quadratischen Abweichung als Verlustfunktion ergibt sich der Modellfehler ϵ bei diesem Verfahren aus dem Durchschnitt des mittleren quadratischen Fehlers der Modelle aus jedem Durchlauf.

$$\epsilon = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_{-k}} \sum_{i=1}^{N_{-k}} (\hat{y}_{i,k} - y_{i,k})^2 \tag{2.7}$$

Ziel dieser mehrfachen, rotierenden Modellierung über den ganzen Datensatz ist die Festlegung der optimalen Modellkomplexität anhand des in Bild 2.4 dargestellten Optimums des Modellfehlers. Nach Festlegen der Modellkomplexität und der Bestimmung des Modellfehlers können die Parameter des Modells abschließend über den gesamten Datensatz trainiert werden. Auf diese Weise können alle Messpunkte des Datensatzes sowohl zum Training als auch zur Validierung des Modells genutzt werden. Als Nachteil der K-fachen Kreuzvalidierung ist der mit steigendem K anwachsende Berechnungsaufwand zu nennen, wobei pro zu testender Modellkomplexität insgesamt $K + 1$ Trainingsdurchläufe notwendig sind.

Die Aufteilung der Daten auf die Teildatensätze D_k kann auf verschiedene Weise erfolgen. Bei der einfachen K-fachen Kreuzvalidierung erfolgt dies in der Regel durch eine zufällige Zuordnung der einzelnen Messpunkte. Nachteil dieser Vorgehensweise ist die möglicherweise stark unterschiedliche Verteilung der Messpunkte in den K Teildatensätzen und der damit verbundenen erhöhten Varianz des Modellfehlers [6],[10]. Um diese in den einzelnen Durchläufen gering zu halten, sollte bei der Aufteilung darauf geachtet werden, dass die Verteilung der Messpunkte in den Teildatensätzen annähernd der Verteilung im Gesamtdatensatz entspricht. Eine solche Aufteilung wird als stratifizierte K-fache Kreuzvalidierung bezeichnet.

Im Extremfall könnte als Validierungsdatensatz nur ein Messpunkt verwendet werden, was als „leave-one-out“ Verfahren bezeichnet wird. Diese Variante der Kreuzvalidierung

erfordert allerdings einen mit steigender Anzahl an Messpunkten anwachsenden sehr hohen Berechnungsaufwand und ist daher für die hier betrachteten hochdimensionalen Prozesse mit einer großen Anzahl an Messwerten weniger geeignet.

Bootstrapping

Eine weiteres Verfahren zur besseren Nutzung des für das Training des Modells und zur Schätzung des Modellfehlers zur Verfügung stehenden Datensatzes ist das sogenannte Bootstrapping [11],[12].

Die Grundidee dieses Verfahrens ist das Erstellen einer beliebigen Anzahl B von Datensätzen D_b durch Ziehen von zufälligen Stichproben mit Zurücklegen aus dem Originaldatensatz D, wobei die Anzahl N der Datenpunkte in D und die Anzahl N_b in D_b gleich ist. Jeder dieser Datensätze D_b wird zum Training der zugehörigen Modellfunktion $f_b(\boldsymbol{u}_b)$ verwendet. Die Validierung erfolgt über den Originaldatensatz D. Damit ergibt sich der Modellfehler

$$\epsilon = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{i=1}^{N} L(y_i, f_b(\boldsymbol{u}_i)) \tag{2.8}$$

Mit der quadratischen Abweichung als Verlustfunktion $L(y, f_k(\boldsymbol{u}))$ ergibt sich:

$$\epsilon_{boot} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_{i,b} - y_i)^2 \tag{2.9}$$

Da beim Ziehen mit Zurücklegen der Stichproben aus dem Originaldatensatz Messpunkte auch mehrfach ausgewählt werden, kommt es beim Bootstrapping zu einer Überanpassung des Modells auf bestimmte Datenpunkte. Der Modellfehler wird daher systematisch zu optimistisch geschätzt.

Zur Lösung dieses Problems wird in [11] vorgeschlagen, bei der Berechnung des Fehlers für den Datenpunkt d_i ein Bootstrap-Datensatz zu verwenden, in dem d_i nicht enthalten ist. Es sei D_{-b} der Bootstrap-Datensatz, in dem d_i nicht enthalten ist und N_{-b} die Anzahl der Datenpunkte in diesem Datensatz. Damit berechnet sich der Modellfehler entsprechend

$$\epsilon_{-b} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N_{-b}}\sum_{j=1, j\neq i}^{N_{-b}} L(y_j, f_b(\boldsymbol{u}_j)). \tag{2.10}$$

Die Methode lehnt sich an die „leave-one-out“-Variante der Kreuzvalidierung an, schätzt den Bias des Modells jedoch zu pessimistisch. Es wird daher ein sogenannter 0.632-Schätzer vorgeschlagen, der den Bootstrap-Modellfehler ϵ_{-b} mit dem Fehler aus dem Training mit allen Datenpunkten gewichtet mittelt:

$$\epsilon_{0.632} = P\{y_i \in D_b\}\epsilon_{-b} + (1 - P\{y_i \in D_b\}\epsilon_N \tag{2.11}$$

mit

$$\epsilon_N = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\boldsymbol{u}_i)). \tag{2.12}$$

Die Wahrscheinlichkeit $P\{y_i \in D_b\}$ mit der ein einzelner Datenpunkt in einem Bootstrap-Datensatz enthalten ist, läuft beim Ziehen mit Zurücklegen für größere Datensätze gegen

$$P\{y_i \in D_b\} = \lim_{N\to\infty}\left(1 - \frac{1}{N}\right)^N = 1 - e^{-1} \approx 0,632. \tag{2.13}$$

Dieser Wert gab dem Schätzer auch seinen Namen. Eine Weiterentwicklung ist der 0.632_+-Schätzer [13], welcher im Fall einer stark überangepassten Modellfunktion $f(\boldsymbol{u})$ bessere Ergebnisse liefert.

Informationskriterien

Eine Alternative zur Kreuzvalidierung und Bootstrapping bei der Auswahl der Modellkomplexität sind Informationskriterien, die einer quantitativen Beurteilung der Güte eines Modells im Vergleich zu seiner Komplexität dienen. Die Grundidee ist, dass ein Modell nach dem Prinzip der Parsimonie nur so komplex wie nötig sein sollte und einfache Modelle komplexen Modellen vorzuziehen sind. Dazu kombinieren Informationskriterien einen Term zur Bewertung der Anpassungsgüte eines Modells, oft als Verlustfunktion $L(\theta)$ definiert, mit der Bewertung der Komplexität C des Modells. Letzteres erfolgt bei den verschiedenen Arten der Informationskriterien über die Anzahl K der Parameter eines Modells.

$$IC = L(\theta) + C(K) \tag{2.14}$$

An dieser Stelle soll auf die beiden am häufigsten verwendeten Informationskriterien, dem Akaikes Informationskriterium (Akaike information criterion, AIC) und dem Bayessches Informationskriterium (Bayesian information criterion, BIC) eingegangen werden. Für eine weiterführende Betrachtung der verschiedenen Arten von Informationskriterien sei auf [14] verwiesen.

Das erste Informationskriterium wurde 1973 von Akaike vorgeschlagen [15]. Er ging von einem statistischem Modell mit einer angenommenen Dichtefunktion $p(\theta)$ und unbekannten Parametern θ aus und konnte zeigen, dass deren negative, maximierte Likelihoodfunktion $-\mathcal{L}(\theta)$ ein verzehrter Schätzer der Kullback-Leibler-Divergenz ist und dass die Verzerrung mit dem Stichprobenumfang gegen die Anzahl der zu schätzenden Parameter konvergiert. Akaike definierte das Informationskriterium als [16], [14]

$$AIC = -2ln(\mathcal{L}(\theta)) + 2K. \tag{2.15}$$

Für den Anwendungsfall der Modellregression betrachtet man zum Modellvergleich die Verteilung der Modellfehler ϵ_i. Ist diese unbekannt, wird häufig von einer Normalverteilung der Modellfehler ausgegangen. Mit

$$g(\epsilon_i|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\epsilon_i-\mu)}{2\sigma}} \tag{2.16}$$

ergibt sich die Likelihoodfunktion zu

$$\mathcal{L}(\epsilon|\mu,\sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N e^{-\frac{\sum_i^N(\epsilon_i-\mu)^2}{2\sigma}} \tag{2.17}$$

Unter der Annahme das σ konstant ist und $\mu = 0$ ergibt sich die maximierte Likelihoodfunktion nach [14] mit

$$\mathcal{L}(\sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N e^{-\frac{1}{2}N} \tag{2.18}$$

und logarithmiert

$$ln(\mathcal{L}(\sigma)) = -\frac{1}{2}N\ln(\sigma^2) - \frac{N}{2}\ln(2\pi) - \frac{N}{2}. \tag{2.19}$$

Da die letzten beiden Terme konstant sind, werden sie zum Modellvergleich in der Regel vernachlässigt und als Maß für die Güte eines Modells wird in vielen Informationskriterien die vereinfachte Form

$$\ln(\mathcal{L}) \approx -\frac{1}{2}N\ln(\sigma^2) \tag{2.20}$$

verwendet. Mit Gleichung 2.15 und 2.20 berechnet sich das AIC für Modelle mit normalverteilten, mittelwertfreien Modellfehlern durch

$$AIC = N\ln(\sigma^2) + 2K \tag{2.21}$$

und mit der Stichprobenvarianz

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2 \tag{2.22}$$

ergibt sich

$$AIC = N\ln\left(\frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2\right) + 2K \tag{2.23}$$

Über den ersten Term in Gleichung 2.23 wird die Modellgüte bewertet. Da bei der Modelloptimierung und der Berechnung des Informationskriterium keine Trennung zwischen Trainings- und Valdierungsdaten stattfindet, wird mit der Erhöhung der Komplexität des Modells eine Überanpassung stattfinden. Dies soll durch den zweiten Term verhindert werden, über den die Anzahl der Parameter des Modells strafend in das AIC einfließen. In der Summe wird es mit steigender Komplexität des Modell ein Minimum des AICs geben, welches das informationstheoretische Optimum der Modellkomplexität darstellt. Bild 2.6 zeigt den prinzipiellen Verlauf des AICs bei steigender Parameteranzahl.

Eine Alternative zum AIC wurde von Schwarz [17] vorgeschlagen, der darauf hinwies, dass der Strafterm im AIC unabhängig von der Anzahl der Beobachtungen sein und damit bei großen Datensätzen komplexe Modelle bevorzugt werden. Im Ergebnis seiner Arbeit schlug er ein Kriterium vor, welches den konstanten Faktor im Strafterm durch einen ersetzte, der

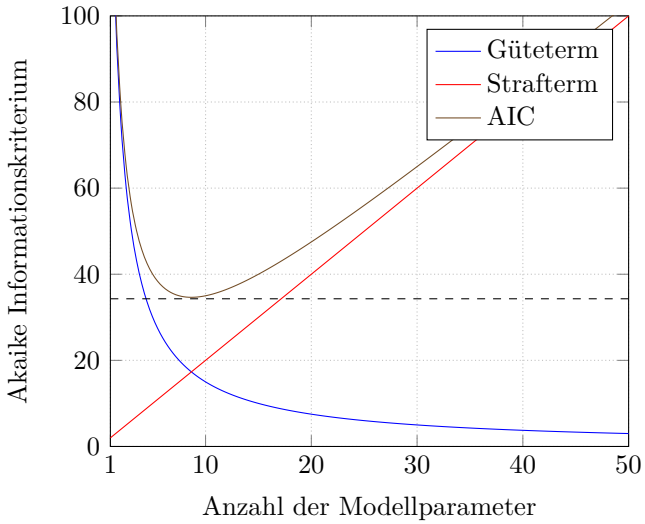

Abbildung 2.6.: Verlauf des AIC, des Güteterms und des Strafterms bei steigender Modellkomplexität

sich mit der Anzahl der Datenpunkte erhöht.

$$BIC = -2ln(\mathcal{L}(\theta)) + \ln(N)K \tag{2.24}$$

Mit normalverteilten, mittelwertfreien Fehlern ergibt sich

$$BIC = N \ln\left(\frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2\right) + \ln(N)K. \tag{2.25}$$

Damit wird im BIC die Anzahl der Parameter schon ab 8 Messpunkten ($\ln(8) \approx 2,079$) stärker bestraft als im AIC. Burnham und Anderson [14] weisen außerdem darauf hin, dass das Ziel bei beiden Kriterien ein anderes ist. Während beim BIC davon ausgegangen wird, dass es ein wahres Modell gibt und dieses zur Auswahl steht, so wird im Ansatz zum AIC davon ausgegangen, dass kein wahres Modell existiert und nur der beste Modellansatz ausgewählt werden soll. Ist das wahre Modell bekannt, so ist das BIC optimal, das heißt mit $N \to 0$ geht die Wahrscheinlichkeit für das wahre Modell gegen 1. Der vermeintliche Vorteil des BIC gegenüber dem AIC wird durchaus kontrovers diskutiert. Eine ausführliche Gegenüberstellung und Diskussion findet man in [18],[19] und [14]: Kapitel 6.4.

Das AIC und BIC beruhen auf der Annahme, dass die Art der Verteilungsdichtefunktion des Modellfehlers bekannt ist. Oben aufgeführte Formeln gehen weiter von der Annahme aus, dass der Modellfehler normalverteilt, mittelwertfrei und bei konstanter Varianz unabhängig von den Eingangsgrößen ist. Dies ist in der Praxis nicht immer gegeben und oft sehr schwer zu überprüfen. Ebenso lässt sich die wahre Verteilungsdichtefunktion des Fehlers oft nicht ermitteln. Sind die Annahmen verletzt, kann es sein, dass die Maximum-Likelihood-Schätzung inkonsistent ist und die Aussagen der Informationskriterien damit unbrauchbar oder diese, im Falle einer konsistenten Schätzung, zumindest ungenau sind. Weiterhin ist zu beachten, dass ein Modellvergleich mit Informationskriterien ausschließlich bei gleichem Datensatz sinnvolle Ergebnisse liefert [14]. Das Hinzufügen von Datenpunkten oder das

Streichen von Ausreißern führt zu verfälschten Ergebnissen. Die Anzahl der Datenpunkte sollte außerdem sehr viel größer als die Anzahl der Parameter des Modells sein.

Zusammenfassung

Die vorgestellten Methoden und Verfahren eignen sich allesamt zur Bewertung des Modellfehlers und zur Bestimmung der optimalen Modellkomplexität. Obwohl alle Verfahren den verfügbaren Datensatz sowohl für die Fehlerbestimmung als auch für das Training des Modells verwenden und damit besser ausnutzen als das klassische Aufteilen, ist ein genügend großer Datensatz essentiell für die datenbasierte Modellierung. Der "Fluch der Dimensionalität"kann so jedoch nur abgemildert, nicht aber umgangen werden.

Die Kreuzvalidierung ist ein robustes und erprobtes Verfahren, welches ohne genaue Kenntnisse des Prozesses und der zur Verfügung stehenden Daten eingesetzt werden kann, während das Bootstrapping zwar eine bessere Performance erzielen kann, in der Praxis aber diverse Fallstricke aufweist und oft eine genaue Analyse der vorhandenen Daten erfordert. In [10] wurden diverse Vergleiche der Kreuzvalidierung und Bootstrapping mit verschiedenen, praktisch relevanten Datensätzen durchgeführt. Dabei wurde die stratifizierte Kreuzvalidierung mit einer moderaten Teilungszahl von $10 \leq K \leq 20$ als robuste Methode für die Modellauswahl mit moderatem Berechnungsaufwand empfohlen.

Das AIC stellt für Prozesse mit normalverteilten, mittelwertfreien Fehlern und konstanter Varianz eine alternative Bewertung der Modellkomplexität da und ist in diesem Fall asymptotisch äquivalent zur „leave-one-out“-Kreuzvalidierung [20]. Im Gegensatz zur „leave-one-out“-Kreuzvalidierung ist der Berechnungsaufwand für das AIC jedoch erheblich geringer und sollte, wenn die Vorbedingungen gesichert erfüllt werden, diesem vorgezogen werden.

Es bleibt anzumerken, dass alle Testverfahren keine Aussage über die Qualität der Datensätze machen. Bei schlecht verteilten Daten oder einer Datenmenge, die für die Komplexität des realen Prozesses nicht ausreicht, erfolgt im besten Falle eine Unteranpassung des Modells mit einem resultierenden großen Modellfehler. Im schlechtesten Fall werden Prozessverläufe partiell fehlerhaft modelliert oder fließen nicht mit in die Bewertung bei der Validierung ein. Anders ausgedrückt, Modellbereiche für die keine Daten vorliegen, werden bei der Validierung über die hier vorgestellten Verfahren nicht berücksichtigt. Infolgedessen können sie einen signifikant besseren Modellfehler vortäuschen als real vorhanden ist.

2.2.2. Modellvalidierung über Prozesswissen

Im letzten Abschnitt wurde auf die Validierung eines optimierten Modells und der Berechnung des Modellfehlers über die verfügbaren Messdaten des Prozesses eingegangen. Mit der oft hohen Eingangsdimensionalität der zu modellierenden Prozesse im Umfeld der Motorsteuerung stellt sich im Zusammenspiel mit dem exponentiell wachsendem Datenbedarf immer wieder das Problem der ungenügenden Datenmenge für die rein datenbasierte Modellierung und Validierung eines Modells. Flankiert wird dieses Dilemma durch begrenzte zeitliche sowie wirtschaftliche Entwicklungsressourcen, welche die Datenausbeute der Messkampagnen entsprechend limitieren. Der im Kapitel 2.1 besprochene „Fluch der Dimensionalität“ lässt das Problem ab ca. 5 Eingangsgrößen soweit ansteigen, dass im hier betrachteten Anwendungsfeld der gesamte Eingangsbereich nicht mehr genügend abgedeckt werden kann. Die im vorherigen Kapitel besprochenen Verfahren sind damit nicht mehr oder nur noch sehr eingeschränkt anwendbar. Unter diesen Voraussetzungen bleibt für die Validierung nur eine

heuristische Überprüfung des optimierten Modells, für dessen praktische Umsetzung allerdings einige Punkte erfüllt sein sollten, deren Details Gegenstand des folgenden Abschnittes sind.

Ziel der heuristischen Überprüfung ist der Test des Modells auf Plausibilität in seinem gesamten Wirkungsbereich. Dabei ist es weniger interessant einzelne Punkte zu testen, die einfacher über einen zusätzlichen Messdatenpunkt adressiert werden können, sondern große Bereiche des Modells anhand einfacher Kriterien zu validieren. Grundsätzlich müssen diese Kriterien zum einen aus Experten- und Prozesswissen heraus ableitbar sein und zum anderen auch in höherdimensionalen Modellstrukturen intuitiv überprüft werden können. In der folgenden Auflistung sind die wesentlichsten Anforderungen an datenbasierte Modellierungsverfahren zur heuristischen Überprüfung erfasst:

1. *Interpretierbarkeit der Parameter des Modells:*
   Datenbasierte Modelle im automotiven Sektor werden in der Regel nicht als reine Blackbox-Modelle entworfen. Oft liegen fundierte Kenntnisse über den Prozess vor und es können prinzipielle Regeln für den grundlegenden Einfluss der Parameter und der Eingangsgrößen angegeben werden. Um diese Regeln in dem datenbasierten Modell überprüfen zu können, ist es wichtig, die physikalisch fundierten Parameter des Prozesses in den Parametern des Modells darstellen zu können.

2. *Interpretierbare Modellstrukturen:*
   Neben der Abbildung der Parameter des Prozesses ist es wünschenswert, dass sich die prinzipiellen Regeln für den grundlegenden Einfluss dieser Parameter in der Modellstruktur abbilden lassen. Veränderungen an den Eingangsgrößen sollten sich einfach und nachvollziehbar in der Ausgangsgröße widerspiegeln.

3. *Unterteilung des Modells in unabhängige Teilbereiche:*
   Komplexe Prozesse unterteilen sich oft funktionell in verschiedene Teilmodelle mit stark unterschiedlichem Verhalten, deren getrennte Betrachtung eine deutliche Vereinfachung gestattet. Ein heuristisch validierbares Modell sollte solch eine Unterteilung strukturell unterstützen beziehungsweise möglich machen. Übergangsbereiche, in denen mehrere dieser Teilmodelle die Ausgangsgröße gleichzeitig beeinflussen, sollten möglichst klein sein und die Art der Verknüpfung mathematisch einfach sein. Um die Unabhängigkeit der Teilbereiche zu gewährleisten, müssen deren Parameter ausschließlich lokal wirken.

4. *Einfache und intuitive Definition der Teilbereichsgrenzen:*
   Die Definition der Teilmodellgrenzen muss einfach, intuitiv und nachvollziehbar erfolgen. Die Grenzen sollten für jede Eingangsgröße getrennt und unabhängig von anderen Eingangsgrößen festgelegt werden können. Eine einfache, unabhängige Teilbereichsdefinition ist mit einer achsenorthogonale Teilung des Eingangsbereiches realisierbar, die sich über einfache UND-Verknüpfungen der Eingangsgrößen darstellen lässt.

5. *Einfache Übergangsfunktionen zwischen den Teilmodellen:*
   Die Aufteilung des Modells in unabhängige Teilbereiche sowie die Forderung nach einem stetigen Verlauf der Ausgangsgröße zwischen diesen Teilmodellen erfordert in der Regel spezielle Übergangsfunktionen. Die Forderung der heuristischen Validierung des Gesamtmodells schließt natürlich auch die Bereiche mit ein, in denen die Übergangsfunktionen gültig sind. In der Konsequenz ergibt sich aus dieser Konstellation

die Forderung nach mathematisch einfachen Übergangsfunktionen, welche die Interpretierbarkeit der Teilmodelle nicht oder nur geringfügig beeinträchtigt.

6. *Inter- und Extrapolationsverhalten:*
   Modellbereiche ohne Messdaten können konzeptbedingt über datenbasierte Verfahren nicht optimiert werden. Das Verhalten, welches das Modell in Bereichen mit wenigen oder keinen Messdaten zeigt, ist dabei von den unterschiedlichen Modellierungsverfahren abhängig. Für den Fall einer heuristischen Validierung des Modells ist es wünschenswert, ein möglichst einfaches, im besten Fall lineares Inter- und Extrapolationsverhalten über diese dünn besetzten Bereiche und in den Randgebieten des Eingangsraumes zu erhalten.

7. *Lokale Optimierung des Modells:*
   Die heuristische Validierung eines Modells ist je nach Komplexität des Prozesses ein aufwendiges Prozedere, welches durch die oben genannte Forderung nach Aufteilung des Modells in unabhängige Teilmodelle, in weniger komplexe Aufgaben zerlegt werden kann. Praktisch wird dabei der Fall eintreten, dass die Modellgüte den Forderungen partiell nicht genügt oder nicht ausreichend validiert werden kann. Die Optimierung der Parameter dieses Teilbereiches, sei es manuell, durch direkte Anpassungen, oder über neue Messdaten für diesen Bereich, sollte dabei keine oder vernachlässigbare Auswirkungen auf die schon validierten Teilmodelle haben. Diese Forderung nach lokaler Optimierbarkeit des Modells schließt die oben genannten Forderungen nach einer unabhängigen Unterteilung der Modellbereiche und einfachen Übergangsfunktionen implizit mit ein.

Mit Modellierungsverfahren, die die genannten Anforderungen erfüllen, ist eine relativ einfache heuristischen Validierung der über die Trainingsdaten erzeugten Modelle möglich. Bei der Modellierung höherdimensionaler Prozesse wird dies in weiten Eingangsbereichen die einzige sinnvolle Validierungsoption sein. Davon unbenommen ist die Validierung über die in Kapitel 2.2.1 diskutierten Verfahren möglich. Diese können bei Erfüllung der Forderung nach der Unterteilung des Modells in unabhängige Teilbereiche, auch für einzelne Teilmodelle, in denen genügend Testdaten zu Verfügung stehen, durchgeführt werden. Datenbasierten Modelle, welche die oben genannten Anforderungen erfüllen, sollen im Folgenden als *interpretierbare* Modelle bezeichnet werden.

2.2.3. Grafisches Tool zur heuristischen Validierung

Im Rahmen dieser Arbeit wurde ein Tool (Bild 2.7) zur grafischen Darstellung hochdimensionaler Modelle entwickelt, welches es erlaubt, bestimmte Arbeitspunkte des Modells gezielt über eine oder zwei Eingangsdimensionen anzuzeigen und durch eine dritte Eingangsdimension zu scrollen. Die Darstellung wird dabei in Echtzeit angepasst, sodass Änderungen im Modell über die dritte Eingangsdimension anschaulich dargestellt werden. Das Tool wurde unter Matlab programmiert und kann für verschiedene Modellstrukturen (LOLIMOT, GMR, ILMON und andere) angepasst werden.

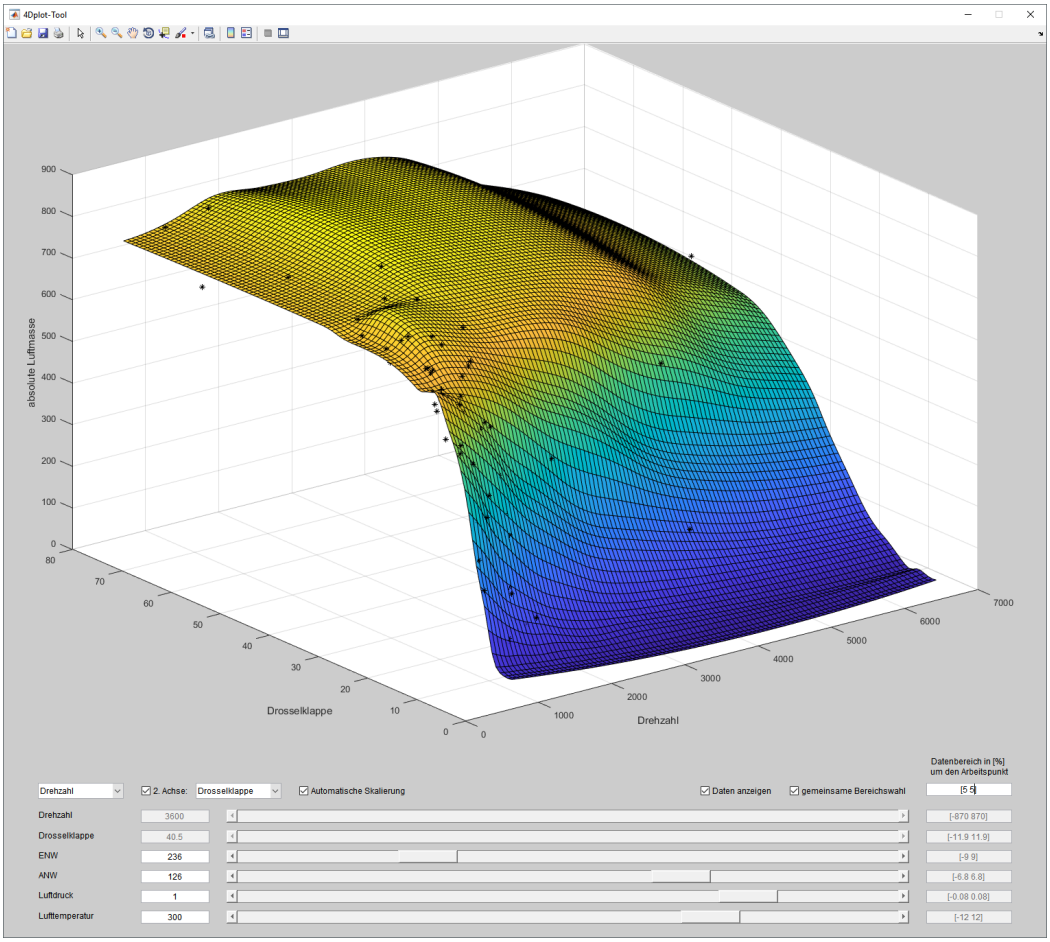

Abbildung 2.7.: 4Dplot-Tool zur grafischen Anzeige hochdimensionaler Modelle

2.3. Datenbasierte Verfahren für den Einsatz in Motorsteuergeräten

Die Notwendigkeit der heuristischen Validierung mit steigender Anzahl von Eingangsdimensionen bei datenbasierten Modellen wurde in Kapitel 2.2.2 aufgezeigt sowie die Anforderungen an interpretierbare Modelle erörtert. Die Eignung bekannter datenbasierter Modellstrukturen und Verfahren für eine interpretierbare Modellierung, soll in diesem Abschnitt genauer untersucht und diskutiert werden.

2.3.1. Kriterien für den Einsatz datenbasierter Verfahren in Motorsteuergeräten

Neben den genannten Kriterien für die Interpretierbarkeit soll nachfolgend die Eignung der Verfahren unter Berücksichtigung der speziellen Erfordernisse in automotiven Anwendungen erörtert werden. Bei der Vielzahl der in diesem Umfeld zu modellierenden Prozesse sind die Anforderungen naturgemäß sehr breit gefächert. Jedoch sind durch die langjährige Entwicklungs- und Forschungsarbeit auf diesem Gebiet, viele Prozesse gut bekannt und

mit hoher Genauigkeit physikalisch modelliert, sodass datenbasierte Modelle auf vergleichsweise wenig Anwendungsfeldern in Frage kommen. Dies sind im allgemeinen hochkomplexe Prozesse, mit einer hohen Anzahl von Eingangsgrößen, bei denen die physikalischen Modellierung nur ungenau oder äußerst rechenintensiv realisierbar ist. Mit dieser Konkretisierung des Einsatzgebietes können einige grundlegende Anforderungen definiert werden:

1. *Eignung für hochdimensionale Eingangsräume*
   In der Regel haben datenbasierten Verfahren zur Modellierung statischer Prozesse keine explizite Beschränkung der Anzahl der Eingangsgrößen. Praktisch skalieren einige Verfahren aber besser als andere mit der Erhöhung der Eingangsdimension, z.B. bezüglich des Datenbedarfs zum Training und Validierung des Modells oder des Ressourcenverbrauchs bei der Anwendung des optimierten Modells. Dies kann verschiedene Ursachen haben, oft ist es durch den „Fluch der Dimensionalität“ begründet.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Im besten Fall liegen die Daten für die Optimierung des Modells gleichmäßig verteilt über den ganzen Eingangsraum vor. In der praktischen Anwendung werden jedoch Hauptarbeitsbereiche eines Verbrennungsmotors mit einer wesentlich höheren Messdatendichte vermessen, als Bereiche, die nur in seltenen Konstellationen durchfahren werden. Einige Modellbereiche werden ganz ausgeschlossen. Mit dieser Clusterung der Messdaten und dem Vorkommen großer Gebiete mit wenig oder gar keinen Messdaten sollte ein Verfahren umgehen können beziehungsweise ein definiertes, vorhersagbares Verhalten zeigen.

3. *Möglichkeit der Integration von a-priori-Wissen*
   Datenbasierte Verfahren sind aus dem systemtheoretischen Blickwinkel betrachtet in Regel keine reinen Blackbox-Modelle. Die Parameter des Modells werden zwar im Allgemeinen über die Trainingsdaten optimiert, Struktur und Komplexität werden jedoch im Vorfeld anhand bestimmter Kriterien festgelegt. Viele Verfahren erlauben hier, über Startwerte und Designparameter weiteres Prozesswissen in dedizierter Form einfließen zu lassen. Weiterhin gestatten einige Verfahren, einzelne Modellparameter manuell festzulegen und von der Optimierung auszuschließen, z.B. weil das Prozessverhalten in Teilbereichen gut bekannt ist. Diese Möglichkeiten Prozesswissen im Vorfeld der Optimierung einfließen zu lassen, ist im Bereich der automotiven Anwendung ein wichtiger Aspekt und mögliche Modellierungsverfahren sollten dies umfassend unterstützen.

4. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Wie in Abschnitt 2.2 dargelegt, hängt der minimal erreichbare Modellfehler auch von der Anzahl der zur Verfügung stehenden Daten ab. Konträr zu der Forderung eines hochgenauen Modells auf Basis eines großen Datensatzes besteht aus wirtschaftlichen Gründen oft der Wunsch, den Aufwand für Messkampagnen zu reduzieren, die Datenanzahl zu verringern und Modelle nur in den Bereichen hochgenau zu optimieren, die im Regelbetrieb relevant sind. Unter diesem Aspekt ist die Möglichkeit wünschenswert, verschiedene Modellbereiche mit verschiedenen Gütevorgaben zu optimieren.

5. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Die a-priori-Vorgabe der Modellkomplexität vor der Optimierung des Modells ist bei den meisten Verfahren ausschlaggebend für den erreichbaren Bias und Varianzfehler

des Modells, siehe Kapitel 2.2. Mit den vorgestellten Verfahren der Modellvalidierung (Kapitel 2.2.1) kann die Modellkomplexität zwar optimiert werden, dies erfordert jedoch, bedingt durch die Vielzahl an notwendigen Optimierungsdurchläufen, einen erhöhten Rechenaufwand. Verfahren, welche die Komplexität des Modells auf Grundlage der Datenbasis innerhalb des Optimierungsprozess anpassen, sind hier im Vorteil.

6. *Geringer Ressourcenbedarf der Schätzgleichung*
   Ziel der meisten Modellierung ist der Einsatz des optimierten Prozessmodells im Fahrzeug und damit die Berechnung der Ausgangsgröße auf einem Steuergerät. Durch die Entwicklung der letzten Jahre werden heutzutage zwar wesentlich leistungsfähigere Steuergeräte eingesetzt, jedoch reizt der Ressourcenbedarf der immer komplexer werdenden Motorsteuerungen und Diagnosefunktionen diese weiterhin stark aus. Unter diesen Voraussetzungen sind Modellierungsverfahren mit geringem Speicherbedarf und moderater Prozessorbelastung zu bevorzugen.

7. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Die Entwicklung neuer Motoren ist im Allgemeinen eine Weiterentwicklung vorhandener Aggregate und Technologien, von denen bestehenden Lösungen übernommen und nur in Teilbereichen verändert werden. Auf die datenbasierte Modellierung übertragen bedeutet dies, dass vorhandene Modelle für neue Projekte adaptiert sowie an die neuen Gegebenheiten angepasst werden. Es soll beurteilt werden, wie gut ein Verfahren eine solche Adaption unterstützt.

Im folgenden Kapitel werden einige Verfahren der datenbasierte Modellierung kurz vorgestellt und hinsichtlich ihrer Eignung nach den aufgeführten Kriterien diskutiert.

2.4. Eigenschaften ausgewählte, datenbasierte Modellierungsverfahren

Für die Modellierung und Identifikation nichtlinearer, statischer Prozesse existieren eine Vielzahl von Verfahren, deren Auswahl nach den gewünschten Eigenschaften bezüglich des konkreten Anwendungsfall stattfindet. In diesem Kapitel sollen einige ausgewählte Verfahren vorgestellt und deren Eigenschaften diskutiert werden. Ein grundlegender systematischen Überblick über statische nichtlineare Verfahren ist in [21] und [8] zu finden. Eine umfassende Übersicht und systematische Einteilung sowie die Darstellung der Vor- und Nachteile verschiedener multipler Modellformen findet sich in [22].

2.4.1. Basisfunktionsmodell

Es sei $f(\boldsymbol{x}, \theta)$ eine statische, nichtlineare Funktion, welches den Eingangsvektor $\boldsymbol{x} \in \mathbb{R}^{q \times 1}$ auf die skalare Ausgangsgröße $\hat{y} \in \mathbb{R}$ abbildet. In den meisten praktisch relevanten Methoden kann die Funktion $f(\boldsymbol{x}, \theta)$ als gewichtete Summe sogenannter Basisfunktionen g_k folgendermaßen formuliert werden [21]:

$$f(\boldsymbol{x}, \theta) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{x}, \beta_k) g_k(\boldsymbol{x}, \gamma_k). \tag{2.26}$$

Die Funktion $\alpha_k(\cdot)$ in dieser Gleichung wird oft als linear in ihren Parametern definiert, sodass die Parameter β_k über lineare Optimierungsmethoden geschätzt werden können, falls die Parameter γ_k der Basisfunktion bekannt sind. In [8] erfolgt weiterhin eine Unterscheidung der Basisfunktionen nach der Reichweite ihrer Wirkung auf die Ausgangsgröße:

- Globale Basisfunktionen beeinflussen die Ausgangsgröße signifikant über den gesamten Eingangsraum. Bei streng globalen Basisfunktionen ist auch die Ableitung global wirksam.
- Lokale Basisfunktionen beeinflussen die Ausgangsgröße nur in einem begrenzten Bereich des Eingangsraumes signifikant und außerhalb dieses Einflussbereiches gilt $g_k \approx 0$. Dieses Kriterium ist bei einer geringen Anzahl K von Basisfunktionen und abhängig von den optimierten Parametern oft etwas unscharf. Als lokal werden Basisfunktionen daher auch dann definiert, wenn sie explizit dem Zweck dienen, den Wirkungsbereich im Eingangsraum klar abzugrenzen. Für streng lokale Basisfunktionen gilt außerhalb ihres Wirkungsbereiches $g_k = 0$ und beeinflussen damit die Ausgangsgleichung an diesen Stellen nicht.

Beispiele für verbreitete Verfahren, die in der Form nach Gleichung (2.26) formuliert werden können, sind Look-up-Tabellen, Polynommodelle, neuronale und neuro-fuzzy Netze, Gaussian-Mixture-Regression oder die Support-Vektor-Regression, auf welche nun kurz eingegangen werden soll. Die Gaussian-Mixture-Regression und der LOLIMOT-Algorithmus werden in den nachfolgenden Kapiteln genauer untersucht.

2.4.2. Polynommodelle

Die Approximation eines Prozesses über eine Polynomfunktion ist eine einfache und weit verbreitete Form der Modellbildung. Die Polynomfunktion l-ten Grades einer q-dimensionalen Eingangsgröße $\boldsymbol{x} = [x_1, x_2, \cdots, x_q]$ lautet

$$\hat{y} = c_0 + \sum_{i_1=1}^{q} c_i x_i + \sum_{i_1=1}^{q} c_{i_1,i_2} x_{i_1} x_{i_2} + \cdots + \sum_{i_1=1}^{q} \cdots \sum_{i_l=i_{l-1}}^{q} c_{i_1 \ldots i_l} x_{i_1} \cdots x_{i_l}. \tag{2.27}$$

c_0 beschreibt den Gleichanteil des Modells, die erste Summe den linearen Anteil und alle weiteren Summen die Anteile höherer Ordnung in aufsteigender Reihenfolge. Die Gesamtzahl der Koeffizienten $c_0, c_i, \ldots, c_{i_1 \ldots i_l}$ ergibt sich nach [8],[23] wie folgt:

$$K = \frac{(l+q)!}{l!q!} \tag{2.28}$$

Die Produkte der Eingangsgrößen in Gleichung (2.27) können auch als Basisfunktionen nach Gleichung (2.26), mit den Koeffizienten als α_k, angesehen werden. Damit lässt sich die Polynomgleichung auch als Summe von Basisfunktionen formulieren [8]

$$\hat{y} = \sum_{j=0}^{K} \alpha_j \tilde{x}_j \quad \tilde{x}_0 = 1 \tag{2.29}$$

mit $\alpha_j \tilde{x}_j$ als j-ter Term in Gleichung (2.27). Mit dem Datensatz $\boldsymbol{D} \in \mathbb{R}^{N \times q}$ ergibt sich somit die Problemstellung, die Koeffizienten $\alpha_0, \alpha_1, \ldots, \alpha_K$ optimal für die angenommene Poly-

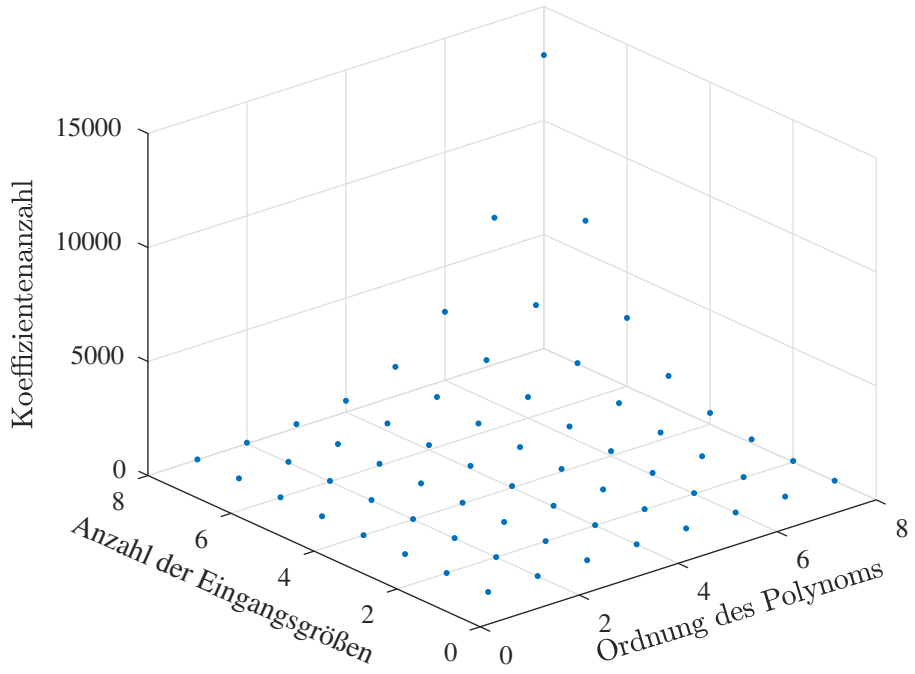

Abbildung 2.8.: Entwicklung der Koeffizientanzahl bei steigender Anzahl an Eingangsgrößen und steigender Ordnung des Polynoms

nomstruktur zu bestimmen. Dies kann als lineares Optimierungsproblem mit der Methode der kleinsten Quadrate gelöst werden. Die zugehörige Designmatrix lautet

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{q,1} & x_{1,1}^2 & \dots & x_{q,1}^l \\ 1 & x_{1,2} & \cdots & x_{q,2} & x_{1,2}^2 & \dots & x_{q,2}^l \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,N} & \cdots & x_{q,N} & x_{1,N}^2 & \dots & x_{q,N}^l \end{bmatrix} \in \mathbb{R}^{N \times K} \tag{2.30}$$

und der Koeffizientenvektor

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{K-1} \end{bmatrix}^T \in \mathbb{R}^{K \times 1}. \tag{2.31}$$

Im Folgenden sollen die Eigenschaften im Hinblick der aufgestellten Kriterien diskutiert werden:

1. *Eignung für hochdimensionale Eingangsräume*
   Wie aus Gleichung (2.28) ersichtlich, steigt die Anzahl der Koeffizienten und damit die Modellkomplexität rapide mit der Anzahl der Eingangsgrößen und der Ordnung des Polynoms, Bild 2.8 veranschaulicht dies. Dieser Nachteil kann durch die Beschränkung auf die Terme reduziert werden, welche einen relevanten Einfluss auf die Ausgangsgröße haben. Trotz solcher Maßnahmen ist der Einsatz eines Polynommodells für komplexe, hochdimensionale Prozesse stark limitiert.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Polynomfunktionen neigen bei hohen Ordnungen zu einem oszillierendem Interpolationsverhalten und streben außerhalb des Trainingsbereiches umso schneller gegen $\pm\infty$, je höher die Ordnung des Polynoms ist. Praktisch bedeutet dies, dass gerade in den Randbereichen eines Modells Trainingsdaten vorhanden sein müssen, welche sehr häufig nicht den Hauptwirkungsbereichen eines Modells entsprechen.

3. *Möglichkeit der Integration von a-priori-Wissen*
   Sind prinzipielle physikalische Zusammenhänge zwischen Eingangsgrößen und Ausgangsgröße bekannt, können die korrespondierenden Terme der Polynomfunktion explizit ausgewählt beziehungsweise nicht relevante Terme entfernt werden. Dieses Vorgehen ist nur global für den gesamten Eingangsbereich möglich, Vorgaben für bestimmte Modellbereiche können nicht gemacht werden.

4. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Die Approximationsfähigkeit der Polynomfunktion ergibt sich ausschließlich über die Definition der Polynomordnung beziehungsweise der Anzahl der Terme. Diese Vorgabe ist global gültig, eine lokale Anpassung ist nicht möglich.

5. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Zur Schätzung der Koeffizienten des Polynoms ist die Vorgabe der Polynomordnung beziehungsweise der gewünschten Terme erforderlich. Verfahren zur Auswahl der relevanten Regressoren können diese Vorgabe zwar anhand der Messdaten vornehmen, sind jedoch an gewisse statistische Voraussetzungen gebunden, die in der Praxis nicht immer einzuhalten sind.

6. *Ressourcenbedarf der Schätzgleichung*
   Der Berechnungsaufwand der Ausgangsgleichung und der Speicherbedarf ist sehr gering und somit gut geeignet für den Einsatz in Motorsteuergeräten.

7. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Sind die relevanten Terme eines Polynoms für eine Applikation gefunden und physikalisch begründet, so ist in den meisten Fällen die Übernahme dieser Struktur für einen ähnlichen Anwendungsfall problemlos möglich. Die globale Gültigkeit bedingt, das die Koeffizienten dafür komplett neu optimiert werden müssen, was auch einen vollständigen Datensatz notwendig macht. Ist das Polynommodell als Blackbox-Modell entworfen, haben also die Koeffizienten keinen physikalischen Bezug, kann die Modellstruktur nicht ohne weiteres als identisch angenommen werden.

Zusammenfassung: Polynommodelle bestechen durch ihre einfache, intuitive Handhabung und sind auf Grund ihres geringen Ressourcenbedarfs bei weniger komplexen Prozessen mit 1 bis 2 Eingangsgrößen oft eine gute Wahl. Ihr problematisches Inter- und Extrapolationsverhalten und die globale Wirksamkeit der Parameter lassen eine heuristische Validierung in höherdimensionalen, komplexen Prozessen jedoch nicht zu.

2.4.3. Lookup-Tabelle

Rasterbasierte Lookup-Tabellen mit ein bis zwei Eingangsgrößen sind die im Motorsteuergerät am häufigsten genutzte Modellform für die Abbildung nichtlinearer Prozesse. Dies ist

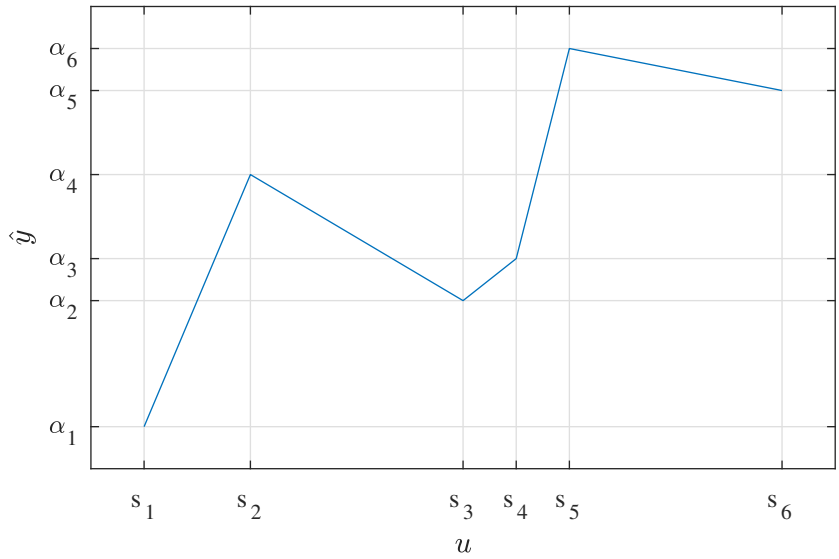

Abbildung 2.9.: Verlauf der Ausgangsgröße einer Lookup-Tabelle mit 6 Stützstellen s_k und den jeweiligen Höhen α_k

zum einen auf die einfache Implementierbarkeit und dem sehr geringen Rechenaufwand als auch auf die sehr gute intuitive Applizierbarkeit zurückzuführen. In rasterbasierten Lookup-Tabellen wird in jeder Eingangsdimension ein festes Raster an Stützstellen vorgegeben und für jede Stützstelle s_k eine Höhe α_k als Parameter definiert, wobei das Raster nicht äquidistant sein muss. Zwischen den Stützstellen erfolgt die Berechnung der Ausgangsgröße mittels Interpolation. In Abbildung 2.9 ist der Verlauf einer Lookup-Tabelle mit einer Eingangsgröße und 6 Stützstellen beispielhaft dargestellt.

Die Ausgangsgleichung einer eindimensionalen Lookup-Tabelle ergibt sich in der Basisfunktionsschreibweise als

$$\hat{y} = \sum_{k=1}^{K} \alpha_k g_k(u, \boldsymbol{s}) \tag{2.32}$$

mit den Höhen α_k an den Stützstellen $\boldsymbol{s} = \{s_1, s_2, \cdots, s_K\}$. Der Verlauf der Basisfunktionen $g_k(u, \boldsymbol{s})$ des im Bild 2.9 gezeigten Ausgangsverlaufes mit linearer Interpolation zwischen den Stützstellen, ist in Bild 2.10 dargestellt.

Die Gesamtzahl der Stützstellen einer q-dimensionalen Lookup-Tabelle ergibt sich mit der Anzahl der Stützstellen K_i in jeder Eingangsdimension zu

$$K = \prod_{i=1}^{q} K_i. \tag{2.33}$$

Wie an Gleichung (2.33) zu sehen ist, wächst die Gesamtzahl der Stützstellen multiplikativ mit der Anzahl der Stützstellen jeder weiteren Eingangsdimension, siehe auch Bild 2.1. Der „Fluch der Dimensionalität“ kommt hier voll zur Geltung. Ist der Prozess stark nichtlinear und werden somit zur genauen Abbildung eine große Anzahl von Stützstellen pro Eingangsgröße benötigt, sind mehr als 3 Dimensionen in der Praxis kaum handhabbar.

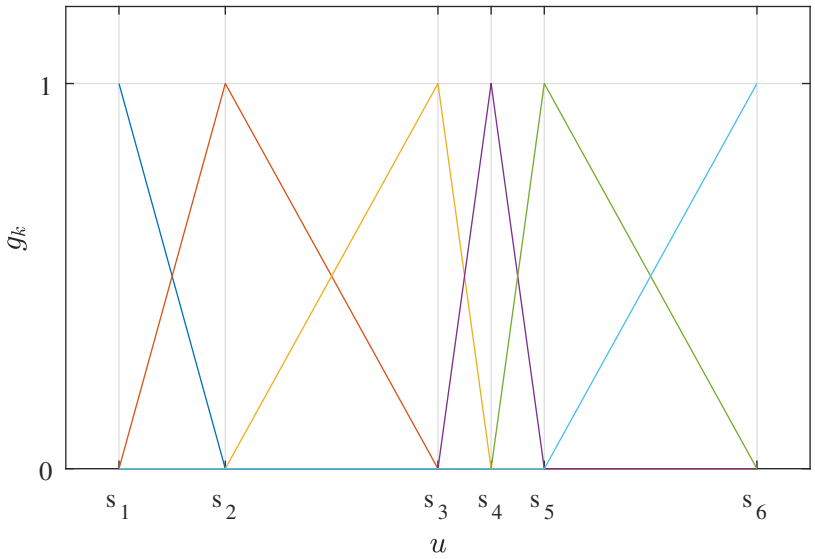

Abbildung 2.10.: Verlauf der Basisfunktionen g_k mit den Stützstellen s_k der Lookup-Tabelle aus Bild 2.9

Zur Interpolation können verschiedene Methoden verwendet werden. Angefangen von der linearen Interpolation über eine stückweise Polynominterpolation verschiedener Grade bis hin zu verschiedenen Arten von Splines. Einen Überblick über die Methoden und weiterführenden detaillierte Ausführungen hierzu findet man in [24], [25] und [26].

Auf Grund der stark eingeschränkten Anwendbarkeit für Prozesse mit mehr als 3 Eingangsgrößen soll an dieser Stelle nicht weiter auf die Modellierung mit Lookup-Tabellen eingegangen werden. Eine detaillierte Betrachtung unter dem Gesichtspunkt der Anwendung in Motorsteuergeräten findet man in [27] und [8].

Die Eigenschaften hinsichtlich der aufgestellten Kriterien sind:

1. *Eignung für hochdimensionale Eingangsräume*
   Durch den rasterbasierten Ansatz kommt der „Fluch der Dimensionalität“ bei Lookup-Tabellen voll zum Tragen, die Eignung für hochdimensionale Eingangsräume ist daher sehr gering. In praktischen Anwendungen sind kaum mehr als 3 Eingangsdimensionen realisierbar.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   In der Regel sollten für alle Stützstellen der Tabelle mindestens eine Messung aufgenommen werden. Für einzelne Stützstellen und kleinere Bereiche ohne Messdaten können die Parameter durch die Methode der kleinsten Quadrate und durch Verfahren der Regularisierung aus den umliegenden Daten geschätzt werden. Durch die streng lokale Struktur der Lookup-Tabellen sind diesem Vorgehen allerdings enge Grenzen gesetzt.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Die Parameter einzelner Teilbereiche der Tabelle können unabhängig voneinander und mit unabhängigen Gütekriterien geschätzt werden.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*

Die Anzahl der Stützstellen wird im Vorfeld der Optimierung fest vorgegeben, die Modellkomplexität wird nicht optimiert.

5. *Ressourcenbedarf der Schätzgleichung*
   Der Aufwand für die Berechnung der Ausgangsgröße ist hauptsächlich vom gewählten Interpolationsverfahren abhängig und ist bei den typischen Verfahren sehr gering. Der Speicherbedarf ist linear abhängig von der Anzahl der Stützstellen.

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Die Möglichkeit der unabhängigen lokalen Anpassung der Parameter ohne Beeinflussung der umliegenden Modellbereiche bietet gute Voraussetzungen, bestehende Lookup-Tabellen für modifizierte Anwendungen zu adaptieren. Insbesondere wenn die Verteilung der Stützstellen ganz oder zum großen Teil übernommen werden kann, sind hier Anpassungen schnell und intuitiv möglich.

Zusammenfassung: Lookup-Tabellen sind eine einfache Möglichkeit, nichtlineare Prozesse intuitiv und ressourcenschonend abzubilden. Sie lassen sich einfach lokal-unabhängig anpassen und für ähnliche Prozesse adaptieren. Durch den massiven Einfluss des „Fluch der Dimensionalität“, sind sie jedoch für eine Modellierung mit mehr als 3 Eingangsgrößen nur stark eingeschränkt geeignet.

2.4.4. Künstliche Neuronale Netze

Die Methodenklasse der künstlichen neuronalen Netze (KNN) hat in den letzten Jahren im Bereich der künstlichen Intelligenz und der Big-Data-Anwendungen große mediale Aufmerksamkeit erfahren. Oft werden die KNN dabei synonym mit der Klassifikation von Objekten, wie die Erkennung von Objekten in Bildern, gleichgesetzt. In dieser Arbeit soll sich allerdings auf den Anwendungsbereich der Regression beschränkt werden sowie auf die in diesem Bereich am häufigsten verwendeten Klassen den Multilayer-Perzeptron-Netze und den neuronalen Netzen mit radialen Basisfunktionen (RBF) [8].

Basis und Grundelemente eines KNN sind die Neuronen und die gerichteten Verbindungen zwischen diesen. Ausgehend von der McCulloch-Pitts-Zelle, der ersten Definition eines künstlichen Neurons, gibt es je nach Verwendungszweck sehr viel unterschiedliche Umsetzungen. Abgeleitet vom biologischen Vorbild ist allen Varianten gemein, dass mehrere Eingangsgrößen zu einem Ausgangsvektor verarbeitet werden können. Eine weit verbreitete Definition ist die des sogenannten Perzeptrons, welches in den Multilayer-Perzeptron-Netzen Anwendung findet. Diese sollen nachfolgend näher beschrieben werden.

Multilayer-Perzeptron-Netze

Nach der klassischen Definition eines Neurons besteht dieses, wie in Bild 2.11 dargestellt, aus drei Grundbestandteilen [28]. Die Propagierungsfunktion

$$y_{net} = f_{prop}(\boldsymbol{u}) \tag{2.34}$$

nimmt die Eingänge entgegen und verarbeitet diese zur Netzeingabe y_{net}. Die Aktivierungsfunktion

$$y_{akt} = f_{akt}(y_{net}) \tag{2.35}$$

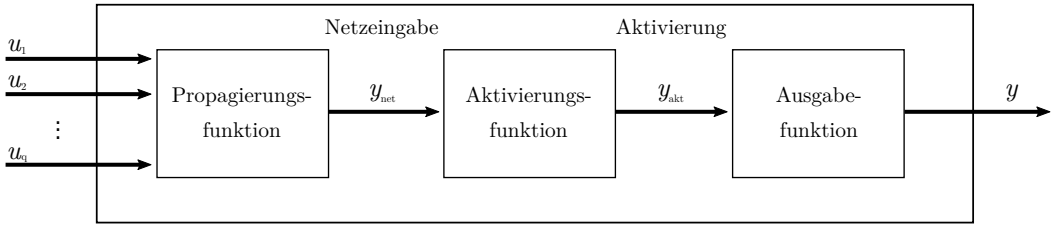

Abbildung 2.11.: Aufbau eines künstlichen Neurons

erzeugt aus der Netzeingabe die Aktivierung y_{akt}, welche die Ausgabefunktion

$$y = f(y_{akt}) \tag{2.36}$$

zum Ausgabewert y verarbeitet. Die einzelnen Neuronen können nun über gerichtete Verbindungen zusammengeschaltet werden, wobei als Eingangsgrößen der einzelnen Neuronen sowohl die Eingangsgrößen des Prozesses als auch die Ausgänge anderer Neuronen möglich sind. Alle Neuronen mit Prozesseingängen als Eingangsgrößen werden zur Eingangsschicht zusammengefasst. Neuronen mit einer Prozessgröße als Ausgang werden in der Ausgangsschicht angeordnet. Alle weiteren Neuronen werden entsprechend der Richtung der Verbindungen in sogenannten verdeckten Schichten strukturiert, siehe Bild 2.12.

Als Propagierungsfunktion wird oft die gewichtete Summe der Eingangsgrößen verwendet. Sie berechnet sich mit

$$y_{net} = \boldsymbol{w}\boldsymbol{u}, \quad \boldsymbol{w} = [w_1, w_2, \cdots, w_q], \quad \boldsymbol{u} = [u_1, u_2, \cdots, u_q]^T \tag{2.37}$$

mit den Gewichten $\boldsymbol{w}$ und dem Vektor der Eingangsgrößen $\boldsymbol{u}$.

Die einfachste Aktivierungsfunktion ist die binäre Schwellenwertfunktion, welche in Abhängigkeit vom Ausgang zwischen zwei konstanten Werten umschaltet. Der Nachteil der fehlenden Differenzierbarkeit am Schwellenwert, die für den Backpropagation-Algorithmus als Lernmethode Voraussetzung ist, wird durch die Verwendung einer Sigmoidfunktion wie z.B. der logistischen Funktion oder der Tangens-hyperbolicus-Funktion vermieden, welche in der Praxis sehr häufig eingesetzt werden [28]. Mit der Erweiterung der logistischen Funktion durch einen Parameter T

$$y_{akt}(y_{net}, T) = \frac{1}{1 + e^{\frac{-ynet}{T}}}, \tag{2.38}$$

kann diese der binären Schwellenwertfunktion beliebig angenähert werden.

Über die Ausgabefunktion kann die Aktivierung nochmals angepasst werden. Da diese mathematische Operation prinzipiell auch schon in der Aktivierungsfunktion erfolgen kann, wird die Ausgabefunktion eher selten genutzt und mit $y = y_{akt}$ meist als Identität definiert.

Die Anzahl der Neuronen und die Struktur der Verbindungen eines MLP-Netzes lassen sich, wie in Bild 2.12 angedeutet, beliebig festlegen. Da MLP-Netze zur Klasse der universellen Approximatoren gehören [29], lassen sich mit der Erhöhung der Anzahl der Neuronen und der verdeckten Schichten beliebig komplexe nichtlineare Funktionen abbilden. Durch die freie Ausrichtung der sigmoiden Aktivierungsfunktionen und deren Kombination in den verdeckten Schichten ist die durch die Datenpunkte angelernte Funktion sehr schwer inter-

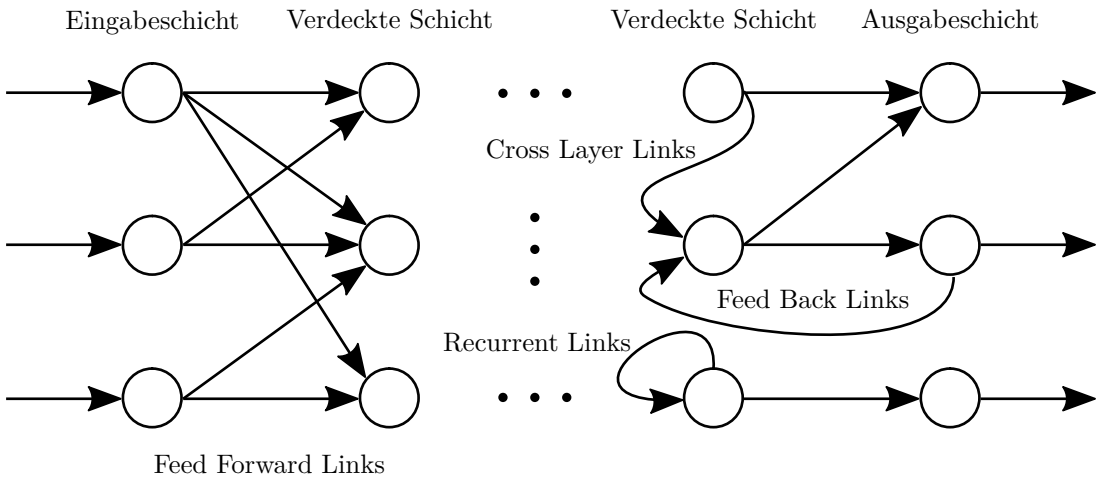

Abbildung 2.12.: Netzstruktur eines künstlichen, neuronalen Netzes

pretierbar. Ebenso lässt sich das Interpolationsverhalten kaum aus den angelernten Parametern ablesen. Nachteilig ist weiterhin, dass die hohe Anzahl an Freiheitsgeraden bei der Wahl der Struktur und der Anzahl der Neuronen eine systematische Festlegung der notwendigen Komplexität anhand der Messdaten oder der a-priori-Kenntnisse über den zu modellierenden Prozess kaum zulässt. In der Praxis wird daher für Anwendungen in der Regression und bei spärlich mit Messdaten besetzten Eingangsräumen oft nur ein MLP-Netz mit einer verdeckten Schicht aus gleichartigen Neuronen und einem einzelnen Ausgangsneuron verwendet [8]. Die Ausgangsgleichung solch eines Modells lautet:

$$\hat{y} = \sum_{k=1}^{K} v_k \phi_k \left(y_{net} \right) \tag{2.39}$$

mit der Netzeingabe y_{net} nach Gleichung 2.37, der Aktivierungsfunktion $\phi = y_{akt}$ und den Gewichten v_k als Ausgabefunktion. Das Ausgangsneuron bildet die Summe über alle Ausgänge der einzelnen Neuronen. Die Struktur solch eines Netzes ist in Bild 2.13 abgebildet. Vorteil dieser vereinfachten Struktur ist die Skalierbarkeit der Komplexität zur Verringerung des Bias ausschließlich über die Erhöhung der Neuronenanzahl. Eine Änderung der Netzstruktur ist nicht notwendig. Die Modellkomplexität kann so quantitativ in Abhängigkeit des Modellfehlers festgelegt werden.

Die Optimierung der Parameter eines MLP-Netzes erfolgen im Allgemeinen nichtlinear, zum Beispiel über den Backpropagation-Algorithmus. Die üblichen Probleme der nichtlinearen Optimierung bezüglich der zu wählenden Initialisierungsparameter als auch der Optimierung auf ein lokales Optimum kommen dabei, verstärkt durch die hohe Komplexität der automotiven Prozesse, voll zum Tragen. Insbesondere die mangelnde Interpretierbarkeit der Parameter des MLP-Netzes lässt eine physikalisch begründete Wahl der Startparameter der Optimierung nicht zu, sodass üblicherweise eine zufällige Festlegung erfolgt.

Die Eigenschaften hinsichtlich der aufgestellten Kriterien lassen sich wie folgt zusammenfassen:

1. *Eignung für hochdimensionale Eingangsräume*
   Die Flexibilität der MLP-Netze lassen eine gute Anpassung auch in hochdimensiona-

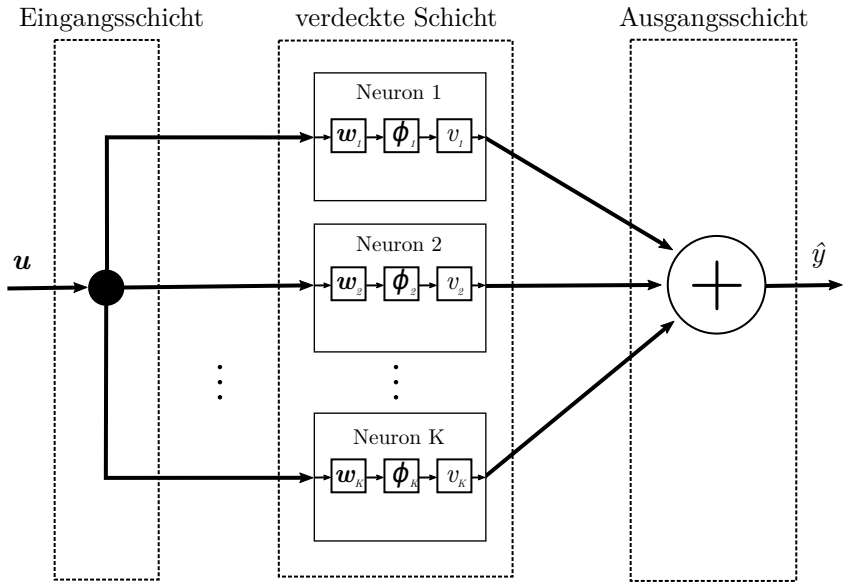

Abbildung 2.13.: Struktur eines skalierbaren MLP-Netzes mit einer verdeckten Schicht und der Summe als Ausgangsneuron

len Eingangsräumen zu. Allerdings machen die fehlende physikalische Interpretierbarkeit der Modellparameter eine systematische Festlegung der Struktur des Netzes sehr schwierig, sodass diese oft nur per Erfahrungswissen festgelegt werden kann.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Das komplexe, schwer kontrollierbare Interpolationsverhalten eines MLP-Netzes machen eine hohe Anzahl von Messdaten mit einer gleichmäßigen Abdeckung des Eingangsraumes, insbesondere bei komplexen Strukturen, notwendig.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Durch die globale Wirksamkeit der einzelnen Neuronen auf den Modellausgang und die komplexen Überlagerungen der Bereiche einzelner Neuronen können keine lokalen Modellbereiche gesondert betrachtet und optimiert werden.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Eine automatische Anpassung der Struktur des Netzwerkes auf Grundlage des aktuellen Modellierungsfehlers lässt sich nicht umsetzen. Bei einer manuellen Definition einer Struktur, insbesondere der vereinfachten Struktur nach Gleichung (2.39) kann aber die Anzahl der Neuronen schrittweise erhöht werden.

5. *Ressourcenbedarf der Schätzgleichung*
   Der Ressourcenbedarf hängt in erste Linie von der Anzahl der Neuronen und der Wahl der Aktivierungsfunktion ab. Die logistische Funktion sowie die Tangens-hyperbolicus-Funktion ist in Motorsteuergeräten nur mit hohem Aufwand zu berechnen. Spezielle Aktivierungsfunktionen, welche über Parabelbögen definiert sind, lassen allerdings auch eine resourcenschonende Berechnung in Motorsteuergeräten zu [30].

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Eine einmal gefundene Struktur eines MLP-Netzes sowie die Anzahl der Neuronen

lassen sich problemlos auf gleichartige Prozesse übertragen. Die Optimierung der Parameter des Netzes muss dagegen komplett neu durchgeführt werden. Dies gilt auch für die Erweiterung der Messdaten mit neuen Datenpunkten.

Zusammenfassung: MLP-Netze sind durch ihre Flexibilität sehr gut für hochdimensionale Prozesse geeignet, stellen jedoch hohe Ansprüche an die Qualität und Quantität der Trainingsdaten. Weiterhin sind aufgrund der Eigenschaften der nichtlinearen Optimierung, die optimalen Parameter des Netzes nur sehr aufwendig zu ermitteln. Die Interpretierbarkeit ist stark limitiert und a-priori-Wissen lässt sich nur schwer in den Modellierungsprozess integrieren. Das Interpolationsverhalten bei Netzen mit nur einer verdeckten Schicht und sigmoiden Aktivierungsfunktionen ist tendenziell monoton, bei mehreren verdeckten Schichten oder komplexen Aktivierungsfunktionen ändert sich dies zu nicht-monotonen bis oszillierenden Verläufen. Insgesamt sind MLP-Netze für die Modellierung automotiver Prozesse und zur Anwendung in Motorsteuergeräten nur bedingt geeignet.

Neuronale Netze mit radialen Basisfunktionen

Radiale Basisfunktionen wurden schon außerhalb des Kontextes der neuronalen KKNs zur Aproximation von Funktionen eingesetzt, insbesondere wenn die Datenbasis nicht rasterbasiert vorliegt [31], [32]. Die Standardform ist dabei eine Summe von K Radialbasisfunktionen φ mit den Zentren $\boldsymbol{c}_k$ und den Gewichten w_k, über welche die zu approximierende Funktion angenähert wird.

$$\hat{y} = \sum_{k=1}^{K} w_k \varphi(\|\boldsymbol{u} - \boldsymbol{c}_k\|, \theta) \tag{2.40}$$

Typischerweise sind die Parameter der Radialbasisfunktionen bis auf die Zentren gleich. Als Norm $\|\cdot\|$ wird häufig die Mahalanobis-Distanz, die euklidische Distanz oder die standardisierte euklidische Distanz gewählt [8]. Letztere sind zwar weniger flexibel, haben jedoch bei einer hohen Anzahl an Eingangsgrößen den Vorteil der erheblich reduzierten Parameteranzahl der RBFs. Alle drei Distanzen berechnen sich nach folgender Gleichung

$$\|\boldsymbol{u} - \boldsymbol{c}_k\|_{\boldsymbol{S}_k} = \sqrt{(\boldsymbol{u} - \boldsymbol{c}_k)^T \boldsymbol{S}_k (\boldsymbol{u} - \boldsymbol{c}_k)} \tag{2.41}$$

und unterscheiden sich in der Wahl der symmetrischen Normierungsmatrix $\boldsymbol{S}$. Für die euklidische Distanz wird hier die Einheitsmatrix gewählt, während es bei der Mahalanobis-Distanz die inverse, vollbesetzte Kovarianzmatrix $\boldsymbol{\Sigma}^{-1}$ ist. Die standardisierte euklidische Distanz für q Eingangsgrößen ergibt sich mit einer Diagonalmatrix als Normierungsmatrix

$$\boldsymbol{S}_k = \begin{bmatrix} \frac{1}{\sigma_{k,1}^2} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_{k,2}^2} & \dots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{\sigma_{k,q}^2} \end{bmatrix}, \tag{2.42}$$

in der die Standardabweichungen σ_k die Breite der RBFs und damit den Wirkungsbereich um die Zentren $\boldsymbol{c_k}$ in den einzelnen Eingangsdimensionen bestimmen. Die Ausrichtung der Basisfunktion ist in diesem Fall ausschließlich achsenorthogonal.

Mit der Verbreitung der KNN wurden auch RBFs in diesem Kontext betrachtet und deren Eigenschaften untersucht [33], [34]. Ein neuronales Netz mit radialen Basisfunktionen (RBF-Netz) wird dabei als KNN mit einem verdeckter Schicht nach Bild 2.13 betrachtet, mit einer radialen Basisfunktion Φ_k als Aktivierungsfunktion. Oft wird diese als Gaußfunktion in der Form

$$\Phi_k = e^{(-\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{c_k}\|_{\boldsymbol{\Sigma_k}})} \tag{2.43}$$

definiert. Die Ausgangsgleichung eines RBF-Netzes ergibt sich damit als

$$\hat{y} = \sum_{k=1}^{K} v_k e^{(-\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{c_k}\|_{\boldsymbol{\Sigma_k}})}. \tag{2.44}$$

Da die Gewichte v_k linear auf die Ausgangsgröße wirken, können sie über lineare Optimierungstechniken bestimmt werden, was einen erheblichen Vorteil gegenüber den nichtlinearen Trainingsmethoden eines MLP-Netzes darstellt. Die Zentren und Standardabweichungen der Basisfunktionen sind hingegen nichtlineare Parameter der Neuronen der verdeckten Schicht. Mit ihnen können Lage und Breite der Basisfunktionen im Eingangsraum sehr gut heuristisch festgelegt werden, wodurch diese Parameter oft manuell definiert und nicht durch die Trainingsalgorithmen optimiert werden.

Das Interpolationsverhalten eines RBF-Netzes wird maßgeblich von der Breite der Basisfunktionen bestimmt. Mit großen Standardabweichungen ergibt sich einen größere Überlagerung der Wirkungsbereiche der Aktivierungsfunktionen und damit ein glatter Übergang zwischen den Neuronen. Mit schmaleren Basisfunktionen können stärkere Nichtlinearitäten besser abgebildet werden. Sie bergen aber auch die Gefahr eines nichtmonotonen Interpolationsverhaltens, insbesondere bei einer geringen Abdeckung des Eingangsraumes mit Neuronen, wie es bei höherdimensionalen Problemen unvermeidlich ist. Bedingt durch den Verlauf der Basisfunktionen extrapolieren RBF-Netze immer gegen Null. Ist dies nicht erwünscht, muss das Verhalten durch zusätzliche Neuronen an den Rändern des Eingangsraumes kompensiert werden, was in höherdimensionalen Konfigurationen zu einer erheblichen Erhöhung der Neuronenanzahl führen kann.

Diese Nachteile können durch die Verwendung sogenannter normalisierter RBF-Netze umgangen werden. Die Aktivierungsfunktionen der Neuronen werden bei dieser Variante der RBF-Netze mit der Summe aller Aktivierungsfunktionen normiert. Die normierte Aktivierungsfunktion ergibt sich mit:

$$\tilde{\Phi}_k = \frac{\Phi_k(\boldsymbol{u}, \boldsymbol{c}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \Phi_k(\boldsymbol{u}, \boldsymbol{c}_k, \boldsymbol{\Sigma}_k)} \tag{2.45}$$

und die Ausgangsgleichung eines normalisierten RBF-Netzes lautet somit

$$\hat{y} = \sum_{k=1}^{K} v_k \tilde{\Phi}_k. \tag{2.46}$$

Die Normierung bewirkt, dass die Summe aller Aktivierungsfunktionen an jeder Stelle des

Eingangsraumes gleich 1 ist

$$\sum_{k=1}^{K} \tilde{\Phi}_k = 1, \tag{2.47}$$

wodurch der Wirkungsbereich der einzelnen Neuronen so verzerrt wird, dass unabhängig von der Lage und der Breite der unnormierten Aktivierungsfunktionen Φ_k, der gesamte Eingangsraum abgedeckt wird. Das Inter- und Extrapolationsverhalten wird durch die Normierung signifikant verbessert, jedoch zieht dies eine starke Verkopplung jedes Neurons mit den umliegenden Neuronen nach sich (vergleiche auch Kapitel 3.2.1). Die ausschließlich lokale Wirkung eines Neurons und die damit einhergehende Möglichkeit der unabhängigen, heuristischen Strukturoptimierung ist damit nicht mehr gegeben und im Hinblick auf die Eignung für höherdimensionale Prozesse als nachteilig zu bewerten.

Im Folgenden sind die Eigenschaften radialer Basisfunktionsnetze hinsichtlich der aufgestellten Kriterien nochmals zusammengefasst:

1. *Eignung für hochdimensionale Eingangsräume*
   Die sehr gute Eignung der Struktur der RBF-Netze für höherdimensionale Probleme wird durch das spezielle Inter- und Extrapolationsverhalten klassischer RBF-Netze stark eingeschränkt. Normierte RBF-Netze verbessern diesen Nachteil zwar signifikant, erschweren jedoch die Möglichkeit, Lage und Breite der RBFs heuristisch festzulegen und nach erfolgter Optimierung, den Wirkungsbereich eines einzelnen Neurons im Kontext des zu modellierenden Prozesses zu betrachten.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Der strukturelle Aufbau klassischer RBFs eignet sich gut für eine geclusterte Datenbasis. Inwiefern auch dünn besetzte Eingangsräume monoton interpoliert werden, wird durch den verwendeten Typ und die Konfiguration der radialen Basisfunktion bestimmt. Normierte RBF-Netze sind durch ihre verkoppelten Parameter und den damit einhergehenden komplexen Verläufen der Basisfunktionen für dünn besetzte Datenbasen nur bedingt geeignet.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Durch die Möglichkeit, die Lage und Anzahl der Neuronen im Eingangsraum genau zu definieren, können unterschiedliche lokale Vorgaben zur Modellgenauigkeit sehr gut realisiert werden. Die nichtlineare Optimierung der Breite der RBFs bringt aber das Problem mit sich, hierfür ein globales Optimum zu finden.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Die Modellkomplexität kann über die Erhöhung der Anzahl der Neuronen in RBF-Netzen schrittweise in Abhängigkeit des Modellfehlers erhöht werden.

5. *Ressourcenbedarf der Schätzgleichung*
   Der Ressourcenbedarf ist im wesentlichen von der Anzahl der Neuronen und von der gewählten Basisfunktion abhängig. Die klassische Gaußfunktion, insbesondere in ihrer normierten Variante, ist für eine Berechnung in Motorsteuergeräten weniger geeignet.

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Die gute Interpretationsmöglichkeit eines optimierten klassischen RBF-Netzes begünstigt eine Adaption eines vorhandenen Netzes auf einen ähnlichen Prozess. Durch die

lokale Wirksamkeit der Neuronen ist auch eine Teiloptimierung möglich. Normierte RBF-Netze bieten diese Möglichkeiten nur sehr eingeschränkt.

Zusammenfassung: Der Struktur, der guten Eignung für ungleichmäßig verteilte Datenbasen und den guten Interpretationsmöglichkeiten klassischer RBF-Netze stehen das spezielle Inter- und Extrapolationsverhalten gegenüber, das sich vor allem in hochdimensionalen Prozessen negativ bemerkbar macht. Normierte RBF-Netze versuchen diese Nachteile auszugleichen, wodurch jedoch auch die positiven Eigenschaften verloren gehen bzw. reduziert werden. Insgesamt gesehen sind RBF-Netze gut für niedrig bis mitteldimensionale Probleme geeignet, bei höherdimensionalen Modellen überwiegen die Nachteile.

2.4.5. Parallele lokal-lineare Modelle

Die Klasse der lokal-linearen Modellstrukturen approximiert Prozesse über die Aufteilung in lineare Teilmodelle. Diese Modellform wurde auf Grundlage unterschiedlicher Strukturen und Optimierungsverfahren im Kontext von stückweise-lineare Modellen, Neuro-Fuzzy-Modellen, statistischen Modellen u.a. entwickelt. Eine ausführliche Übersicht und Diskussion findet man in [22], [8], [35], [36].

Ausgehend von der allgemeinen Formulierung als Summe gewichteter Basisfunktionen nach Gleichung (2.26) kann ein lokal-lineares Teilmodell $\alpha_k(\boldsymbol{u})$ als eine Linearkombination der Eingänge und einem Offset beschrieben werden

$$\alpha_k(\boldsymbol{u}, \boldsymbol{c}_k) = \boldsymbol{c}_k \tilde{\boldsymbol{u}} \tag{2.48}$$

mit

$$\boldsymbol{c}_k = [c_0, c_1, c_2, \cdots, c_q], \quad \text{und} \quad \tilde{\boldsymbol{u}} = [1, u_1, u_2, \cdots, u_q]^T. \tag{2.49}$$

Die Ausgangsgleichung für ein lokal-lineares Modell ergibt sich so zu

$$\hat{y}(\boldsymbol{u}) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{u}, \boldsymbol{c}_k) \cdot \varphi_k(\boldsymbol{u}, \theta_k). \tag{2.50}$$

Die Basisfunktion φ_k definiert, abhängig vom Eingangsvektor $\boldsymbol{u}$, den Gültigkeitsbereich des Teilmodells im Eingangsraumes sowie die Übergänge zwischen den Teilmodellen. Der Eingangsraum von φ muss dabei nicht notwendigerweise dem vollständigen Eingangsraum des linearen Modells α_k entsprechen, sondern kann diesen auch als Unterraum abbilden, was oft für eine Reduktion der Modellkomplexität genutzt werden kann. In den meisten Verfahren liegt der Wertebereich der Basisfunktionen im Intervall $[0, 1] \in \mathbb{R}$ und die Summe aller Basisfunktionen ergibt sich für alle Eingangsvektoren $\boldsymbol{u}$ zu:

$$\sum_{k=1}^{K} \varphi_k(\boldsymbol{u}, \theta_k) = 1. \tag{2.51}$$

Mit der Bedingung nach Gleichung (2.51) lässt sich die Gültigkeit eines Teilmodells anhand des Funktionswerts der zugehörigen Basisfunktion direkt ablesen, was die Interpretierbarkeit der Modellstruktur deutlich verbessert.

Das Strukturbild eines solchen lokal-linearen Modells ist in Bild 2.14 dargestellt. Grundi-

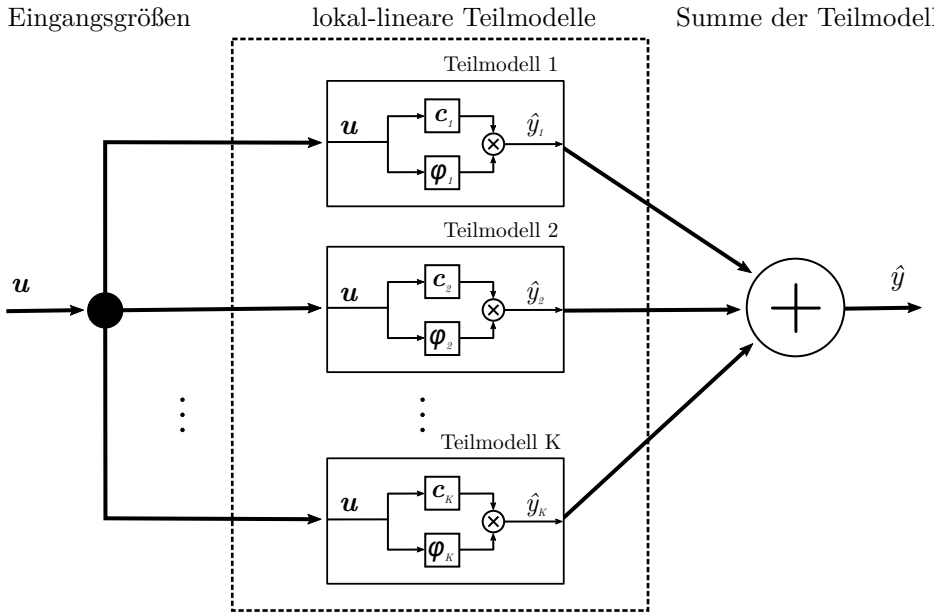

Abbildung 2.14.: Struktur eines lokal-linearen Modellnetzes mit der Anzahl K an Teilmodellen

dee dieser Struktur ist die Approximation einer nichtlinearen Funktion durch lineare Teilmodelle und damit die Reduktion der Komplexität eines Prozesses auf lokal geltende lineare Zusammenhänge. Die Beschränkung auf lineare Funktionen bietet den Vorteil eines monotonen Inter- und Extrapolationsverhaltens innerhalb des Geltungsbereiches eines Teilmodells.

Die verschiedenen Arten lokal-linearer Modelle unterscheiden sich in der Definition, dem Entwurf und der Art der Parameterbestimmung der Basisfunktion $\varphi_k(\boldsymbol{u}, \theta)$, als auch in der Art der Strukturkonstruktion und -optimierung, wodurch sich auch ein Großteil ihrer Eigenschaften ergibt. In [22] wird eine fließende Klassifizierung in Splines und stückweise-lineare Modelle, statistische Ansätze, Fuzzy Modelle und Neuronale Netzwerke vorgenommen und weiterführende Literatur angegeben.

Typische Vertreter lokal-linearer Netze in den einzelnen Klassen sind zum Beispiel:

- *Splines und stückweise-lineare Modelle:* Adaptive spline modelling of observation data (ASMOD) [37] und Multivariate adaptive regression splines (MARS) [38]. Die Übergänge zwischen den Teilmodellen werden bei diesen Verfahren als unstetige Umschaltung oder als B-Splines definiert.

- Fuzzy-Modelle: Takagi-Sugeno Fuzzy Modelle und Adaptive Neuro Fuzzy Inferenzsystem (ANFIS). In der Fuzzy-Terminologie wird die Basisfunktion als Zugehörigkeitsfunktion bezeichnet, über deren Definition die Abgrenzung der einzelnen Teilmodelle erfolgt.

- Neuro-Fuzzy-Modelle: Lokal-Linear-model-Trees (LOLIMOT) und Hinging-Hyperplane-Netze. Diese Modellklasse kombiniert Eigenschaften der neuronalen Netze und der Fuzzy-Modelle. Während beim LOLIMOT-Algorithmus die Fuzzy-typische Definition der Basisfunktionen als UND verknüpfte IF-THEN-Bedingungen eine wesentliche

Eigenschaft darstellt, steht beim Hinging-Hyperplane-Verfahren die KNN-typische, flexible Ausrichtung der Basisfunktionen im Eingangsraum im Vordergrund.

- statistische Ansätze: Gaussian-Mixture-Regression (GMR). Statistische Ansätze verfolgen das Ziel, Lage und Parameter der linearen Teilmodelle über wahrscheinlichkeitstheoretische Grundsätze aus den Messdaten zu ermitteln. Dies erlaubt oft elegante mathematische Lösungen, bedingt aber in der Regel eine hohe Qualität der Messdaten und die Erfüllung spezieller statistischer Eigenschaften des Datensatzes.

Die GMR und der LOLIMOT-Algorithmus wurden im Rahmen dieser Arbeit genauer untersucht und werden in den Kapitel 2.4.7 und 2.4.8 ausführlich dargestellt.

Nachfolgend seien die Eigenschaften lokal-linearer Modelle bezüglich der aufgestellten Kriterien zur Interpretierbarkeit datenbasierter Modelle aufgeführt:

1. *Eignung für hochdimensionale Eingangsräume*
   Die klassische Struktur eine lokal-linearen Netzes nach Gleichung (3.2) eignet sich sehr gut für höherdimensionale Probleme. Die Beschränkung auf lineare Teilmodelle erlaubt ein gut validierbares Inter- und Extrapolationsverhalten, welches jedoch grundlegend von der Wahl der Basisfunktion und deren Entwurfsverfahren abhängt.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Die lineare Definition der Teilmodelle erlaubt eine Abdeckung großer Gebiete des Eingangsraumes mit nur wenigen Modellen und die Optimierung der Parameter dieser Modelle mit einer kleinen, ungleich verteilten Anzahl an Messdaten. Die Anforderungen an die Verteilung der Daten können daher als gering eingestuft werden.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Je nach gewählter Basisfunktion und einer damit verbundenen Überlagerung der Wirkungsbereiche können die Güteanforderungen für jedes der einzelnen Teilmodelle gesondert festgelegt werden.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Auf dem Themenfeld der lokal-linearen Modelle gibt es eine Vielzahl von Verfahren zur Strukturoptimierung. Einige dieser Methoden erlauben die iterative Erhöhung der Modellkomplexität auf Grundlage des erreichten Modellfehlers.

5. *Ressourcenbedarf der Schätzgleichung*
   Der Ressourcenbedarf zur Berechnung eines linearen Teilmodells ist sehr gering und der Hauptbedarf wird durch die verwendete Basisfunktion bestimmt. Die in vielen Ansätzen zum Einsatz kommenden Exponentialfunktionen benötigen einen sehr großen Rechenaufwand und sind für den Einsatz auf Motorsteuergeräten weniger geeignet. Polynomfunktionen dagegen erlauben eine effektive Berechnung und bieten sich mit ihrem geringen Ressourcenbedarf für dieses Anwendungsfeld an.

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Die Möglichkeit lokaler Anpassungen einzelner Teilmodelle ohne die Beeinflussung anderer Bereiche des Modells hängt ganz wesentlich von der Art der Basisfunktion und der Größe der Überlagerung der einzelnen Teilmodelle ab. Sind die Überlagerungsbereiche klein, sind Adaptionen durch lokale Anpassungen, unterstützt durch die gute Interpretierbarkeit der linearen Teilmodelle, in der Regel sehr gut möglich.

Zusammenfassung: Lokale-lineare Modellansätze zeigen eine breite Palette unterschiedlicher Eigenschaften, die im Wesentlichen von der Art der Basisfunktion und der Strukturoptimierung des gewählten Verfahrens abhängen. Sich wenig überlagernde Teilmodelle und einfache Basisfunktionen resultieren in einem vorhersehbaren Inter- und Extrapolationsverhalten und einer sehr guten physikalischen Interpretierbarkeit auch bei höherdimensionalen Prozessen. Ungleichmäßig verteilte Messdaten stellen in der Regel kein Problem da, ebenso wie unterschiedliche Güteforderungen für Teilbereiche des Modells. Der Ressourcenbedarf richtet sich nach der Wahl der Basisfunktion.

2.4.6. Support-Vektor-Regression

Das Prinzip der Support Vektor Regression (SVR) beruht auf der Annahme, einen Datensatz $D = \{\{\boldsymbol{x_1}, y_1\}, \{\boldsymbol{x_2}, y_2\}, \ldots, \{\boldsymbol{x_p}, y_p\}\}, D \in \mathbb{R}^{N\times(q+1)}$ über ein Hyperebene

$$\hat{y} = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b \tag{2.52}$$

mit dem Fehler ε linear approximieren zu können, wobei der Abstand der Datenpunkte zur Ebene minimiert werden soll, was über die Minimierung der Norm $\|\boldsymbol{w}\|^2$ erreicht werden kann [39]. Datenpunkte, die einen größeren Fehler als ε aufweisen, fließen über eine Schlupfvariable ξ und einem Strafterm in die Optimierung ein. Das Optimierungsproblem lautet so:

$$\min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)\right) \tag{2.53}$$

mit den Nebenbedingungen

$$\begin{gathered} y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b - y_i \leq \varepsilon + \xi^* \\ \xi_i, \xi_i^* \geq 0 \end{gathered} \tag{2.54}$$

Die Schlupfvariablen ξ_i, ξ_i^* berechnet sich in Abhängigkeit des Fehlers ε und des Abstandes des Messpunktes zur Hyperebene mit

$$|\xi_i| = \begin{cases} 0, & \text{wenn } |\zeta_i| \leq \varepsilon \\ |\zeta| - \varepsilon, & \text{wenn } |\zeta_i| > \varepsilon \end{cases} \tag{2.55}$$

Grafisch wird der Zusammenhang in Bild 2.15 dargestellt.

Dieses Optimierungsproblem kann über eine Langrange-Funktion als duales Problem formuliert werden, mit der Lösung [39]

$$\boldsymbol{w} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i \tag{2.56}$$

mit den Lagrange-Multiplikatoren α_i, α_i^*, woraus sich die Ausgangsgleichung

$$\hat{y} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b \tag{2.57}$$

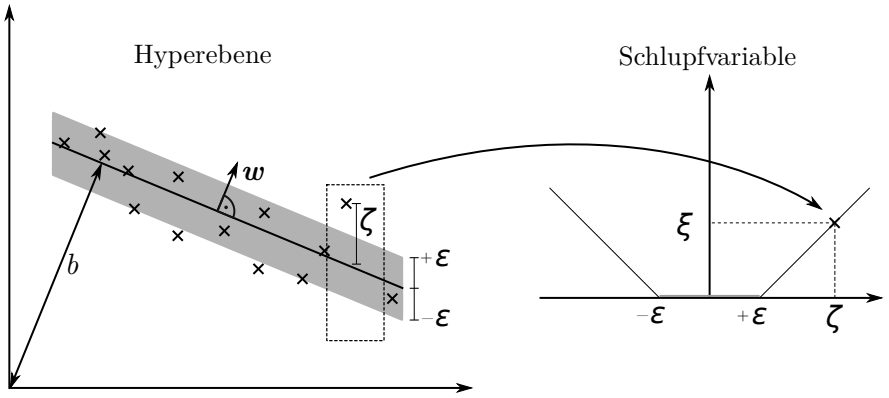

Abbildung 2.15.: Prinzip der Approximation über eine Hyperebene mit dem Fehler ε und den Schlupfvariablen ξ in der Support-Vektor-Regression

ergibt. Die Lagrange-Multiplikatoren für alle Datenpunkte mit $|\hat{y}_i - y_i| \leq \varepsilon$ (in Abbildung 2.15 alle Datenpunkte innerhalb des grau markierten Bereiches) sind dabei gleich Null, sodass diese Summanden verschwinden. Die verbleibenden Datenpunkte, die in die Ausgangsgleichung eingehen, werden als Stützvektoren bezeichnet.

Diese Lösung hat einige bemerkenswerte Eigenschaften. So ist die Komplexität des Ausgangsgleichung unabhängig von der Dimension des Eingangsraumes und nur abhängig von der Anzahl der Stützvektoren. Weiterhin muss der Normalenvektor $\boldsymbol{w}$ nicht explizit nach Gleichung (2.56) berechnet werden, da die Datenpunkte nur als Vektorprodukt in die Ausgangsgleichung eingehen. Dies ermöglicht auch die nichtlineare Erweiterung des Algorithmus, deren Grundidee es ist, den Eingangsraum in einen höherdimensionalen Raum $\mathcal{M}$ zu überführen, in dem sich die Daten linear approximieren lassen. Dazu wird eine Kernfunktion

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}) \rangle \tag{2.58}$$

eingeführt, die sich wie ein Skalarprodukt der in den Raum $\mathcal{M}$ transformierten Datenpunkte verhält und für das Skalarprodukt in Gleichung (2.57) eingesetzt werden kann. Eine explizite Überführung der Daten in den Raum $\mathcal{M}$ ist damit nicht mehr notwendig. Die Ausgangsgleichung für den nichtlinearen Fall ergibt sich so mit

$$\hat{y} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{2.59}$$

Als Kernfunktionen können alle Funktionen eingesetzt werden, die die Mecer-Bedingungen erfüllen [40]. Oft verwendet werden lineare Kerne

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{x}_i^T \boldsymbol{x} + c, \tag{2.60}$$

polynominale Kerne

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = (c + \beta \boldsymbol{x}_i^T \boldsymbol{x})^d \tag{2.61}$$

sowie Gauß-Kerne

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|^2}{2\sigma^2}} \tag{2.62}$$

und sigmoide Kerne

$$k(\boldsymbol{x}_i, \boldsymbol{x}) = tanh(c + \beta \boldsymbol{x}_i^T \boldsymbol{x}). \tag{2.63}$$

Setzt man die Kernfunktionen in die Ausgangsgleichung (2.59) ein und vergleicht mit den Ausgangsgleichungen der oben besprochenen Verfahren, so werden die Ähnlichkeiten zu den RBF-Netzen, den MLP-Netzen und den lokalen Funktionsnetzen sichtbar. Bezüglich der Basisfunktionsdefinition nach Gleichung (2.26) kann die Kernfunktion als Basisfunktion interpretiert werden, die über die Faktoren $\nu = \alpha_i - \alpha_i^*$ gewichtet wird. Die gewählte Kernfunktion bestimmt damit maßgeblich die Eigenschaften eines SVR-Modells hinsichtlich der Interpretierbarkeit und des Inter- und Extrapolationsverhaltens, sodass darüber wenig allgemeingültige Aussagen getroffen werden können. Da sich die Wirkungsbereiche der einzelnen Stützvektoren jedoch weit überlagern können, ist die Interpretierbarkeit in der Regel deutlich eingeschränkt.

Die Wahl der Kernfunktion wird im Allgemeinen heuristisch durchgeführt und ist stark von den zur Verfügung stehenden Daten beziehungsweise von dem zu modellierenden Prozess abhängig. Eine automatische Auswahl aus einer Reihe von Standardkernfunktionen wird in [41] vorgestellt. Die Optimierung über die Lagrangefunktion erlaubt weiterhin keine Vorgabe von A-priori-Wissen und keine unabhängige lokale Anpassung von Teilbereichen. Weiterhin können neue Datenpunkte nur durch eine vollständige Neuoptimierung in das Modell einfließen.

Nachfolgend sind die Eigenschaften der SVR bezüglich der in Kapitel 2.3.1 aufgestellten Kriterien zusammengefasst:

1. *Eignung für hochdimensionale Eingangsräume*
   Durch die von der Eingangsdimension unabhängigen Komplexität der Ausgangsgleichung und des Optimierungsalgorithmus eignet sich die SVR sehr gut für hochdimensionale Prozesse.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Prinzipiell ist die SVR auch für die Optimierung bei ungleichmäßig verteilte Daten geeignet und kann dabei gute Ergebnisse liefern. Dies ist jedoch sehr stark von der Wahl der Kernfunktion und dem zu modellierenden Prozess abhängig. Die Inter- und Extrapolationseigenschaften mit Gauß-Kernen sind mit denen der RBF vergleichbar. SVR mit sigmoiden Kernen hat ähnliche Eigenschaften wie MLP-Netze.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Die geschlossene analytische Optimierung der SVR lässt keine getrennten Vorgaben der Modellgüte für verschiedene Bereiche des Eingangsraumes zu, diese kann nur global über die Anzahl der Stützvektoren und der Art der Kernfunktion bestimmt werden.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Die Modellkomplexität für eine vorgegebene Modellgüte wird bei der SVR durch die notwendige Anzahl der Stützvektoren bestimmt, welche wiederum von der gewählten Kernfunktion abhängt. Die Anzahl der Stützstellen lässt sich inkrementell bis zur

gewünschten Modellgüte erhöhen, die optimale Wahl der Kernfunktion ist in der Regel ein heuristischer Prozess.

5. *Ressourcenbedarf der Schätzgleichung*
   Die Anzahl der Stützvektoren und die Art der Kernfunktion bestimmen auch den Ressourcenbedarf der Schätzgleichung. In Motorsteuergeräten sind lineare und Polynomfunktionen wesentlich effizienter zu berechnen als Exponentialfunktionen wie z.B. in Gauß-Kernen.

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Vorhandene Modellierungen lassen sich auf einen neuen Prozess nur durch eine vollständig neue Optimierung des Modells anpassen. Eine Anpassung von Teilbereichen unter Beibehaltung der restlichen Modellteile ist nicht möglich. Die Wahl der Kernfunktion als ein wesentlicher Entwurfsschritt kann bei ähnlichen Prozessen meist übernommen werden.

Zusammenfassung: Die SVR eignet sich sehr gut für höherdimensionale Prozesse, besitzt einen effizienten Optimierungsalgorithmus und ist über die Wahl der Kernfunktionen flexibel für unterschiedlichste Anforderungen einsetzbar. Das Extra- und Interpolationsverhalten ist dabei stark von der verwendeten Kernfunktion abhängig. Die Wirkungsbereiche der einzelnen, über die Stützvektoren positionierten Kernfunktionen können sich stark überlagern, wobei der Optimierungsalgorithmus keine direkte Beeinflussung dieser Überlagerung zulässt. Die Interpretierbarkeit wird durch dieses Eigenschaft stark eingeschränkt. Der Ressourcenbedarf ist ebenfalls direkt von der gewählten Kernfunktion abhängig.

2.4.7. Gaussian Mixture Regression

Die Grundidee der Gaussian mixture regression (GMR) ist die Approximation der statistischen Verteilung der Messdaten $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\} \in \mathbb{R}^{N\times q}$, mit der Messpunkteanzahl N und der Dimension q, als eine Summe von K gewichteten q-dimensionalen Normalverteilungen

$$p(\boldsymbol{d}|\boldsymbol{\theta}) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.64}$$

mit den Mittelwertvektoren $\boldsymbol{\mu}_k \in \mathbb{R}^{q\times 1}$, den Kovarianzmatrizen $\boldsymbol{\Sigma}_k \in \mathbb{R}^{q\times q}$ und den Gewichten $w_k \in \mathbb{R}$. Zu jeder dieser Verteilungen $M_k = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ wird ein korrespondierendes, lineares Modell $m_k(\boldsymbol{u})$ als Funktion des p-dimensionalen Eingangsvektors $\boldsymbol{u} \in \mathbb{R}^{p\times 1}$ angenommen

$$\hat{y}_k = m_k(\boldsymbol{u}) = \boldsymbol{c}_k \boldsymbol{u}, \tag{2.65}$$

welches über den Koeffizientenvektor $\boldsymbol{c}_k \in \mathbb{R}^{1\times p}$ die lokale Ausgangsgröße $\hat{y}_k$ der einzelnen Komponenten definiert. Die Schätzung der lokalen Ausgangsgröße wird mit der Wahrscheinlichkeitsdichte $p(M_k|\boldsymbol{u})$ gewichtet und als Summe aller lokal-linearen Komponenten ergibt sich die Ausgangsgleichung des Modells

$$\hat{y}(\boldsymbol{u}) = \sum_{k=1}^{K} p(M_k|\boldsymbol{u}, \boldsymbol{\theta}_k) m_k(\boldsymbol{u}), \qquad \hat{y} \in \mathbb{R}. \tag{2.66}$$

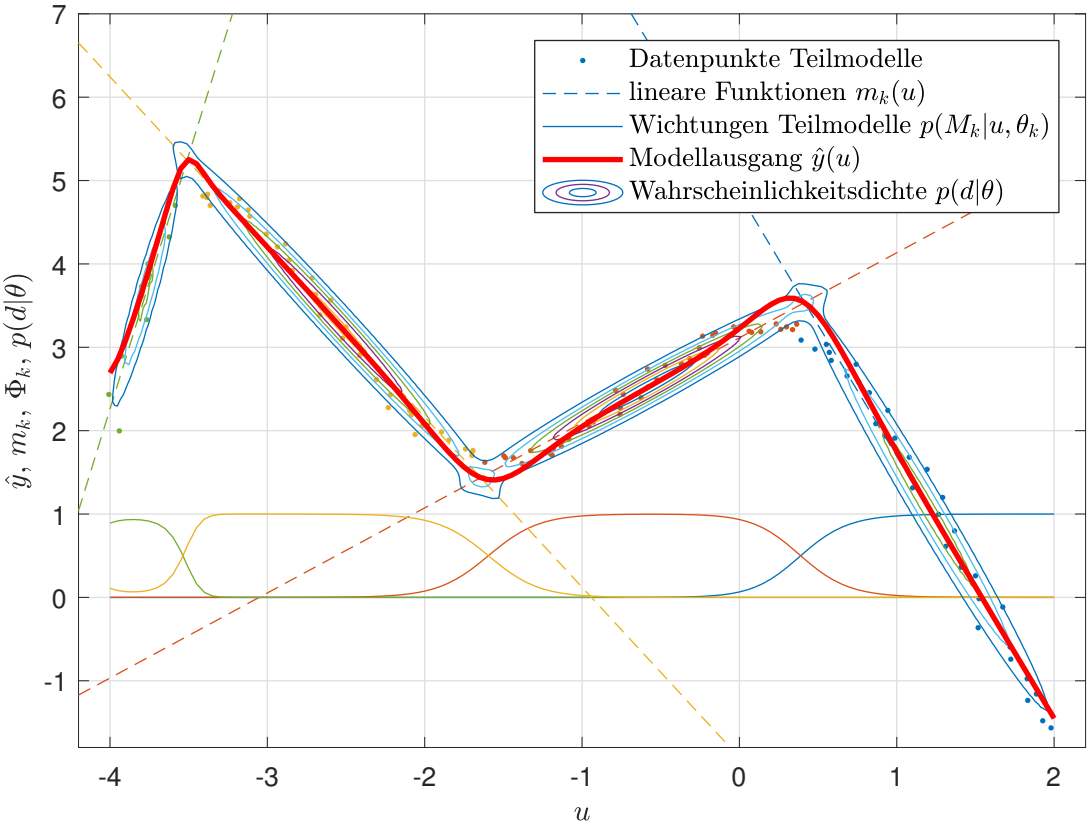

Abbildung 2.16.: Datenpunkte mit resultierenden Wahrscheinlichkeitsverteilungen, lineare Funktionen, Wichtungen und Verlauf der modellierten Ausgangsgröße eines Gaussian-Mixrure-Regression-Modells mit 4 Komponenten. Datenpunkte, lineare Funktionen und Wichtungen gleicher Farbe stehen für jeweils eine Teilkomponente k

Die bedingten Wahrscheinlichkeiten $p(M_k|\boldsymbol{u}, \boldsymbol{\theta}_k)$ mit den Parametern der Normalverteilungsdichten $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ definieren hierbei den Gültigkeitsbereich jeder linearen Komponente und können somit als Basisfunktionen $g(\boldsymbol{u}, \boldsymbol{\theta_k})$ nach Gleichung (2.26) interpretiert werden. Die resultierende Ausgangsgleichung (2.66) kann in jeder dieser Komponenten als stückweise Linearisierung eines nichtlinearen Modells aufgefasst werden. Abbildung 2.16 veranschaulicht das Prinzip an einem einfachen Beispiel. Eine ausführliche Herleitung und Diskussion findet man in [42], [43] und [44].

Für die Schätzung der Ausgangsgröße müssen die unbekannten Parameter in Gleichung (2.66) anhand der Messdaten identifiziert werden. Dies erfolgt über den Expectation-Maximization-Algorithmus (EM) [45], welcher im anschließenden Kapitel beschrieben wird.

Expectation-Maximization-Algorithmus (EM-Algorithmus)

In der Trainingsphase des GMR-Modells werden die unbekannten Parameter $\boldsymbol{c}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, w_k$ der Gleichung (2.66) aus den Messdaten geschätzt. Jeder Messpunkt $\boldsymbol{d}_i = \{\boldsymbol{u}_i, y_i\}$ enthält die Eingangsgrößen und die Ausgangsgröße des Prozesses. Der Prozess wird, wie oben beschrieben, als eine Summe gewichteter Normalverteilungen angenommen und dessen Ver-

teilungsdichtefunktion kann nach Gleichung (2.64) ausgedrückt werden, mit $\mathcal{N}(\boldsymbol{d}|\boldsymbol{\theta}_k)$ als multivariate normalverteilte Wahrscheinlichkeitsdichtefunktion

$$\mathcal{N}(\boldsymbol{d}|\boldsymbol{\theta}_k) = \frac{1}{(2\pi)^{\frac{p+1}{2}}\sqrt{|\boldsymbol{\Sigma}_k|}} \exp\left[-\frac{1}{2}(\boldsymbol{d}-\boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{d}-\boldsymbol{\mu}_k)^T\right] \tag{2.67}$$

und

$$\sum_{k=1}^{K} w_k = 1. \tag{2.68}$$

Zur Schätzung der Parameter $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K\}$ für alle Teilkomponenten $k = \{1, 2, \cdots, K\}$ kann die Likelihood-Funktion folgendermaßen definiert werden

$$\begin{aligned}\log p(\boldsymbol{D}, \boldsymbol{Z}|\boldsymbol{\theta}) &= \log \prod_{i=1}^{N} p(\boldsymbol{d}_i, z_i|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \log \sum_{k=1}^{K} \mathbb{I}(z_i = k) w_k p(\boldsymbol{d}_i|\boldsymbol{\theta}_k).\end{aligned} \tag{2.69}$$

$\boldsymbol{Z} = \{z_1, z_2, \ldots, z_K\}$ sind die Werte der gesuchten Zuordnungen der Beobachtungen zu den einzelnen Komponenten und $\mathbb{I}(z_i = k)$ die Indikatorfunktion. Da das Maximum der Likelihood-Funktion (2.69) nicht analytisch bestimmt werden kann, wird zur Lösung des Problems der Expectation-Maximization-Algorithmus (EM-Algorithmus) eingesetzt, welcher ein lokales Maximum durch das iterative Ausführen der folgenden beiden Schritte findet [43]:

1. Im Expectation-Schritt des EM-Algorithmus wird die bedingte Wahrscheinlichkeit der logarithmierten Likelihood-Funktion für den kompletten Datensatz inklusive der Zuordnungen zu den Komponenten $\boldsymbol{C} = \{\boldsymbol{D}, \boldsymbol{Z}\}$ nach Gleichung (2.69) berechnet. Gegeben sind dabei der Datensatz $\boldsymbol{D}$ und die Parameter der vorhergehenden Iteration $\boldsymbol{\theta}_{old}$ beziehungsweise die Initialisierungswerte der Iteration.

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old}) = E\{\log p(\boldsymbol{D}, \boldsymbol{Z}|\boldsymbol{\theta})|\boldsymbol{D}, \boldsymbol{\theta}_{old}\}. \tag{2.70}$$

2. Im Maximization-Schritt des EM-Algorithmus werden die neuen Parameter $\boldsymbol{\theta}$ über eine Maximum-Likelihood-Schätzung bestimmt.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}}\{Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{old})\}. \tag{2.71}$$

Zur Initialisierung des Algorithmus können die Mittelwertvektoren $\boldsymbol{\mu}_k$ per Zufall bestimmt werden. Die Kovarianzmatrizen $\boldsymbol{\Sigma}_k$ können als Diagonalmatrizen mit den Varianzen der Messgrößen in den Diagonalelementen aufgestellt werden. Die Iteration wird beendet, wenn die Änderung der Likelihood-Funktion innerhalb eines Schrittes

$$r = |\log p(\boldsymbol{D}, \boldsymbol{Z}|\boldsymbol{\theta}) - \log p(\boldsymbol{D}, \boldsymbol{Z}|\boldsymbol{\theta}_{old})| \tag{2.72}$$

unter einem vorgegebenen Schwellwert sinkt oder eine bestimmte Anzahl an Iterationen erreicht ist. Das Ergebnis des Algorithmus sind die Schätzungen der Parameter der Normalverteilungen $\boldsymbol{\mu}_k$ und $\boldsymbol{\Sigma}_k$ sowie die Gewichte w_k für die K Komponenten.

Regressionsfunktion einer Gaußschen Mischverteilung

Es sei $p(\boldsymbol{d}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ eine multivariate Normalverteilung

$$p(\boldsymbol{d}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{d}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.73}$$

mit

$$\boldsymbol{d} = \begin{bmatrix} \boldsymbol{u} \\ y \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_U \\ \mu_Y \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{UU} & \boldsymbol{\Sigma}_{UY} \\ \boldsymbol{\Sigma}_{YU} & \Sigma_{YY} \end{bmatrix}. \tag{2.74}$$

Diese Verteilungsfunktion kann in zwei Normalverteilungen aufgeteilt werden [44], [42]:

$$p(\boldsymbol{u}, y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(y|\boldsymbol{u}, m(u), \boldsymbol{\sigma}^2)\mathcal{N}(\boldsymbol{u}, \boldsymbol{\mu}_U, \boldsymbol{\Sigma}_{UU}) \tag{2.75}$$

mit

$$m(u) = E\{Y|U = \boldsymbol{u}\} = \mu_Y + \boldsymbol{\Sigma}_{YU}\boldsymbol{\Sigma}_{UU}^{-1}(\boldsymbol{u} - \boldsymbol{\mu}_U), \tag{2.76}$$

und

$$\sigma^2 = Var\{Y|U = \boldsymbol{u}\} = \Sigma_{YY} - \boldsymbol{\Sigma}_{YU}\boldsymbol{\Sigma}_{UU}^{-1}\boldsymbol{\Sigma}_{UY}. \tag{2.77}$$

Unter der Bedingung, dass der Prozess als eine Summe von Normalverteilungen nach Gleichung (2.64) beschrieben werden kann, kann nun die gemeinsame Wahrscheinlichkeitsverteilung für die Ein- und Ausgangsgrößen des Prozesses geschrieben werden als:

$$p(\boldsymbol{u}, y) = \sum_{k=1}^{K} w_k \mathcal{N}(y|\boldsymbol{u}, m_k(\boldsymbol{u}), \sigma_k^2)\mathcal{N}(\boldsymbol{u}, \boldsymbol{\mu}_{k,U}, \boldsymbol{\Sigma}_{k,UU}) \tag{2.78}$$

mit

$$m_k(u) = \mu_{k,Y} + \boldsymbol{\Sigma}_{k,YU}\boldsymbol{\Sigma}_{k,UU}^{-1}(\boldsymbol{u} - \boldsymbol{\mu}_{k,U}) \tag{2.79}$$

und

$$\sigma_k^2 = \Sigma_{k,Y} - \boldsymbol{\Sigma}_{k,YU}\boldsymbol{\Sigma}_{k,UU}^{-1}\boldsymbol{\Sigma}_{k,UY}. \tag{2.80}$$

Die Randwahrscheinlichkeit $p(\boldsymbol{u})$ und die bedingte Wahrscheinlichkeit $p(y|\boldsymbol{u})$ kann ebenfalls als Summe aus Normalverteilungen geschrieben werden:

$$p(\boldsymbol{u}) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{k,U}, \boldsymbol{\Sigma}_{k,UU}) \tag{2.81}$$

$$p(y|\boldsymbol{u}) = \sum_{k=1}^{K} \varphi_k(u)\mathcal{N}(y|\boldsymbol{u}, m_k(\boldsymbol{u}), \sigma_k^2) \tag{2.82}$$

Die Wichtungsfunktion $\varphi_k(u)$ kann mit Hilfe des Bayes-Theorems formuliert werden:

$$\varphi_k(u) = \frac{w_k \mathcal{N}(u, \mu_{k,U}, \mathbf{\Sigma}_{k,UU})}{\sum_{k=1}^{K} w_k \mathcal{N}(u, \mu_{k,U}, \mathbf{\Sigma}_{k,UU})}. \tag{2.83}$$

Mit m_k und φ_k aus Gleichung (2.79), (2.83) kann nun die Regressionsfunktion in der Form

$$\hat{y}(u) = E\{Y|U = \boldsymbol{u}\} = \sum_{k=1}^{K} \varphi_k(\boldsymbol{u}) m_k(\boldsymbol{u}). \tag{2.84}$$

geschrieben werden.

Die Regressionsfunktion nach (2.84) stellt die Basis der Gaussian-Mixture-Regression da. Die Parameter der Funktion ($\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k$ and w_k in (2.74)) können über den im letzten Abschnitt vorgestellten EM-Algorithmus aus den Messdaten geschätzt werden.

Eigenschaften der Gaussian-Mixture-Regression

Die Ausgangsgleichung der GMR nach Gleichung (2.84) entspricht in ihrer Struktur dem Basisfunktionsmodell mit $\varphi(\boldsymbol{u})$ als Basisfunktion und kann mit der linearen Funktion $m_k(\boldsymbol{u})$ auch als lokal-linearer Modellansatz betrachtet werden. Die geschlossene statistische Formulierung des Optimierungsproblems skaliert sehr gut mit der Erhöhung der Eingangsdimensionen und ermöglicht auf Grund der vollbesetzten Kovarianzmatrizen eine hohe Flexibilität bei einer geringen Komponentenanzahl. Damit eignet sich die GMR sehr gut für stark nichtlineare, hochdimensionale Prozesse. Gleichzeitig bringen diese Flexibilität und die statistische Grundlagen des Verfahrens diverse praktische Probleme mit sich, auf die im Folgenden näher eingegangen werden soll.

Zur Initialisierung des EM-Algorithmus werden für die erste Berechnung des Parametersatzes θ Startwerte benötigt, welche über verschiedene Vorgehensweisen festgelegt werden können. Da es sich bei dem Optimierungsproblem nach Gleichung (2.69) um eine nichtlineare Optimierung handelt, sind die Startwerte entscheidend für das Auffinden des globalen Optimums. Die hohe Parameteranzahl und die bei vielen realen Prozessen große Anzahl von lokalen Optima machen bei einer zufälligen Wahl der Initialisierungswerte eine große Anzahl an verschiedenen Optimierungsdurchläufen für eine gute Näherung an das globale Optimum notwendig, ohne jedoch den Güteverlust der Näherung beziffern zu können. Die Ineffizienz dieses Vorgehens steigt mit der Erhöhung der Komponentenanzahl und der Eingangsdimensionen.

Ein populärer Ansatz zur Bestimmung geeigneter Startparameter ist die Verwendung des k-mean-Algorithmus' zur Zuordnung der einzelnen Datenpunkte zu den Komponenten [46]. Mit der Indizierung können Mittelwert und Kovarianzmatrix der jeweiligen Komponenten berechnet und zur Initialisierung genutzt werden. Der k-mean-Algorithmus benötigt jedoch zur Berechnung der Clustermittelpunkte ebenfalls Startwerte, für deren Bestimmung zwar verschiedene Methoden publiziert wurden [47], [48], [49], welche jedoch die Ermittlung des globalen Optimums, gerade unter der Vorgabe einer großen Clusterzahl, nicht garantieren.

Neben den Initialisierungswerten des EM-Algorithmus' ist weiterhin die Vorgabe der Komponentenanzahl notwendig. Üblicherweise werden dazu Informationskriterien wie die in Kapitel 2.2.1 vorgestellten AIC und BIC verwendet [50], [51]. Deren Berechnung ist jedoch gleichfalls von den gewählten Startparametern des EM-Algorithmus' abhängig. Eine Metho-

de zur kombinierten Schätzung der Komponentenanzahl und der Initialisierungswerte des EM-Algorithmus, welche die Cluster hierarchisch definiert, finden sich in [46].

Der statistische Ansatz zur Modellierung des Prozesses über die GMR gestattet zwar eine elegante mathematische Formulierung und Optimierung, bedingt jedoch auch die Einhaltung der mathematischen Voraussetzungen. Konkret muss sich zur Anwendung des Verfahrens die statistische Verteilung der Messdaten sowohl in den Eingangsgrößen als auch in der Ausgangsgröße als Summe von Normalverteilungen darstellen lassen. Weiterhin müssen für eine konsistente Schätzung der Parameter der Verteilungsdichtefunktionen die Datenpunkte der zugrundeliegenden Stichprobe unabhängig, unkorreliert und zufällig ermittelt werden. Sind diese Bedingungen nicht gegeben, beeinflusst die gewählte Stichprobe die Parameter der Verteilungsdichte und letztlich wird das Modell den Prozess fehlerhaft widerspiegeln.

Die genannten Voraussetzungen sind im Anwendungsgebiet der Motorenentwicklung nur schwer einzuhalten. Während die Zielgröße, bedingt durch ein vorhandenes Messrauschen, oft als statistische Größe angesehen werden kann, sind viele Eingangsgrößen Steuergrößen des Prozesses, die keiner statistischen Verteilung unterliegen und entsprechend der Prozessvorgaben gewählt werden. Die Messpunkte des Datensatzes entsprechen so nicht den Forderungen nach Unabhängigkeit und Zufälligkeit der Stichprobe, womit die Schätzung der Verteilungsparameter einem systematischen Fehler unterliegt. Sollen verschiedene Eingangsbereiche mit unterschiedlicher Intensität vermessen werden, verschärft sich die Diskrepanz zwischen mathematischen Forderungen und praktischer Umsetzung weiter. Eine ausschließlich zufällige Wahl der Eingangswerte ist zwar möglich, erfordert jedoch die gleichmäßige Verteilung im Eingangsraum, wodurch der „Fluch der Dimensionalität“ voll zum tragen kommt.

Die Berechnung der Wichtungsfunktionen nach Gleichung (2.83) erfordert für jeden Ausgabewert die Bestimmung der Normalverteilungsdichten aller Komponenten. Mit dem erheblichen Rechenaufwand für die Berechnung der multivariaten Normalverteilungsdichten ist die Umsetzung mit den Ressourcen aktueller Motorsteuergeräte nicht zu realisieren. Ein Vergleich des Rechenaufwands mit einer ILMON-Modellierung wird im Kapitel 5.1.5 durchgeführt.

Nachfolgend werden die Eigenschaften einer GMR-Modellierung unter den geforderten Kriterien zusammengefasst:

1. *Eignung für hochdimensionale Eingangsräume*
   Die Struktur der Ausgangsgleichung und der Optimierungsalgorithmus der GMR eignen sich gut für die Modellierung höherdimensionaler Prozesse.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Prinzipiell kommt die Multimodellstruktur eines GMR-Modells einer Clusterung der Messdaten sehr entgegen, dies allerdings nur, wenn die lokale Verteilung der Messdaten in den Clustern einer Normalverteilung entspricht bzw. aus diesen zusammengesetzt werden kann. Ist dies nicht der Fall, kommt es zu großen systematischen Fehlern. Das Interpolationsverhalten in Bereichen ohne Messdaten neigt zu nichtmonotonem Verhalten. Im Zusammenhang mit höheren Eingangsdimensionen ist eine Vorhersage des Verlaufes in solchen Gebieten kaum noch möglich.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Die Modellgüte kann durch die Erhöhung der Anzahl der Modellkomponenten verbessert werden, welche global festgelegt werden muss. Eine lokale Vorgabe der Kompo-

nentenanzahl in einem definierten Modellbereich und damit einer lokalen Gütevorgabe ist im EM-Algorithmus nicht möglich.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Die Anzahl der Komponenten muss als Startparameter des EM-Algorithmus zur Optimierung des Modells festgelegt werden. Die Anzahl der Komponenten kann aber in mehreren Durchläufen schrittweise erhöht werden, bis die optimale Komponentenanzahl für die geforderte Modellgüte erreicht ist.

5. *Ressourcenbedarf der Schätzgleichung*
   Die Berechnung der Normalverteilungsdichten mit vollbesetzten Kovarianzmatrizen benötigt erhebliche Ressourcen und ist für eine Berechnung auf Motorsteuergeräten wenig geeignet

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Über die Vorgabe der Startwerte für den Optimierungsalgorithmus lassen sich vorhandene Modellierungen begrenzt an neue Prozesse anpassen. Eine Übernahme von Teilmodellen ist nicht möglich, da die Optimierung ausschließlich über alle Komponenten und Parameter erfolgt.

Zusammenfassung: Die hohe Flexibilität der GMR ermöglicht eine sehr genaue Approximation bei einer geringen Komplexität des Modells. Gleichzeitig wird durch die freie Ausrichtung der Normalverteilungsdichten im Eingangsraum und die Möglichkeit der Überlagerung der verschiedenen Komponenten die Interpretierbarkeit stark eingeschränkt. Das Interpolationsverhalten bei ungleichmäßig im Eingangsraum verteilten Daten ist schwer vorhersehbar. Die statistischen Grundlagen des Verfahrens erfordern eine besondere Güte der Messdaten, welche in der Praxis der Motorenentwicklung oft nicht eingehalten werden kann. Der Ressourcenbedarf steigt mit der Anzahl der Komponenten und der Dimension des Eingangsraumes erheblich.

2.4.8. Local-Linear-Model-Tree-Algorithmus (LOLIMOT)

Struktur eines LOLIMOT-Modell

Ein allgemeiner Ansatz zur Modellierung eines komplexen, nichtlinearen Prozesses ist die Aufteilung des Gesamtprozesses in mehrere lokal-lineare Teilmodelle und die Approximation als stückweise lineares System. Diese Idee wird in vielen verschiedenen Ansätzen und Anwendungen umgesetzt, z.B. Takagi-Sugeno-Fuzzy-Modell, Local Model Network oder Operating Regime Based Model [52].

Basierend auf diesem Konzept hat Nelles und Isermann [53] mit LOLIMOT (local linear model tree) einen iterativen Ansatz zur Modellierung vorgeschlagen, indem eine iterative Aufteilung, des von den Eingangsgrößen aufgespannten Eingangsraumes, in K verschiedene Teilbereiche erfolgt und jeder einzelnen dieser Bereiche über eine lineare Funktion $y_k(\boldsymbol{u})$ mit $\boldsymbol{u} \in \mathbb{R}^{1\times p}, k = \{1, \ldots, K\}$ approximiert wird. Die Definition, in welchen Bereichen des Eingangsraumes die linearen Teilfunktionen gültig sind, erfolgt über sogenannte Validierungsfunktionen $\Phi_k(\boldsymbol{u})$ mit einem Wertebereich von $0 \leq \Phi_k \leq 1$. Der Modellausgang ergibt sich aus der Summe der mit der Validierungsfunktion $\Phi_k(\boldsymbol{u})$ gewichteten, linearen

Teilmodelle

$$\hat{y}(\boldsymbol{u}) = \sum_{k=1}^{K} \Phi_k(\boldsymbol{u}) y_k(\boldsymbol{u}). \tag{2.85}$$

Der Ausgang der linearen Teilmodelle wird berechnet mit

$$\begin{aligned} y_k(\boldsymbol{u}) &= \begin{bmatrix} 1 & \boldsymbol{u} \end{bmatrix} \boldsymbol{c}_k^T \\ &= c_{k0} + c_{k1}u_1 + c_{k2}u_2 + \ldots + c_{kp}u_p \end{aligned} \tag{2.86}$$

mit dem Vektor der Regressionskoeffizienten $\boldsymbol{c}_k \in \mathbb{R}^{1\times p+1}$ des Teilmodells k. Für einen stetigen Übergang zwischen den linearen Teilmodellen ist die Validierungsfunktion als achsenorthogonale, normierte Gauß-Funktion definiert [8]:

$$\Phi_k(\boldsymbol{u}) = \frac{\varphi_k(\boldsymbol{u})}{\sum_{j=1}^{K} \varphi_j(\boldsymbol{u})} \tag{2.87}$$

mit

$$\varphi_k(\boldsymbol{u}) = exp\left(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{u} - \boldsymbol{\mu}_k)^T\right) \tag{2.88}$$

als Aktivierungsfunktion. $\boldsymbol{\Sigma}_k \in \mathbb{R}^{p\times p}$ ist als Diagonalmatrix mit unterschiedlichen Standardabweichungen $\sigma_{k,i}$ in den einzelnen Dimensionen $i = \{1, \ldots, p\}$

$$\boldsymbol{\Sigma}_k(\boldsymbol{u}) = \begin{bmatrix} \sigma_{k,1} & 0 & \ldots & 0 \\ 0 & \sigma_{k,2} & \ldots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \ldots & 0 & \sigma_{k,p} \end{bmatrix}$$

definiert und $\boldsymbol{\mu}_k \in \mathbb{R}^{1\times p}$ der Vektor der Zentrumskoordinaten der Gaußfunktion. Das Modellierungs- und Optimierungsproblem in LOLIMOT ist in zwei Teilbereiche gegliedert:

1. Schätzen der Parameter der lokal-linearen Teilmodelle $y_k(\boldsymbol{u})$
2. Identifikation der Aktivierungsfunktion $\varphi_k(\boldsymbol{u})$.

Lokale Parameterschätzung

Zur Schätzung der Ausgangsgröße $\hat{y}$ eines Teilmodells müssen die Modellstruktur und deren Parameter nach Gleichung (2.85), (2.86), (2.87), (2.88) bestimmt werden. Sind die Validierungsfunktionen Φ_k bekannt, reduziert sich die Parameterschätzung auf ein lineares Optimierungsproblem der Parameter $\boldsymbol{c}_k$. Hierfür werden in [54] und [8] mit der lokalen und der globalen Schätzung zwei mögliche Ansätze vorgeschlagen. Bei der globalen Schätzung werden die Parameter $\boldsymbol{c}_k$ aller Teilmodelle gemeinsam optimiert. Die assoziierte Designmatrix enthält dabei die Koeffizienten aller Teilmodelle für alle Datenpunkte. Der Berechnungsaufwand des Algorithmus' erhöht sich bei diesem Ansatz kubisch mit der Anzahl der Parameter $p + 1$ und der Anzahl der Teilmodelle K [8].

Im lokalen Ansatz wird die gegenseitige Beeinflussung der Teilmodelle vernachlässigt und die Parameter jedes einzelnen Teilmodells werden separat geschätzt. Der Teilmodellausgang

$$\hat{y}_k = \boldsymbol{X}_k \boldsymbol{c}_k \tag{2.89}$$

und der Parametervektor $\boldsymbol{c}_k$ werden über die Designmatrix

$$\boldsymbol{X}_k = \begin{bmatrix} 1 & u_{1,1} & u_{2,1} & \dots & u_{p,1} \\ 1 & u_{1,2} & u_{2,2} & \dots & u_{p,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_{1,N} & u_{2,N} & \dots & u_{p,N} \end{bmatrix} \in \mathbb{R}^{N \times (p+1)} \tag{2.90}$$

berechnet, mit der Anzahl der Messpunkte N. Die Schätzung des Parametervektors erfolgt über die Minimierung der Verlustfunktion

$$J_k = \sum_{i=1}^{N} \Phi_k(\boldsymbol{u}_i)(y_i - \hat{y}_i)^2. \tag{2.91}$$

Die Validierungsfunktion Φ_k wichtet in Gleichung (2.91) den Modellfehler in Abhängigkeit der Entfernung der Datenpunkte von den Zentren der Validierungsfunktion. Datenpunkte in angrenzenden Teilmodellen oder weiter entfernten Modellbereichen haben damit praktisch keine Auswirkungen auf die lokalen Parameter eines Teilmodells. Die Minimierung der Gleichung (2.91) ergibt

$$\boldsymbol{c}_k = \left(\boldsymbol{X}_k^T \boldsymbol{Q}_k \boldsymbol{X}_k \right)^{-1} \boldsymbol{X}_k^T \boldsymbol{Q}_k \boldsymbol{y} \tag{2.92}$$

mit der Wichtungsmatrix

$$\boldsymbol{Q}_k = \begin{bmatrix} \Phi_k(\boldsymbol{u}_1) & 0 & \dots & 0 \\ 0 & \Phi_k(\boldsymbol{u}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Phi_k(\boldsymbol{u}_N) \end{bmatrix}. \tag{2.93}$$

Im Hinblick auf die geforderten Kriterien bietet die lokale Parameterschätzung einige Vorteile. Zum einen ergibt sich im Vergleich mit der globalen Schätzung ein kleinerer Varianzfehler bei verrauschten Messdaten und dünn besetzten Modellbereichen. Da Letzteres gerade in hochdimensionalen Eingangsräumen kaum zu vermeiden ist, hat die lokale Parameterschätzung unter diesen Bedingungen Vorteile. Weiterhin kann ein einzelnes Teilmodell, als linearisierter Arbeitspunkt des Prozesses interpretiert werden. Mit dieser Sichtweise ist eine einfache Integration von Expertenwissen über die Festlegung der Aktivierungsfunktion (2.88) möglich. Neben diesen Aspekten verringert sich auch der Berechnungsaufwand im Vergleich zur globalen Schätzung. Der Aufwand erhöht sich kubisch mit der Anzahl der Parameter $(p+1)$, jedoch linear mit der Teilmodellanzahl K.

Nachteil der lokalen Parameterschätzung ist der erhöhte Bias des optimierten Modells, welcher durch die Vernachlässigung der Interaktion der Teilmodelle hervorgerufen wird. Für eine weiterführende Betrachtung der Eigenschaften dieser Ansätze sei auf [8] verwiesen. Aufgrund der Vorteile in Modellen mit hochdimensionalen Eingangsräumen wird sich in

dieser Arbeit auf die lokale Parameterschätzung beschränkt.

LOLIMOT-Partitionierungsalgorithmus

Wie im letzten Abschnitt dargestellt, ist die Schätzung der Modellparameter $\boldsymbol{c}_k$ ein lineares Optimierungsproblem und mathematisch gut handhabbar. Die Schätzung der Parameter $\boldsymbol{\mu}_k$ und $\boldsymbol{\Sigma}_k$ der Aktivierungsfunktionen aus Gleichung (2.88) ist dagegen ein nichtlineares Optimierungsproblem mit den bekannten Schwierigkeiten in Hinblick auf das Finden des globalen Optimums. In [53] wird für diese Optimierung ein heuristischer, iterativer Algorithmus für ein lokal-lineares Modellnetzwerk vorgeschlagen, welches eine achsenorthogonale Aufteilung des Eingangsraumes auf Grundlage des Modellfehlers vornimmt.

Grundprinzip dieses „local linear model tree"-Algorithmus' ist die Erhöhung der Komplexität des Modells durch Aufteilung eines Teilmodells in zwei neue Teilmodelle. Die zugehörigen Aktivierungsfunktionen werden in Abhängigkeit der Grenzen dieser Teilmodelle definiert. Das Vorgehen soll im Folgenden kurz beschrieben werden, weiterführende Informationen sind in [8], [53] zu finden.

Der Algorithmus startet mit der Annahme eines globalen linearen Modells über den Grenzen des gesamten Eingangsraumes und schätzt den Parametervektor $\boldsymbol{c}_0$ mit der Methode der kleinsten Quadrate aus dem zur Verfügung stehenden Datensatz. Die Validierungsfunktion wird als $\Phi_0(\boldsymbol{u}) = 1$ definiert. Nach dieser Initialisierung werden folgende Schritte iterativ ausgeführt:

1. Bestimmung des lokalen Teilmodells y_l mit dem höchsten lokalen Fehler in der Verlustfunktion (2.91) über alle Teilmodelle: $\max(J_k) \rightarrow l = k$.

2. Teilung des Teilmodells y_l mit dem schlechtesten Fehlerwert in zwei Teilmodelle, jeweils achsenorthogonal in allen Eingangsdimensionen p. Die Teilung erfolgt mittig zwischen den Grenzen des ursprünglichen Teilmodells in der jeweiligen Eingangsdimension $b_{min} \in \mathbb{R}^{1\times p}$, $b_{max} \in \mathbb{R}^{1\times p}$. Im Ergebnis gibt es $d = \{1 \dots p\}$ verschiedene Teilungen.

3. Berechnung der Aktivierungsfunktion $\varphi_{1,d}$ und $\varphi_{2,d}$ für alle Teilungen d.

4. Lokale Schatzung der Parameter $\boldsymbol{c}_{1,d}$ und $\boldsymbol{c}_{2,d}$ für alle Teilungen d.

5. Auswahl der besten Teilung anhand des kleinsten Wertes der Verlustfunktion

$$J_d = \sum_{i=1}^{N} (y - \hat{y}_d)^2$$

   für das jeweilige Gesamtmodell y_d: $D = d \quad \text{für} \quad \min(J_d)$.

6. Übernahme der beiden aus der besten Teilung D resultierenden Teilmodelle $y_{1,D}$ und $y_{2,D}$ mit den Parametern $\Phi_{1,D}$, $\Phi_{2,D}$, $c_{1,D}$, $c_{2,D}$ in das Gesamtmodell und Erhöhung der Anzahl der Teilmodelle von $K \rightarrow K + 1$.

7. Abbruch des Algorithmus bei Konvergenz, sonst Beginn einer neuen Iteration bei Schritt 1

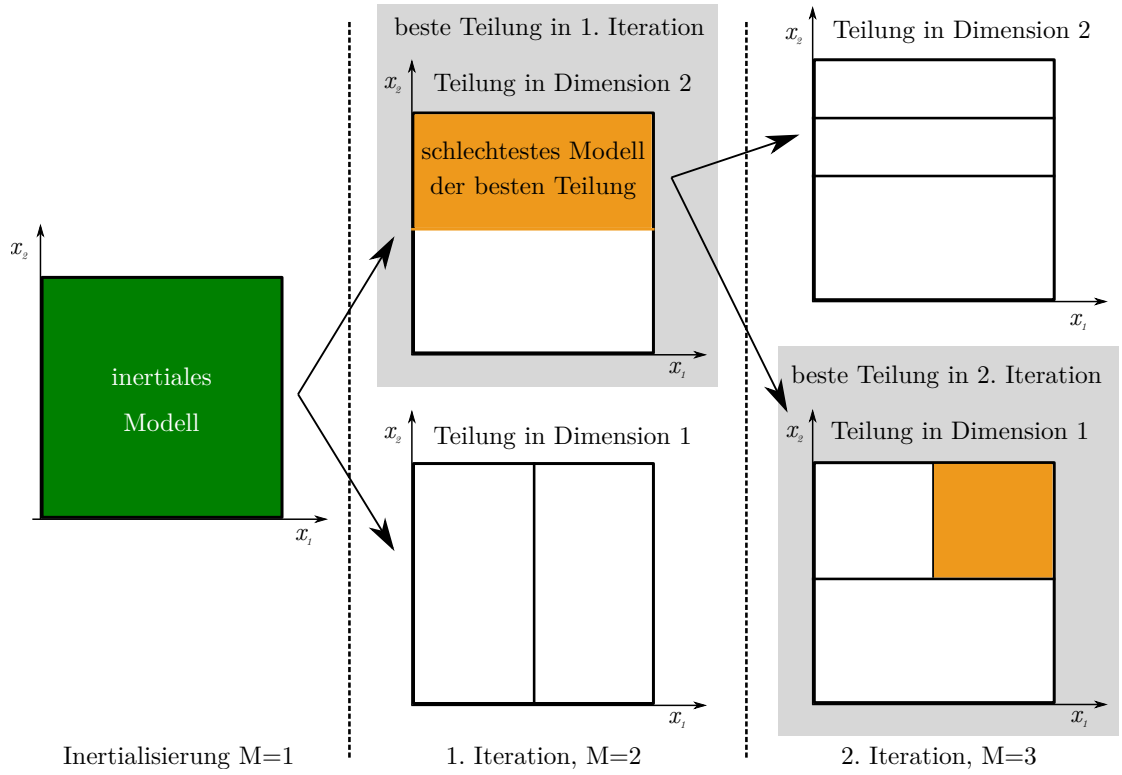

Abbildung 2.17.: Prinzip der Partitionierung unter LOLIMOT bei einem Modell mit 2 Eingangsgrößen $p = 2$

Das Prinzip des Partitionierungsalgorithmus' ist am Beispiel eines Prozesses mit 2 Eingangsgrößen in Bild 2.17 dargestellt.

Wie oben erwähnt, erfolgt die Berechnung des Parameters $\mathbf{\Sigma}_k$ der Aktivierungsfunktionen $\varphi_{1,d}$ und $\varphi_{2,d}$ in Schritt 3 in Abhängigkeit der Grenzen $\mathbf{\Delta}_k$ des korrespondierenden Teilmodells y_k:

$$\mathbf{\Sigma}_k = s_\Sigma \cdot \mathbf{\Delta}_k \tag{2.94}$$

mit

$$\mathbf{\Delta}_k = \begin{bmatrix} \Delta_{1,k} & 0 & \dots & 0 \\ 0 & \Delta_{2,k} & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \Delta_{p,k} \end{bmatrix}. \tag{2.95}$$

Die einzelnen Grenzen $\Delta_{i,k}$ des Teilmodells k in der Eingangsdimension i berechnen sich mit

$$\Delta_{i,k} = b_{i,k,max} - b_{i,k,min} \tag{2.96}$$

In Gleichung (2.94) ist s_Σ der Designparameter, mit dem das Maß der Überlagerung der Teilmodelle festgelegt werden kann. Ein großes s_Σ bedeutet einen glatteren Übergang zwischen den Teilmodellen, jedoch fließen die umliegenden Datenpunkte stärker in die Parameter-

schätzung eines Teilmodells mit ein, wodurch die Lokalität der Parameter reduziert wird. Der optimale Wert von s_Σ ist abhängig vom konkreten Anwendungsfall und kann über die Berechnung des Modellfehler für verschiedene s_Σ bestimmt werden. Ein guter Praxiswert für die initiale Berechnung ist $s_\Sigma = 0.33$ [8].

Obwohl der Algorithmus bei der Partitionierung des Eingangsraumes hierarchisch arbeitet, entsteht als Ergebnis eine parallele Modellstruktur, die sich, im Zusammenspiel mit der achsenorthogonalen Teilung, gut für eine nachträgliche Interpretation des Modells eignet. Weiterhin kann die Teilmodellanzahl indirekt über den Modellfehler bestimmt werden, indem als Abbruchkriterium für die Iteration in Schritt 5 eine Schwelle für den Modellfehler definiert wird.

Nachfolgend sind die Eigenschaften des LOLIMOT-Algorithmus im Hinblick auf den in Kapitel 2.3.1 aufgestellten Kriterien zusammengefasst.

1. *Eignung für hochdimensionale Eingangsräume*
   Die Struktur der Ausgangsgleichung gestattet auch die Modellierung höherdimensionaler Probleme. Durch die ausschließlich diagonal besetzten Wichtungsmatrizen Σ_k wird die Flexibilität, insbesondere bei der Modellierung nichtachsenorthogonaler Nichtlinearitäten, jedoch deutlich beschränkt. Bei nichtmonotonen Verläufen in Randbereichen des Modells beziehungsweise zur Approximation achsenschräger Nichtlinearitäten ist oft eine größere Anzahl von Teilmodellen notwendig.

2. *Eignung bei geclusterter und dünn besetzter Datenbasis*
   Die Approximation über lineare Teilmodelle ermöglicht es, auch große, spärlich mit Messdaten besetzte Bereiche des Prozesses zu modellieren und ohne weitere Testdaten zu validieren. Die Normierung der Aktivierungsfunktionen führt in bestimmten Konstellationen zu einem unerwarteten Extra- und Interpolationsverhalten.

3. *Vorgabe unterschiedlicher Modellgenauigkeit für verschiedene Modellbereiche*
   Es können für jedes Teilmodell unterschiedliche Güteanforderungen definiert werden, bei deren Erreichen das Teilmodell für eine weitere Teilung innerhalb des Algorithmus gesperrt wird.

4. *Modellierung auch ohne Vorgabe der Modellkomplexität*
   Die iterative Erhöhung der Teilmodellanzahl zur Verbesserung des Modells ermöglicht es, die Komplexität in Abhängigkeit der erreichten Modellgüte zu definieren.

5. *Ressourcenbedarf der Schätzgleichung*
   Die Berechnung der Gauß-Funktionen und deren Normierung erfordert eine hohe Rechenleistung. Im Vergleich zur Gaussian-Mixture-Regression oder der Support-Vektor-Regression mit Gauß-Kernen reduziert sich der Ressourcenbedarf durch die nicht besetzen Nebenelemente der Wichtungsmatrix allerdings erheblich.

6. *Adaptionsmöglichkeit vorhandener Modellierungen*
   Die Möglichkeit des Starts des Optimierungsalgorithmus mit einer heuristisch festgelegten Partitionierung gestattet eine gute Adaption bestehender Modelle. Weiterhin können in solchen Strukturen auch Teilmodelle gesperrt werden, sodass eine Optimierung nur in bestimmten Bereichen des Eingangsraumes erfolgt.

Zusammenfassung: Der LOLIMOT-Algorithmus eignet sich gut zur Modellierung höherdimensionaler Prozesse und bietet durch die achsenorthogonale Partitionierung des Eingangsraumes in lineare Teilmodelle eine gute Interpretationsmöglichkeit des optimierten Modells. Diese kann jedoch bei einer stärkeren Überlagerung mehrerer Teilmodelle erheblich beeinträchtigt werden. Die Flexibilität des Verfahrens ist durch die achsenorthogonale Teilung und dem starren Partitionierungsalgorithmus erheblich eingeschränkt. Die Berechnung der normierten Gaußfunktionen erfordert bei einer großen Anzahl an Teilmodellen einen erheblichen Rechenaufwand.

3. ILMON - ein gut interpretierbarer Modellansatz

Ausgehend von den in Abschnitt 2.3.1 aufgestellten Kriterien zur Interpretierbarkeit einer Modellstruktur sollen in diesem Kapitel Vorschläge zur Verbesserung bekannter Verfahren erarbeitet werden. Nachfolgend wird ein neues, gut interpretierbares datenbasiertes Modellierungsverfahren für hochdimensionale Prozesse vorgestellt, welches die Integration von a-priori-Wissen zulässt und für den Einsatz in Motorsteuergeräten geeignet ist. Dabei soll ein Hauptaugenmerk auf die Robustheit bezüglich des Verhaltens bei wenigen, ungleichmäßig verteilten Messdaten gelegt werden.

3.1. Struktur eines interpretierbaren Modellansatzes

In dem vorangegangenen Kapitel wurde verschiedene Verfahren zur datenbasierten Modellierung vorgestellt und ihre Eigenschaften sowie die Vor- und Nachteile bezüglich der Modellierung hochdimensionaler Prozesse aufgezeigt. In Kapitel 2.2.2 wurde außerdem dargestellt, dass in den komplexen Problemstellungen der Motorsteuerung, für Prozesse mit einer hohen Anzahl an Eingängen, praktisch immer mit einer sehr geringen Anzahl an Messdaten ausgekommen werden muss. Die Sicherheitsanforderungen der Automobilindustrie verlangen weiterhin ein über alle Modellbereiche validiertes Modell, was in der Wahl der Modellstruktur sein Berücksichtigung finden muss und zu den in Kapitel 2.2.2 getroffenen Forderung nach hoher Interpretierbarkeit mit den dort aufgeführten Kriterien führt. An dieser Stelle soll nun die Wahl einer Modellstruktur diskutiert werden, die diesen Anforderungen genügt.

Komplexe Prozesse werden in der Praxis oft in Teilprobleme aufgesplittet, die sich isoliert besser untersuchen, modellieren und regeln lassen. Da viele Probleme einfacher unter der Annahme einer linearen Abhängigkeit der Ein- und Ausgangsgrößen zu lösen sind, spielt die Linearisierung von nichtlinearen Prozessen in vielen Anwendungen eine große Rolle. Ein wichtiger Aspekt dieser lokal begrenzten Linearisierung ist die einfache mathematische Darstellung der Verknüpfungen zwischen Einganggrößen $\boldsymbol{u}$ und der Ausgangsgröße in der Form

$$\alpha_k(\boldsymbol{u}, \boldsymbol{\gamma_k}) = \boldsymbol{\gamma}_k \tilde{\boldsymbol{u}} \tag{3.1}$$

mit

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} 1 & \boldsymbol{u} \end{bmatrix}^T = [1, u_1, u_2, \cdots, u_q]^T$$

und den konstanten Proportionalitätsfaktoren $\boldsymbol{\gamma}_k = [\gamma_0, \gamma_1, \gamma_2, \cdots, \gamma_q]$, über welche die einzelnen Eingänge auf den Ausgang wirken. Komplexe physikalische Zusammenhänge lassen sich oft lokal begrenzt in diesen Proportionalitätsfaktoren zusammenfassen und gestatten in dieser Form die einfache Prüfung auf Plausibilität mittels physikalischem Vorwissen. Lineare

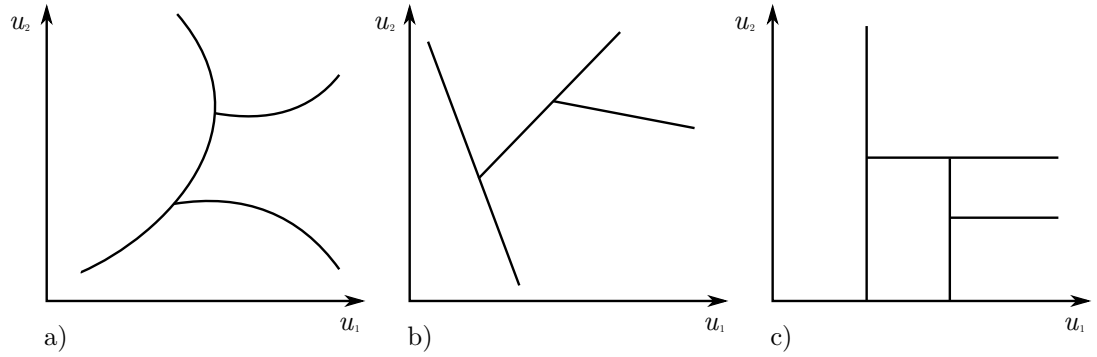

Abbildung 3.1.: Verschiedene Arten der Partitionierung des Eingangsraumes, nichtlineare Trennung (a), lineare Trennung (b), achsenortogonale Trennung (c)

Teilmodelle bieten somit eine sehr gute Möglichkeit der Interpretation der Parameter des optimierten Modells.

Die Linearisierung eines nichtlinearen Prozesses erfolgt in abgegrenzten, lokal definierten Bereichen des Modells. Die Verläufe der Grenzen dieser Teilmodelle werden durch die Wahl des Modellierungsverfahrens bzw. der Basisfunktion grundlegend vorgegeben. Deren Parameter werden wiederum in Abhängigkeit der Messdaten optimiert. Im Ergebnis dieser Optimierung kann es, je nach Verfahren, zu sehr komplexen Verläufen der Wirkungsgrenzen eines Teilmodells sowie zu einer Überlagerung mehrerer Teilmodelle kommen, was die einfache Interpretierbarkeit der Parameter aufhebt. Möchte man diesen Vorteil aufrecht erhalten, bedarf es neben der linear definierten Struktur der Teilmodelle auch eine einfache Definition der Teilmodellgrenzen.

Allgemein gilt, je mehr Freiheitsgrade bei der Definition dieser Bereichsgrenzen vorhanden sind, um so flexibler kann sich das Modell an den Prozess anpassen. Nachteil dieser Flexibilität ist die komplexe Beschreibung dieser Bereichsgrenzen, die einer einfachen physikalischen Interpretation entgegen stehen.

Bild 3.1 zeigt drei mögliche Verläufe der Teilmodellgrenzen: In Bild 3.1a sind stark nichtlineare Verläufe, wie sie bei der Support-Vektor-Regression oder der Gaussian-Mixture-Regression auftreten, dargestellt. Bild 3.1b zeigt die linearen Teilmodellgrenzen, z.B. des Hinging-Hyperplane Verfahrens [55] und in Bild 3.1c sind achsenorthogonalen Grenzen, wie sie im LOLIMOT-Verfahrens auftreten, abgebildet.

Während bei der nichtlinearen und linearen Trennung der Teilmodelle eine einfache Definition der Wirkungsbereiche schwierig ist, ermöglicht die achsenorthogonale Trennung eine intuitive Festlegung über eine Konjunktion der Geltungsbereiche in den jeweiligen Eingangsgrößen, z.B.

$$\text{Teilmodell 1} \coloneqq \begin{cases} & \text{wenn: } 700 \text{ U/min} \leq \text{Drehzahl} \leq 2500 \text{ U/min} \\ \wedge & \text{wenn: } 1 \text{ bar} \leq \text{Saugrohrdruck} \leq 2\text{bar} \\ \wedge & \text{wenn: } 30^\circ \text{ C} \leq \text{Saugrohrtemperatur} \leq 50^\circ \text{ C} \end{cases}$$

Diese Art der Definition garantiert auch in höherdimensionalen Modellierungen eine intuitive Abgrenzung der Wirkungsbereiche der Teilmodelle und erfüllt so eine Voraussetzung zur Validierung eines Modells auf Grundlage von Prozesswissen. Für die Umsetzung einer interpretierbaren Modellierung wurde sich daher in dieser Arbeit auf ein paralleles lokal-lineares

Modellnetz mit einer achsenorthogonalen Partitionierung des Eingangsraumes festgelegt. Die Ausgangsgleichung dieser Modellstruktur ergibt sich als gewichtete Summe aller Teilmodelle, siehe Gleichung (2.50) und sei hier aus Gründen der Übersichtlichkeit nochmals aufgeführt:

$$\hat{y}(\boldsymbol{u}) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{u}, \boldsymbol{\gamma_k}) \cdot \varphi_k(\boldsymbol{u}, \boldsymbol{\theta}_k) \tag{3.2}$$

mit $\alpha_k(\boldsymbol{u}, \boldsymbol{c_k})$ aus Gleichung (3.1) als lineares Teilmodell, den Basisfunktionen $\varphi_k(\boldsymbol{u}, \boldsymbol{\theta}_k)$ und der Nebenbedingung:

$$\sum_{k=1}^{K} \varphi_k(\boldsymbol{u}, \boldsymbol{\theta}_k) = 1 \tag{3.3}$$

Die achsenorthogonale Trennung ergibt sich aus Gleichung (3.2) und (3.3) nicht zwangsläufig, sondern muss in der Konstruktion der Basisfunktion berücksichtigt werden, auf die im folgenden Kapitel ausführlich eingegangen wird.

3.2. Basisfunktionen für interpretierbare lokal-lineare Netze

Ziel der Basisfunktion $\varphi(\boldsymbol{u}, \boldsymbol{\theta}_k)$ aus Gleichung (3.2) ist es, die global über den gesamten Eingangsraum wirkenden Teilmodelle $\alpha_k(\boldsymbol{u}, \boldsymbol{c}_k)$ auf einen bestimmten Wirkungsbereich zu beschränken. Dabei können für bestimmte Bereiche durchaus mehrere Teilmodelle Anteile zur Berechnung des Schätzwertes liefern. Für die Konstruktion einer multivariaten Basisfunktion gibt es verschiedene Methoden, deren wichtigste Vertreter sich nach [8] in drei Kategorien einteilen lassen:

- vektorbasierte Konstruktion
- radiale Konstruktion
- Tensorproduktkonstruktion

Vektorbasierte Konstruktion In der vektorbasierte Konstruktion wird eine Nichtlinearität nur in einer Richtung des Eingangsraumes abgebildet, welche durch einen Parametervektor $\boldsymbol{\nu}$ definiert wird. Die Projektion des erweiterten Eingangsvektors $\tilde{\boldsymbol{u}}$ auf den Parametervektor

$$x_k = \boldsymbol{\nu_k}\tilde{\boldsymbol{u}} = \nu_0\tilde{u}_0 + \nu_1\tilde{u}_1 + \cdots + \nu_q\tilde{u}_q \tag{3.4}$$

mit

$$\boldsymbol{\nu} = \begin{bmatrix} \nu_0 & \nu_1 & \dots & \nu_q \end{bmatrix}$$

geht als Argument in die jeweilige Basisfunktionsdefinition $\varphi_k(\boldsymbol{u}, \boldsymbol{\nu}_k)$ ein. Als Beispiel für die häufig genutzte Sigmoidfunktion ergibt sich dann als Basisfunktion

$$\varphi_k(\boldsymbol{u}, \boldsymbol{\nu}_k) = \frac{1}{1 + e^{-x_k}}. \tag{3.5}$$

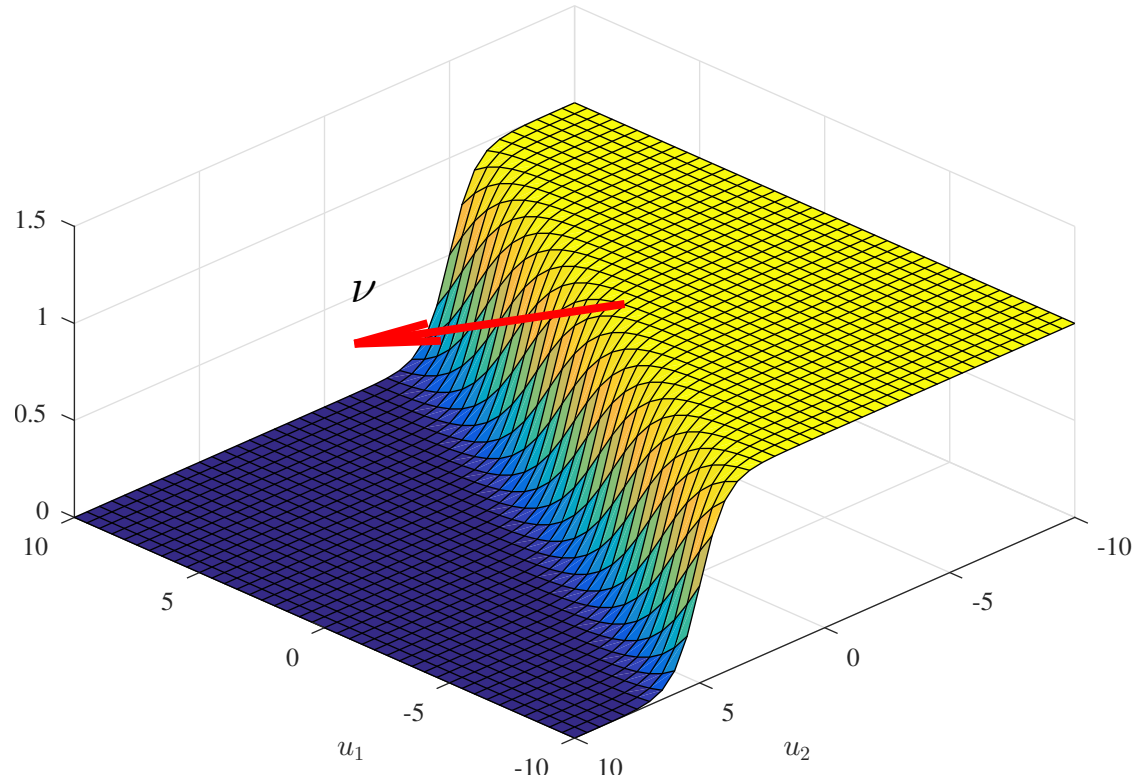

Abbildung 3.2.: Verlauf einer sigmoiden, vektorbasiert konstruierten Basisfunktion. Der Parametervektor $\boldsymbol{\nu} = [0 \quad 1 \quad 2]$ gibt die Richtung und Lage des nichtlinearen Verlaufes vor.

Der Verlauf einer so konstruierten Basisfunktion ist in Bild 3.2 dargestellt. Der Parametervektor gibt die Richtung der Nichtlinearität vor und steht in diesem Beispiel senkrecht auf der Kante der Sigmoidfunktion. Diese Art der Konstruktion der Basisfunktion wird oft in MLP-Netzen verwendet.

Radiale Konstruktion Die radiale Konstruktion nutzt als Argument zur Berechnung der Basisfunktion den Abstand des Eingangsvektors $\boldsymbol{u}$ zum Zentrum der Basisfunktion:

$$x_i = \|\boldsymbol{u} - \boldsymbol{c}_i\| \tag{3.6}$$

mit dem Vektor der Zentrumsposition

$$\boldsymbol{c}_i = \begin{bmatrix} c_1 & c_2 & \dots & c_q \end{bmatrix}.$$

Die Wahl der Norm in Gleichung (3.6) ist dabei freigestellt. Häufig findet die euklidische Norm Verwendung, welche die Abstände in allen Dimensionen symmetrisch um das Zentrum bewertet. Mehr Freiheitsgrade bietet die Mahalanobis-Distanz

$$x(\boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\Sigma}) = \|\boldsymbol{u} - \boldsymbol{c}\| = \sqrt{(\boldsymbol{u} - \boldsymbol{c})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{u} - \boldsymbol{c})}, \tag{3.7}$$

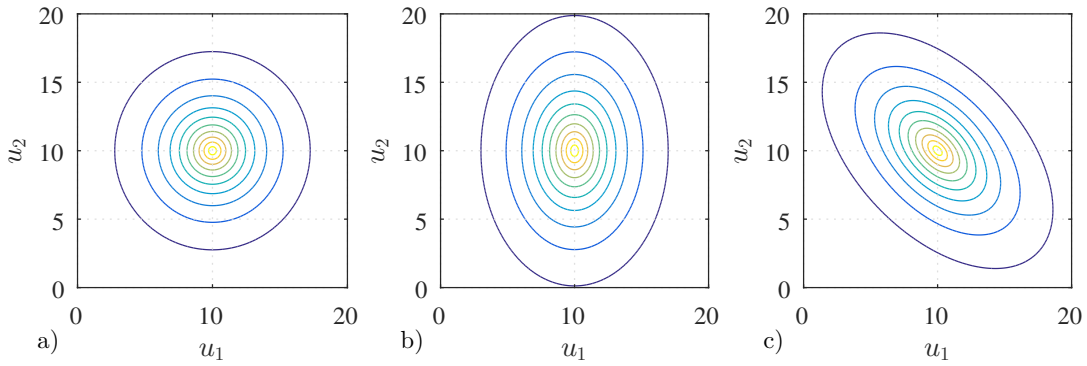

Abbildung 3.3.: Isolinien radial konstruierter Basisfunktionen: a) Euklidische Norm b) Mahalanobis-Distanz mit Σ als Diagonalmatrix c) Mahalanobis-Distanz mit voll besetzter Matrix Σ

.

welche über die Kovarianzmatrix $\mathbf{\Sigma}$ die Skalierung und Ausrichtung der nun elliptische Basisfunktion separat in allen Eingangsdimensionen zulässt. Ist ausschließlich die Hauptdiagonale der Kovarianzmatrix besetzt, wird nur eine achsenorthogonale Skalierung durchgeführt. Mit den besetzten Nebendiagonalen erfolgt eine Rotation um das Zentrum.

In Bild 3.3 sind die prinzipiellen Verläufe der Basisfunktionen mit euklidischer Norm und Mahalanobis-Distanz dargestellt. Die achsenorthogonale Skalierung über eine Diagonalmatrix findet im LOLIMOT-Verfahren Anwendung, siehe Gleichung (2.94) und (2.95). Die Konstruktion mit vollständig besetzter Kovarianzmatrix wird z.B. in der Gaussian-Mixtur-Regression verwendet, siehe Gleichung (2.67) und (2.83).

Tensorproduktkonstruktion Die Tensorproduktkonstruktion ermöglicht es univariate Funktionen multivariat zu erweitern. Breite Anwendung findet diese Art der Basisfunktionskonstruktion in Neuro-Fuzzy-Modellen bei der Konstruktion von multidimensionalen Zugehörigkeitsfunktionen und bei der Definition von mehrdimensionalen Spline-Funktionen. Die multivariate Basisfunktion φ_k wird dabei aus dem Produkt der univariaten Basisfunktionen $\varphi_{k,i}$ aller Eingangsdimensionen q gebildet:

$$\varphi_k(\boldsymbol{u}) = \prod_{i=1}^{q} \varphi_{k,i}(u_i). \tag{3.8}$$

In Bild 3.4 wird die Konstruktion einer zweidimensionalen Basisfunktion als Produkt aus zwei unterschiedlichen univariaten Funktionen der beiden Eingangsgrößen illustriert. Die Einzelfunktionen unterliegen dabei keinen Beschränkungen und können auch unstetig gewählt werden. Bei der Definition der Funktionen im abgeschlossenen Einheitsintervall $[0, 1]$ ergibt sich die resultierende Basisfunktion ebenfalls im Einheitsintervall. Durch die Abhängigkeit der univariaten Basisfunktionen von jeweils nur einer Eingangsdimension werden über die Tensorproduktkonstruktion immer achsenorthogonale multivariate Basisfunktionen erzeugt. Die Flexibilität gegenüber der radialen und vektorbasierten Konstruktion wird dadurch deutlich eingeschränkt.

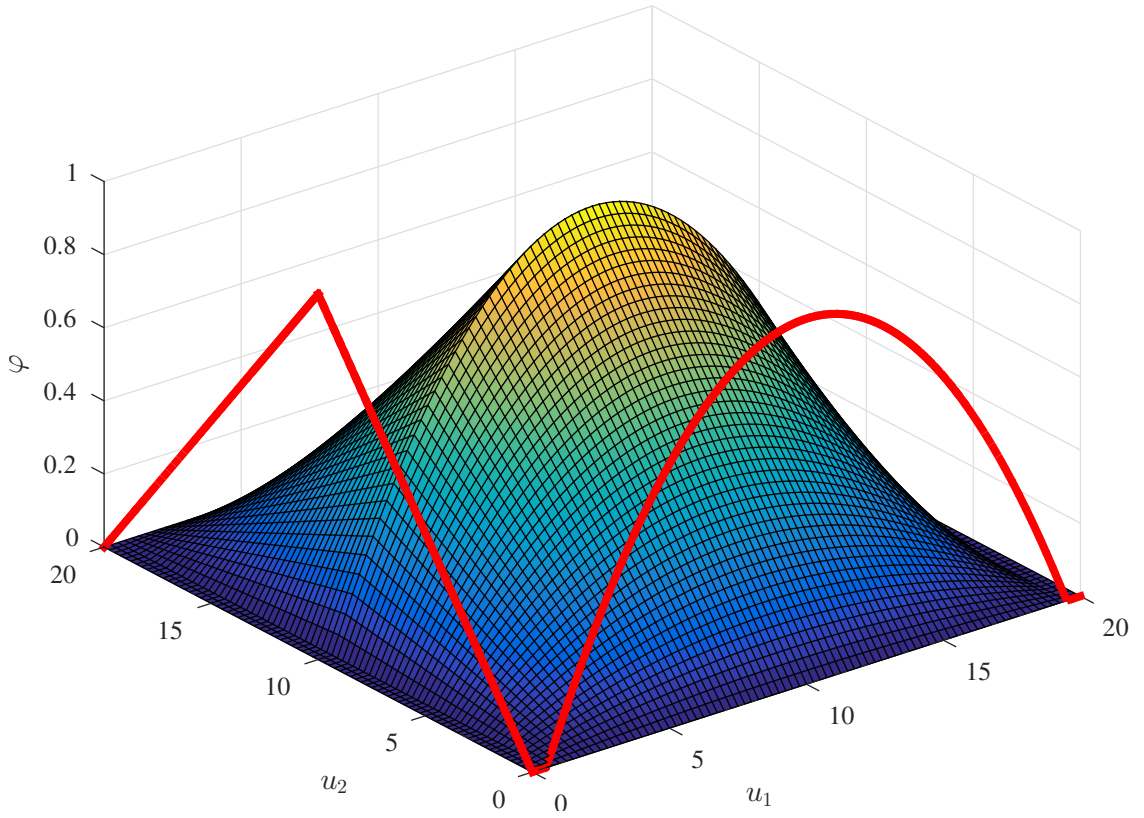

Abbildung 3.4.: Verlauf einer zweidimensionalen Basisfunktion als Produkt zweier univariater Funktionen, hier rot eingezeichnet

3.2.1. Normierte multivariate Gaußfunktion als Basisfunktion

In parallelen lokal-linearen Modellen werden zur Definition des Geltungsbereichs oft Gaußsche Funktionen verwendet, welche den Vorteil haben, beliebig oft ableitbar zu sein. Diese Funktionen können definiert werden als:

$$\mu(\boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\Sigma}) = \exp\left(\sqrt{(\boldsymbol{u}-\boldsymbol{c})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{u}-\boldsymbol{c})}\right) \tag{3.9}$$

mit dem Eingangsvektor $\boldsymbol{u}$, dem Vektor der Zentrumsparameter $\boldsymbol{c} = [c_1, c_2, \ldots, c_q]$ und der Kovarianzmatrix $\boldsymbol{\Sigma}$. Die Funktionen μ_k werden in der Terminologie der Neuronal-Fuzzy-Modelle als Zugehörigkeitsfunktionen bezeichnet, was an dieser Stelle zur besseren sprachlichen Unterscheidung übernommen werden soll.

Mit dem Wertebereich im Intervall $(0, 1]$ ist das mit Gleichung (3.9) gewichtete Teilmodell nur im Zentrum vollständig gültig, weshalb eine Normierung über die Summe aller Gauß-Funktionen eingeführt wird:

$$\varphi_k = \frac{\mu_k(\boldsymbol{u}, \boldsymbol{c_k}, \boldsymbol{\Sigma_k})}{\sum_{k=1}^{K} \mu_k(\boldsymbol{u}, \boldsymbol{c_k}, \boldsymbol{\Sigma_k})} \tag{3.10}$$

Mit dieser Normierung ist sichergestellt, dass die Summe aller normierten Basisfunktionen φ_k und damit die Wichtung an jedem Punkt des Eingangsraumes gleich 1 und somit der

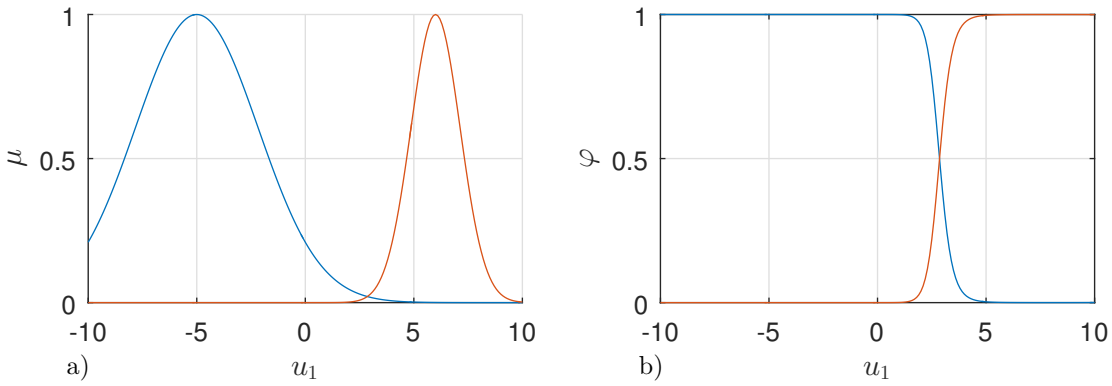

Abbildung 3.5.: a) Verlauf zweier Gaußscher Zugehörigkeitsfunktionen μ und b) der zugehörigen normierten Basisfunktionen φ

Schätzwert aus der Summe aller linearen Teilmodelle nicht verfälscht wird. Es gilt also:

$$\sum_{k=1}^{K} \varphi_k = 1 \tag{3.11}$$

In Bild 3.5 ist die Normierung für zwei univariate Gaußsche Zugehörigkeitsfunktionen und den resultierenden normierten Basisfunktionen dargestellt. Ein Nachteil dieser Normierung ist die Verkopplung der Parameter aller Gauß-Funktionen und damit die implizite Festlegung der Gültigkeitsbereiche durch das Verhältnis der Zugehörigkeitsfunktionen. Zum einen erschwert dies die Einbringung von a priori Wissen in die Modellierung und zum anderen führt dies zu einigen unerwünschten Nebeneffekten, auf die im Folgenden genauer eingegangen werden soll.

Mit der Berechnung der Basisfunktionen nach Gleichung (3.10) ergeben sich die Gültigkeitsbereiche bzw. die Teilmodellgrenzen sowie die Übergangsbereiche zwischen den Teilmodellen aus den Parametern $\mathbf{\Sigma}_k$ und $\boldsymbol{c}_k$ der Zugehörigkeitsfunktionen. Die Definition der Kovarianzmatrix bestimmt dabei, ob diese Aufteilung achsenorthogonal oder achsenschräg erfolgt. In Bild 3.6 ist die Partitionierung eines zweidimensionalen Eingangsraumes durch Gauß-Funktionen mit einer vollbesetzten Matrix $\mathbf{\Sigma}$ dargestellt. Die achsenschräge Aufteilung bietet eine sehr flexible Anpassung an Nichtlinearitäten, hat jedoch den Nachteil, dass sich die Teilmodellgrenzen nicht mehr direkt in den einzelnen Eingangsdimensionen abbilden. Variieren die Breiten der verwendeten Gauß-Funktionen erheblich, kommt es weiterhin zu sehr stark verrundeten Verläufen und in den Grenzgebieten mehrere Teilmodelle überlagern sich diese in schwer vorhersehbarer Weise. In hochdimensionalen Eingangsräumen mit einer großen Anzahl an Teilmodellen ergeben sich so sehr komplexe Grenzverläufe, welche die Interpretierbarkeit stark einschränken, siehe Bild 3.6d.

Mit einer Definition von $\mathbf{\Sigma}$ als Diagonalmatrix resultiert eine Aufteilung des Eingangsraumes in achsenorthogonale Teilmodelle. Da die Eingangsgrößen über die nicht besetzten Nebenelemente der Matrix entkoppelt werden, kann die Lage der Teilmodelle im Eingangsraum über simple und intuitive Konjunktionen der einzelnen Eingangsgrößen definiert werden. Diese ist so auch bei hohen Eingangsdimensionen gut adressierbar. In Bild 3.7 ist solch eine Partitionierung mit drei Gauß-Funktionen beispielhaft dargestellt. Die Zentren der Gauß-Funktionen sind hier mittig in die gewünschten Teilmodelle gelegt bzw. die Lage der

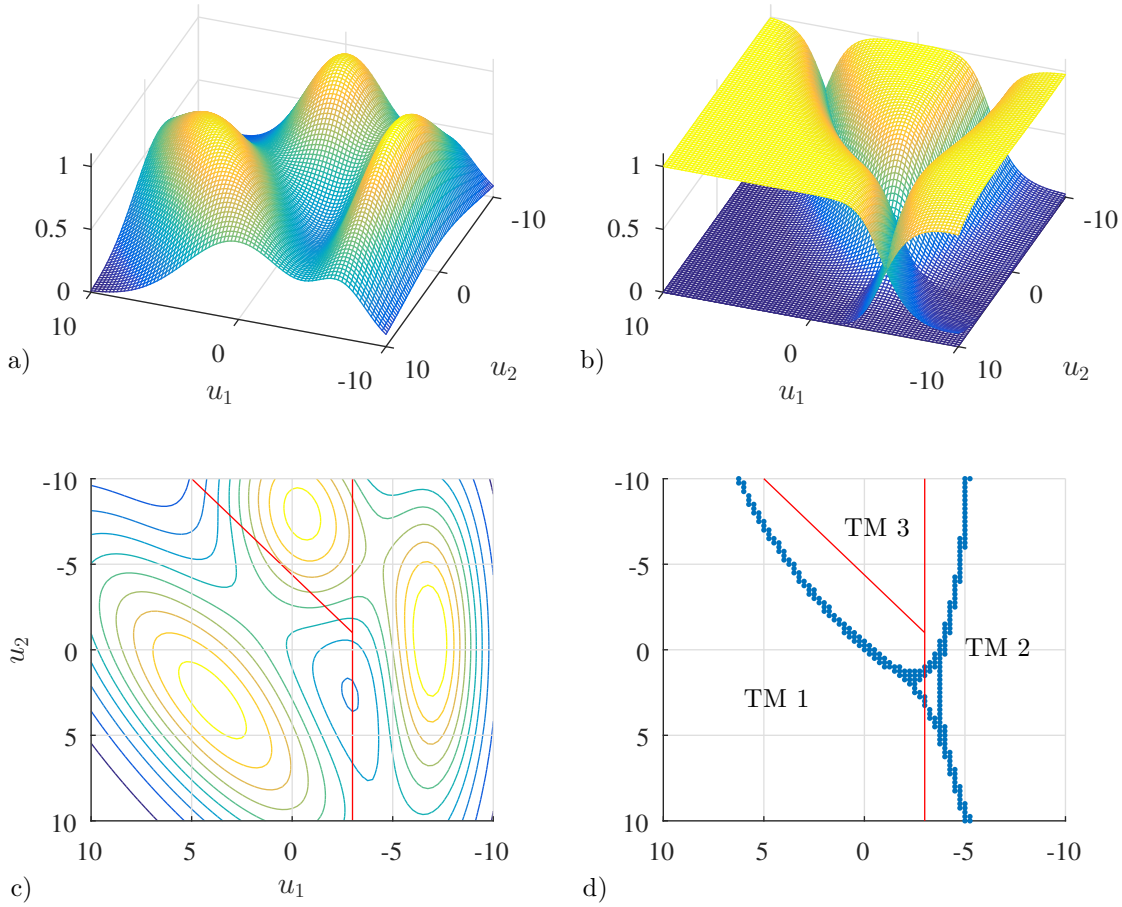

Abbildung 3.6.: Achsenschräge Partionierung mit Gaußschen Zugehörigkeitsfunktionen, a) Achsenschräge Gauß-Funktionen, b) zugehörige normierte Basisfunktionen, c) Projektion der Gauß-Funktionen, d) resultierende Teilmodellgrenzen $\varphi = 0,5$

Teilmodelle kann über die optimierten Parameter der Gauß-Funktionen bestimmt werden. Die Abbildung 3.7d zeigt, dass es bedingt durch die unterschiedlichen Breiten der Zugehörigkeitsfunktionen auch im achsenorthogonalen Fall zu Abrundungen und Verschiebungen der Teilmodellgrenzen kommt. Diese fallen jedoch moderater aus als bei der achsenschrägen Anordnung der Gaußschen Zugehörigkeitsfunktionen.

Überlagert eine Gaußsche Zugehörigkeitsfunktion mit großer Varianz andere Zugehörigkeitsfunktionen, kann es auf Grund der Normierung in unerwarteten Bereichen zu einer Reaktivierung von Teilmodellen kommen. Die ausschließlich lokale Gültigkeit eines Teilmodells ist so nicht mehr garantiert. In Bild 3.8 ist dieses Szenario als Beispiel dargestellt. Die Gefahr eine Reaktivierung besteht hauptsächlich bei stark unterschiedlichen Varianzen der Gauß-Funktionen, wie sie bei der Modellierung von starken Nichtlinearitäten in lokallinearen Modellen auftreten können. In [56] wird als Bedingung für eine Reaktivierung

$$\frac{\sigma_{q,1}}{\sigma_{q,2}} < \frac{|u_q - c_{q,1}|}{|u_q - c_{q,2}|} \tag{3.12}$$

angegeben. Bei einer achsenorthogonalen Definition der Zugehörigkeitsfunktionen kann die-

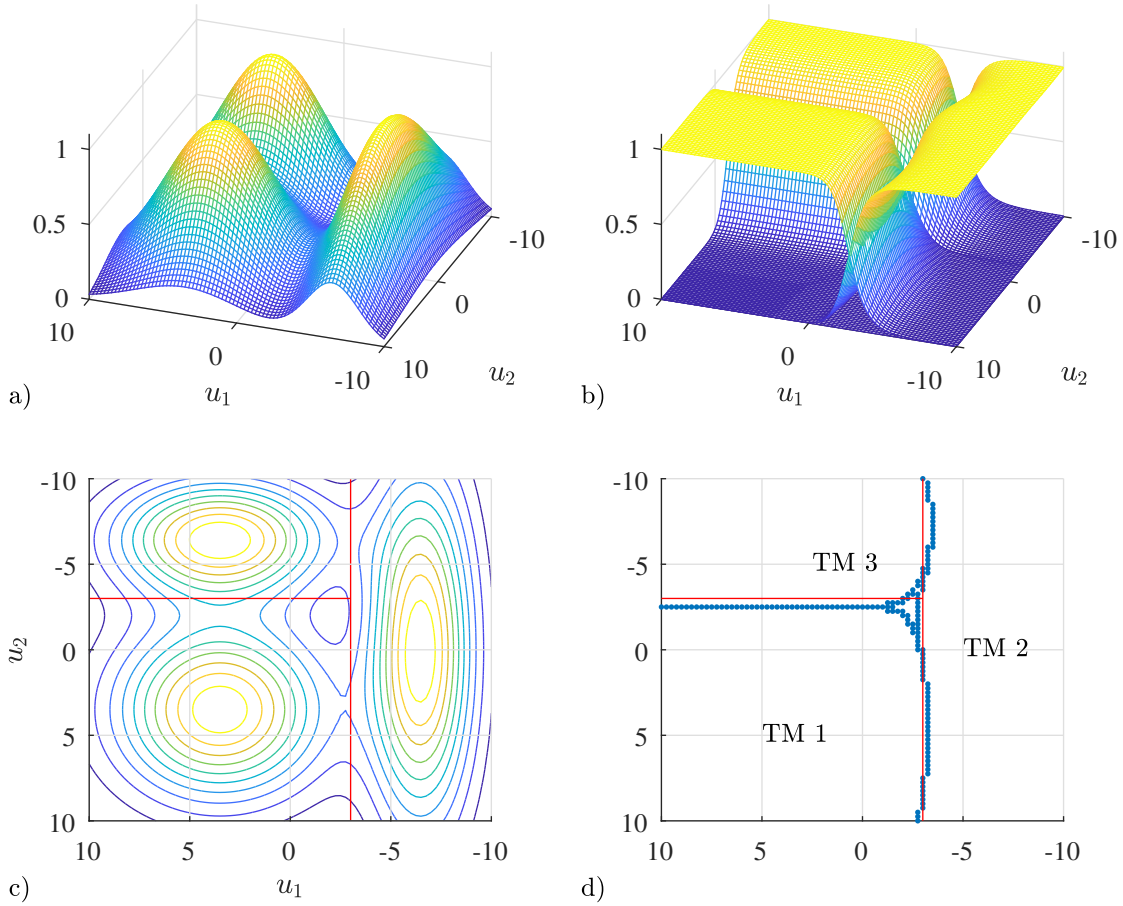

Abbildung 3.7.: Achsenortogonale Partionierung mit Gaußschen Zugehörigkeitsfunktionen, a) Achsenorthogonale Gaußfunktionen, b) zugehörige normierte Basisfunktionen, c) Projektion der Gaußfunktionen, d) resultierenden Teilmodellgrenzen $\varphi = 0,5$

se Bedingung relativ einfach für jede Eingangsgröße überprüft werden. Für achsenschräge Partitionierungen ist die Vermeidung der Reaktivierung in der Regel nur schwer möglich. Weitergehende Erörterungen zu den Effekten der Normierung von Zugehörigkeitsfunktionen finden sich in [56].

Zu den genannten unerwünschten Eigenschaften der Normierung kommt weiterhin der hohe Berechnungsaufwand. Ausgehend von der Ausgangsgleichung (3.2) ergibt sich mit Gleichung (3.10) die Ausgangsgleichung mit normierten Gaußschen Basisfunktionen

$$\hat{y}(\boldsymbol{u}, \boldsymbol{c}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \alpha_k(\boldsymbol{u}) \cdot \frac{\mu_k(\boldsymbol{u}, \boldsymbol{c_k}, \boldsymbol{\Sigma_k})}{\sum_{k=1}^{K} \mu_k(\boldsymbol{u}, \boldsymbol{c_k}, \boldsymbol{\Sigma_k})}. \tag{3.13}$$

Aus dieser ist ersichtlich, dass für jeden Schätzwert $\hat{y}_i(\boldsymbol{u_i}, \boldsymbol{c}, \boldsymbol{\Sigma})$ alle K Zugehörigkeitsfunktionen μ_k an der Stelle $\boldsymbol{u}_i$ berechnet werden müssen, auch wenn die zugehörigen Teilmodelle an dieser Stelle nicht wirksam sind. Für die limitierte Rechenleistung von Motorsteuergeräten ist dieser Aufwand erheblich und steigt mit der Anzahl an Teilmodellen. Weiterhin ist die Berechnung der Exponentialfunktionen über die Festkommaarithmetik der Motorsteu-

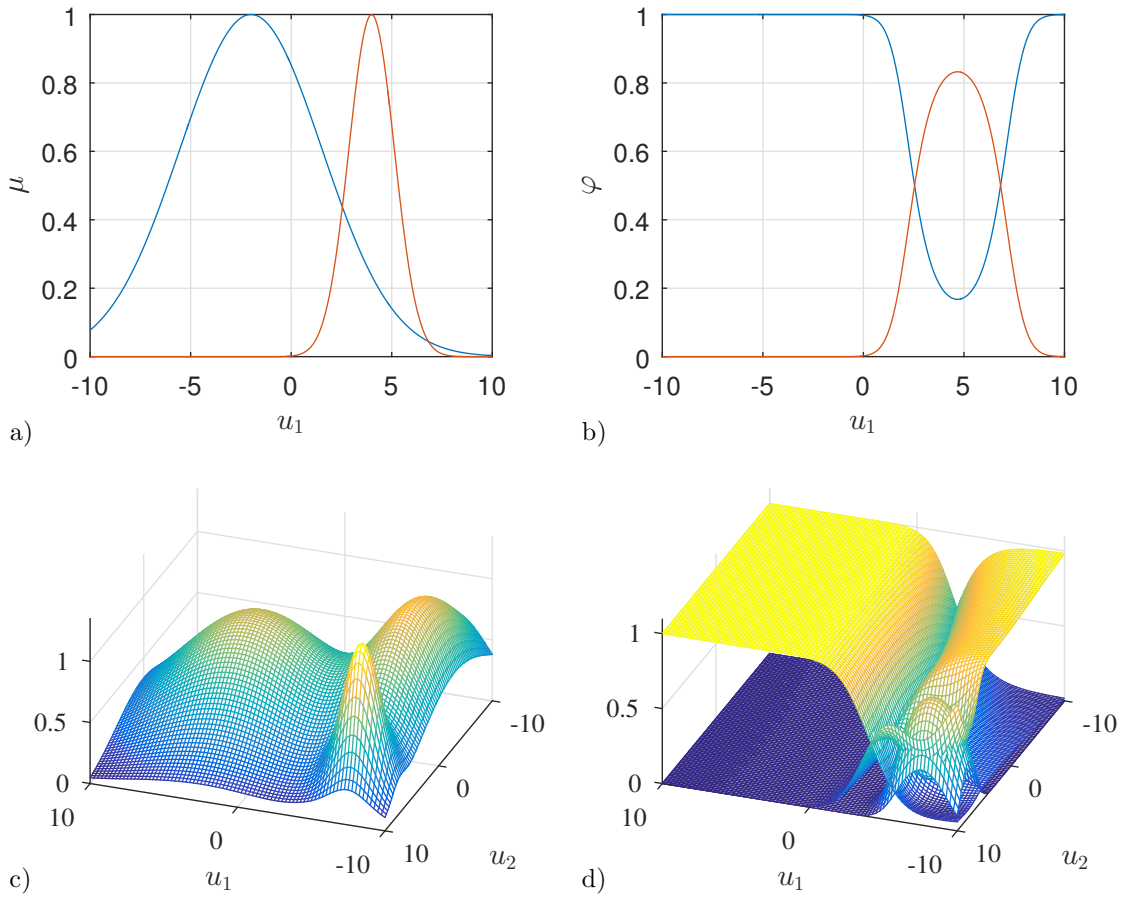

Abbildung 3.8.: Effekt der Reaktivierung von Gaußschen Zugehörigkeitsfunktionen mit großer Varianz, a) eindimensional und c) zweidimensional, sowie den resultierenden normierten Basisfunktionen b) und d)

ergeräte ebenfalls sehr rechenintensiv.

Zusammenfassend kann gesagt werden, dass sich die unerwünschten Effekte der Normierung in höherdimensionalen Modellen verstärken und zu einer schlechteren Interpretierbarkeit führen, auch wenn bei einer ausschließlich achsenorthogonalen Ausrichtung der Zugehörigkeitsfunktionen die Nachteile größtenteils vermieden werden können. Der Ressourcenbedarf zur Berechnung der Ausgangsgleichung ist bei heutigen Motorsteuergeräten ebenfalls erheblich und erschwert den Einsatz eines Modells in der Praxis. Für die Verwendung in gut interpretierbaren Modellansätze sind normierte Gauß-Funktionen als Basisfunktionen somit weniger gut geeignet. Im Folgenden soll daher eine Tensorproduktkonstruktion als Basisfunktion vorgestellt werden, welche die genannten Nachteile vermeidet.

3.2.2. Multivariate Pi-shape Basisfunktion

Die Eigenschaften und insbesondere die Interpretierbarkeit eines lokal-linearen Modells hängt im entscheidenden Maße von der verwendeten Basisfunktion ab. Wie im letzten Kapitel aufgezeigt wurde, haben die in vielen Verfahren, wie zum Beispiel in LOLIMOT oder der Gaussian-Mixture-Regression genutzten normierten Gauß-Funktionen, wesentliche Nachtei-

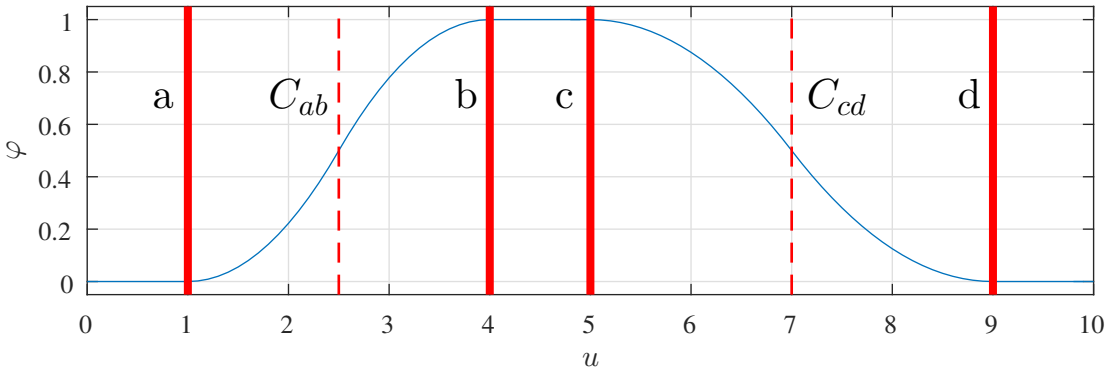

Abbildung 3.9.: Verlauf der univariaten Π-shape-Funktion mit den Parametern $a = 1, b = 4, c = 5, d = 9$ und den Zentren der Flanken C_{ab} und C_{cd}

le, welche die Interpretierbarkeit in höherdimensionalen Modellen einschränken. In diesem Kapitel soll die Konstruktion einer multivariaten Basisfunktion vorgenommen werden, die eine gute Interpretierbarkeit sicherstellt. Um dies zu erreichen sollen folgende Eigenschaften realisiert werden:

- *Keine Normierung notwendig:* Der Geltungsbereich eines Teilmodells soll direkt über die Parameter der Basisfunktion vorgegeben werden können und unabhängig von der Berechnung der weiteren Teilmodelle sein.

- *Unabhängige Definition der Übergangsbereiche*: Die Übergangsbereiche zwischen den Teilmodellen sollen unabhängig voneinander sowohl in jeder Eingangsdimension, als auch an jeder Flanke der Basisfunktion definierbar sein.

- *Achsenorthogonale Ausrichtung:* Der Geltungsbereich der Teilmodelle soll auch in höherdimensionalen Eingangsräumen einfach definierbar und aus dem optimierten Modell ablesbar sein. Um dies zu erreichen, soll die Ausrichtung der Basisfunktionen achsenorthogonal erfolgen.

- *stetig und differenzierbar:* Um den häufigen Anforderungen aus der Regelungstechnik nach einem stetigen Verlauf der Ausgangsgröße und der Differenzierbarkeit an jedem Punkt der Ausgangsfunktion gerecht zu werden, muss auch die Basisfunktion stetig und differenzierbar sein.

- *Ressourcenschonende Berechnung der Modellausgangsfunktion:* Zur praxisnahen Verwendung muss die Berechnung der Ausgangsgleichung des Modells ressourcenschonend auf einem typischen Motorsteuergerät erfolgen können.

Ausgehend von diesen Kriterien wurde eine univariate Π-shape-Funktion, wie sie in Neuro-Fuzzy-Modellen als Zugehörigkeitsfunktion zum Einsatz kommt, als Grundlage für die Konstruktion verwendet. Diese Funktion verläuft im Wertebereich $[0, 1]$ und besitzt zwei Anstie-

ge, die aus jeweils zwei Parabelästen zusammengesetzt sind. Sie ist folgendermaßen definiert:

$$\varphi(x,a,b,c,d) = \begin{cases} 0, & a < x \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x < \frac{a+b}{2} \\ 1-2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x < b \\ 1, & b \leq x < c \\ 1-2\left(\frac{x-c}{d-c}\right)^2, & c \leq x < \frac{c+d}{2} \\ 2\left(\frac{x-d}{d-c}\right)^2, & \frac{c+d}{2} \leq x \leq d \\ 0, & x > d \end{cases} \tag{3.14}$$

mit

$$a < b \leq c < d. \tag{3.15}$$

Der Verlauf dieser Funktion ist in Bild 3.9 dargestellt. Mit den vier Parametern a, b, c, d werden die Lage und Breite der Anstiege definiert. Die Parabeläste sind kongruent und gedreht um die Zentren $C_{ab}(\frac{a+b}{2}; 0, 5)$ bzw. $C_{cd}(\frac{c+d}{2}; 0, 5)$ angeordnet, wodurch die Flanken jeweils symmetrisch sind. Der Anstieg beider Äste in den Zentrumspunkten ist identisch, womit der Übergang zwischen den Parabelästen stetig erfolgt. Die Funktion ist somit über den gesamten Definitionsbereich stetig als auch differenzierbar und die quadratische Struktur ermöglicht eine effektive Berechnung auf Steuergeräten.

Das Extrapolationsverhalten der Funktion kann direkt über die Parameter gesteuert werden. Mit $a = -\infty$ oder $d = \infty$ kann die uneingeschränkte Gültigkeit des jeweiligen Teilmodells über die Modellgrenzen hinaus festgelegt werden. Mit einem endlichen Wert für a und d extrapoliert der Modellausgang, wie auch bei den unnormierten radialen Basisfunktionsnetzen, gegen Null.

Bedingt durch die Forderung einer achsenorthogonalen Ausrichtung der Teilmodellgrenzen, kann die multivariate Erweiterung der Funktion über eine Tensorproduktkonstruktion erfolgen. Mit der unabhängigen Definition der univariaten Funktionen in jeder Eingangsdimension ergibt sich die multivariate Π-shaped-Funktion nach Gleichung (3.8)

$$\varphi_k(\boldsymbol{x}, \boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{c_k}, \boldsymbol{d_k}) = \prod_{i=1}^{q} \varphi_k(x_{ik}, a_{ik}, b_{ik}, c_{ik}, d_{ik}) \tag{3.16}$$

mit den Vektoren der Koeffizienten $\boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{c_k}, \boldsymbol{d_k} \in \mathbb{R}^{1\times q}$

$$\begin{aligned} \boldsymbol{a_k} &= \begin{bmatrix} a_{1k} & a_{2k} & \dots & a_{qk} \end{bmatrix} \\ \boldsymbol{b_k} &= \begin{bmatrix} b_{1k} & b_{2k} & \dots & b_{qk} \end{bmatrix} \\ \boldsymbol{c_k} &= \begin{bmatrix} c_{1k} & c_{2k} & \dots & c_{qk} \end{bmatrix} \\ \boldsymbol{d_k} &= \begin{bmatrix} d_{1k} & d_{2k} & \dots & d_{qk} \end{bmatrix}. \end{aligned} \tag{3.17}$$

Die Konstruktion bietet die Möglichkeit, den Anstieg jeder Flanke unabhängig voneinander zu definieren und damit die Breite des Übergangsbereiches zwischen zwei Teilmodellen

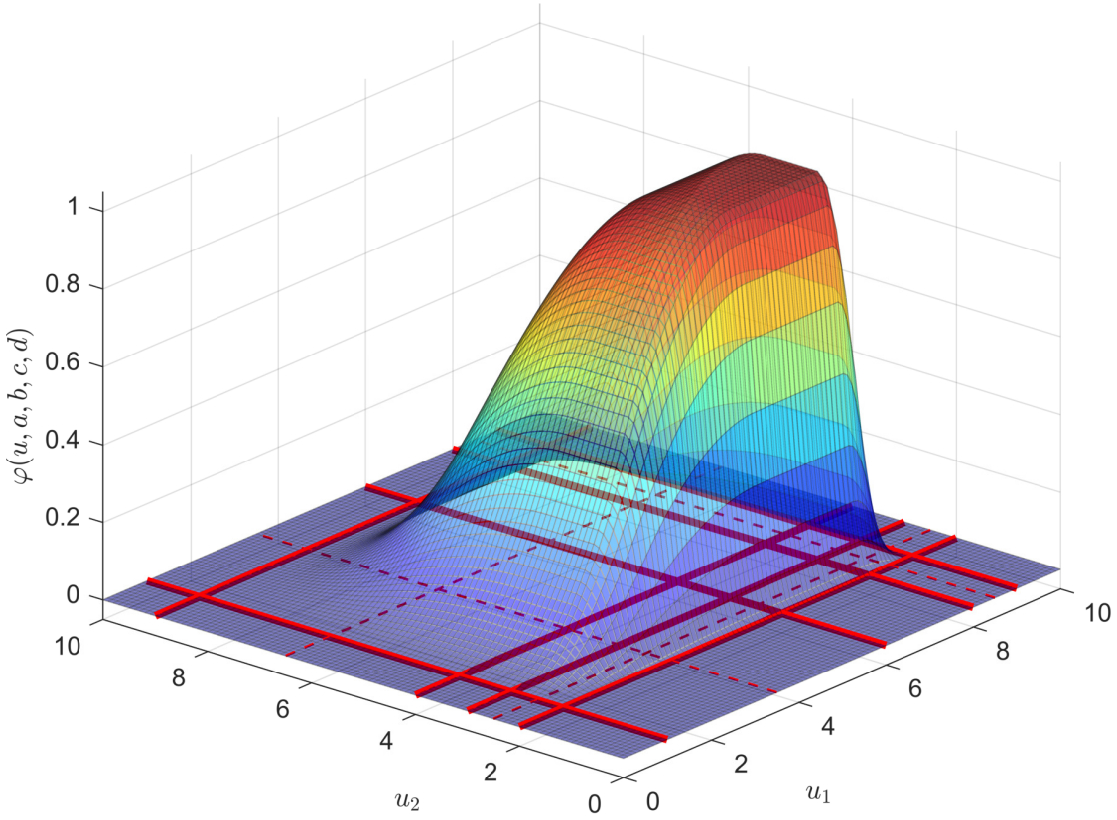

Abbildung 3.10.: Verlauf der multivariaten Π-shape-Funktion mit den Parametervektoren $\boldsymbol{a} = [1, 2]$,$\boldsymbol{b} = [6, 3]$,$\boldsymbol{c} = [8, 4]$,$\boldsymbol{d} = [9, 9]$. Die Zentren C der Übergangszonen sind gestrichelt dargestellt.

individuell festlegen zu können. Über die Parameter können die Gültigkeits- und Übergangsbereiche der Teilmodelle explizit vorgegeben oder im optimierten Modell direkt abgelesen werden. Als Beispiel ist eine zweidimensionale Basisfunktion nach Gleichung (3.16) in Abbildung 3.10 dargestellt.

Mit dem Wertebereich der Basisfunktion im Intervall $[0, 1]$ ist nur eine Voraussetzung für den Verzicht auf eine Normierung erfüllt. Um dies zu gewährleisten ist es weiterhin notwendig, die Flankenbreite der Basisfunktionen φ_1 und φ_2 zweier aneinanderliegender Teilmodelle deckungsgleich zu realisieren. Bild 3.11 veranschaulicht dies. Für die Parameter der Basisfunktion muss daher gelten:

$$\begin{aligned} c_{i1} &= a_{i2} \\ d_{i1} &= b_{i2}. \end{aligned} \tag{3.18}$$

Mit dieser Bedingung ergibt sich die Summe zweier angrenzender Basisfunktionen im Übergangsbereich immer zu Eins und zusammen mit den Bedingungen aus Gleichung (3.15) gilt

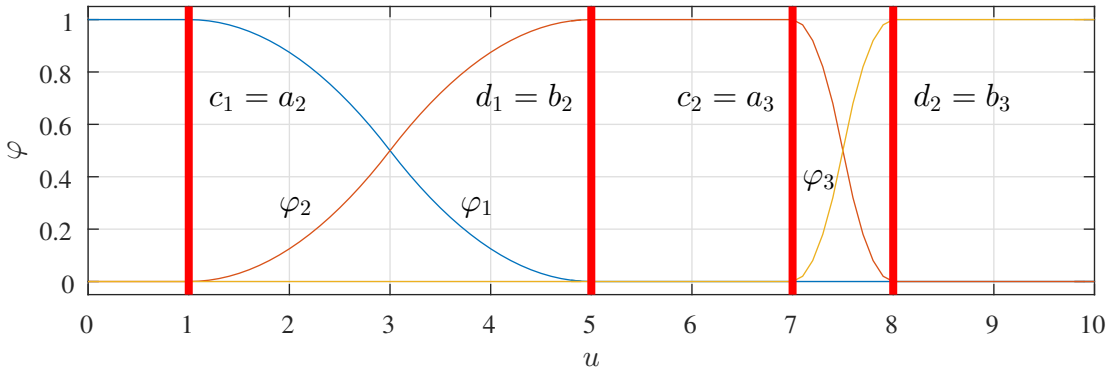

Abbildung 3.11.: Übergangsbereiche dreier univariaten Π-shape-Funktionen $(\varphi_1, \varphi_2, \varphi_3)$ unter den in Gleichung (3.18) aufgeführten Bedingungen

somit für jeden Punkt $\boldsymbol{u}$ im Eingangsraum

$$\sum_{k=1}^{K} \varphi_k(\boldsymbol{u}) = 1. \tag{3.19}$$

womit eine Normierung entfallen kann. Weiterhin ist eine einfache und eindeutige Definition der Teilmodellgrenzen über die Lage der Zentrumspunkte C an den Stellen $\varphi_k = 0,5$ der Übergangszonen möglich. In Bild 3.12a ist als Beispiel der Verlauf von drei Π-shaped-Basisfunktionen in einem zweidimensionalen Eingangsraum dargestellt. Die resultierenden Teilmodellgrenzen sowie die Bereiche der Übergangszonen sind in Bild 3.12b abgebildet.

Mit der multivariaten Π-shaped-Funktion wurde eine Basisfunktion konstruiert, die sich gut für eine interpretierbare Modellierung in hochdimensionalen Eingangsräumen eignet und zugleich eine ressourcenschonende Berechnung auf einem Motorsteuergerät ermöglicht. Die wesentlichen Eigenschaften sind eine achsenorthogonale Ausrichtung im Eingangsraum, die unabhängige Definition der Übergangsbereiche zwischen den Teilmodellen sowie die Stetigkeit und Differenzierbarkeit. Letzteres ist ein wichtiger Aspekt für die Anwendung in regelungstechnischen Aufgaben.

3.3. Strukturoptimierung paralleler Modellnetze

Nachdem in den vorangegangenen Kapiteln die Eignung von parallelen lokal-linearen Modellstrukturen für interpretierbare Modelle herausgearbeitet wurde, soll in diesem Kapitel untersucht werden, welche Verfahren zur Strukturoptimierung dieser Modelle geeignet sind. Ein Hauptaugenmerk liegt dabei auf der Wahl des Partitionierungsalgorithmus und dessen Eignung für hochdimensionale Eingangsräume. Weiterhin soll eine Integration der im letzten Kapitel vorgestellten multivariaten Π-shape-Basisfunktion möglich sein.

Ziel der Strukturoptimierung ist, die optimalen Parameter der Basisfunktionen zu finden, was im Allgemeinen eine nichtlineare Optimierung darstellt. Im nachfolgenden Kapitel soll ein Überblick über die verschiedenen Methoden und Verfahren der Partitionierung des Eingangsraumes gegeben werden. Danach soll der für das interpretierbare, lokal-lineare Modellnetz gewählte Optimierungsansatz erläutert werden.

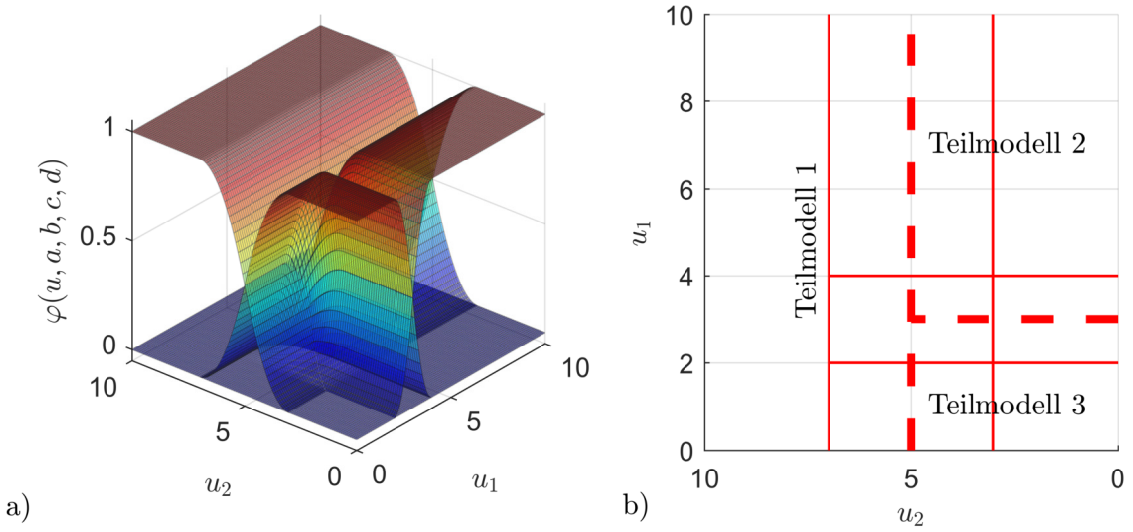

Abbildung 3.12.: a) Überlagerung von 3 multivariaten Π-shape-Funktionen in einem zweidimensionalen Eingangsraum, b) resultierende Partitionierung des Eingangsraumes und die entstehenden Übergangszonen

3.3.1. Strukturoptimierung lokal-linearer Modelle

Die Eigenschaften lokal-linearer Modelle hängen ganz wesentlich von der Art der Aufteilung in einzelnen Teilmodelle ab. Dabei lassen sich prinzipiell zwei Arten der Konstruktion unterscheiden:

Parallele Konstruktion: Die parallele Konstruktion definiert zur Initialisierung eine feste Anzahl an Komponenten, deren Parameter mit verschiedenen Verfahren der Optimierung angepasst werden. Die Modellkomplexität ist somit beim Start der Optimierung festgelegt und die Parameter der Basisfunktionen werden parallel optimiert. Ein typischer Vertreter dieser Strategie ist die Gaussian-Mixture-Regression, welches in Kapitel 2.4.7 vorgestellt wurde. Auf Grund des nichtlinearen Optimierungsproblems ist das Ergebnis dieses Vorgehens stark von der Wahl der Initialisierungsparameter abhängig. Die Güte des gefundenen Optimums kann in der Regel nur über den Vergleich von mehreren, mit verschiedenen Startparametern durchgeführte Optimierungsdurchläufe abgeschätzt werden. Das Problem verschärft sich bei einer hohen Anzahl an Eingangsgrößen oder einer hohen Anzahl an Teilmodellen.

Neben der Parameterbestimmung stellt die optimale Wahl der Teilmodellanzahl ein weiteres Optimierungsproblem dar, welches zudem die Parameter der Basisfunktionen beeinflusst. Eine Abschätzung kann über die wiederholte Optimierung mit verschiedenen Teilmodellanzahlen und die Bewertung dieser Durchläufe über ein Informationskriterium wie dem AIC oder BIC erfolgen, siehe Kapitel 2.2.1.

Hierachische Konstruktion: Im Gegensatz zu einer parallelen, startet eine hierarchische Konstruktion mit einem oder wenigen Teilmodellen. Die Erweiterung der Struktur erfolgt durch das Aufteilen und Ersetzen einzelner Bereiche, während die Parameter der anderen Komponenten unverändert bleiben. Die hierarchische Konstruktion folgt damit der Idee, dass Modell lokal dort zu erweitern, wo die größte Verbesserung der Modellgüte zu erwarten ist.

Bekannte Vertreter dieses Lokalitätsprinzips sind das LOLIMOT-Verfahren (siehe Kapitel 2.4.8) oder das bisecting-K-means Verfahren [57]. Die Modellkomplexität wird bei dieser Art der Konstruktion iterativ bis zum Erreichen einer vorgegebenen Modellgüte erhöht, wobei die Unterteilung in den einzelnen Iterationen in der Regel nicht global optimal ist. Gehört das Verfahren zur Klasse der universellen Approximatoren, wird die Modellgüte jedoch mit jeder Erhöhung der Modellkomplexität an das globale Optimum angenähert [8], [58].

Neben der Kategorisierung der Entwurfsverfahren in parallele und hierarchische Konstruktionen können diese auch über die Art der Parameteroptimierung eingeteilt werden. Grob kann hier zwischen einer empirischen Definition, der Bestimmung der Parameter auf Grundlage der Messdaten sowie einer freien Parameteroptimierung unterschieden werden.

Empirische Verfahren Zu den empirischen Verfahren zählen die in der Praxis weit verbreiteten Lookup-Tabellen mit ihrer Gitterstruktur, deren Raster sowohl systematisch und äquidistant mit einer definierten Anzahl, als auch flexibel über a-priori-Wissen festgelegt werden kann. Letztere Vorgehensweise wird bei den Neuro-Fuzzy-Systeme angewendet, deren Zugehörigkeitsfunktionen auf Grundlage physikalischen oder empirischen Wissens definiert werden und deren Gültigkeitsfunktionen als Tensorproduktkonstruktion in einer Gitterstruktur resultieren [58]. Wie in Kapitel 2.4.3 aufgeführt, sind alle Verfahren, die auf solchen Gitterstrukturen basieren, für hochdimensionale Probleme weniger geeignet. Vor allem gilt dies, wenn auf Grund geringer Prozesskenntnisse die Anzahl der Rasterstellen nicht verringert werden kann.

Messdatenbasierte Verfahren Ist über den Prozess nur wenig bekannt, sind Verfahren vorteilhaft, welche die Parameter der Basisfunktionen auf Grundlage der Messdaten bestimmen. Diese lassen sich in zwei Kategorien aufteilen:

1. *Clustering:* Beim Clustering erfolgt eine Zuordnung der zur Modellierung verwendeten Messdaten zu verschiedenen Gruppen, deren Eigenschaften zur Bestimmung der Parameter der Basisfunktionen genutzt werden. Für das Clustering gibt es eine Vielzahl verschiedener Verfahren, einen Überblick und Kategorisierung findet man in [59], [60] und [61].

   Die Zuordnung der Messdaten zu Clustern kann über die Datenverteilung im Eingangsraum erfolgen. Dieses Vorgehen hat den Vorteil, dass nur in den Bereichen Teilmodelle geschätzt werden, in denen genügend Daten vorhanden sind. Andererseits ist die Modellgüte mit dieser Strategie sehr stark von der Platzierung der Messpunkte abhängig und die Komplexität des Prozesses wird gegebenenfalls nicht ausreichend abgebildet.

   Eine bessere Berücksichtigung der Modellkomplexität kann durch das Einbeziehen der Ausgangsgröße in das Clustering erreicht werden. Dazu wird der Eingangsraum um die Ausgangsgröße zum Prozessraum erweitert. Teilbereiche die im Eingangsraum dicht beieinander liegen, sich jedoch im Prozessverhalten stark unterscheiden, können im Prozessraum besser separiert und abgebildet werden. Ein typischer Vertreter dieser Clustering-Variante ist die Gaussian-Mixture-Regression, siehe auch Kapitel 2.4.7. Der Nachteil der tendenziellen Ausrichtung der Teilmodelle an der Verteilung der Messdaten im Eingangsraum ist allerdings auch bei diese Gruppe der Clustering-Verfahren gegeben. Damit hängt die Güte des Modells, ohne eine genaue Kenntnis des zu modellierenden Prozesses und der damit möglichen optimalen Wahl der Messpunkte, wesentlich von der Verteilung der Datenpunkte ab.

Ein weiterer Nachteil des Clusterings im Prozessraum besteht in der Möglichkeit von Mehrdeutigkeiten bei der Abbildung der Cluster auf die eingangsraumbasierten Parameter der Basisfunktionen, was die Interpretierbarkeit des resultierenden Modells erheblich verschlechtert.

2. *Datenpunktbasiert:* Datenpunktbasierte Ansätze nutzen zur Parameterbestimmung der Basisfunktionen einzelne Datenpunkte aus dem Messdatensatz. Klassische Vertreter dieser Kategorie sind die RBF, bei denen häufig die Zentren der Basisfunktionen durch konkrete Datenpunkte aus dem Messdatensatz definiert werden und die SVR, welche die Stützvektoren aus dem Datensatz zur Parametrisierung der Kernfunktionen verwendet. In lokal-linearen Modellnetzen finden datenpunktbasierte Ansätze im Zusammenhang mit radialen Basisfunktionen Anwendung. Oft werden alle Datenpunkte als potentielle Zentren für die möglichen Basisfunktionen betrachtet und über geeignete Verfahren erfolgt eine Selektion nach den Kandidaten, welche die Modellgüte am stärksten beeinflussen [58].

Freie Strukturoptimierung Als direktester Ansatz zur Strukturoptimierung können die Parameter der Basisfunktionen frei und ohne Vorgaben gleichzeitig mit den Koeffizienten der linearen Teilmodelle optimiert werden. Nachteil dieses Ansatzes ist die, bedingt durch große Anzahl an Parametern, häufig auftretende Vielzahl an lokalen Optima. Dies führt zu einer großen Abhängigkeit der Modellgüte von den gewählten Initialisierungsparametern, wodurch Methoden für eine geeignete Vorauswahl der Anzahl und Parameter der Basisfunktionen bei dieser Art der Strukturoptimierung eine erhebliche Bedeutung zukommt. Für komplexe, hochdimensionale Prozesse treten diese Nachteile besonders stark in den Vordergrund, weshalb sie für das angestrebte Ziel einer interpretierbaren Modellierung weniger geeignet erscheinen.

3.3.2. Strukturoptimierungsalgorithmus in ILMON

Unter Abwägung der im letzten Kapitel dargestellten Eigenschaften der verschiedenen Strukturoptimierungsmethoden wurde sich im Rahmen der Umsetzung eines interpretierbaren lokal-linearen Modellnetzes (ILMON) für einen hierarchischen Konstruktionsalgorithmus entschieden. Als wesentliche Eigenschaft dieser Methode sollen neue Teilmodelle, ausgehend von einem initialem Basismodell, aus den Parametern eines bestehenden Teilmodells konstruiert werden und dieses ersetzen. Weiterhin wird gefordert, dass die nicht beteiligten Komponenten nicht beeinflusst werden und sich die Erweiterung der Modellstruktur damit nur lokal auswirkt, was im Zusammenspiel mit der vorgestellten multivariaten Π-shape-Basisfunktion möglich wird. Dieses Vorgehen bietet einige wesentliche Vorteile im Hinblick auf die Interpretierbarkeit des optimierten Modells:

1. Die Strukturoptimierung kann auf Grundlage der Messdaten sowohl für das komplette Modell als auch lokal begrenzt durchgeführt werden. Besteht nur in bestimmten Teilmodellen des Modells weiterer Optimierungsbedarf, können neue Messdaten partiell für den gewünschten Bereich aufgenommen werden und die Optimierung kann lokal begrenzt ausschließlich in den involvierten Teilmodellen erfolgen.

2. Einmal validierte Modellbereiche bleiben bestehen. Bedingt durch das Lokalitätsprinzip müssen aufwendig validierte Teilmodelle bei einer Änderung der Modellstruktur

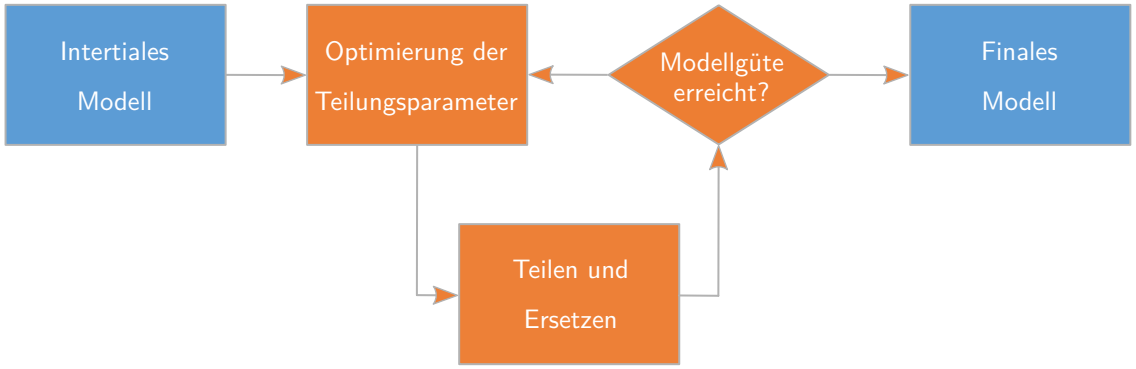

Abbildung 3.13.: Iterative Erweiterung der Modellstruktur

nicht neu überprüft werden, solange sich die Parameter dieser Teilmodelle nicht geändert haben.

3. Es kann auf einfache Weise A-priori-Wissen in die Modellierung einfließen. Mit der Definition der Teilmodellgrenzen im initialen Basismodell können physikalische und heuristische Kenntnisse über den Verlauf von Nichtlinearitäten schon vor dem Start der Optimierung in das Modell integriert und der Eingangsraum in unabhängige Teilbereiche aufgetrennt werden.

Iterationsschleife der Strukturoptimierung Für die Umsetzung der hierarchischen Konstruktion eignet sich ein iteratives Vorgehen, in dem in jedem Iterationsschritt die Auswahl des zu verbessernden Teilmodells vorgenommen wird und die Optimierung der neuen Parameter, inklusive die notwendige Schätzung der Koeffizienten der linearen Modellgleichungen, stattfindet. Ausgehend von einer initialen Modellstruktur werden in jeder Iteration folgende Schritte ausgeführt:

1. Optimierung der Teilungsparameters
2. Erweiterung der Struktur durch Splitten eines Teilmodells und Schätzen der linearen Koeffizienten aller beteiligter Teilmodelle auf Grundlage der neuen Basisfunktionsparameter. Die Teilmodellanzahl wird in diesem Schritt um Eins erhöht.
3. Überprüfen der Modellgüte anhand des gewählten Gütekriteriums

Die Iterationsschleife wird wiederholt, bis die gewünschte Modellgüte erreicht ist. Der Ablauf der Iteration ist in Abbildung 3.13 dargestellt.

Teilen der Basisfunktion Das Splitten der multivariaten Basisfunktion eines Teilmodells nach Gleichung (3.16) erfolgt immer achsenorthogonal, an der Stelle τ_j in der Eingangsdimension u_j. Die zwei neu entstehenden Basisfunktionen werden nachfolgend als linkes $(b_{j,left}, c_{j,left} < \tau_j)$ und rechtes $(b_{j,left}, c_{j,left} > \tau_j)$ Teilmodell bezeichnet. Die Teilung ist schematisch in Abbildung 3.14 dargestellt.

Unter der Vorgabe, dass beide entstehenden Teilmodelle das ursprüngliche Modell ersetzen, müssen die Parameter der aktuellen Basisfunktion in die Parameter der entstehenden

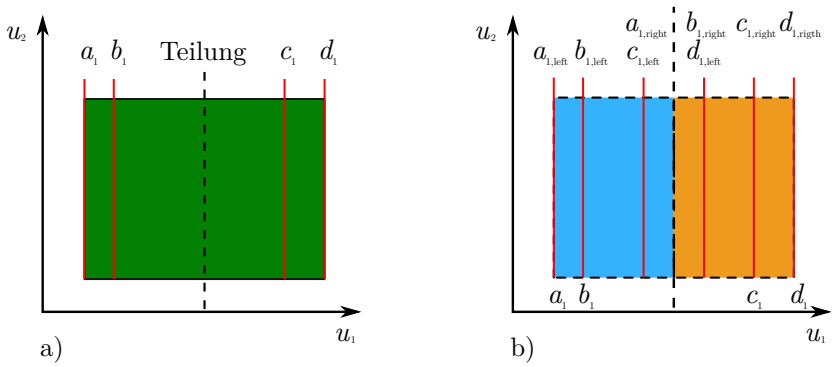

Abbildung 3.14.: Teilung eines Teilmodells mit Pi-shaped-Basisfunktion in der Dimension u_j mit $j = 1$, a) Ausgangsteilmodell mit den Parametern der Basisfunktion a, b, c, d , b) beide neu definierte Teilmodelle ersetzen das ursprüngliche Modell vollständig

Basisfunktionen überführt werden. Die Splittung in der Dimension j erfolgt nach folgendem Schema:

$$\varphi_k(\boldsymbol{a}_k, \boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k) \underset{j}{\Rightarrow} \varphi_{left}(\boldsymbol{a}_k, \boldsymbol{b}_k, \boldsymbol{c}_{left}, \boldsymbol{d}_{left}) + \varphi_{right}(\boldsymbol{a}_{right}, \boldsymbol{b}_{right}, \boldsymbol{c}_k, \boldsymbol{d}_k) \tag{3.20}$$

mit

$$\begin{aligned} \boldsymbol{a}_{right} &= [a_{1,right}, a_{2,right}, \ldots, a_{q,right}], & a_{i\neq j,right} &= a_{j,k} \\ \boldsymbol{b}_{right} &= [b_{1,right}, b_{2,right}, \ldots, b_{q,right}], & b_{i\neq j,right} &= b_{j,k} \\ \boldsymbol{c}_{left} &= [c_{1,left}, c_{2,left}, \ldots, c_{q,left}], & c_{i\neq j,left} &= c_{j,k} \\ \boldsymbol{d}_{left} &= [d_{1,left}, d_{2,left}, \ldots, d_{q,left}], & d_{i\neq j,left} &= d_{j,k} \end{aligned} \tag{3.21}$$

Die Parametervektoren a_k und b_k werden in das linke Teilmodell übernommen, die Parametervektoren c_k und d_k in das rechte Teilmodell. Um die Voraussetzungen für den Verzicht auf die Normierung der Basisfunktionen zu erfüllen, müssen die Bedingungen aus Gleichung (3.15) und (3.18) eingehalten werden, womit für die restlichen Parameter gilt:

$$\begin{aligned} c_{j,left} &= a_{j,right} \\ d_{j,left} &= b_{j,right}. \end{aligned} \tag{3.22}$$

sowie

$$\begin{aligned} b_{j,k} &\leq c_{j,left} \\ b_{j,right} &\leq c_{j,k} \end{aligned} \tag{3.23}$$

und damit folglich für die Teilungslinie

$$\tau_j = \frac{c_{j,left} + d_{j,left}}{2}. \tag{3.24}$$

Diese Bedingungen können auf die vier Parameter Teilmodell k_t, Teilungsdimension j, Tei-

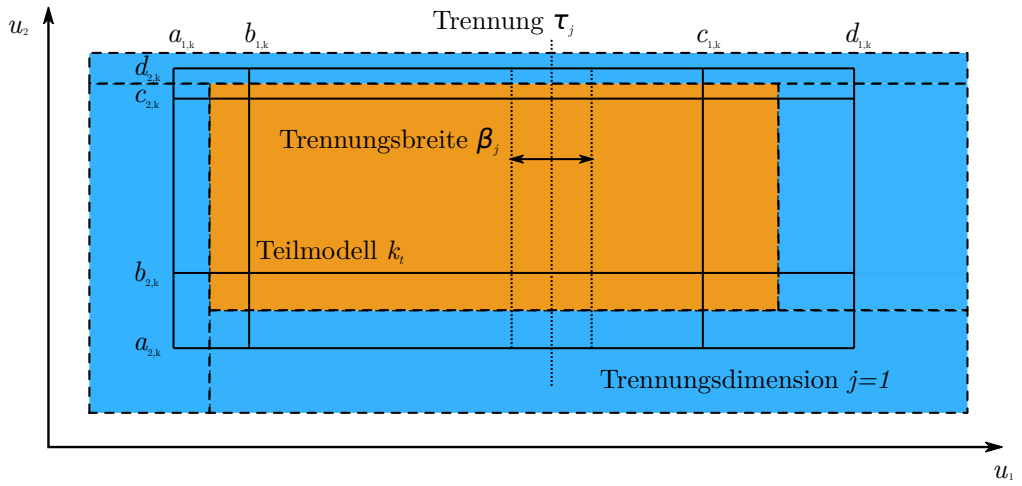

Abbildung 3.15.: Teilungsparameter τ_j und β_j eines Teilmodells k_t in der Dimension $j = 1$

lungslinie τ_j sowie die Breite des Überlagerungsbereiches

$$\beta_j = \frac{c_{j,left} - d_{j,left}}{2} \tag{3.25}$$

reduziert werden, welche die Teilung eindeutig beschreiben und mit denen diese auch intuitiv erfasst werden kann, was der Interpretierbarkeit des Modells zugute kommt, siehe Abbildung 3.15.

Die Optimierungsaufgabe kann nun derart formuliert werden, dass diejenigen Teilungsparameter gesucht werden, welche den Modellfehler ϵ minimieren:

$$\{k_{opt}, j_{opt}, \tau_{opt}, \beta_{opt}\} = \underset{k_t,j,\tau,\beta}{\arg\min}\, \epsilon, \quad k_t = \{1, \ldots, K\}, \quad j = \{1, \ldots, q\} \tag{3.26}$$

Da die Teilungsdimension und die Teilmodellnummer diskrete Parameter sind, erfolgt die Optimierung der Teilung τ_j und der Teilungsbreite β_j über mehrere Durchläufe in jeweils einem Teilmodell und einer Eingangsdimension, an deren Ende der Durchlauf mit dem minimalen Modellfehler ausgewählt wird. Die eigentliche Optimierung findet somit über nur noch 2 Parameter statt. Auf Grund der Unstetigkeitsstellen der Ableitung der Basisfunktion und den damit verbundenen Fallunterscheidungen wurde sich für das Nelder-Meat-Verfahren [62],[63] als ableitungsfreiem und robustem Optimierungsalgorithmus entschieden. Zur Bestimmung der Modellgüte wird der Modellfehler als Summe der quadratischen Fehler über alle Datenpunkte N bestimmt

$$\epsilon = \sum_{i=1}^{N} \left(\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i)\right)^2, \tag{3.27}$$

mit dem Modellausgang $\hat{y}$, der sich aus der Ausgangsgleichung eines lokal-linearen Modell-

netzes (3.2) und der Gleichung der Basisfunktion (3.16) ergibt:

$$\hat{y}(\boldsymbol{u}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}) = \sum_{k=1}^{K} \alpha(\boldsymbol{u}) \varphi_k(\boldsymbol{u}, \boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{c_k}, \boldsymbol{d_k}). \tag{3.28}$$

Mit einer vollzogenen Teilung des Teilmodells k_t in der Dimension j berechnet sich der Modellausgang mit

$$\begin{aligned} \hat{y}_{k+1} = & \sum_{k=1}^{k_t-1} \alpha_k(\boldsymbol{u}) \varphi_k(\boldsymbol{x}, \boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{c_k}, \boldsymbol{d_k}) \\ & + \alpha_{left}(\boldsymbol{u}) \varphi_{left}(\boldsymbol{a}_{k_t}, \boldsymbol{b}_{k_t}, \boldsymbol{c}_{left}, \boldsymbol{d}_{left}) \\ & + \alpha_{right}(\boldsymbol{u}) \varphi_{right}(\boldsymbol{a}_{right}, \boldsymbol{b}_{right}, \boldsymbol{c}_{k_t}, \boldsymbol{d}_{k_t}) \\ & + \sum_{k=k_t+1}^{K} \alpha_k(\boldsymbol{u}) \varphi_k(\boldsymbol{x}, \boldsymbol{a_k}, \boldsymbol{b_k}, \boldsymbol{c_k}, \boldsymbol{d_k}). \end{aligned} \tag{3.29}$$

Die Koeffizienten der linearen Komponenten $\alpha_k(\boldsymbol{u})$ werden als lokale Schätzung über die Messdaten per linearer Regression bestimmt. Diese müssen nach jeder Teilung für alle Teilmodelle neu berechnet werden. Auf die Eigenschaften und Details der Umsetzung wird im nachfolgenden Kapitel genauer eingegangen.

Damit ergibt sich für den Schritt der Erweiterung des Modellnetzes folgender Ablauf:

1. Start mit Teilmodell $k_t = 1$ und Teilungsdimension $j = 1$
2. Optimierung der Teilung τ_j und der Teilungsbreite β_j
3. Schätzen der linearen Koeffizienten γ
4. Berechnung und Speichern des Modellfehlers ϵ für das aktuelle Teilmodell und Teilungsdimension
5. Falls die letzte Teilungsdimension q nicht erreicht ist: Erhöhung von j um eins und Wiederholung ab Schritt 2
6. Falls das letztes Teilmodell K nicht erreicht ist: Erhöhung von k_t um eins und Wiederholung ab Schritt 2
7. Bestimmung des niedrigsten Modellfehlers ϵ_{min} aus allen Durchläufen
8. Übernahme der optimalen Parameter aus dem Durchlauf mit dem niedrigsten Modellfehler

Mit den ermittelten optimalen Parametern k_{opt}, j_{opt}, τ_{opt} und β_{opt} kann nun die Teilung durchgeführt werden. Wurde die angestrebte Modellgüte mit dieser Teilung erreicht, ist die iterative Erweiterung der Modellstruktur nach Abbildung 3.13 abgeschlossen. Andernfalls wird eine neue Iteration zur Strukturoptimierung gestartet.

3.4. Lokale Parameterschätzung und implizite Regularisierung

Neben der Strukturoptimierung müssen die Koeffizienten der linearen Komponenten für jedes Teilmodells geschätzt werden, wofür sich als ein Standardverfahren die lokal gewichtete Methode der kleinsten Fehlerquadrate anbietet. Bei dieser können mit der lokalen sowie der globalen Parameterschätzung zwei Ansätze unterschieden werden [8], [54], [64], [65].

Die globale Schätzung bestimmt die Koeffizienten aller Teilmodelle gleichzeitig. Dabei fließt auch die gegenseitige Beeinflussung der Teilmodelle in den Überlagerungsbereichen mit in die Schätzung ein. Dieses Vorgehen bietet eine hohe Flexibilität des Modells, was zu einem minimalen Bias führt. Es birgt allerdings auch die Gefahr einer Überanpassung und einer schlechten Schätzung bei stark verrauschten Messwerten.

Im Gegensatz zur globalen Schätzung werden bei der lokalen Schätzung die Parameter der Einzelmodelle separat geschätzt. Dies schränkt die Freiheitsgrade des Modells ein und bedingt, dass die Überlagerungen zwischen den Teilmodellen weitgehend vernachlässigt werden, woraus ein höherer Bias-Fehler resultiert. Der Vorteil dieser Vorgehensweise ist die Möglichkeit der unabhängigen Analyse der Teilmodelle. Weiterhin ist der Varianzfehler im Vergleich zur globalen Schätzung verringert, was Vorteile bei spärlich verteilten Messdaten in hochdimensionalen Eingangsräumen mit sich bringt und Anwendungsfällen mit stark verrauschten Messdaten zugute kommt. Da im Rahmen dieser Arbeit der Hauptaugenmerk auf die Interpretierbarkeit von hochdimensionalen Modellen liegt, wurde sich für die Umsetzung der lokalen Parameterschätzung entschieden. Einen ausführlichen Vergleich der Eigenschaften beider Methoden findet man in [8] und [54].

Die linearen Komponenten jedes Teilmodells sind Linearkombinationen der Eingangsgrößen $\boldsymbol{u}$ und eines Offsets der Form

$$\alpha_k(\boldsymbol{u}) = \boldsymbol{\gamma}_k \tilde{\boldsymbol{u}} \tag{3.30}$$

mit dem Vektor der Koeffizienten

$$\boldsymbol{\gamma}_k = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_q \end{bmatrix} \quad \in \mathbb{R}^{1\times(q+1)} \tag{3.31}$$

und dem erweiterten Eingangsgrößenvektor

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} 1 & u_1 & u_2 & \cdots & u_q \end{bmatrix}^T \quad \in \mathbb{R}^{(q+1)\times 1}. \tag{3.32}$$

Als Grundlage für die Schätzung der Koeffizienten eines Teilmodells sollen alle Datenpunkte dienen, die innerhalb des Wirkungsbereiches des Teilmodells liegen. Dies schließt auch die Punkte in den Überlagerungsbereichen mit ein, deren Einfluss mit der Lage in dieser Zone variiert. Die prozentuale Zugehörigkeit eines Datenpunktes zu einem Teilmodell lässt sich direkt an der Basisfunktion des Teilmodells ablesen, weshalb diese auch zur Wichtung des Modellfehlers und der Designmatrix herangezogen werden kann. Mit dem quadratischen Fehler kann die gewichtete Verlustfunktion eines Teilmodells k als

$$J_k = \sum_{i=1}^{N} \varphi_k(\boldsymbol{u}_i)(\hat{y}_i - y_i)^2. \tag{3.33}$$

formuliert werden. Die Basisfunktion φ_k wichtet hier den quadratischen Modellfehler in Abhängigkeit der Lage innerhalb der Überlagerungszone. Je dichter ein Datenpunkt am

Kernbereich des Teilmodells $b_{i,k} \leq u_i \leq d_{i,k}$ liegt, umso mehr Einfluss hat der Modellfehler an diesem Punkt auf das Optimierungskriterium. Für Datenpunkte außerhalb des Wirkungsbereiches des Basisfunktion $a_{i,k} > u_i \vee d_{i,k} < u_i$ gilt $\varphi_k = 0$, womit diese keine Auswirkungen auf die Verlustfunktion haben.

Sind die Parameter der Basisfunktion φ_k bekannt, reduziert sich die Schätzung der Koeffizienten auf ein lineares Optimierungsproblem. Die Schätzung der Ausgangsgröße der linearen Teilmodellkomponente berechnet sich durch

$$\hat{\boldsymbol{y}}_k = \boldsymbol{X}\boldsymbol{\gamma}_k \tag{3.34}$$

mit der Designmatrix

$$\boldsymbol{X} = \begin{bmatrix} 1 & u_{1,1} & u_{2,1} & \dots & u_{p,1} \\ 1 & u_{1,2} & u_{2,2} & \dots & u_{p,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_{1,N} & u_{2,N} & \dots & u_{p,N} \end{bmatrix} \in \mathbb{R}^{N \times (q+1)} \tag{3.35}$$

welche in jeder Zeile den erweiterten Eingangsgrößenvektor $\tilde{\boldsymbol{u}}$ von einem der N Messpunkte enthält.

Die Koeffizienten $\boldsymbol{\gamma}_k$ ergeben sich aus der Minimierung der Verlustfunktion (3.33)

$$\boldsymbol{\gamma}_k = \left(\boldsymbol{X}^T \boldsymbol{Q}_k \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{Q}_k \boldsymbol{y} \tag{3.36}$$

mit der Wichtungsmatrix

$$\boldsymbol{Q}_k = \begin{bmatrix} \varphi_k(\boldsymbol{u}_1) & 0 & \dots & 0 \\ 0 & \varphi_k(\boldsymbol{u}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varphi_k(\boldsymbol{u}_N) \end{bmatrix}. \tag{3.37}$$

Ein interessanter Aspekt dieser Wichtung der Designmatrix ist die daraus resultierende implizite Regularisierung. Dies kann als einer Reduzierung der effektiven Parameteranzahl des Modells und damit der Anzahl der Freiheitsgrade interpretiert werden. Anschaulich dargestellt erfolgt diese Regularisierung durch den Einfluss der Datenpunkte in den Übergangsbereichen, welche in die Parameterschätzung mehrerer Teilmodelle einfließen. Wie in [52] und [54] ausgeführt, kann die effektive Anzahl der Parameter eines Modells ν_{eff} mit Hilfe der Glättungsmatrix $\boldsymbol{S}$ bestimmt werden, welche den Zusammenhang zwischen dem Prozessausgang und dem Modellausgang auf Grundlage der Messwerte ausdrückt:

$$\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{y} \tag{3.38}$$

mit

$$\boldsymbol{S} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T. \tag{3.39}$$

Die effektive Parameteranzahl eines Teilmodells k ergibt sich damit zu

$$\nu_{eff} = \text{Spur}\left(\boldsymbol{S}^T\boldsymbol{S}\right). \tag{3.40}$$

Für ein lokales Modellnetz mit K Teilmodellen berechnet sich die Glättungsmatrix aus der Summe der lokalen Glättungsmatrizen $\boldsymbol{S}_k$ [8]

$$\boldsymbol{S} = \sum_{k=1}^{K} \boldsymbol{S}_k \tag{3.41}$$

mit

$$\boldsymbol{S}_k = \boldsymbol{Q}_k\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{Q}_k\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Q}_k \tag{3.42}$$

und die effektive Parameteranzahl mit

$$\nu_{eff} = \sum_{k=1}^{K}\nu_{eff,k} = \sum_{k=1}^{K}\text{Spur}\left(\boldsymbol{S}_k^T\boldsymbol{S}_k\right). \tag{3.43}$$

Sind alle Diagonalelemente der Wichtungsmatrix Q_k mit $\varphi_k = 1$ definiert, liegen also keine Messwerte in einem Übergangsbereich zwischen zwei Teilmodellen, ergibt sich nach Gleichung (3.43) die höchste effektive Parameteranzahl. Mit einer Wichtung der Datenpunkte $\varphi_k < 1$ reduziert sich die effektive Parameteranzahl des Modells entsprechend. Je größer also der Anteil der Überlagerungsbereiche im Gesamtmodell ist, umso kleiner ist die effektive Parameteranzahl und damit die Anzahl der Freiheitsgrade des Modells.

Wie in Kapitel 2.2 ausgeführt, erhöht sich mit der Verringerung der Freiheitsgrade auch der Bias des Modells. Gleichzeitig wird der Varianzfehler reduziert, wodurch das Modell robuster gegenüber Überanpassungen wird. Gerade bei hochdimensionalen, komplexen Modellen mit einer großen Anzahl an Parametern und relativ wenig Messdaten kann dieser Effekt vorteilhaft sein und entspricht zugleich dem Ziel dieser Arbeit.

3.5. ILMON - Realisierungsaspekte

Mit den in den vorangegangenen Kapiteln vorgestellten Methoden zur Parameterschätzung und Strukturoptimierung soll in diesem Abschnitt der komplette Modellierungsalgorithmus des interpretierbaren lokal-linearem Modellnetz definiert werden. Insbesondere soll auf die Wahl der Startparameter, die Möglichkeiten der Integration von Prozesswissen und zusammenfassend auf die Eigenschaften des Modellierungsverfahrens eingegangen werden. Der Ablauf der Modellierung ist in Bild 3.16 dargestellt. Auf die einzelnen Bereiche des Ablaufs soll nachfolgend im Detail eingegangen werden.

Initialisierung, Extrapolation und Integration von a-priori-Wissen Der Start der Modellierung erfolgt mit einem initialem Modell, welches im einfachsten Fall aus nur einem Teilmodell mit einer Basisfunktion und einer linearen Komponente besteht.

$$\begin{aligned} \hat{y}(\boldsymbol{u},\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}) &= \varphi(\boldsymbol{u},\boldsymbol{a_0},\boldsymbol{b_0},\boldsymbol{c_0},\boldsymbol{d_0})\cdot\boldsymbol{\gamma}\tilde{\boldsymbol{u}} \\ \boldsymbol{a_0},\boldsymbol{b_0},\boldsymbol{c_0},\boldsymbol{d_0} \in \mathbb{R}^{1\times q},\quad \boldsymbol{\gamma} \in \mathbb{R}^{1\times q+1},\quad \tilde{\boldsymbol{u}} &\in \mathbb{R}^{q+1\times 1} \end{aligned} \tag{3.44}$$

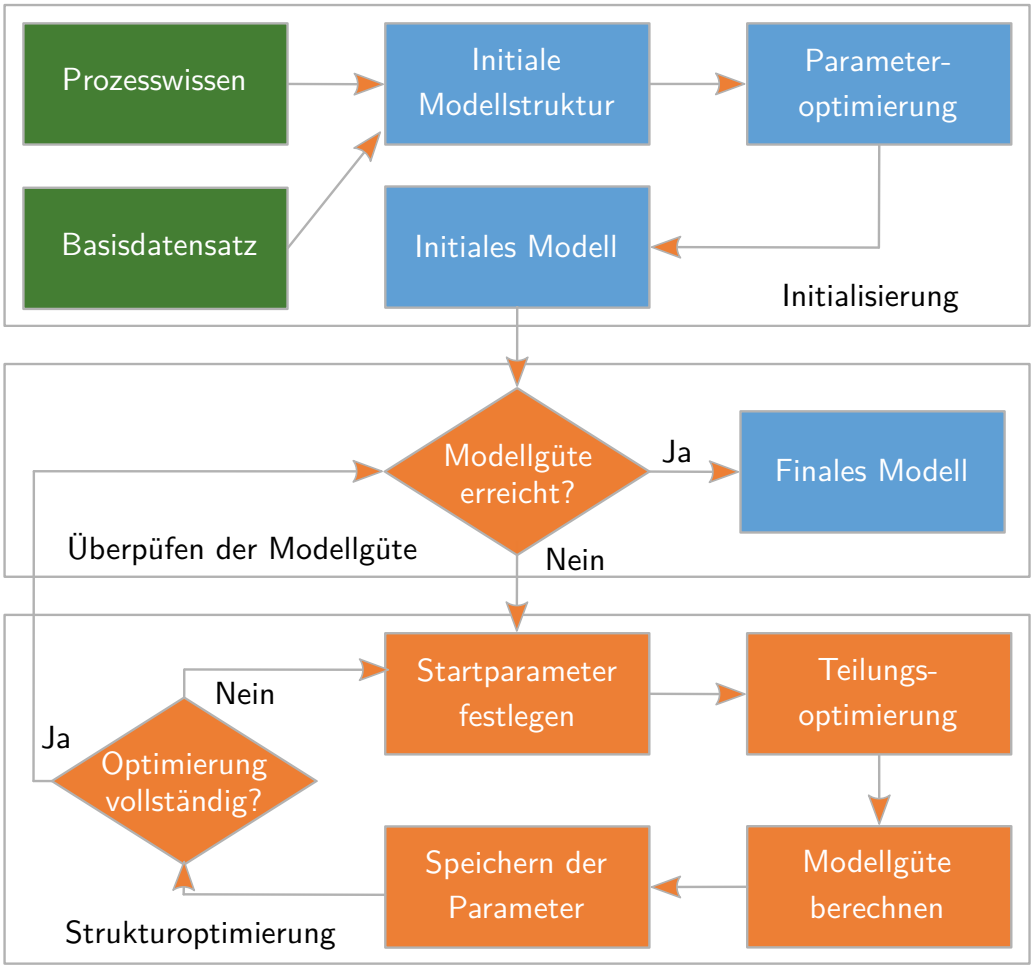

Abbildung 3.16.: Ablauf der Modellierung mit einer ILMON-Modellstruktur

Die initiale Basisfunktion definiert mit ihren Parametern $\boldsymbol{a_0}, \boldsymbol{b_0}, \boldsymbol{c_0}, \boldsymbol{d_0}$ die Grenzen des Gesamtmodells im Eingangsraum, welche entweder entsprechend dem Wertebereich der Messdaten gewählt oder per Vorwissen definiert werden.

Über die Definition der Basisfunktionsparameter wird auch das Extrapolationsverhalten gesteuert, für das zwei mögliche Varianten eingestellt werden können. Die erste Option ist eine lineare Extrapolation, welche über das Setzen der Parameter $\boldsymbol{a}_0 = -\infty, \boldsymbol{d}_0 = \infty$ realisiert werden kann. Die linearen Komponenten der an den Grenzen des Modells liegenden Teilmodelle gelten so uneingeschränkt auch außerhalb der mit Messungen abgedeckten Modellbereiche.

Eine zweite Möglichkeit ist die Wahl eines endlichen Wertes für $\boldsymbol{a}_0, \boldsymbol{d}_0$, wodurch der Modellausgang für $\boldsymbol{u} < \boldsymbol{a}_0$ sowie $\boldsymbol{u} > \boldsymbol{d}_0$ und damit außerhalb der Modellgrenzen zu null wird. Da im Strukturoptimierungsprozess die außenliegenden Teilmodellgrenzen mit ihren Parametern weitervererbt werden, ist diese Festlegung auch für das optimierte Modell gültig. Weiterhin kann das Extrapolationsverhalten auch nach den Optimierungsläufen geändert werden, ohne das Prozessverhalten in den Kernregionen zu verändern.

Neben der Definition des initialen Modells mit nur einem Teilmodell ist auch die Vorgabe

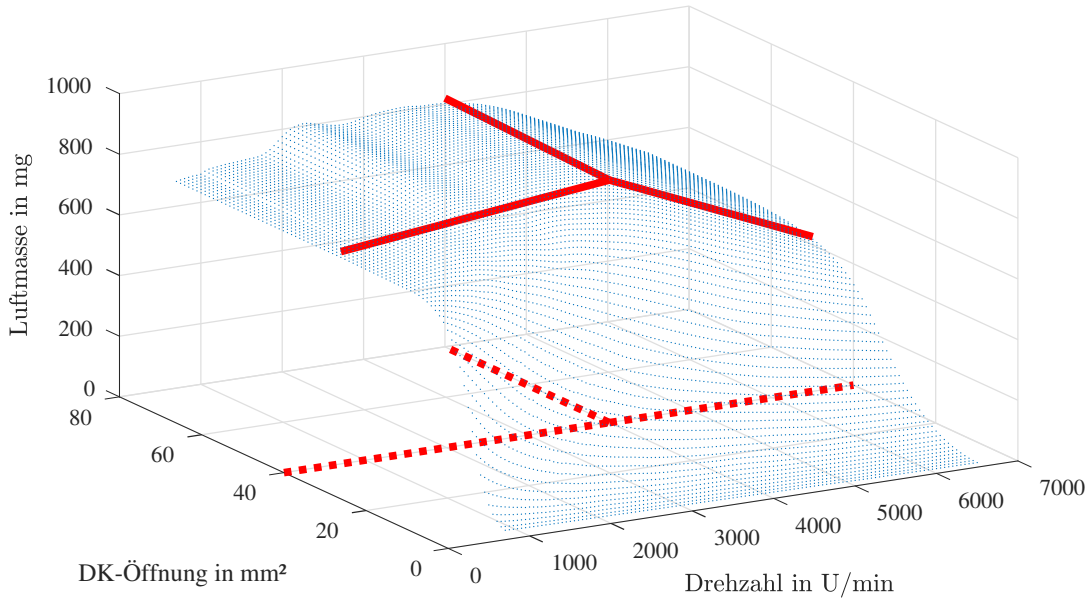

Abbildung 3.17.: Definition von initialen Modellstrukturen durch Prozesswissen am Beispiel der Füllungserfassung eines 3,2l-Saugmotors. Die bekannten Bereiche mit weitgehend linearem Verhalten können vorab in das Modell integriert werden.

mehrerer Teilmodelle durch ihre Übergangsverläufe möglich, die aus Prozesswissen hergeleitet worden sein können. Ein typischer Anwendungsfall hierfür sind bekannte starke Nichtlinearitäten wie einsetzende Turboverdichter bei bestimmten Drehzahlen oder Zustands- bzw. Stellgrößenbegrenzungen, welche sich direkt auf die Ausgangsgröße auswirken und oft in einer über weite Bereiche konstant oder linear verlaufenden Ausgangsgröße resultieren. Für die Vorgabe solcher Bereiche ist nur die Definition der Basisfunktionen φ_k notwendig, die linearen Koeffizienten $\boldsymbol{\gamma}_k$ werden aus den Messdaten geschätzt.

Bei dieser Integration von Prozesswissen kommen die Vorteile der vorgestellten Basisfunktion voll zu Geltung. Durch die achsenorthogonale Definition ist eine Vorgabe der Bereiche über eine einfache Konjunktion der Eingangsgrößen möglich. Dies soll an einem Beispiel der Füllungserfassung eines 3,2l-Saugmotors deutlich gemacht werden.

In Abbildung 3.17 ist das Kennfeld der eingeschlossenen Luftmasse pro Arbeitsspiel in Abhängigkeit von der Drosselklappenöffnung (DK) und der Drehzahl dargestellt. Der bei großen Drosselklappenöffnungen und hohen Drehzahlen weitgehend lineare Zusammenhang kann vorab in die Modellstruktur integriert werden. In diesem Beispiel lautet die Formulierung als Eingangsgrößenkonjunktion für die drei Modellbereiche M_1, M_2, M_3:

$$
\begin{aligned}
M_1 &:= \left(4000\tfrac{U}{min} < Drehzahl < 7000\tfrac{U}{min}\right) \wedge \left(40mm^2 < DK < 80mm^2\right) \\
M_2 &:= \left(0\tfrac{U}{min} < Drehzahl < 4000\tfrac{U}{min}\right) \wedge \left(40mm^2 < DK < 80mm^2\right) \\
M_3 &:= \left(0\tfrac{U}{min} < Drehzahl < 7000\tfrac{U}{min}\right) \wedge \left(0mm^2 < DK < 40mm^2\right)
\end{aligned}
\tag{3.45}
$$

Aus diesen Formulierungen resultieren die in Abbildung 3.17 gezeigten Trennlinien wobei der hierarchischen Konstruktion folgend die erste Trennung bei $\tau_1 = 40mm^2$ erfolgt und das

rechte daraus resultierende Teilmodell bei $\tau_2 = 4000^{U}/_{min}$ geteilt wird. Mit der Definition der Übergangsbreiten $\beta_1 = 20mm^2$ und $\beta_2 = 2000^{U}/_{min}$ sowie der Vorgabe einer linearen Extrapolation ergeben sich die drei Basisfunktionen mit den Parametern

$$\begin{array}{llll} \varphi_1 : \boldsymbol{a} = [3000 \quad 30], & \boldsymbol{b} = [5000 \quad 50], & \boldsymbol{c} = [7000 \quad 80], & \boldsymbol{d} = [\infty \quad \infty] \\ \varphi_2 : \boldsymbol{a} = [-\infty \quad 30], & \boldsymbol{b} = [0 \quad 50], & \boldsymbol{c} = [3000 \quad 80], & \boldsymbol{d} = [5000 \quad \infty] \\ \varphi_3 : \boldsymbol{a} = [-\infty \quad -\infty], & \boldsymbol{b} = [0 \quad 0], & \boldsymbol{c} = [7000 \quad 30], & \boldsymbol{d} = [\infty \quad 50]. \end{array} \tag{3.46}$$

Die per Vorwissen definierten Trennungsparameter müssen nicht exakt sein, sondern können als Startparameter für einen Optimierungsdurchlauf dienen und auf Basis der Messdaten angepasst werden.

Basisdatensatz Zur Parameterschätzung der linearen Komponenten $\alpha_k(\boldsymbol{u})$ wird ein Basisdatensatz benötigt, der pro Teilmodell mindestens einen Umfang von $q + 1$ Messwerten haben muss. Um einer schlechten Kondition der Parameterschätzung vorzubeugen, ist in der praktischem Umsetzung eine höhere Anzahl an Messpunkten empfehlenswert. Da durch die Wichtung der Designmatrix mit der Wichtungsmatrix $\boldsymbol{Q}$ nach Gleichung (3.36) die Datenpunkte in den Übergangsbereichen weniger stark in die Schätzung einfließen, erhöht sich der Einfluss der Messwerte innerhalb der Kerngebiete eines Teilmodells entsprechend. Damit einhergehend erhöht sich auch der Einfluss von Störungen auf die Parameterschätzung, was zu einer Verringerung der Konditionszahl führt. Um diesem Effekt entgegenzuwirken, empfiehlt es sich, die Mindestzahl ν_{min} der zur Parameterschätzung eines Teilmodells notwendigen Daten über den Wert der jeweiligen Basisfunktion an der Stelle $\boldsymbol{u}_i$ zu berechnen:

$$\nu_{min} = \sum_{i=1}^{N} \varphi_k(\boldsymbol{u}_i) \geq q + 1. \tag{3.47}$$

Die Überprüfung dieser Mindestanzahl erfolgt im Algorithmus vor jeder Parameterschätzung eines Teilmodells. Wird für eine Teilung die Mindestanzahl ν_{min} in einem der entstehenden Teilmodelle unterschritten, wird diese nicht ausgeführt.

Mit dem Basisdatensatz erfolgt die initiale Schätzung der Koeffizienten $\boldsymbol{\gamma}_k$, womit die Initialisierung des Modells abgeschlossen ist und der Start des Iterationsschleife zur Verbesserung des Modells durch Erhöhung der Modellkomplexität starten kann.

Überprüfung der Modellgüte Die Entscheidung zur Erhöhung der Modellkomplexität wird anhand eines Gütekriterium getroffen, welches z.B. als Summe der quadratischen Fehler

$$\epsilon = \sum_{i=1}^{N} \left(\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i)\right)^2 \tag{3.48}$$

über alle Datenpunkte N definiert werden kann. Ist die geforderte Güte erreicht, wird die aktuelle Struktur als finales Modell ausgegeben, andernfalls erfolgt die Erhöhung der Modellkomplexität durch Teilung eines Teilmodells.

Während die Definition der Verlustfunktion als Summe der quadratischen Fehler bei der Parameteroptimierung alle Vorteile der linearen Optimierung mit sich bringt, ist diese Beschränkung bei der Strukturoptimierung nicht notwendig. Hier können auch Gütekriterien verwendet werden, welche besser an das vom Anwender angestrebte Ziel der Optimierung

ausgerichtet sind. Eine in der Praxis populäre Fehlerfunktion ist die Summe der quadratischen relativen Fehler

$$\epsilon = \sum_{i=1}^{N} \left(\frac{\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i)}{y(\boldsymbol{u}_i)} \right)^2, \quad y(\boldsymbol{u}_i) \neq 0, \tag{3.49}$$

da hier der Fehler unabhängig vom absoluten Wert der Ausgangsgröße beurteilt wird. Bedingt durch die Division mit der Ausgangsgröße kann diese Funktion allerdings nicht direkt angewendet werden, wenn der Wertebereich des Prozessausgangs eine Teilmenge mit $\{y \approx 0\}$ enthält, da sonst eine starke Optimierung des Modells innerhalb dieses Teilbereichs stattfindet. Diese Einschränkung kann leicht mit der Einführung eines Offsets und der Verschiebung der Ausgangsgröße umgangen werden, sodass keine kritische Teilmenge mehr enthalten ist.

Neben der Optimierung auf den relativen Fehler ist es in vielen Anwendungen erwünscht, die einzelnen Modellfehler in einem bestimmten Toleranzband zu halten. Die Summe aller Fehler ist dabei nicht relevant. Hierfür kann die Verlustfunktion als Betrag des maximalen Fehlers über alle Messwerte des Datensatzes D definiert werden.

$$\epsilon = \max_{i \in D} |\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i)| \tag{3.50}$$

Dies kann äquivalent zu Gleichung (3.49) auch mit dem maximalen relativen Fehler definiert werden.

In der praktischen Anwendung kommt es häufig vor, dass bestimmte Bereiche eines Modells von der Optimierung ausgeschlossen werden sollen, z.B. weil diese nicht zu den Hauptbetriebsbereichen mit gehobenen Güteforderungen gehören, per Vorwissen definiert oder aus einem erprobten Modell übernommen wurden. In diesem Fall kann ein Ausschluss der Bereiche B über eine Indikatorfunktion $I(\boldsymbol{u})$ erfolgen, welche den Fehler entsprechend wichtet:

$$\epsilon = \sum_{i=1}^{N} I_k(\boldsymbol{u}) \left(\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i) \right)^2, \quad I_k(\boldsymbol{u}) = \begin{cases} 0; & \text{falls} \quad u \notin B \\ 1; & \text{falls} \quad u \in B \end{cases}. \tag{3.51}$$

Der Modellbereich B muss dabei nicht mit einem oder mehreren Teilmodellen übereinstimmen und kann eine beliebigen Grenzverlauf haben, sodass auch nichtachsenorthogonale Bereiche unabhängig von den Teilmodellgrenzen definiert werden können.

Wird eine der hier aufgeführten abweichenden Verlustfunktionen eingesetzt, muss diese auch in der Optimierung der Teilung der Basisfunktion verwendet werden und Gleichung (3.27) ersetzen, siehe Kapitel 3.3.2.

Bei allen von Gleichung (3.48) abweichenden Gütekriterien ist zu beachten, dass das Ergebnis der Modellierung nur noch bedingt optimal ist, da die Parameter der linearen Komponenten der Teilmodelle weiterhin unter dem Kriterium der kleinsten Fehlerquadratsumme geschätzt werden, welches die eingeführten Begrenzungen und Transformationen nicht berücksichtigt. Infolgedessen kann ein erweitertes Gütekriterium zu einer höheren Modellkomplexität führen. Ob und in welchem Umfang dies auftritt, wurde im Rahmen dieser Arbeit nicht untersucht und könnte Gegenstand zukünftiger Forschungsarbeiten sein.

Startparameter der Strukturoptimierung Wie bei jeder nichtlinearen Optimierung ist das Ergebnis der Strukturoptimierung stark von der Wahl der Startparameter abhängig. Wie in Kapitel 3.3.2 dargestellt, ist die optimale Teilung durch das zu teilenden Modell k_t und der

Teilungsdimension j als diskrete Parameter definiert sowie durch die Teilungslinie τ_j und die Breite des Überlagerungsbereiches β_j. Einige Verfahren bestimmen das zu teilende Modell hilfsweise über eine lokale Verlustfunktion und verwenden feste Teilungsverhältnisse für die Definition der Trennungslinie. Dies kann jedoch zu einer suboptimalen Teilung und damit der Erhöhung der Teilmodellanzahl im finalen Modell führen [8], [66].

Da eine höhere Teilmodellanzahl auch die Interpretierbarkeit des Modells verschlechtert, wird im ILMON-Algorithmus auf die Bestimmung der zu teilenden Komponente über die lokale Verlustfunktion sowie ein festes Teilungsverhältnis verzichtet und es wird pro Eingangsdimension und Teilmodell jeweils ein Optimierungsdurchlauf ausgeführt. Damit verbleiben in jedem Durchlauf als zu definierende Startparameter die Teilungslinie τ_j und die Teilungsbreite β_j, welche folgendermaßen festgelegt werden:

Teilungsline τ_j: Es wird eine bestimmte Anzahl ν_τ von Startwerten für die Trennungslinie festgelegt, welche gleichverteilt in dem zu teilenden Bereich angeordnet werden. In diesem wird der in der jeweilige Eingangsdimension j definierte, halbe Übergangsbereich der Basisfunktion mit einbezogen. Ausgehend von den Parametern der Basisfunktion erfolgt die Anordnung somit innerhalb des Bereiches

$$\frac{a_j + b_j}{2} < \tau_j < \frac{c_j + d_j}{2}. \tag{3.52}$$

Die ν_τ Startwerte berechnen sich damit durch

$$\tau_j = \frac{a_j + b_j}{2} + i\left(\frac{c_j + d_j}{2} - \frac{a_j + b_j}{2}\right)\frac{1}{\nu_\tau - 2} \quad \text{mit} \quad i \in \{1, 2, \ldots, \nu_\tau - 2\}. \tag{3.53}$$

Insgesamt ergeben sich durch dieses Vorgehen mit K Teilmodellen und j Eingangsdimensionen $\nu_\tau K j$ Optimierungsdurchläufe.

Teilungsbreite β_j: Da sich die Breiten der Teilmodelle mit jeder neuen Trennungslinie und in jeder Dimension unterscheiden, ist eine absolute Definition der Startwerte für die Teilungsbreite nicht sinnvoll. Es wurden daher relative Teilungsbreiten β_{rel} als Startwerte definiert, aus denen die absolute Breite des Teilungsbereichs in Abhängigkeit der Breite des Teilmodells in der jeweiligen Dimension berechnet werden kann

$$\beta_j = \beta_{rel} \cdot \min\left\{\tau_j - \frac{a_j + b_j}{2}; \frac{c_j + d_j}{2} - \tau_j\right\}. \tag{3.54}$$

Da die Übergangszone nicht breiter als eines der beiden entstehenden Teilmodelle sein kann, wird als Basis zur Berechnung die Breite des schmaleren Teilmodells verwendet. Die ν_β Startwerte können dem Algorithmus als Vektor übergeben werden.

$$\boldsymbol{\beta}_{rel} = \begin{bmatrix}\beta_{rel,1}, & \beta_{rel,2}, & \cdots & \beta_{rel,\nu_\beta}\end{bmatrix}, \quad 0 < \beta_{rel,i} < 0,5 \tag{3.55}$$

Werden weichere Übergänge zwischen den Modellbereichen erwartet können hier größere Übergangsbreiten definiert werden, was sich durch die höheren Regularisierungseffekte auch bei einer geringen Anzahl von Messdaten empfiehlt. In der praktischen Realisierung haben sich 3 bis 5 Startwerte für β_{rel} von 5 % bis 45% als vorteilhaft erwiesen.

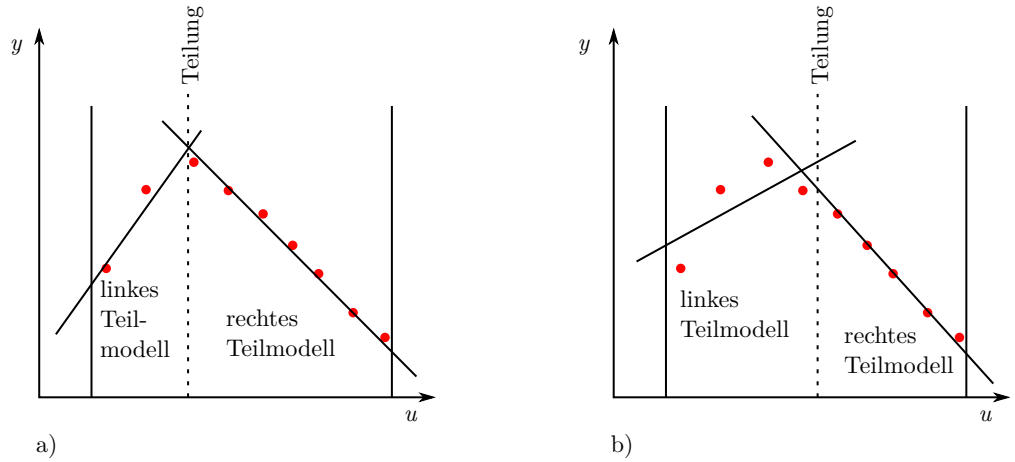

Abbildung 3.18.: Einfluss der Datenverteilung auf die Teilung eines Teilmodells, a) Optimale Teilung, welche auf Grund fehlender Messdaten im linken Teilmodell nicht ausgeführt werden kann. b) Aus der Datenverteilung resultierende suboptimale Teilung des Teilmodells

Zusammen mit den Startwerten der Teilungslinien ergeben sich für den Algorithmus insgesamt

$$\nu_{opt} = \nu_\tau \nu_\beta j K \tag{3.56}$$

Optimierungsdurchläufe. Zur Reduzierung des Rechenaufwands bei steigenden Teilmodellanzahlen, wurde eine Option in den Algorithmus integriert, welche nur die Teilmodelle mit den höchsten lokalen Verlustfunktionen in die Optimierung einbezieht. Hier hat sich bewährt, die Anzahl auf die schlechtesten 5 bis 10 Teilmodelle zu begrenzen. Diese Option erhöht die Gefahr einer nichtoptimalen Teilung nur unwesentlich, verringert den Rechenaufwand in Optimierungen mit großer Teilmodellanzahl jedoch erheblich.

Für alle Optimierungsdurchläufe wird die zugehörige Modellgüte bestimmt und mit den Parametern gespeichert. Sind alle Optimierungsdurchläufe abgeschlossen, erfolgt die Teilung des Modells mit den Parametern des besten Durchlaufs.

Weitere Eigenschaften des ILMON-Algorithmus Voraussetzung für die Teilung eines Teilmodells ist das Vorhandensein der nach Gleichung (3.43) definierten Mindestanzahl von Datenpunkten in beiden entstehenden Teilmodellen. Ist diese Bedingung für eine Teilung nicht erfüllt, wird diese im Optimierungsdurchlauf verworfen. Bei wenigen oder ungünstig verteilten Messdaten birgt dieses Vorgehen die Gefahr, dass die Modellanpassungen nur in Bereichen stattfinden, die mit genügend Messdaten besetzt sind. Das Modell passt sich der Verteilung der Messdaten an und nicht dem Verlauf der Ausgangsgröße. Der Sachverhalt ist in Abbildung 3.18 dargestellt.

Um die Gefahr dieser Fehlanpassung zu reduzieren, ist es empfehlenswert, die effektive Mindestanzahl an Datenpunkten in einem zu teilenden Teilmodell höher zu wählen als für die Parameterschätzung erforderlich. Bei Verwendung realer Messdaten hat sich eine Mindestanzahl von $3 \cdot \nu_{eff}$ bewährt. Wird diese unterschritten, wird der Modellierungsalgorithmus abgebrochen und es können weitere Messdaten für den betreffenden Modellbereich aufgenommen werden. Auf die beschriebene Problematik wird unter dem Gesichtspunkt der

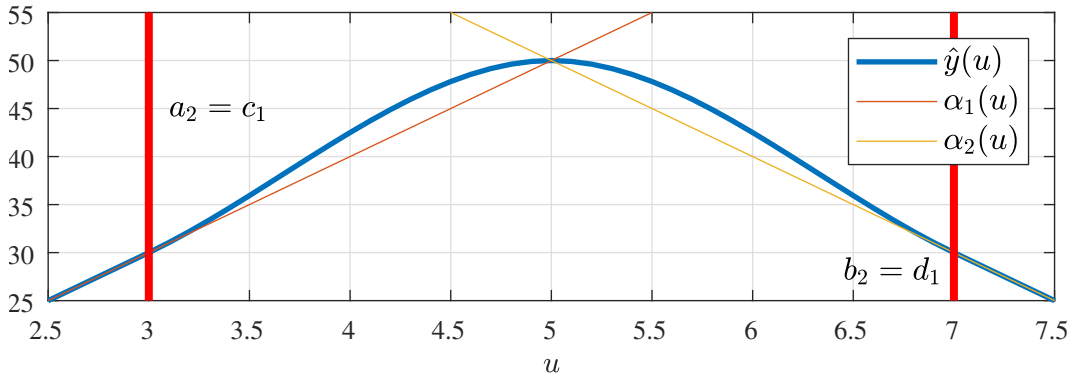

Abbildung 3.19.: Überhöhung des Modellausgangs $\hat{y}$, wenn sich der Schnittpunkt der linearen Komponenten α im Übergangsbereich $a_2 < u_s < b_2$ befindet

Versuchsplanung in Kapitel 4.2.3 ausführlicher eingegangen.

Liegen innerhalb eines Übergangsbereiches keine Datenpunkte, so ist der Modellfehler unabhängig von den Parametern der betreffenden Basisfunktionen. Deren Werte werden somit unabhängig von den Messdaten geschätzt und nehmen im Verlauf der Modellierung zufällige Werte an. Um unerwartete Verläufe zu vermeiden und alle Parameter auf Grundlage des Datensatzes zu bestimmen, wurde als Bedingung das Vorhandensein mindestens eines Datenpunktes in jeder Übergangszone in den Optimierungsablauf integriert. Ist dies bei einer optimierten Teilung nicht der Fall, werden die Parameter verworfen. Bei dünn mit Messdaten besetzten Eingangsräumen empfiehlt sich daher, mindestens einen relativ hohen Startwert für die Teilungsbreite, z.B. $\beta_j \approx 0.45$ zu definieren.

Mit dem Aufteilen eines Teilmodells werde die Parameter der angrenzenden Bereiche zwar nicht direkt beeinflusst, da sich der Verlauf der Ausgangsgröße in den Übergangszonen aber aus zwei Teilmodellen ergibt, können die Parameter der angrenzenden Teilmodelle nach der Teilung nicht mehr optimal bezüglich der Messwerte sein. Um hier eine höhere Modellgüte zu erreichen, wurde nach jeder vollzogenen Teilung und vor der Überprüfung der Modellgüte ein zusätzlicher Optimierungsdurchlauf eingeführt, der die Parameter aller Basisfunktionen in ihr Optimum verschiebt. Als Startwerte für diese Optimierung werden die jeweils aktuellen Parameter der Basisfunktionen verwendet.

Interpolationsverhalten Das Interpolationsverhalten eines ILMON-Modells ist über die linearen Funktionen klar definiert und auch in hochdimensionalen Modellen gut nachvollziehbar. In den Übergangsbereichen zwischen den Teilmodellen kommt es zu zwei verschiedenen Verlaufvarianten. Liegt die Schnittpunkt der linearen Komponenten α_1, α_2 zweier Teilmodelle M_1, M_2 innerhalb der Übergangszone einer Eingangsdimension j, gilt also

$$(a_{2,j} = c_{1,j}) \leq \boldsymbol{u}_{j,s} \leq (b_{2,j} = d_{1,j}) \quad \text{mit} \quad \alpha_1(\boldsymbol{u}_s) = \alpha_2(\boldsymbol{u}_s) \quad \text{und} \quad \varphi(\boldsymbol{u}_s) > 0, \tag{3.57}$$

so kommt es im Bereich um den Schnittpunkt $\boldsymbol{u}_s$ zu einer Überhöhung des Ausgangsgrößenverlaufs. Die Ausgangsgröße $\hat{y}$ an der Stelle $\boldsymbol{u}_s$ stimmt dabei mit dem Schnittpunkt der linearen Komponenten überein, siehe Abbildung 3.19.

In realen Prozessen sind solche Verläufe der Ausgangsgröße eher ungewöhnlich und stellen damit einen Nachteil des ILMON-Modellierung dar. Der Effekt ist um so größer, je stärker

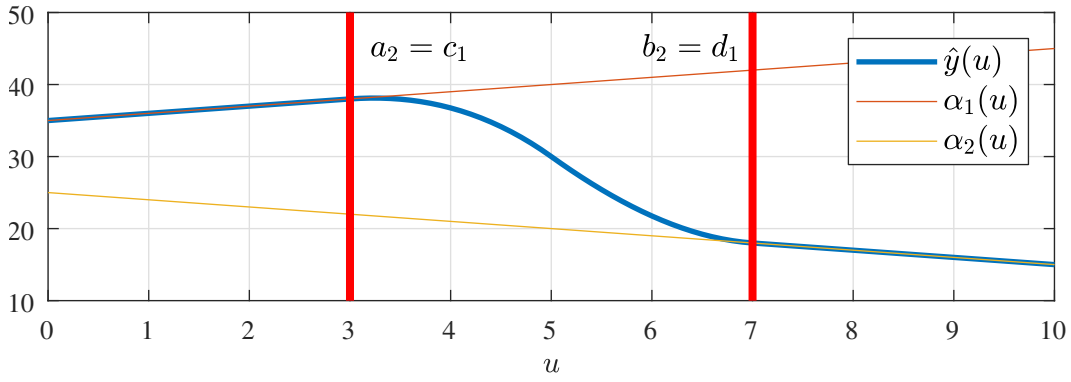

Abbildung 3.20.: Interpolation der Ausgangsgröße eines ILMON-Modells zwischen zwei Teilmodellen für $u_s < a_2$

die Anstiege der linearen Funktionen in der Dimension j voneinander abweichen. Da sich allgemein mit Erhöhung der Teilmodellanzahl auch die Anstiege der benachbarten Bereiche annähern, verliert sich diese Überhöhung mit zunehmender Modellkomplexität. Es finden sich in der Literatur verschiedene Vorschläge, diese Überhöhung für lokal-linearen Modelle zu vermeiden [67], [68], [69]. Nachteil all dieser Ansätze ist die starke Einschränkung der Interpretierbarkeit des resultierenden Modells, weshalb im Rahmen dieser Arbeit auf die Integration dieser Lösungen verzichtet wurde.

Liegt der Schnittpunkt der linearen Komponenten außerhalb des Übergangsbereiches, so ergibt sich ein S-förmiger Verlauf entsprechend der Flanke der Basisfunktion, welcher nahe der direkten Interpolation zwischen den Rändern des Übergangsbereiches liegt und weitgehend dem intuitiv erwartetem Verlauf entspricht. In Abbildung 3.20 ist dies beispielhaft dargestellt.

Ressourcenbedarf Der Entwurf des ILMON-Algorithmus erfolgte unter der Maßgabe einer ressourcenschonende Berechnung des Modells zum Einsatz in aktuellen Motorsteuergeräten. Die Berechnung der Ausgangsgröße $\hat{y}$ bei einer gegebenen Eingangsgröße $\boldsymbol{u}$ aus einem bestehenden Modell erfolgt nach Gleichung (3.1), (3.2) und (3.14) mit

$$\hat{y} = \sum_{k=1}^{K} \alpha_k(\boldsymbol{u}, \boldsymbol{\gamma}_k)\varphi_k(\boldsymbol{u}, \boldsymbol{a}_k, \boldsymbol{b}_k, \boldsymbol{c}_k, \boldsymbol{d}_k). \tag{3.58}$$

Nach dieser sind zur Berechnung folgende Schritte notwendig:

1. Berechnung aller K Basisfunktionen φ_k an der Stelle $\boldsymbol{u}$
2. Berechnung aller K Komponenten α_k an der Stelle $\boldsymbol{u}$
3. Multiplikation der Werte aus Basisfunktion und linearer Komponente sowie Aufsummieren aller Werte

Mit der vorgestellten Basisfunktion sind zur Berechnung nur Multiplikationen notwendig, was eine effiziente Umsetzung in der in Steuergeräten hauptsächlich eingesetzten Festkommaarithmetik möglich macht. Da auch keine Normierung der Basisfunktionen notwendig ist, hängt die benötigte Rechenleistung hauptsächlich von der Anzahl der Teilmodelle ab.

Unter Ausnutzung der speziellen Eigenschaften der Basisfunktion, lässt sich die Effizienz der Berechnung weiter steigern. Die in Gleichung (3.14) vorhandenen Fallunterscheidungen können für alle Basisfunktionen im Modell ohne Berechnung über einen simplen Parametervergleich durchgeführt werden. Befindet sich $\boldsymbol{u}$ im Kernbereich einer Basisfunktion oder außerhalb des Gültigkeitsbereichs, ist deren Wert per Definition $\varphi_k = 1$ oder $\varphi_k = 0$ und muss nicht berechnet werden. Damit reduziert sich der Rechenaufwand im ungünstigsten Fall auf die Bestimmung der Basisfunktionen, in deren Übergangsbereich der Eingangswert $\boldsymbol{u}$ liegt.

Analog dazu müssen nur die relevanten linearen Komponenten $\alpha_k(\boldsymbol{u})$ bestimmt werden, für deren Basisfunktion $\varphi_k(\boldsymbol{u}) \neq 0$ gilt. Zusammengefasst ergibt sich damit folgendes Vorgehen:

1. Filtern aller Basisfunktionen nach $(b_j < u_j < c_j)$, mit der Ergebnismenge aller Basisfunktionen $V = \{\varphi_k | \varphi_k = 1\}$; u_j ist der Eingangswert in der jeweiligen Eingangsdimension j
2. Filtern aller Basisfunktionen nach $(a_j \leq u_j \leq b_j) \wedge (c_j \leq u_j \leq d_j)$ mit der Ergebnismenge $W = \{\varphi_k | 0 < \varphi_k < 1\}$
3. Berechnung der Basisfunktionswerte der Mengen V und W sowie aller linearen Komponenten $\alpha_k(\boldsymbol{u})$ die zu diesen Basisfunktionen gehören.
4. Berechnen aller Produkte $\alpha_k \varphi_k$ der Mengen V und W sowie Aufsummieren der Ergebnisse

Mit diesem Vorgehen ist die Berechnung unabhängig von der Teilmodellanzahl und erfordert hauptsächlich simple Variablenvergleiche, die in der aktuellen Steuergerätehardware mit geringem Aufwand erfolgen können. Abhängig vom Anteil der Übergangsregionen am gesamten Eingangsraum liegen in der praktischen Anwendung die meisten Eingangswerte innerhalb von Kernregionen der Basisfunktionen, sodass pro Datenpunkt $\boldsymbol{u}$ in der Regel nur eine lineare Komponente $\alpha_k(\boldsymbol{u})$ eines Teilmodells berechnet werden muss.

Die benötigte Rechenleistung für die Berechnung der Ausgangsgleichung des ILMON-Modells ist damit äußerst gering. Eine quantitative Betrachtung sowie ein Vergleich mit LOLIMOT und GMR erfolgt in Kapitel 5.1.5 anhand eines konkreten Beispiels.

Neben der Rechenleistung ist der notwendige Speicherbedarf eine praktisch relevante Größe. Dieser ergibt sich für ein ILMON-Modell aus der Parameteranzahl der Basisfunktionen und der linearen Komponenten. Mit der Anzahl der Eingangsdimensionen q werden $4q$ Parameter pro Basisfunktion φ_k gespeichert. Pro linearer Komponente α_k werden $q+1$ Parameter benötigt. Damit ergibt sich für die Teilmodellanzahl K die Gesamtparameteranzahl mit

$$\nu_p = K\,(5q + 1)\,. \tag{3.59}$$

Im Vergleich hat ein ILMON-Modell einen etwas höheren Speicherbedarf als ein LOLIMOT-Modell mit $K\,(3q + 1)$ Parametern und einen sehr viel niedrigeren Speicherbedarf als ein Gaussian-Mixture-Regression-Modell mit $K\,(q^2 + 2q + 1)$.

3.6. Erweiterung auf quadratische Regressionsterme

Neben den lokal-linearen Komponenten in den Teilmodellen können auch komplexere Funktionen mit einer größeren Anzahl an Freiheitsgraden eingesetzt werden. Die so erhöhte Fle-

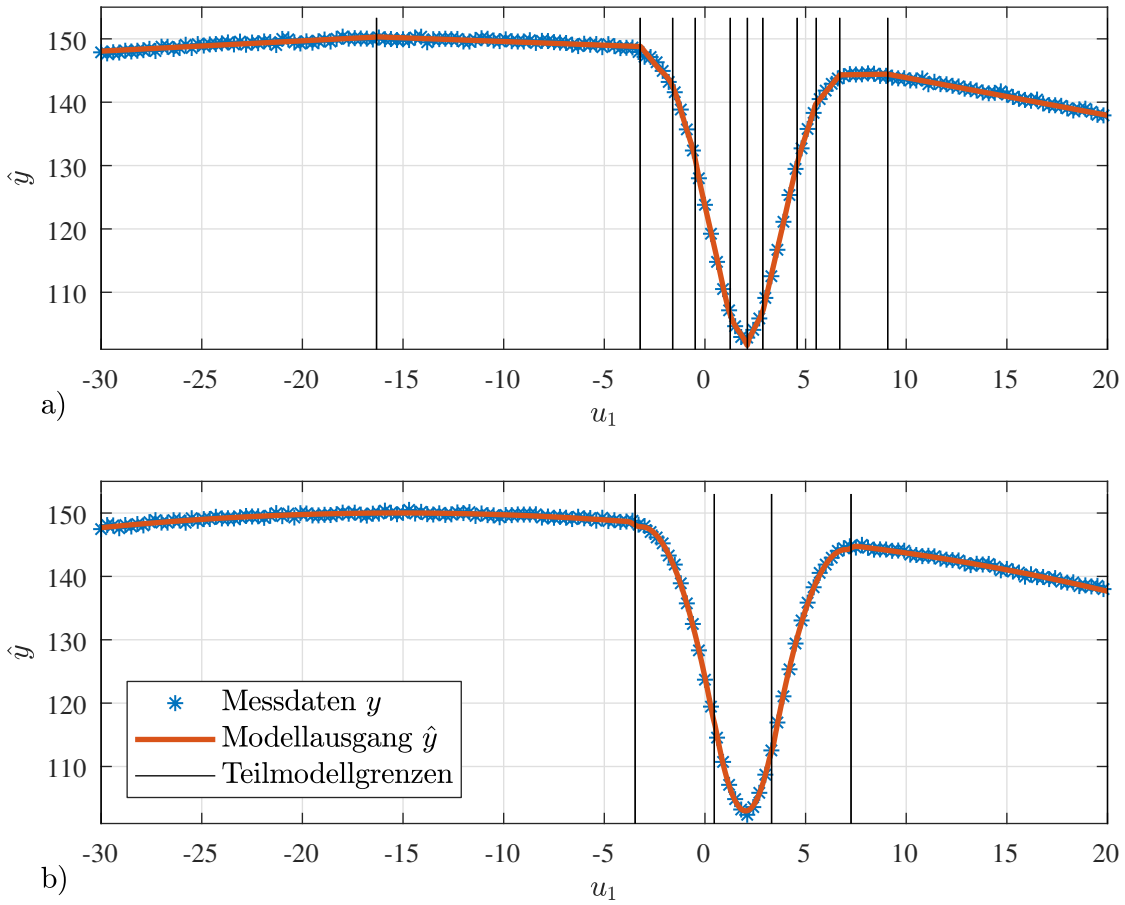

Abbildung 3.21.: Vergleich zweier ILMON-Modelle mit linearen und zusätzlichen quadratischen Termen. a) ILMON-Modell mit 12 Teilmodellen und ausschließlich linearen Regressoren, b) ILMON-Modell mit 5 Teilmodellen und mit zusätzlichem quadratischen Regressor

xibilität des Modells ermöglicht es, komplexe Prozesse mit weniger Teilmodelle abzubilden, was der Interpretierbarkeit entgegen kommt. Andererseits werden mit der Erhöhung der Freiheitsgrade auch die in Kapitel 2.3 aufgeführten Nachteile in die Modellierung integriert, insbesondere die Neigung zur Überanpassung und ein nachteiliges Inter- und Extrapolationsverhalten. Abwägend zwischen diesen Nachteilen und den Vorteilen einer niedrigeren Teilmodellanzahl stellt der Einsatz quadratischer Polynommodelle einen Kompromiss dar, der in einigen Anwendungen von Nutzen sein kann.

Mit der Erweiterung der linearen Definition der lokalen Komponenten um die vollständigen quadratischen Regressionsterme, ergibt sich die lokale Funktion zu

$$\alpha_k(\boldsymbol{u}) = \gamma_{0,k} + \sum_{i=1}^{q} \gamma_{ik} u_{ik} + \sum_{i=1}^{q} \sum_{j=i}^{q} \gamma_{ijk} u_{ik} u_{jk}. \tag{3.60}$$

Damit erhöht sich die Anzahl der Parameter einer lokalen Funktion proportional um $q/2$ von

$q+1$ auf

$$\nu_{p,\alpha} = \frac{q}{2}(q+1), \tag{3.61}$$

was bei höherdimensionalen Modellen die Parameterzahl und damit auch den Varianzfehler erheblich erhöht.

Die quadratischen Regressionsterme nach Gleichung (3.60) müssen aber nicht vollständig übernommen werden, sondern können und sollten auf die Terme reduziert werden, mit denen die Ausgangsgröße tatsächlich korreliert. Für die Auswahl können Verfahren der Variablenselektion, zum Beispiel Stepwise-Regression [70] oder Lasso (Least absolute shrinkage and selection operator [71]) genutzt werden. Diese Verfahren stellen allerdings oft erhöhte Anforderungen an die Datenbasis, wie eine konstante Varianz des Messfehlers oder haben spezielle Voraussetzungen an das Modell, welche bei Nichteinhaltung zu einer suboptimalen Auswahl der Terme führen. Da in der Praxis diese Anforderungen oft nicht gegeben sind oder zugesichert werden können, ist es empfehlenswert, die Auswahl der Regressionsterme manuell auf Grund von Prozesswissen vorzunehmen.

Ist die Ausgangsgröße über den gesamten Eingangsbereich oder in Teilbereichen mit einer Eingangsgröße quadratisch korreliert, kann dies erhebliche Vorteile für die Reduzierung der Teilmodellanzahl bei gleichen Güteforderungen bedeuten. In Bild 3.21 ist das Ergebnis zweier Modellierungen mit den Messdaten, dem Ausgangsgrößenverlauf und den Teilmodellgrenzen dargestellt. Während für die vorgegebene Modellgüte mit ausschließlich linearen Termen im lokalen Modell 12 Teilmodelle notwendig sind, können die selben Güteforderungen mit zusätzlichem quadratischem Term in der linearen Komponente

$$\alpha_k(u) = \gamma_{0,k} + \gamma_{1,k}u + \gamma_{2,k}u^2 \tag{3.62}$$

mit nur 5 Teilmodellen erreicht werden.

3.7. Zusammenfassung

In diesem Kapitel wurde eine im Rahmen dieser Arbeit entwickelte, gut interpretierbare lokale Modellstruktur vorgestellt, welche insbesondere den Anforderungen der datenbasierten Modellierung in der Motorenentwicklung im Hinblick auf eine gute Validierbarkeit entgegen kommt.

Dafür wurde ausgehend von den im Vorfeld aufgestellten Kriterien eine geeigneten Modellstruktur diskutiert und ausgewählt. Aufbauend auf dieser lokal-linearen Struktur erfolgte die Herleitung und Definition einer multivariaten Basisfunktion, welche die Lokalität der Parameter sowie ein einfaches Interpolationsverhalten garantiert. Neben diesen Kriterien ermöglicht die hier vorgestellte Basisfunktion eine variables Extrapolationsverhalten und die Möglichkeit der einfachen Integration von A-priori-Wissen. Die achsenorthogonale Definition der Gültigkeitsbereiche gestattet eine einfache Abgrenzung der Modellbereiche auch im optimierten Modell.

Es wurde ein iterativer Optimierungsalgorithmus entwickelt, welcher über eine hierarchische Partitionierung des Eingangsraumes die Modellkomplexität bis zu einer vorgegebenen Modellgüte erhöht. Das zugrunde liegende Gütekriterium kann zudem frei gewählt werden. Die Partitionierung erfolgt weitgehend optimal und ermöglicht eine Approximation des Prozesses mit geringer Teilmodellanzahl.

Ein geringer Ressourcenbedarf zur Berechnung der Ausgangsgleichung ist eine Voraussetzung für den Einsatz auf Motorsteuergeräten und konnte im vorgestellten Verfahren umgesetzt werden. Neben der unter diesem Gesichtspunkt entworfenen Basisfunktion, wurden weitere Wege zur effektiven Berechnung der Ausgangsgröße aufgezeigt.

Mit der Erweiterung der lokal-linearen Komponenten um quadratische Regressoren konnte die Flexibilität der Modellstruktur deutlich erhöht werden, ohne die Interpretierbarkeit zu stark einzuschränken.

4. Iterative Versuchsplanung und Modellierung

Die Qualität der für die Modellierung notwendigen Datenbasis ist entscheidend für die erreichbare Güte des Modells. Umso mehr gilt dies, wenn die Quantität der Daten, bedingt durch den Fluch der Dimensionalität oder durch zeitliche, personelle und finanzielle Einschränkungen, limitiert ist. Verfahren der Versuchsplanung dienen zur effizienten und ressourcenschonenden Bestimmung der für den Prozess optimalen Lage der Messpunkte.

In diesem Kapitel soll der Fragestellung nachgegangen werden, wie die Messpunkte einer ILMON-Modellierung optimal bestimmt werden können, um neben der Erhöhung der Modellgüte, deren Anzahl zu minimieren. Dabei stand die Idee im Vordergrund, den stetigen Informationsgewinn der iterativen Modellierung für eine kontinuierliche Anpassung der Versuchsplanung an den Prozess zu nutzen. Zu diesem Zweck soll eine serielle Versuchsplanung in den iterativen ILMON-Modellierungsalgorithmus eingebunden werden, welche in jedem Iterationsschritt eine optimale Messpunkteverteilung hinsichtlich des aktuellen Modells realisiert.

In Kapitel 4.1 wird zunächst ein Überblick über die klassischen Verfahren der Versuchsplanung gegeben und auf die grundsätzlichen Unterschiede zwischen geometrischen Versuchsplänen, modelloptimalen Verfahren sowie sequentiellen Versuchsabläufen eingegangen. Die Idee der Verknüpfung von Modellierungsprozess und Versuchsplanung wird im nachfolgenden Kapitel diskutiert. In Abschnitt 4.3 wird schließlich die Umsetzung der in die ILMON-Modellierung integrierten Versuchsplanung sowie deren Eigenschaften erläutert.

4.1. Verfahren der statistischen Versuchsplanung

Die statistischen Versuchsplanung dient unter der Vorgabe von Genauigkeits- und Zuverlässigkeitsanforderungen der Bestimmung einer optimale Anzahl und Verteilung von Messpunkten. Weiterführend wird damit eine Minimierung des Versuchsaufwands angestrebt. Gerade bei hochdimensionalen Prozessen ist es oft erst durch diese Optimierung möglich, einen praktikablen Versuchsumfang zu erreichen.

Ein weiteres Aufgabenfeld der statistischen Versuchsplanung ist die Erkennung von Zusammenhängen zwischen Ein- und Ausgangsgrößen. Es lassen sich vier große Gruppen von Verfahren unterscheiden [72], [73]:

1. *Klassische Versuchspläne*: Die klassischen Varianten der Versuchsplanung haben als Ausgangspunkt den sogenannten Vollfaktorplan, indem alle Variationen einer Eingangsgröße, in der Versuchsplanung Faktoren genannt, berücksichtigt werden und deren Auswirkungen auf die Ausgangsgröße vollständig vermessen wird. Um diesen Aufwand zu reduzieren, werden in Teilfaktorplänen nur Kombinationen der Eingangsgrößenvariation getestet, aus denen die Wirkung der einzelnen Eingangsgrößen auf die Ausgangsgröße mit möglichst geringem Informationsverlust ebenfalls bestimmbar ist.

Diese Screening-Versuchspläne basieren auf einem linearen Beschreibungsmodell, wodurch der Einsatzbereich eng eingegrenzt wird. Sie werden nach bestimmten Schemen aufgebaut, zum Beispiel dem Yates-Standard oder der Plackett-Burman-Konstruktion, welche den Anforderungen nach Orthogonalität und Ausgewogenheit genügen. Erweiterungen auf ein quadratisches Beschreibungsmodell führen zu geometrischen Anordnungen der Versuchspunkte, wie dem Central-Composite-Design oder dem Box-Behnken-Design [72]. Höhere Modellordnungen lassen sich zwar ebenfalls umsetzen, sind mit solchen festen Anordnungen aber nur noch in speziellen Fällen sinnvoll.

2. *Optimale Versuchspläne*: Kommen in realen Prozessen Begrenzungen und Nebenbedingungen ins Spiel oder soll das absolute Minimum an Versuchen erreicht werden, können Versuchspläne eingesetzt werden, welche auf ein bestimmtes Kriterium hin optimiert wurden. Diese Optimierung erfordert neben der Definition des Kriteriums auch die Festlegung auf ein genaues Beschreibungsmodell, meist ein quadratisches oder kubisches Polynommodell. Weicht diese Modellbeschreibung von dem realen Prozess ab, ist auch die Allokation der Versuchspunkte nicht mehr optimal, was einen großen Nachteil dieser Verfahren darstellt. Trotzdem haben diese Versuchspläne im Bereich der Motorenentwicklung eine große Verbreitung gefunden, vor allem auf Grund der dort gut etablierten Polynommodelle und der einfachen praktischen Handhabung. Insbesondere die einfache Möglichkeit, bestimmte nicht anfahrbare oder nicht relevante Modellbereiche aus der Versuchsplanung auszuschließen, ist in der praktischen Anwendung ein großer Vorteil.

   Bei der Vermessung und Applikation von Verbrennungsmotoren werden häufig D-optimale Versuchspläne eingesetzt [74], [75], [76]. Bei diesen wird die Varianz der Parameter eines linearen Regressionsmodells minimiert, was gleichbedeutend mit der Maximierung des Informationsgehaltes der Messdaten ist. Dies wird über die Maximierung der Determinante (D-optimal) der Designmatrix $|\boldsymbol{X}^T\boldsymbol{X}|$ erreicht, was dem Verfahren seinen Namen gab [77].

   Zur Veranschaulichung des Prinzips ist in Bild 4.1 die Änderung die Parametervarianz einer linearen Funktion auf Grund der unterschiedlichen Lage der Messpunkte dargestellt. Bei gleicher Fehlervarianz, in der Abbildung als Fehlerbalken dargestellt, variiert die Lage der geschätzten linearen Funktion bei einem großen Abstand der Messpunkte im Eingangsraum nur wenig. Die Varianz der Funktionsparameter ist gering. Liegen die beiden Messpunkte dichter beieinander, wirkt sich die Fehlervarianz wesentlich stärker auf den Schätzwert der Parameter aus. Dieser Umstand wird durch den Wert der Determinante der Designmatrix beziffert. In Abbildung 4.1a ist der Informationsgehalt der Messpunkte mit $|\boldsymbol{X}^T\boldsymbol{X}| = 82$ hoch, während er in 4.1b mit $|\boldsymbol{X}^T\boldsymbol{X}| = 52$ niedriger ist.

   Neben dem D-Kriterium gibt es weitere häufig eingesetzte Designs zur optimalen Allokation der Versuchspunkte [78], [72]:

   - A-optimal: Dieses Kriterium minimiert die mittlere Varianz der Regressionskoeffizienten durch Minimierung der Summe der Hauptdiagonalelemente der inversen Designmatrix $\text{Spur}((\boldsymbol{X}^T\boldsymbol{X})^1)$
   - G-optimal: Hier wird die maximal auftretende Varianz der Schätzwerte $\hat{y}$ minimiert. Dies geschieht über die Minimierung des größten Hauptdiagonalelements

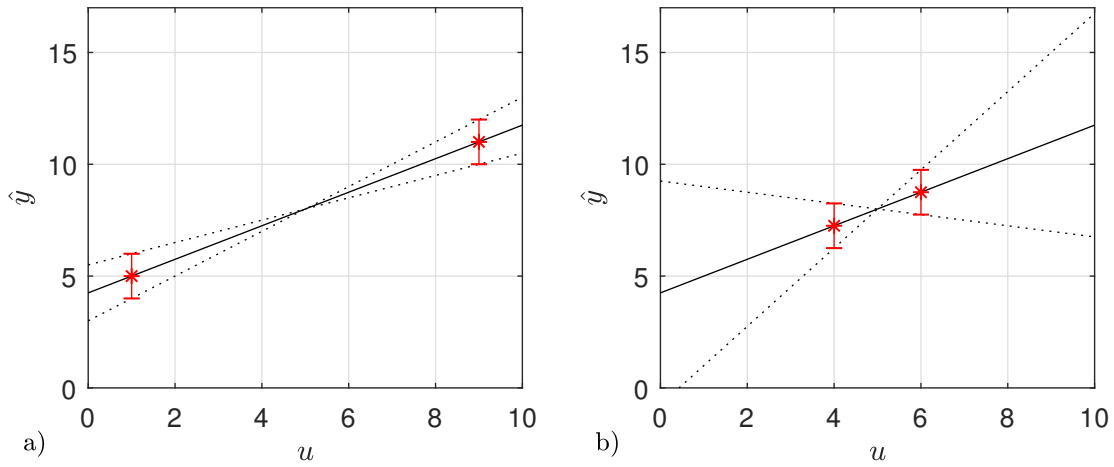

Abbildung 4.1.: Erhöhung der Parametervarianz einer linearen Funktion bei veränderter Messpunkteverteilung und gleicher Fehlervarianz. a) niedrige Parametervarianz durch hohen Informationsgehalt der Messpunkte ($|\boldsymbol{X}^T\boldsymbol{X}| = 82$), b) hohe Parametervarianz durch niedrigen Informationsgehalt der Messpunkte ($|\boldsymbol{X}^T\boldsymbol{X}| = 52$)

der Glättungsmatrix $\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$, welche den Zusammenhang zwischen den Schätzwerten und dem Prozessausgang definiert, siehe Gleichung (3.38).

- I-optimal: Optimiert den Versuchsplan auf die minimale, mittlere Varianz der Schätzwerte $\hat{y}$
- V-optimal: Minimiert wird hier ebenfalls auf die mittlere Varianz der Schätzwerte, wobei dies über eine Menge spezifischer Punkte erfolgt.

Bei allen optimalen Versuchsplänen muss neben dem Optimierungskriterium auch die Modellbeschreibung und die Anzahl der Versuche festgelegt werden. Mit falschen Modellannahmen und einer zu geringen Anzahl an Messpunkten ist die resultierenden Messpunkteallokation nicht mehr optimal und abweichende Prozessverläufe können unerkannt bleiben. Optimale Versuchspläne eigenen sich daher gut für weniger komplexe Prozesse mit einer bekannten Struktur der Modellbeschreibung. Für stark nichtlineare Ausgangsgrößenverläufe und unbekannte Abhängigkeiten der Ausgangsgröße von den Eingängen sind diese Verfahren weniger gut geeignet [72].

3. *Versuchspläne für komplexe Zusammenhänge:* Bei der Vermessung komplexer nichtlinearer Funktionen stoßen klassische und optimale Versuchspläne mit ihren linearen Regressionsmodellen an ihre Grenzen. Ist A-Priori-Wissen über den zu modellierenden Prozess vorhanden oder liegen qualitative Informationen zum Verlauf der Ausgangsgröße vor, können spezielle Versuchsabläufe entwickelt werden. Fehlen diese Informationen, so können Versuchspläne nur auf Grundlage der Verteilung in den Eingangsgrößen aufgestellt werden und zielen somit auf einen gleichmäßigen und hohen Informationsgewinn im gesamten Eingangsraum. Praktisch bedingt dies eine gleichmäßige und dichte Abdeckung des Eingangsraumes mit Messpunkten, um alle Nichtlinearitäten abbilden zu können. Zu diesem Zweck kommen verschiedene Verfahren zum Einsatz, welche eine optimale Verteilung auf Grund statistischer Annahmen ausschließlich auf Grundlage der Eingangsgrößen vornehmen. Bekannte Methoden hierfür

sind Monte-Carlo-Verfahren [72], Latin-Hypercubes [79] , Maximin-Distance-Designs [80] und Uniform-Designs [81].

Diese Konstruktionsmethoden setzen das Ziel eines gleichverteilten Messfeldes, je nach Anwendungsfall, unterschiedlich gut und nicht immer optimal um. Zur Bewertung der erzeugten Datensätze werden daher verschiedene Gütekriterien für die Gleichverteilung der Messdaten verwendet. Gängig sind hier das Minmax- und Maxmin-Kriterium [80], das mit verschiedenen Distanzmaßen die Abstände der Messpunkte zueinander bewertet sowie das Entropie-Kriterium [82], welches den Informationsgehalt der Messdaten über die Entropie quantifiziert. In [83] wird die Gleichverteilung über die sogenannte Diskrepanz bewertet. Diese setzt die Messpunkte in einem Teilbereich des Eingangsraumes zur Gesamtzahl der Messpunkte ins Verhältnis und vergleicht dies mit dem Volumen des jeweiligen Teilbereichs. Mit der L_p-Diskrepanz sowie der zentrierten und einhüllenden Diskrepanz werden verschiedene Kriterien definiert, welche die Gleichverteilung des Datensatzes bewerten. Ausführliche Informationen hierzu findet man in [72] und [82].

Auf Grund des Ziels einer gleichmäßigen und dichten Verteilung der Messdaten ist die Anzahl der notwendigen Messpunkte bei komplexen Zusammenhängen entsprechend hoch. Mit einer hohen Anzahl an Eingangsdimensionen ist dieses Ziel aus praktischen Gesichtspunkten oft nicht mehr ohne Informationen über den Prozess und der damit möglichen Konzentration auf die wesentlichen Teilbereiche des Eingangsraumes zu erfüllen, da der zeitliche und finanzielle Testaufwand erheblich wird. Dies stellt einen entscheidenden Nachteil dieser Verfahren dar. Stehen diese A-priori-Informationen nicht zur Verfügung, können sequentielle Versuchsverfahren eine Alternative bieten.

4. *Sequentielle Versuchsplanung* In der klassischen Versuchsplanung erfolgt die komplette Auswahl der Datenpunkte vor der eigentlichen Versuchsdurchführung und der Auswertung der gewonnenen Messdaten. Charakteristisch für sequentielle Verfahren ist, dass diese drei Schritte für jeden oder mehrere Messpunkte nacheinander abgearbeitet werden. Dabei beeinflusst das Ergebnis eines Versuchs je nach Verfahren sowohl die Auswahl der nachfolgenden Messpunkte als auch die Entscheidung, ob weitere Messpunkte aufgenommen werden sollen. Der Versuchsumfang wird somit erst auf Grundlage der Messungen bestimmt und nicht wie bei den obigen Verfahren im Vorfeld festgelegt. Je nach Vorgehen werden drei prinzipielle Methoden unterschieden [84]:

   - Offene sequentielle Versuchspläne, bei denen nach jedem Versuch die Auswertung erfolgt, ob ein weiterer Versuch erforderlich ist. Eine Maximalzahl an Messungen wird hier nicht festgelegt
   - Geschlossene sequentielle Versuchspläne, die zur Reduktion des Aufwandes eine maximale Anzahl von Versuchen vorgeben und bei Erreichen dieser Grenze den Versuchsplan abbrechen
   - Gruppierte sequentielle Versuchspläne, bei denen pro Versuchsdurchgang eine bestimmte Anzahl an Messungen durchgeführt wird. Nach der Auswertung der in einem Versuchsdurchgang aufgenommenen Gruppe an Messpunkten wird gegebenenfalls ein neuer Versuchsdurchgang gestartet.

   Die Auswahl des nächsten Messpunktes kann von verschiedenen Kriterien abhängig gemacht werden. Typisch sind Entscheidungen nach statistischen Eigenschaften des

aktuell vorliegenden Datensatzes, der erreichten Güte des hinterlegten Modells sowie nach der bestmöglichen Unterscheidung verschiedener alternativer Modellansätzen [84], [85], [86].

Größter Vorteil der sequentiellen Verfahren ist die Nutzung der aus den vorangegangenen Versuchen gewonnenen Informationen für die Wahl des nächsten Messpunktes. Insbesondere bei Verfahren zur Modelldiskriminierung kann so die Auswahl der Versuchspunkte direkt auf diejenige Modellstruktur adaptiert werden, welche sich im Verlauf der Messkampagne bestätigt. Die Vorgabe einer Auswahl an Modellansätzen ist hier allerdings weiterhin notwendig.

Im Bereich des Maschinenlernens werden ebenfalls sequentielle Methoden verwendet, die dort unter dem Begriff „Active learning“ zusammengefasst werden [87]. Hier finden sich auch Verfahren, die eine Adaption der Modellkomplexität im Rahmen der vorgegebenen Modellstruktur an die Messdaten durchführen, z.B. in Neuro-Fuzzy-Modellen [88] oder Support-Vektor-Maschinen [89]. Die damit mögliche Reduktion der Versuchszahl auf die minimal notwendigen Messdaten macht dieses Vorgehen auch für den Einsatz in höherdimensionalen Modellen interessant, welche in der Praxis, bedingt durch den „Fluch der Dimensionalität“, oft das Problem einer sehr dünnen Datenbasis haben.

Im Vergleich der unterschiedlichen Ansätze zur Versuchsplanung lässt sich zusammenfassend sagen, dass für die Modellierung hochdimensionaler und komplexer Prozesse in der Motorentwicklung, die traditionellen optimalen Versuchspläne mit ihren linearen Regressionsmodellen nur sehr eingeschränkt geeignet sind. Klassische gleichverteilte Testfelder unterliegen wiederum stark dem Fluch der Dimensionalität und sind auf Grund des immensen Messaufwandes praktisch oft nicht vollständig umsetzbar. Sequentielle Verfahren, die Prozessinformationen aus vorangegangenen Messungen nutzen, um diese in die Bestimmung der neuen Messpunkte einfließen zu lassen, können dieses Problem abmildern und bieten sich als interessante Ergänzung des iterativen Modellierungsprozesses an. Eine Erweiterung der vorgestellten ILMON-Modellierung mit einer sequentiellen Versuchsplanung soll im nachfolgenden Abschnitt ausführlich behandelt werden.

4.2. Optimale Versuchsplanung für die iterative Modellierung

Die Versuchsplanung für eine iterative Modellierung, wie sie im Kapitel 3 vorgestellt wurde, unterscheidet sich in einigen wesentlichen Punkten von der klassischen statistischen Versuchsplanung. Zum einen existiert vor der Optimierung keine feste Modellstruktur mit einer definierten Anzahl an Parametern und zum anderen wird die Komplexität im Laufe der Modellierung kontinuierlich und in Abhängigkeit der Messwerte erhöht. Mit dieser Erweiterung auf Basis der Messdaten wird das Modell mit jeder Iteration dem realen Prozessverhalten angenähert. Damit steht nach jedem Durchlauf mehr Prozesswissen zur Verfügung, das für eine angepasste Messpunkteallokation genutzt werden kann.

Die Idee der Integration einer Versuchsplanung in diesen iterativen Prozess besteht nun darin, die Allokation der Versuchspunkte nicht komplett vor der Vermessung und der Modelloptimierung vorzunehmen. Statt dessen soll eine Versuchsplanung nach jeder Iteration auf Grundlage des jeweils aktuellen Modells erfolgen, um so das für den jeweiligen Zeitpunkt maximale Prozesswissen für eine optimale Verteilung der neuen Versuchspunkte zu nutzen. Weiterhin ermöglicht dieses Vorgehen, die Messdaten auch optimal für die Struktur-

und Parameteroptimierung des ILMON-Modells zu wählen und damit Fehlanpassungen des Modells auf Grund ungünstiger Messdaten zu vermeiden.

Die Unabhängigkeit der Teilmodelle einer ILMON-Modellierung lässt es zu, die Versuchsplanung getrennt für die einzelnen Teilbereiche zu betrachten. Ausgehend von dieser Modellstruktur ergeben sich verschiedenen Ziele und Anforderungen für die Allokation der Messpunkte:

1. Optimale Wahl der Messpunkte zur Parameteroptimierung der lokalen Funktionen $\alpha_k(\boldsymbol{u})$
2. Optimale Wahl der Messpunkte zur Erkennung von Nichtlinearitäten und zur Strukturoptimierung eines ILMON-Modells, welche sich in der Parameteroptimierung der Basisfunktionen $\varphi(\boldsymbol{u})$ widerspiegelt.
3. Berücksichtigung von Begrenzungen und nicht anfahrbaren beziehungsweise ausgeschlossenen Modellbereichen
4. Berücksichtigung von diskreten Stellgrößen und unterschiedlichen Stellgrößenrastern in den einzelnen Eingangsgrößen

Im Folgenden sollen die Möglichkeiten zur Umsetzung dieser Punkte diskutiert werden.

4.2.1. Versuchsplanung zur Parameteroptimierung

Die lokalen Funktionen $\alpha_k(\boldsymbol{u}|\boldsymbol{\gamma}_k)$ der Teilmodelle in einem ILMON-Modell nach Gleichung (3.30) sind als Linearkombinationen der Eingangsgrößen definiert und werden über die lineare Regression geschätzt. Ziel der Versuchsplanung soll es sein, die Varianz der Parameter $\boldsymbol{\gamma}_k$ zu minimieren. Dies kann über die Maximierung der Determinante der Designmatrix

$$\max\left(|\boldsymbol{X}^T\boldsymbol{X}|\right) \tag{4.1}$$

erreicht werden, siehe Abbildung 4.1. Gleichzeitig soll über die Berechnung des lokalen Modellfehlers

$$\epsilon_k = \sum_{i=1}^{N} \varphi_k(\boldsymbol{u}_i)\left(\hat{y}(\boldsymbol{u}_i) - y(\boldsymbol{u}_i)\right)^2 \tag{4.2}$$

erkennbar sein, wie gut sich der Prozess in diesem Bereich über die lokale Funktion des Teilmodells beschreiben lässt. Dies macht es notwendig, mehr Versuchspunkte aufzunehmen als für die Berechnung der Parameter nach Gleichung (3.36) notwendig ist, da abweichende Ausgangsgrößenverläufe sonst unerkannt bleiben könnten.

Mit der achsenorthogonalen Definition der Basisfunktionen ergeben sich im Eingangsraum Hyperquader als Teilgebiete, die im einfachsten Fall über klassische geometrische Versuchspläne wie dem Central-Composite- oder dem Box-Behnken-Design vermessen werden können, siehe Abbildung 4.2. Diese Pläne garantieren die minimale Varianz der Parameter der lokalen Funktion, setzen jedoch voraus, dass das Regressionsmodell den wahren Verlauf der Ausgangsgröße in diesem Bereich gut widerspiegelt, was zumindest bei Start der Optimierung in der Regeln nicht den Gegebenheiten entspricht. Damit eignen sie sich auch nur sehr eingeschränkt für die Bewertung der Modellgüte in diesem Bereich. Hinzu kommt, dass solche festen Verteilungsmuster eventuelle Begrenzungen und Ausschlüsse in den Teilbereichen

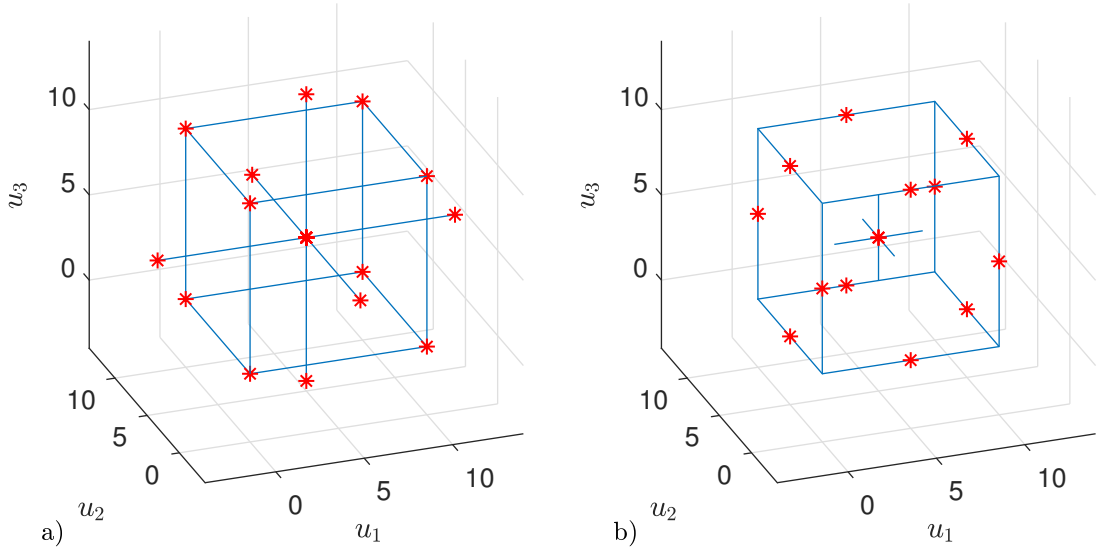

Abbildung 4.2.: Beispiel eines Versuchsplans im a) Central-Composite-Design und b) im Box-Behnken-Design für 3 Eingangsgrößen

nicht berücksichtigen können und die Rasterung von diskreten Eingangsgrößen in verschieden großen Bereichen nicht skaliert werden kann. In der Konsequenz wären Anpassungen in den Plänen notwendig, die deren Optimalität stark beeinträchtigen.

D-optimale Versuchspläne, welche die Varianz der Parameter nach Gleichung (4.1) minimieren, bieten die Möglichkeit die Messpunkteallokation für ein Teilmodell flexibel unter Berücksichtigung von Rasterungen, Begrenzungen und Ausschlüsse zu optimieren. Sie liefern jedoch schlechte Ergebnisse, wenn realer Prozess und lokale Funktion zu stark voneinander abweichen. Damit eignen sie sich nur eingeschränkt für den Einsatz zu Beginn einer ILMON-Modellierung, wenn Gebiete mit starken Nichtlinearitäten noch über wenige lineare Modelle approximiert werden. Zudem werden mit der Optimierung auf die minimale Varianz der Parameter immer nur die Punkte im Eingangsraum als Kandidaten ausgewählt, welche den höchsten Informationsgewinn bezüglich der Modelldefinition erzeugen. Eine realistische Bewertung des Modellfehlers bei einem von der Modelldefinition abweichenden Prozessverhalten ist darüber jedoch nicht möglich. Hierfür müssen zusätzliche Versuchspunkte manuell definiert werden, die nicht der optimalen Auswahl im Sinne des hinterlegten Modells entsprechen.

Besser als mit geometrischen und optimalen Versuchsplänen können die oben definierten Rahmenbedingungen und Ziele mit raumfüllenden Versuchsplänen nach dem Maxmin-Kriterium

$$\max \min_{i \neq j} \|x_i - x_j\| \tag{4.3}$$

erreicht werden, welches den minimalen Abstand über alle Messpunkte maximiert. Als Distanzmaß $\|\cdot\|$ wird häufig die normierte euklidische Distanz oder die Mahalanobis-Distanz verwendet

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j)}, \tag{4.4}$$

welche sich in der Wahl der Wichtungsmatrix $\boldsymbol{W}$ unterscheiden. Mit einer Diagonalmatrix,

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{\sigma_{u_1}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_{u_2}} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sigma_{u_q}} \end{bmatrix} \tag{4.5}$$

mit den reziproken Varianzen $1/\sigma_u$ der Eingangsgrößen auf der Hauptdiagonale, werden in der normierten euklidischen Distanz die Abstände achsenorthogonal, entsprechend der Eingangsgrößenbereiche verzehrt, was einer Normierung der Eingangsgrößen entspricht. Der Einheitskreis ergibt sich folglich als Ellipse mit zu den Koordinatenachsen parallelen Hauptachsen.

Sollen auch die Korrelationen der Eingangsgrößen untereinander berücksichtigt werden, wird als Wichtungsmatrix die inverse Kovarianzmatrix

$$\boldsymbol{W} = \mathrm{Cov}\,(\boldsymbol{X})^{-1} \tag{4.6}$$

genutzt, womit sich die Mahalanobis-Distanz ergibt. Der Einheitskreis kann hier auch achsenschräg verzehrt sein. Die Mahalanobis-Distanz bewertet über die Kovrianzmatrix den Abstand zweier Punkte in Abhängigkeit der gesamten Messmatrix. Mit auftretende Korrelationen verringert sich der Abstand der Messpunkte in dieser Richtung und neue Punkte werden automatisch unkorreliert gewählt. Diese Eigenschaft ist im Rahmen der Versuchsplanung sehr vorteilhaft, womit die Mahalanobis-Distanz als Abstandsmaß für den hier diskutierten Anwendungsfall die bessere Wahl ist.

Über das Maxmin-Kriterium erzeugte Versuchspläne minimieren, wie auch die D-optimalen Messpunkteallokationen, die Varianz der Parameter einer lokalen Funktion mit linearen oder quadratischen Termen [80]. Ebenso ist die Berücksichtigung von Begrenzungen und Ausschlüssen möglich. Anders als bei den optimalen Versuchsplänen ermöglicht das Kriterium jedoch eine beliebige Erweiterung des Messdatensatzes, sodass über die gleichverteilte Anordnung zusätzlicher Messpunkte eine Bewertung der Modellgüte über den Modellfehler möglich ist. Die Anzahl der Messpunkte pro Teilmodell kann dabei direkt als Maß für die Zuverlässigkeit des berechneten Modellfehlers verwendet werden. Je mehr Versuchspunkte im Teilbereich liegen, um so sicherer zeigt ein kleiner Modellfehler eine hohe Modellgüte an.

In Abbildung 4.3 ist der Vergleich eines D-optimalen Versuchsplans mit einem raumfüllenden Versuchsplan nach dem Maxmin-Kriterium für ein quadratisches Modell dargestellt. Der D-optimale Versuchsplan ist mit dieser Modellannahme eine Untermenge des raumfüllenden Versuchsplans, womit dieser bezüglich der Varianzminimierung die gleichen Eigenschaften hat.

Nachteil der Maxmin-Anordnung ist die schlechte Projektionseigenschaft in die niederdimensionalen Unterräume des Eingangsraumes, welche sich in einer Häufung der Messpunkte in den einzelnen Eingangsdimensionen und speziell an dessen Grenzen zeigt. In Abbildung 4.3a bilden sich die 13 Messpunkte in nur jeweils 5 Einstellungen in den Eingangsgrößen u_1 und u_2 ab. Mit einer Erhöhung der Eingangsdimension verschärft sich dieses Problem erheblich.

Monte-Carlo-Methoden, welche eine Gleichverteilung der Messpunkte anstreben, weisen mit stetigen Eingangsgrößen und einer hohen Zahl an Versuchspunkten bessere Projekti-

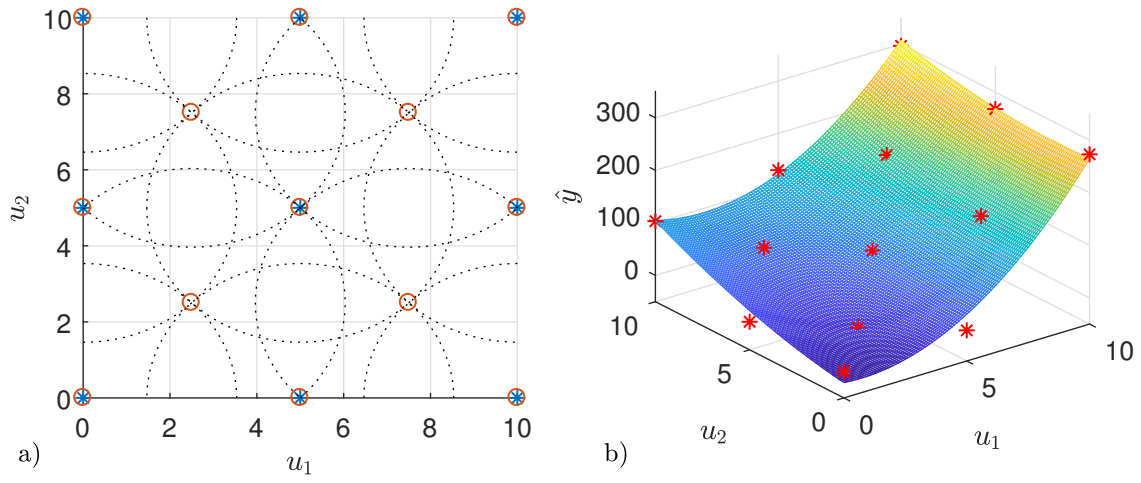

Abbildung 4.3.: Vergleich von D-optimaler Versuchsplanung und raumfüllendem Versuchsplan für eine quadratische Modellannahme, a) Die Messpunkte des D-optimalen Versuchsplans (Sterne) sind vollständig in der Maxmin-optimierten Anordnung (Kreise) enthalten. Die gestrichelten Linien stellen die maximierten Abstände zwischen den Punkten da. b) Mit den zusätzlichen Messpunkten kann die Modellannahme über den resultierenden Modellfehler besser überprüft werden.

onseigenschaften auf. Wie eingangs festgehalten, ist es jedoch das erklärte Ziel der Versuchsplanung, mit möglichst wenig Messdaten auszukommen. In realen Anwendungen an Verbrennungsmotoren sind zudem viele Stellgrößen nicht völlig frei einstellbar, sondern oft einer Rasterung unterworfen. Diese Vorgaben führen zu einer Verschlechterung der Eigenschaften. In Abbildung 4.4a ist eine Gleichverteilung im Eingangsraum bei einer Rasterung der Eingangsgrößen dargestellt. Wie zu sehen ist, kann es in Folge einer geringen Anzahl an Messdaten zu Clusterbildung und der wiederholten Messung an einer Rasterstelle in einer Eingangsgröße kommen. Vermeidet man die Häufung in einer Eingangsdimension durch eine Gleichverteilung in jeder einzelnen Stellgröße (Abbildung 4.4b), steigt die Gefahr der Clusterbildung und der Korrelation zwischen den Eingangsgrößen. Eine unkorrelierte Gleichverteilung im Eingangsraum ist damit nicht sichergestellt.

Latin-Hypercube-Verfahren sind von der mathematischen Spielerei des magischen Quadrates inspiriert und verteilen die Messpunkte in jeder Eingangsdimension in n gleichgroße Abschnitte unter der Bedingung, dass in jedem Abschnitt nur ein Messpunkt liegen darf. Sie garantieren eine gute Projektion dieser Verteilung in alle Unterräume. Abbildung 4.5 zeigt beispielhaft solch eine Messpunkteallokation. Durch eine zusätzliche Optimierung der Abstände über das Maximin-Kriterium kann weiterhin eine unkorrelierte Gleichverteilung der Messpunkte über den gesamten Eingangsraum gesichert werden. Nachteilig für die Anwendung im Rahmen dieser Arbeit ist die fehlende Möglichkeit gesperrte Bereiche zu berücksichtigen. Weiterhin ist es schwierig, Rastervorgaben der Stellgrößen in der Optimierung mit einzubeziehen. Letztlich können diese Verfahren auch nicht für die Erweiterung schon vorhandener Datensätze genutzt werden, was sie für den Einsatz in der sequentiellen Versuchsplanung ausschließt.

Im Vergleich der Verfahren weist die Erzeugung eines Versuchsplanes über das Maxmin-Kriterium mit Mahalanobis-Distanz einen Großteil der gewünschten Eigenschaften für die

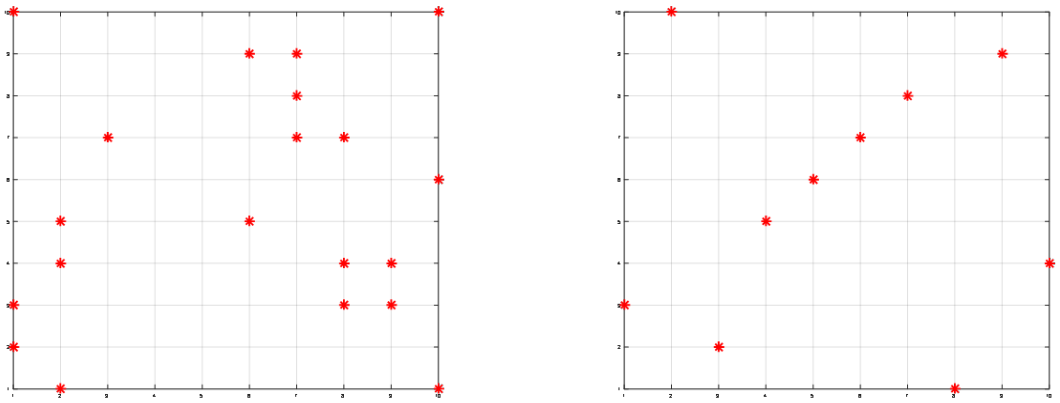

Abbildung 4.4.: Verteilung der Messdaten per Monte-Carlo-Verfahren bei Rasterung der Eingangsgrößen, a) Gleichverteilt im Eingangsraum, b) Gleichverteilt in den Eingangsgrößen

Parameteroptimierung auf und soll daher als Grundlage für die Versuchsplanung dienen. Auf die Besonderheiten der sequentiellen Variante diese Kriteriums und der praktischen Umsetzung wird im Kapitel 4.3 eingegangen. Dort werden auch das Vorgehen zur Vermeidung der schlechten Projektionseigenschaften und anderer praktischen Probleme erläutert.

4.2.2. Versuchsplanung zur Strukturoptimierung

Ziel der Strukturoptimierung innerhalb des ILMON-Algorithmus' ist das Finden der optimalen Teilungsparameter $k, j, \beta_{j,k}, \tau_{j,k}$ zur Verbesserung der Güte des Gesamtmodells. Dies kann nur in Abhängigkeit der Messdaten erfolgen, womit diese großen Einfluss auf das Ergebnis der Optimierung haben. Im Falle der ILMON-Modellierung mit seinen unabhängigen und nebeneinanderliegenden, linearen Teilmodellen ist offensichtlich, dass eine optimale Teilung im Bereich großer Anstiegsänderungen liegen sollte. Die Abbildung dieser Nichtlinearitäten des Prozesses in den Messdaten muss damit Ziel der Versuchsplanung zur Strukturoptimierung sein.

Neben diesem Ziel sollten von der Versuchsplanung die Voraussetzungen gewährleistet werden, welche für eine Aufteilung eines Teilmodells notwendig sind. Konkret erfordert dies eine ausreichend große Anzahl an Messpunkten in jedem Teilmodell, sodass sowohl die Teilung an sich durchgeführt werden kann, als auch eine hinreichend genaue Bewertung der Teilung über die Messdaten in diesem Bereich möglich ist.

Existiert für den zu modellierenden Prozess keine Abschätzung der Lage nichtlinearer und weitgehend linearer Bereiche, führt dies zur Notwendigkeit einer gleichmäßigen Abdeckung der Modellbereiche mit Messdaten. Erst mit der Beurteilung der lokalen Modellgüte auf Grundlage der in diesen Bereichen liegenden Datenpunkte ist es möglich, bestimmte Bereiche als linear oder quadratisch approximierbar zu deklarieren und diese, als versuchstechnisch ausreichend bestimmt, für die weitere Versuchsplanung nicht mehr zu berücksichtigen.

Damit ergeben sich für die Versuchsplanung zur Strukturoptimierung die gleichen Anforderungen und Aussagen, wie sie schon für die Parameteroptimierung formuliert wurden:

1. Die Versuchspunkte sollten innerhalb eines Teilmodells gleichverteilt, raumfüllend und unkorreliert sein.

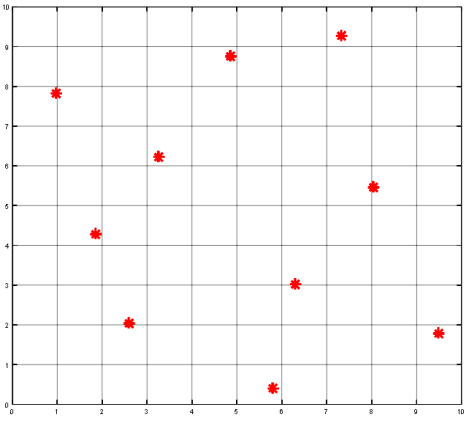

Abbildung 4.5.: Verteilung der Messdaten per Latin-Hypercube-Algorithmus

2. Die Anzahl der Versuchspunkte kann als Maß für die Zuverlässigkeit der Modellgüte betrachtet werden.

Bilden sich die Nichtlinearitäten des Prozesses über die Messdaten in der Modellgüte ab, so sorgt der ILMON-Algorithmus für eine Teilung in diesem Bereich. Werden weiterhin für jedes Teilmodell die gleiche Anzahl an Messdaten erhoben, ergibt sich bedingt durch die lokal-lineare Struktur der ILMON-Modellierung ein interessanter Effekt. In den nichtlinearen Bereichen findet eine Konzentration der Messpunkte statt, während die weitgehend linearen Bereiche, welche eine gute lokale Modellgüte aufweisen, mit einer geringeren Messpunktedichte vermessen werden. Anders ausgedrückt erfolgt lokal in allen Teilmodellen eine gleichverteilte und damit gleichrangige Messung. Global werden mit Steigerung der Modellkomplexität, die nichtlinearen Modellbereiche dichter vermessen als die annähernd linearen Gebiete. Abbildung 4.6 zeigt dieses Prinzip.

Die damit verbundene Reduktion des Messaufwandes in den linearen Bereichen stellt den Hauptvorteil der Verbindung von Versuchsplanung und iterativer Modellierung dar. Ohne Prozesskenntnisse werden die im jeweiligen Stadium der Modellbildung gewonnenen strukturellen Informationen zur optimalen Platzierung der nächsten Messpunkte genutzt. Wie oben erwähnt, stellt sich dieser Effekt bei gleicher Messpunkteanzahl pro Teilmodell ein, was somit als zusätzliche Anforderung zu den genannten Punkten formuliert wird.

Zusammengefasst ergeben sich somit für die optimale Parameterschätzung und Strukturoptimierung in der iterative Versuchsplanung folgende Anforderungen:

1. Alle Teilmodelle sollen mit der gleichen Anzahl an Versuchspunkten vermessen werden.

2. Die Versuchspunkte sollten innerhalb eines Teilmodells gleichverteilt, raumfüllend und unkorreliert sein.

3. Die Anzahl der Versuchspunkte kann als Maß für die Zuverlässigkeit der Modellgüte betrachtet werden. Je mehr Messpunkte in einem Teilmodell gleichverteilt vorhanden sind, um so sicherer spiegeln die Daten den tatsächlichen Verlauf der Ausgangsgröße wider.

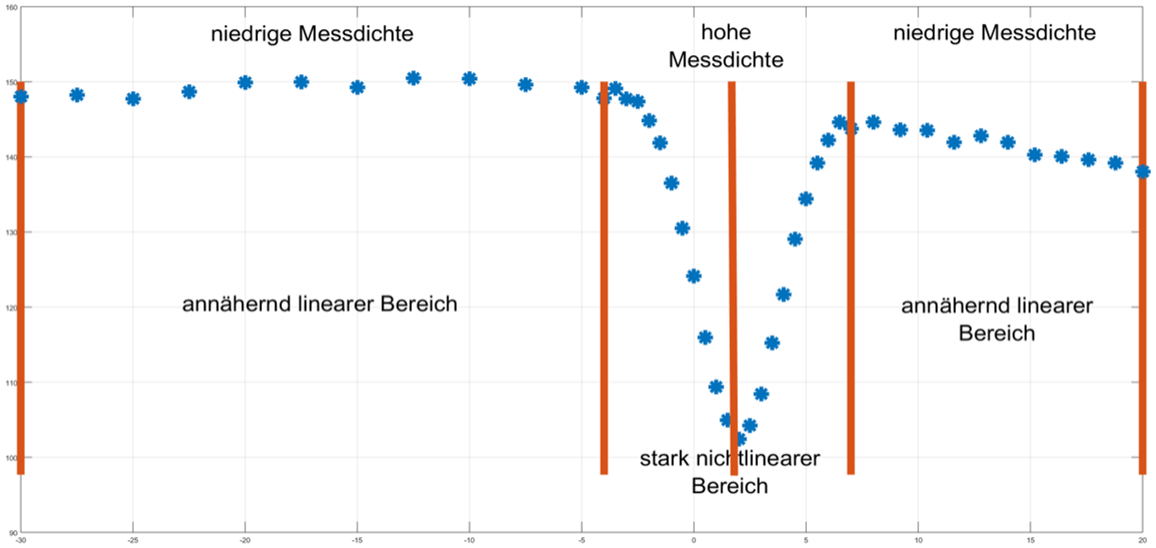

Abbildung 4.6.: Verteilung der Messdaten bei konstanten Anzahl von Messdaten pro Teilmodell. Die roten Linien stellen die Teilmodellgrenzen dar.

4.2.3. Bestimmung der Messpunkteanzahl pro Teilmodell

Mit den genannten Anforderungen ergibt sich die Bedingung zum Wechsel zwischen der Modellierung und der Versuchsplanung einzig aus der genügend großen Anzahl von Versuchspunkten in jedem Teilmodell. Was dabei als genügend groß bewertet wird, hängt von verschiedenen Faktoren ab und soll im Folgenden diskutiert werden.

Die minimale Versuchspunkteanzahl innerhalb eines Teilmodells wird durch die Anzahl der Parameter bestimmt, welche sich aus der effektiven Parameteranzahl ergibt, die über die Glättungsmatrix $\boldsymbol{S}_k$ nach Gleichung (3.42) bestimmt werden kann

$$\nu_{\mathrm{eff},k} = \mathrm{Spur}(\boldsymbol{S}^T \boldsymbol{S}). \tag{4.7}$$

Die Berechnung nach Gleichung (4.7) ist allerdings für die praktische Umsetzung sehr rechenintensiv. Als Alternative wird in [8] vorgeschlagen, stattdessen die effektive Messpunkteanzahl über die Summe der Basisfunktionswerte an den Messstellen zu berechnen:

$$n_{\mathrm{eff},k} = \sum_{i=1}^{N} \varphi_k(\boldsymbol{u}_i). \tag{4.8}$$

Diese sollte mindestens der Gesamtzahl der Parameter ν_k eines Teilmodells entsprechen, welche sich aus der Anzahl der Parameter der lokalen Funktion α_k und der Basisfunktion φ_k zusammensetzt:

$$\nu_k = \nu_{k,\alpha} + \nu_{k,\varphi} \leq n_{\mathrm{eff},k}. \tag{4.9}$$

Die Parameterzahl der Basisfunktion hängt nach Gleichung (3.16) ausschließlich von der Anzahl der Eingangsdimensionen q ab und ergibt sich mit

$$\nu_{k,\varphi} = 4q. \tag{4.10}$$

Wie viel Parameter die lokale Funktion α_k beinhaltet, ist neben der Anzahl der Eingangsdimensionen q auch von den verwendeten quadratischen Regressionstermen abhängig und liegt nach Gleichung (3.61) im Intervall

$$q + 1 \leq \nu_{k,\alpha} \leq \frac{q}{2}(q + 1). \tag{4.11}$$

Die effektive Messpunkteanzahl nach Gleichung (4.9) stellt das Minimum an Versuchspunkten dar, welche für eine robuste Schätzung der Parameter ausreicht. Das Splitten eines Teilmodells ist mit dieser Messpunkteanzahl allerdings nicht möglich und muss diesbezüglich mindestens doppelt so hoch sein, um in beiden neu entstehenden Teilmodellen die minimal notwendige Anzahl an Messpunkten zu realisieren. Bedingt durch die angestrebte Gleichverteilung wäre jedoch auch damit nur eine Teilung in der Mitte jedes Teilmodells möglich, was die Optimierung der Modellparameter stark limitiert und zu einer Erhöhung der Teilmodellanzahl führt. Um die Minimalanforderung einer asymmetrischen Teilung mit zwei Teilungsvarianten realisieren zu können, ist die dreifache Menge an Daten notwendig

$$3\nu_k \leq n_{\text{eff},k}, \tag{4.12}$$

was als Minimalforderung für die Anzahl der Versuchspunkte definiert wurde.

Je nach erwarteten maximalen Anstiegen der Ausgangsgröße des zu modellierenden Prozesses kann dieser Wert auch wesentlich größer gewählt werden, um in großen Teilmodellen eine sichere Erkennung von Nichtlinearitäten zu gewährleisten. Insbesondere in unbekannten Prozessen kann dies die Zuverlässigkeit der Modellgütebestimmung erhöhen und damit eine genauere Modellierung ermöglichen. Praktisch spielt natürlich auch der zeitliche und finanzielle Aufwand für eine Messung eine Rolle, sodass hier Werte zwischen $3\nu_k$ bis $10\nu_k$ einen guten Kompromiss darstellen.

4.3. Iterative Versuchsplanung mit ILMON

Nachdem im letzten Kapitel die grundlegenden Strategien für eine iterative Versuchsplanung besprochen wurden, soll in diesem Abschnitt auf die konkrete Umsetzung der kombinierten Versuchsplanung und Modellierung mit der ILMON-Struktur eingegangen werde. Beginnend mit der strukturellen Integration der Versuchsplanung in den iterativen Modellierungsprozess wird im zweiten Abschnitt auf die Konstruktion des Versuchsplans mit den vorgestellten Kriterien eingegangen.

4.3.1. Integration in den Modellierungsprozess

Bevor die strukturelle Verkopplung von Versuchsplanung und ILMON-Modellierung besprochen wird, soll zunächst der Ablauf der Versuchsplanung skizziert werden. Folgende Schritte sind dazu notwendig:

1. *Ermitteln der Messpunkteanzahl in jedem Teilmodell:* Die Bestimmung der Anzahl der Messpunkte, welche im Einflussbereich eines jeden Teilmodell liegen, erfolgt als effektive Messpunkteanzahl $n_{\text{eff},k}$ nach Gleichung (4.8).

2. *Konstruktion des Versuchsplans für jedes Teilmodell:* Die notwendige Anzahl neuer Versuchspunkte in jedem Teilmodell ergibt sich als Differenz aus der geforderten und

vorhandenen Anzahl. Da die Mindestanzahl der Messpunkte durch die Eingangsdimension des Modells und die Anzahl der Regressionsterme nach Gleichung (4.9) bestimmt ist, hat es sich als praktisch erwiesen, die geforderte Anzahl an Messpunkten pro Teilmodell $n_{k,m}$ nicht absolut zu definieren, sondern relativ über einen Faktor v_m aus der Mindestanzahl an Messpunkten pro Teilmodell zu berechnen.

$$n_{k,m} = v_m \nu_k - n_{\mathrm{eff},k}, \quad \text{mit} \quad v_m \geq 3 \tag{4.13}$$

Die Konstruktion des Versuchsplans erfolgt gesondert für jedes Teilmodell, worauf in Kapitel 4.3.2 ausführlich eingegangen wird. Die Versuchspläne aller Teilmodelle werden als gruppierter Versuchsplan zu einem Versuchsdurchgang zusammengefasst.

3. *Ermitteln der Messwerte*

4. *Korrektur des Versuchsplanung für nicht vermessbare Punkte:* Bei der Durchführung des Versuchsplanes kann es aus verschiedenen Gründen dazu kommen, dass einzelne geplanten Versuchspunkte nicht angefahren werden können, zum Beispiel weil Begrenzungen nicht korrekt berücksichtigt wurden. Diese Punkte müssen markiert und in der weiteren Versuchsplanung ausgeschlossen werden. Um die geforderte Anzahl an Messpunkten einzuhalten, muss eine korrigierte Version des Versuchsplans mit alternativen Messpunkten aufgestellt und vermessen werden. Dieser iterative Vorgang über die Punkte 2 und 3 wird durchgeführt, bis die geforderte Anzahl an Messpunkten in jedem Teilmodell aufgenommen wurde.

5. *Erweiterung der Datenbasis* Die Datenbasis wird um die Messwerte und die ausgeschlossenen Versuchspunkte erweitert.

Die in Abschnitt 4.2.2 definierten Primärziele, die Gleichverteilung der Messdaten sowie die Parität der Versuchspunkteanzahl in allen Teilmodellen garantieren die zuverlässige Bewertung der Modellgüte über den Modellfehler sowie die Strukturerweiterung des Modells durch Teilung eines Teilmodells. Als Konsequenz daraus, muss der skizzierte Ablauf der Versuchsplanung und -durchführung vor der Berechnung des Gütekriteriums im Modellierungsprozess integriert werden.

Mit der Erweiterung des Messdatensatzes in Punkt 5 sind die Parameter des zu diesem Zeitpunkts aktuellen Modells nicht mehr optimal. Die Anpassung an den erweiterten Datensatz erfolgt über einen Optimierungsdurchlauf ohne Erweiterung der Struktur, das heißt ohne Erhöhung der Teilmodellanzahl. Da sich durch die Optimierung der Basisfunktionsparameter auch die Lage der Teilmodellgrenzen verschieben kann und somit die Anzahl der in den Teilmodellen liegenden Versuchspunkte variiert, muss nach dieser Optimierung eine erneute Prüfung der Messpunkteanzahl stattfinden und gegebenenfalls ein erneuter Durchlauf der Versuchsplanung.

Aus dieser Überlegung resultiert die in Bild 4.7 dargestellte Struktur der Versuchsplanung und deren Einbindung in die iterative Modellierung eines ILMON-Modells. Kern dieser Struktur sind die beiden Iterationsschleifen zur Erweiterung der Modellstruktur sowie zur Aufnahme neuer Messpunkte, welche in Abhängigkeit von der Messpunkteanzahl in jedem Teilmodell durchlaufen werden.

1. *Initialisierung:* Der Algorithmus startet mit einem initialen Modell, das aus dem definierten Eingangsbereich und eventuellem Vorwissen über den zu modellierenden Pro-

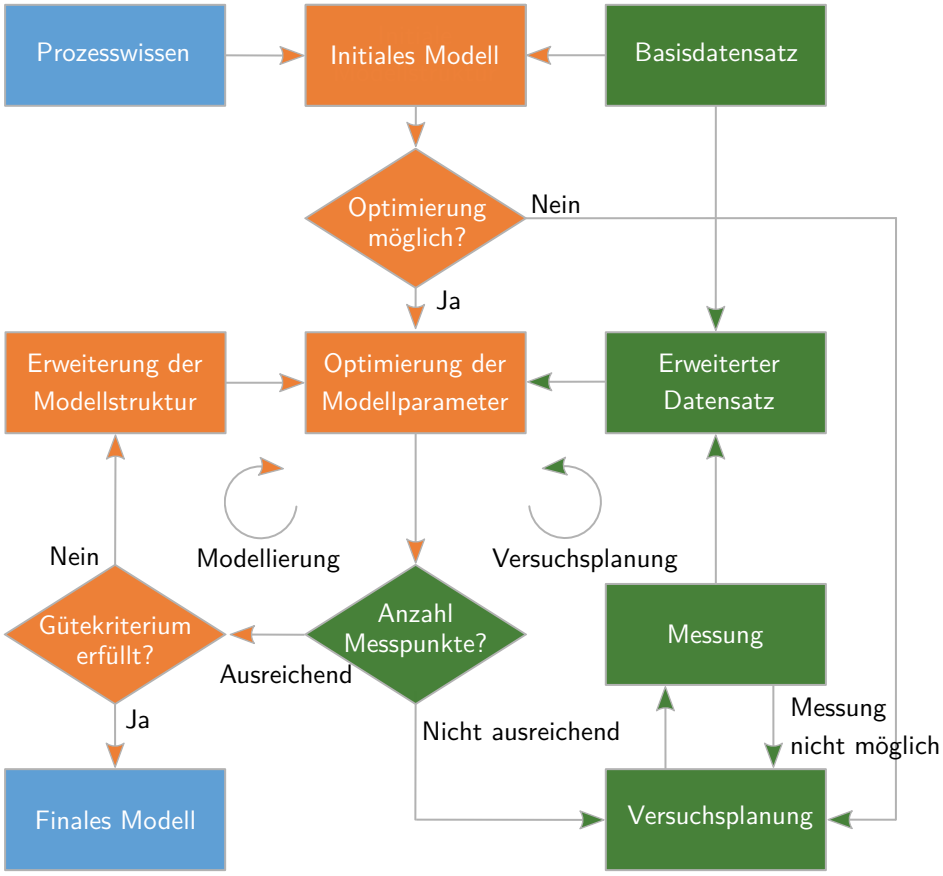

Abbildung 4.7.: Ablauf der gekoppelten, iterativen Versuchsplanung und Modellierung mit einer ILMON-Modellstruktur. Die Schritte der Messdatengewinnung sind grün dargestellt, Orange symbolisiert die Schritte des Modellierungsprozesses.

zess generiert wird. Die Parameter dieses Modells werden anhand eines Basisdatensatzes geschätzt, wobei diese Optimierung ohne eine Erhöhung der Teilmodellanzahl erfolgt. Der Basisdatensatz sollte weitgehend gleichverteilt und raumfüllend sein sowie das initiale Schätzen der Parameter erlauben. Ist dies nicht möglich bzw. sind keine oder nicht ausreichend Messdaten vorhanden, kann der Basisdatensatz auch erstellt oder vervollständigt werden, indem direkt zur Versuchsplanung gewechselt wird.

2. *Modellierung:* Mit dem Wechsel in die Optimierung der Modellparameter beginnt der Durchlauf der Iterationsschleifen. Ist in allen Teilmodellen die vorab definierte Anzahl an Messdaten vorhanden, erfolgt die Berechnung des Gütekriteriums und falls dieses nicht erfüllt wird, die Erweiterung der Modellstruktur durch Teilung eines Teilmodells. Mit der anschließenden Optimierung der Modellparameter ist die Iterationsschleife der Modellierung vollständig durchlaufen. Diese wird wiederholt bis entweder das Gütekriterium erfüllt wird oder sich, bedingt durch die Erhöhung der Teilmodellanzahl in jedem Durchlauf die Anzahl der Messdaten pro Teilmodell soweit verringert hat, dass der Wechsel in die Iterationsschleife der Versuchsplanung erfolgt.

3. *Versuchsplanung und -durchführung:* Ist auf Grund fehlender Messdaten in einem Teil-

modell die Voraussetzung für die Bewertung des Modells über die Modellgüte nicht mehr gegeben, erfolgt der Wechsel in die Iterationsschleife zur Ermittlung neuer Messwerte. Diese startet mit der Konstruktion der raumfüllenden und gleichverteilten Versuchspläne. Sind diese für jedes Teilmodell erstellt, erfolgt das Vermessen der Versuchspunkte. Nicht durchführbare Messungen werden markiert und nach Abschluss des Versuchsdurchgangs erfolgt eine erneute Planung unter Ausschluss der markierten Punkte. Ist die geforderte Anzahl an Messpunkten in allen Teilmodellen erreicht, werden neben den durchgeführten Messungen auch die als undurchführbar markierten Versuchspunkte im Messdatensatz gespeichert. Mit der Optimierung der Modellparameter wird die Erfassung neuer Messdaten abgeschlossen.

Der Wechsel zwischen Modellierung und Versuchsplanung wiederholt sich, bis das gewählte Gütekriterium erfüllt ist und das finale Modell ausgegeben wird.

4.3.2. Konstruktion des Versuchsplans

In Kapitel 4.2.1 wurde die Konstruktion von Versuchsplänen nach dem Maximin-Kriterium unter Anwendung der Mahalanobis-Distanz vorgestellt und deren Eigenschaften als vorteilhaft für die Anwendung in der kombinierten Modellierung mit einem ILMON-Modell erachtet. Die schlechten Projektionseigenschaften in die niederdimensionalen Unterräume wurden jedoch als nachteilig angemerkt. Im Rahmen dieser Arbeit wurde ein Konstruktionsalgorithmus entwickelt, welcher diese negative Eigenschaft vermeidet sowie die gewünschte raumfüllende und gleichverteilte Anordnung der Versuchspunkte weitgehend aufrecht erhält.

Ausgangspunkt der Konstruktion eines Versuchsplanes für ein ILMON-Teilmodell ist der sogenannte Vollfaktorplan, der alle möglichen Werte der Stellgrößen in allen Kombinationen abbildet. Dazu wird für jede kontinuierliche Stellgröße eine bestimmte Anzahl von Rasterpunkten definiert. Sinnvoll ist hier eine äquidistante Verteilung, um in der weiteren Konstruktion eine Gleichverteilung realisieren zu können. Zwingend für das Funktionieren des Algorithmus ist dies jedoch nicht. So können Hauptbetriebsbereiche auch feiner unterteilt werden als für den Regelbetrieb weniger relevante Bereiche. In vielen Fällen wird die Unterteilung auch durch die Gegebenheiten des Prozesses vorgegeben. Die Definition der Rasterpunkte erfolgt als Menge für jede Eingangsgröße u_i

$$R_i = \{r_{i,1}, r_{i,2}, \ldots, r_{i,N_{R_i}}\} \quad \text{mit} \quad i = \{1 \ldots q\} \tag{4.14}$$

und der Anzahl der Rasterpunkte N_{R_i} in der Eingangsdimension i. Die Anzahl der Versuche V in einem Vollfaktorplan berechnet sich dann durch

$$N_V = \prod_{i=1}^{q} R_i. \tag{4.15}$$

Bei einer hohen Eingangsdimension und feiner Rasterung der Stellgrößen ergibt sich in der Regel eine so hohe Anzahl an Faktorkombinationen, dass ein Vollfaktorplan praktisch nicht mehr erstellt werden kann. Zur Lösung dieses Problems wurde daher eine zweistufige Konstruktion realisiert, welche im ersten Schritt eine Reduktion des möglichen Versuchspunkte auf eine maximale Anzahl $N_{V,max}$ vornimmt und so die praktische Umsetzung ermöglicht.

Eingangsgrößenbezogene Filterung der Rasterpunkte: Im ersten Schritt der Versuchsplanung werden in jeder Eingangsgröße die Rasterpunkte bestimmt, die für den nächsten Messpunkt in Betracht kommen. Dabei wird neben einer raumfüllenden Gleichverteilung, welche bestehende Messpunkte berücksichtigt, auch die Vermeidung von wiederholten Messungen an einzelnen Rasterstellen angestrebt. Hierfür wird schrittweise vorgegangen:

1. Filterung der Menge der Rasterpunkte R_i, die innerhalb des Geltungsbereiches der Basisfunktion $\varphi(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d})$ des Teilmodells liegen:

$$R_{i,1} = \{r_{i,j} \in R_i | r_{j,i} < a_i \wedge r_{j,i} > d_i\}, \quad i = \{1 \dots q\}. \tag{4.16}$$

2. In jeder Eingangsgröße wird die Anzahl $\boldsymbol{N}_i = \{N_{i,1}, N_{i,2}, \dots, N_{i,R_i}\}$ der vorhandenen Messpunkte für jeden Rasterpunkt erfasst und alle Rasterpunkte $r_{i,j}$ mit der minimalen Anzahl an Messpunkten ausgewählt. Daraus resultiert die gefilterte Teilmenge an Rasterpunkten

$$R_{i,2} = \{r_{i,j} \in R_{i,1} | j = \arg\min_j N_{ij}\}, \quad i = \{1 \dots q\}. \tag{4.17}$$

   Dieses Vorgehen ist aus dem Latin-Hypercube-Sampling abgeleitet und verhindert eine Häufung von Messwerten in den einzelnen Eingangsgrößen, wodurch der Nachteil der schlechten Projektionseigenschaften des Maximin-Kriteriums ausgeglichen wird.

3. Ist die Anzahl der Versuchspunkte mit den gefilterten Rasterpunkten

$$N_{V,R_2} = \prod_{i=1}^{q} R_{i,2} \tag{4.18}$$

   größer als die maximal mögliche Menge $N_{V,max}$, erfolgt eine Reduktion der einzelnen Rastermengen $R_{i,2}$. Diese werden hierfür um den jeweils gleichen prozentualen Anteil reduziert, welcher aus dem Verhältnis $N_{V,R_2}/N_{V,max}$ bestimmt wird. Die berechnete Anzahl der Rasterpunkte wird auf den nächsten ganzzahligen Wert abgerundet. Damit berechnet sich die reduzierte Anzahl der Rasterpunkte in den einzelnen Eingangsdimensionen durch

$$N_{R_i,max} = \left\lfloor \sqrt[q]{\frac{N_{V,R_2}}{N_{V,max}}} N_{V,R_i} \right\rfloor, \quad i = \{1 \dots q\}. \tag{4.19}$$

   Die zu entfernenden Rasterpunkte werden aus jeder Rastermenge $R_{i,2}$ zufällig ausgewählt, womit sich die Teilmengen

$$R_{i,3} \subseteq \mathcal{U}(R_{i,2}) \quad \text{mit} \quad |R_{i,3}| = N_{R_{i,max}}, \quad i = \{1 \dots q\} \tag{4.20}$$

   ergeben.

Aus den reduzierten Rasterpunkten $R_{i,3}$ wird ein Vollfaktorplan erstellt, aus dem im zweiten Schritt der nächste optimale Versuchspunkt ermittelt wird.

Ermittlung des optimalen Versuchspunktes: Nach der Filterung der einzelnen Rasterpunkte der Eingangsgrößen muss der resultierende Vollfaktorplan V_f auf nicht anfahrbare

oder gesperrte Bereiche innerhalb des Eingangsraumes untersucht werden, um diese für die Bestimmung der nächsten Versuchspunkte auszuschließen. Danach erfolgt die eigentliche Berechnung der optimalen Versuchspunkte. Es wird wiederum schrittweise vorgegangen:

1. Gesperrte Bereiche des Versuchsraums können über eine beliebige Anzahl von Bedingungen $B_i(\boldsymbol{u})$ in Abhängigkeit von den Eingangsgrößen definiert werden. Dabei ist auch eine Überlagerung der Geltungsbereiche mehrerer Bedingungen möglich, sodass komplex geformte Gebiete aus einfacheren Bedingungen zusammengesetzt werden können. Alle Bedingungen werden nacheinander entsprechend ihrer Reihenfolge abgearbeitet und alle Messpunkte, für die eine Bedingung zutrifft, werden aus der Versuchsplanung ausgeschlossen. Damit ergibt sich die Menge der zugelassenen Versuchspunkte zu

$$V_z = \{v \in V_f | v \not\rightarrow B_1 \vee B_2 \vee \ldots\}. \tag{4.21}$$

2. Aus der Menge der Versuchspunkte V_z wird über das Maximin-Kriterium mit Mahalanobis-Distanz der nächste Versuchspunkt ausgewählt

$$V_{next} = \{v(\boldsymbol{x}) \in V_z | v(\boldsymbol{x}) = \max \min_{i \neq j} \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \operatorname{Cov}(\boldsymbol{X}_k)^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j)}\}. \tag{4.22}$$

   In der Designmatrix $\boldsymbol{X}_k$ sind alle Versuchspunkte enthalten, die innerhalb des jeweiligen Geltungsbereiches des Teilmodells k liegen. Da in der Regel pro Durchlauf und Teilmodell mehrere Versuchspunkte vermessen werden und die ermittelten Punkte Einfluss auf die Wahl der nachfolgenden Messpunkte haben, erfolgt die Bestimmung nach Gleichung (4.22) iterativ. Nach der Berechnung eines Versuchspunktes wird dieser der Designmatrix $\boldsymbol{X}_k$ hinzugefügt. Diese beiden Schritte werden solange wiederholt, bis die gewünschte Anzahl an Versuchspunkten erreicht ist.

Die Konstruktion der Versuchspläne aller Teilmodelle erfolgt unabhängig und sequentiell, nach der in der ILMON-Struktur gespeicherten Reihenfolge. Einzelne Versuchspunkte in den Überlagerungsbereichen sind dabei für mehrere Teilmodelle gültig. Jedoch sind diese nur für das Teilmodell optimal nach Gleichung (4.22), in dem die Bestimmung stattfand. Da die weitere Berechnung der Versuchspunkte die schon vorhandenen Messpunkte berücksichtigt, wird die angestrebte raumfüllende Gleichverteilung dennoch realisiert.

4.4. Zusammenfassung

In diesem Kapitel wurde die Grundlagen der statistischen Versuchsplanung umrissen und die Möglichkeit der Einbindung einer Versuchsplanung in die iterative Modellierung eines ILMON-Modells diskutiert. Es wurde der Vorteil der sequentiellen Versuchsplanung herausgearbeitet, das mit den vorhergehenden Messungen wachsende Prozesswissen für die Ermittlung weiterer Versuchspunkte zu nutzen.

Weiterhin wurden die verschiedenen etablierten Verfahren zur Versuchsplanung vorgestellt sowie deren Eigenschaften in Hinblick auf eine Nutzung mit ILMON-Modellen und der Einbindung in den iterativen Modellierungsprozess untersucht. Die Verteilung der Messpunkte nach dem Maximin-Kriterium mit Mahalanobis-Distanz zeigte dabei die besten Eigenschaften und wurde als grundlegende Methodik umgesetzt. Die Nachteile der schlechten Projek-

tionseigenschaften konnten in der praktischen Umsetzung durch eine geeignete Vorauswahl der Messpunkte vermieden werden.

Mit dem in dieser Arbeit entworfenen Algorithmus erfolgt eine optimierte Auswahl der Versuchspunkte sowohl für die Schätzung der linearen Parameter der Teilmodelle als auch für die Basisfunktionsparameter eines ILMON-Modells. Zugleich unterstützt die Versuchsplanung die iterative Erhöhung der Modellkomplexität in dieser Struktur. Weiterhin ermöglicht die so erzeugte Datenbasis eine sichere Bewertung der Modellgüte über die schon zur Modellierung verwendeten Datenpunkte. Damit entfällt die Notwendigkeit eines Validierungsdatensatzes und die Gesamtzahl der Versuchspunkte sowie der damit verbundene Messaufwand kann reduziert werden. Auf Grund der Eigenschaften der ILMON-Modellierung und der über die Versuchsplanung realisierten gleichen Anzahl von Versuchspunkten pro Teilmodell ergibt sich eine automatische Erhöhung der Messdichte in stark nichtlinearen Bereichen des Modells. Die notwendigen Informationen ergeben sich im Laufe der sequentiellen Versuchsplanung durch die stetige Annäherung des Modells an den Prozess.

Realisierungsprobleme auf Grund der hohen theoretischen Messpunkteanzahl in hochdimensionalen Prozessen wurden über eine sinnvolle Reduktion der Rasterpunkte in den einzelnen Eingangsdimensionen beseitigt. Praktische Anforderungen wie unterschiedliche, nichtäquidistante Rasterungen der Eingangsgrößen oder komplexe Ausschlüsse von Gebieten im Versuchsraum sind in die Versuchsplanung integriert.

Zusammengefasst weist das Verfahren folgende Eigenschaften auf:

- Rasterpunkte in den einzelnen Stellgrößen sind frei wählbar. Der aus den Rasterpunkten resultierende Vollfaktorplan kann bei Bedarf auf eine Größe reduziert werden, die den Ressourcen des Versuchsequipment entspricht.
- Begrenzungen und Ausschlüsse im Versuchsraum können als von den Eingangsgrößen abhängige Bedingungen frei definiert werden.
- Die Versuchsplanung und -durchführung erfolgt als gruppierter sequentieller Versuchsplan und wird für jedes Teilmodell unabhängig durchgeführt
- Versuchspunkte, die nach den definierten Bedingungen erlaubt sind, jedoch in der Versuchsdurchführung nicht angefahren werden können, werden online als gesperrte Punkte markiert und in der weiteren Versuchsplanung berücksichtigt.
- Die Versuchspunkte werden gleichverteilt und raumfüllend über dem Eingangsraum der Teilmodelle platziert. In der Projektion auf die einzelnen Stellgrößen erfolgt die Auswahl gleichverteilt auf alle Rasterpunkte innerhalb eines Teilmodells.
- Bereiche mit starken Nichtlinearitäten werden automatisch mit einer größeren Messpunktedichte vermessen als weitgehend lineare Bereiche. Für diesen Mechanismus sind keine Prozesskenntnisse notwendig.

In der Summe wurde mit der im Rahmen dieser Arbeit entwickelten iterativen Versuchsplanung und die Einbindung in den Modellierungsalgorithmus ein Instrument zur effektiven Auswahl der optimalen Versuchspunkte geschaffen, das den praktischen Anforderungen aus den üblichen Abläufen von Messkampangen in der Motorenentwicklung gerecht wird.

5. Ergebnisse und Realisierungsaspekte

In diesem Kapitel soll die Anwendung der vorgestellten Verfahren an verschiedenen Beispielen demonstriert und die Ergebnisse diskutiert werden. In Abschnitt 5.1 wird dazu die ILMON-Modellierung an einer synthetischen Beispielfunktion mit den verschiedenen Parametrierungsmöglichkeiten untersucht und mit den etablierten Verfahren der LOLIMOT- sowie der GMR-Modellierung verglichen. Neben der Modellierung mit gleichverteilten Datensätzen kommt die iterative Versuchsplanung zur Messdatenbestimmung zum Einsatz. In Abschnitt 5.2 wird eine ILMON-Modellierung mit iterativer Versuchsplanung am Beispiel der Füllungserfassung eines 3,2l-Saugmotor mit variablem Ventiltrieb vorgestellt.

5.1. Ergebnisse an synthetischen Testfunktionen

Für die Untersuchung und anschauliche Darstellung der Eigenschaften eines ILMON-Modells wurde eine Testfunktion definiert, welche typische Eigenschaften der in der Motorentwicklung auftretenden funktionalen Zusammenhänge beinhaltet und geeignet ist, die Vor- und Nachteile der ILMON-Modellierung inklusive der Versuchsplanung aufzuzeigen. Dafür wurden folgende Eigenschaften angestrebt:

- Es sollen mehrere Bereiche mit geringen Anstiegsänderungen vorhanden sein, die voneinander scharf abgegrenzt sind. Solche Verläufe treten durch Umschaltvorgänge und Aktivierungen bestimmter Komponenten auf, wie z.B. die Aktivierung des Turboladers ab einer bestimmten Drehzahl.
- Die Testfunktion soll Bereiche mit starken Nichtlinearitäten als auch weitläufige Anstiegsänderungen aufweisen, welche sich mit linearen Teilmodellen nur aufwendig approximieren lassen.
- Für eine mögliche visuelle Darstellung als 3D-Grafik wurde die Testfunktion auf zwei Eingangsgrößen beschränkt.

Aus diesen Vorgaben wurde folgende Testfunktion erstellt

$$y = 100\left(1 - \frac{1}{u_2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{u_1-1}{u_2}\right)^2}\right) - \frac{(u_1+15)^2}{10000} - \frac{u_1 u_2}{5} + 20|u_2 - 1.3| + 50, \qquad (5.1)$$

welche im Wertebereich $-30 \leq u_1 \leq 20$ und $0.9 \leq u_2 \leq 1.5$ den in Abbildung 5.1 dargestellten Verlauf aufweist.

Die Funktion nach Gleichung (5.1) setzte sich aus einer gespiegelten und über die zweite Eingangsgröße verzerrten Gaußfunktion als Nichtlinearität und diversen quadratischen und linearen Termen zusammen, die sowohl leicht gewölbte Bereiche als auch scharfe Übergänge zwischen einzelnen Gebieten ergeben und damit verschiedenen in der Praxis auftretende Verläufe aufweist.

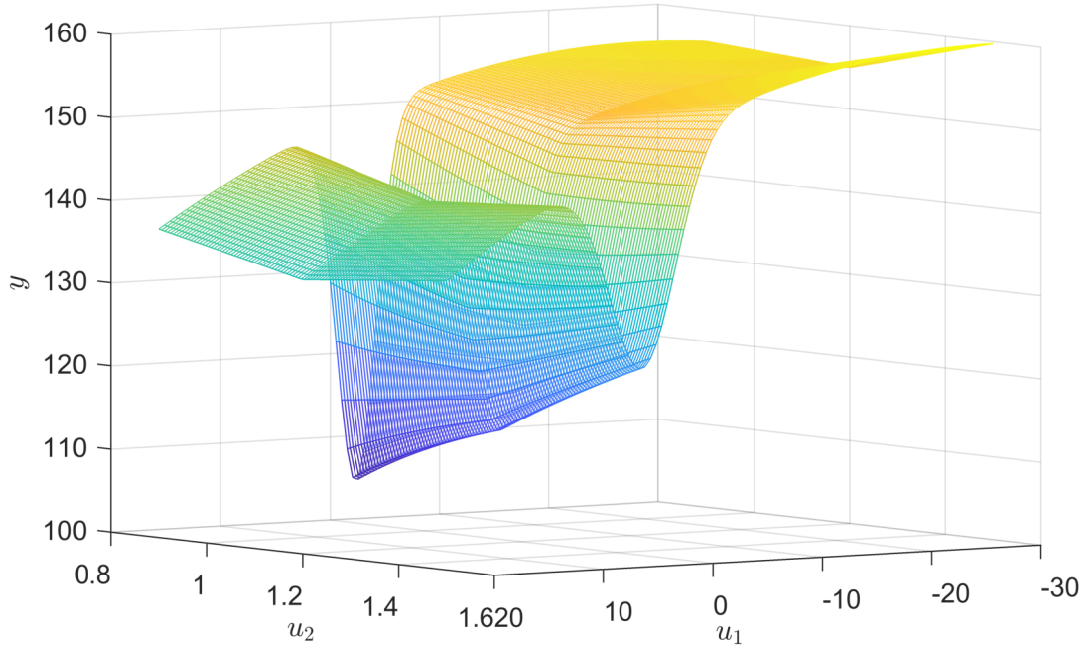

Abbildung 5.1.: Verlauf der Testfunktion nach Gleichung (5.1)

Zur Validierung der Modellierungen wurde ein Referenzdatensatz aus $N = 10.000$ über dem Eingangsraum gleichverteilten Messpunkten erzeugt, der in den folgenden Auswertungen als Basis für die Berechnung des Modellfehlers dient. Für die Generierung der Messdaten wurde die Funktion mit einem normalverteilten, mittelwertfreien Rauschen mit einer Standardabweichung von $\sigma = 0,3$ beaufschlagt. Damit stellt dieser Wert auch die untere Schranke des als Wurzel der mittleren Fehlerquadratsumme (Root mean square error, RMSE) definierten Modellfehlers dar:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \geq \sigma. \tag{5.2}$$

5.1.1. ILMON-Modellierung

Für die Demonstration der ILMON-Modellierung anhand der Testfunktion wurden 6 Trainingsdatensätze mit über dem Eingangsraum gleichverteilten Messpunkten generiert. Die Anzahl der Messpunkte variiert in den Datensätzen von $N = 100$ bis $N = 1000$ Datenpunkten. Als Modellfehler wurde die Wurzel des durchschnittlichen, quadratischen Fehlers nach Gleichung (5.2) mit dem Validierungsdatensatz berechnet, im Folgenden wird dieser Fehler als Validierungsfehler ϵ_{val} bezeichnet.

Im ILMON-Algorithmus wurde als Abbruchbedingung ein Modellfehler von $RMSE = 0,4$ vorgegeben. Die Berechnung erfolgt während der Modellierung über dem jeweiligen Trainingsdatensatz und wird im Folgenden als Trainingsfehler ϵ_{train} bezeichnet. In Abbildung 5.2 ist der Validierungs- und Trainingsfehler in Abhängigkeit der Teilmodellanzahl im Verlauf der Modellierung dargestellt.

Wie aus dieser Darstellung ersichtlich, erfolgt bei sehr kleinen Trainingsdatensätzen eine starke Überanpassung des Modells an die Trainingsdaten. Der Modellierungsalgorithmus

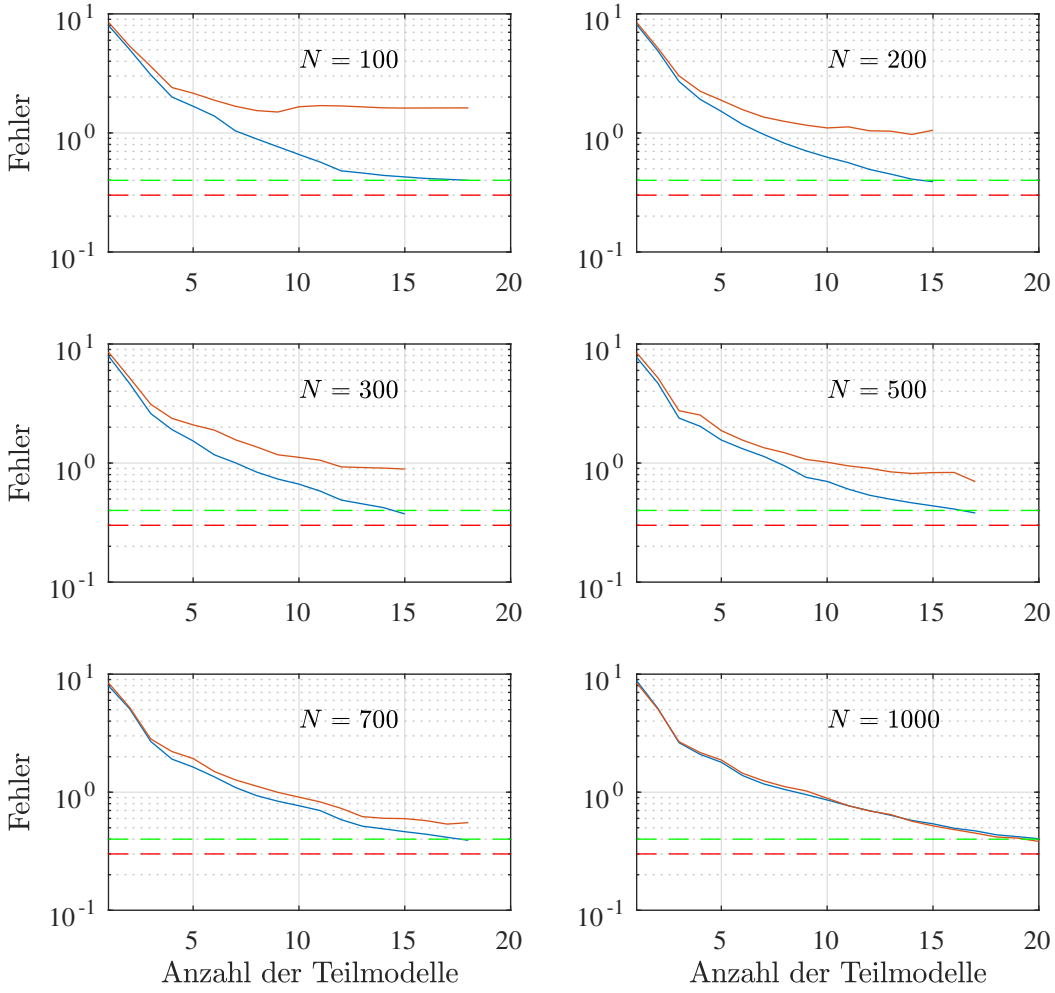

Abbildung 5.2.: Verlauf des Trainings- (blau) und Validierungsfehlers (rot) einer ILMON-Modellierung für verschieden große Trainingsdatensätze. Als Abbruchkriterium wurde $RMSE = 0.4$ (grün-gestrichelte Linie) definiert. Die Standardabweichung des Messrauschens $\sigma = 0,3$ ist als rot-gestrichelte Linie eingezeichnet. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

wird zwar bis zum Erreichen des gewünschten Trainingsfehler fortgeführt, der Validierungsfehler stagniert jedoch ab einer bestimmten Teilmodellanzahl. Erst ab einem Datensatzumfang von ca. 1000 Datenpunkten verlaufen beide Fehler parallel und der Trainingsfehler entspricht der wahren Modellgüte. Der im Vergleich zur LOLIMOT-Modellierung effektivere Partitionierungsalgorithmus und die flexibleren Basisfunktionen reduzieren die Anzahl der Teilmodelle erheblich. Mit den gleichen Modellfehlervorgaben werden bei der ILMON-Modellierung nur ca. 60 % der Teilmodellanzahl benötigt, vergleiche Abbildung 5.7.

Die auf die Minimierung des Trainingsfehlers abzielende iterative Strukturoptimierung garantiert mit jeder Teilung einen sinkenden Trainingsfehler. Eine Überanpassung des Modells

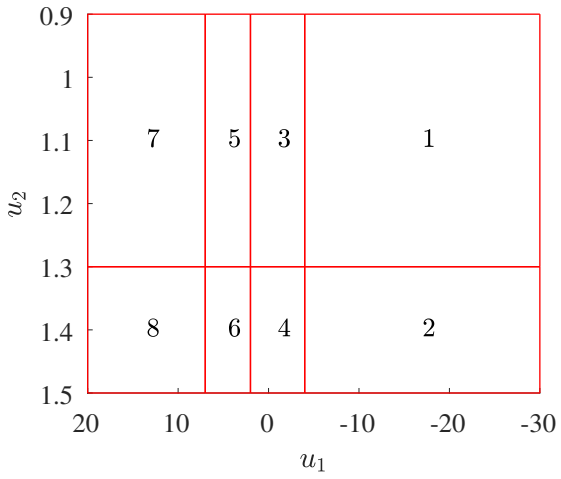

Abbildung 5.3.: Partitionierung des initialen Modells per Prozesswissen mit Teilungen an den Stellen $u_1 = -4$, $u_1 = 2$, $u_1 = 7$ und $u_2 = 1,3$

an die Trainingsdaten wird nur durch die für die Parameterschätzung notwendige, minimale Anzahl an Messpunkten pro Teilmodell beschränkt. Können keine Teilungsparameter gefunden werden, die den aktuellen Trainingsfehler verbessern, wurde in der Strukturoptimierung nur ein lokales Optimum gefunden. In diesem Fall, wie auch bei Unterschreitung der minimalen Messpunkte in einem Teilmodell, erfolgt ein Abbruch des Algorithmus'. Eine Erhöhung der Startparameteranzahl für die Optimierung (siehe Kapitel 3.5) und die Aufnahme weiterer Messpunkte in den betroffenen Teilmodellen kann dies vermeiden.

5.1.2. Integration von Prozesswissen

Wie in Kapitel 3.5 ausgeführt, kann bei vorhandenem Prozesswissen eine Aufteilung des initialen Modells in verschiedene Teilmodelle erfolgen. Die Teilungsgrenzen sollten dabei entlang starker, achsenorthogonaler Gradientenänderungen liegen, wobei die Teilungsparameter nicht exakt bekannt sein müssen. Da diese im ersten Optimierungsdurchlauf entsprechend der Trainingsdaten angepasst werden, sind leichte Abweichungen in der Regel nicht relevant. Mit der Aufteilung des initialen Modells in die wesentlichen Teilbereiche ist oft eine Approximation mit weniger Teilmodellen möglich. Bei der hier untersuchten Testfunktion wurden Teilungen entlang der stärksten Gradientenänderungen an den Stellen $u_1 = -4$, $u_1 = 2$, $u_1 = 7$ und $u_2 = 1,3$ durchgeführt, was zu einem initialen Modell mit 8 Teilmodellen führte, siehe Abbildung 5.3.

Die Optimierung wurde mit dem oben verwendeten Trainingsdatensatz aus 1000 Messpunkten durchgeführt. Der erste Optimierungsdurchlauf erfolgte ohne eine Erhöhung der Teilmodellanzahl. In diesem wurden die Basisfunktionsparameter der im initialen Modell vorgegebenen Teilungen anhand der Messdaten neu geschätzt. Die resultierenden Teilmodellgrenzen sind in Abbildung 5.4 dargestellt. Der Validierungsfehler ergibt sich mit dieser Optimierung zu $\epsilon_{val} = 1,08$. In der weiterführenden Strukturoptimierung mit einer Gütevorgabe von $RMSE = 0.4$ konnte die Anzahl der Teilmodelle von 21 ohne Vorpartitionierung auf 18 reduziert werden.

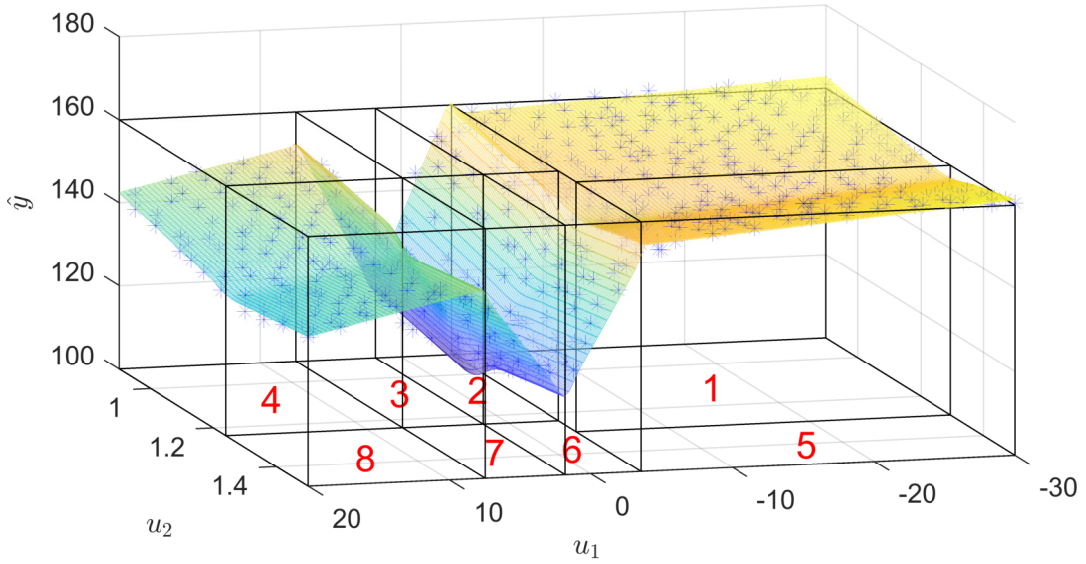

Abbildung 5.4.: ILMON-Modell nach Vorpartitionierung und erster Optimierung ohne Teilung über einen Trainingsdatensatz mit $N = 1000$ Messpunkten. Der Validierungsfehler beträgt nach dieser Optimierung $\epsilon_{val} = 1,08$

.

5.1.3. Interpretierbarkeit der Modellstruktur

Einer der Vorteile der ILMON-Struktur ist die Interpretierbarkeit des optimierten Modells aufgrund einer gut realisierbaren unabhängigen Validierung der einzelnen Teilmodelle. Beispielhaft sind im folgenden 3 Teilmodelle der oben durchgeführten Optimierung über den Trainingsdatensatz von $N = 700$ aufgeführt. Diese können über die Gültigkeitsbereiche ihrer Basisfunktionen in den einzelnen Eingangsgrößen sehr einfach spezifiziert werden:

$$\begin{array}{lrcrcrcr} \text{Teilmodell 3:} & 0,81 & \leq u_1 \leq & 2,81 & \text{und} & 0,9 & \leq u_2 \leq & 1,5 \\ \text{Teilmodell 5:} & -30 & \leq u_1 \leq & -19,93 & \text{und} & 1,29 & \leq u_2 \leq & 1,5 \\ \text{Teilmodell 6:} & 3,2 & \leq u_1 \leq & 6,07 & \text{und} & 0.9 & \leq u_2 \leq & 1.33 \end{array}$$

In jedem dieser drei Teilmodelle beschreiben die zugehörigen linearen Funktionen den Modellverlauf innerhalb des Gültigkeitsbereiches. Deren Parameter können direkt aus dem optimierten Modell ausgelesen werden und lauten in diesem Beispiel:

$$\begin{array}{ll} \text{Teilmodell 3:} & y_3 = 109,31 - 4,97u_1 + 13,10u_2 \\ \text{Teilmodell 5:} & y_5 = 120,59 - 0.09u_1 + 25,03u_2 \\ \text{Teilmodell 6:} & y_6 = 116,89 + 8,03u_1 - 20,57u_2 \end{array}$$

Die Plausibilität dieser Funktionen können über die linearen Koeffizienten in jeder Eingangsdimension sehr einfach überprüft werden.

Weiterhin ist auch eine grafische Auswertung möglich. Als Beispiel ist die lineare Funktion des Teilmodell 3 in Abbildung 5.5 zusammen mit den relevanten Trainingsdaten dargestellt. Für höherdimensionale Prozesse kann für eine grafische Validierung das im Rahmen dieser Arbeit entwickelte und im Kapitel 2.2.3 vorgestellte Tool genutzt werden.

Eine weitere Möglichkeit der Validierung besteht in der Auswertung der Fehlerverteilungen

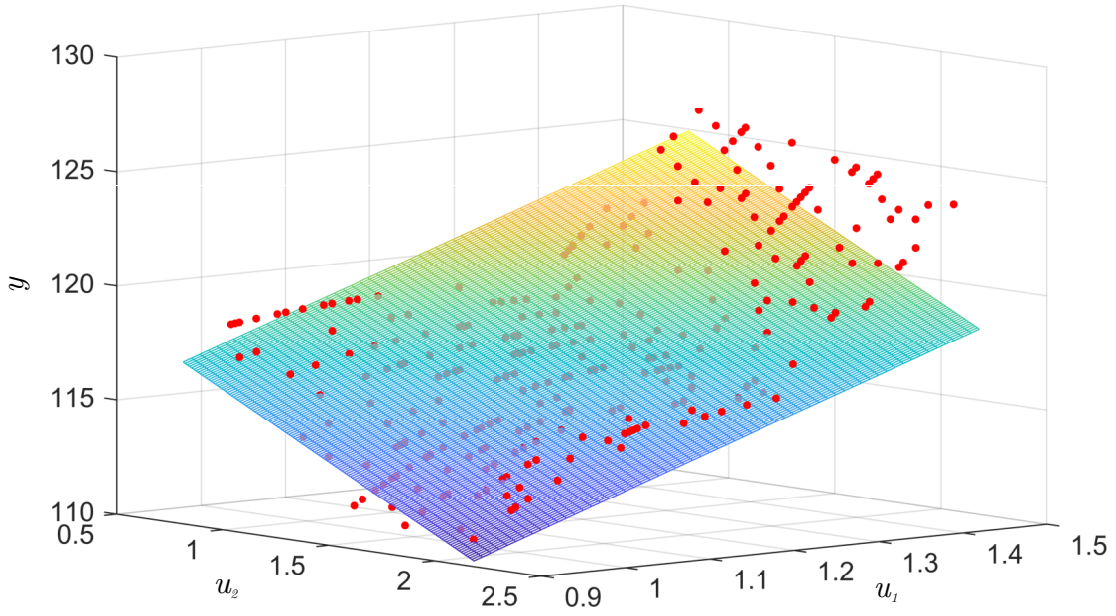

Abbildung 5.5.: Lineare Komponente des Teilmodells 3 mit Trainingsdaten.

in den separierten Teilmodellen. In Abbildung 5.6 sind die Histogramme des Modellfehlers für die aufgeführten Teilmodelle dargestellt. Eine schmale, symmetrische Verteilung wie in Teilmodell 5 weist dabei tendenziell auf eine gute Approximation des Modells im jeweiligen Teilbereich hin. Teilmodelle mit einer breiten oder unsymmetrischen Verteilung bedürfen einer genaueren Betrachtung. Pauschale Aussagen zur Qualität der Modellierung auf Grund der Fehlerverteilung sind allerdings nicht möglich, da in der Regel die Anzahl der Messpunkte in den einzelnen Teilmodellen für fundierte statistische Aussagen zu gering und die Art der Verteilungsdichte der Messfehler sowie deren Parameter in realen Prozessen nicht bekannt ist. Trotzdem kann diese Auswertung Hinweise zur Überprüfung bestimmter Modellbereiche liefern.

5.1.4. Vergleich mit LOLIMOT-Modell

Zur Einschätzung der Leistungsfähigkeit der ILMON-Modellierung wurde die Testfunktion zum Vergleich mit Hilfe des LOLIMONT-Algorithmus sowie der Gaussian-Mixture-Regression approximiert. Die Optimierungen wurden identisch zum obigen Vorgehen an den Trainingsdatensätzen mit verschiedenen Messdatenanzahlen vorgenommen und der Trainings- und Validierungsfehler für die jeweilige Teilmodell- bzw. Komponentenanzahl berechnet.

Der Designparameter s_Σ des LOLIMOT-Modells wurde für diesen Vergleich empirisch auf $s_\Sigma = 0,22$ festgelegt, womit die niedrigste Teilmodellanzahl in den Optimierungsdurchläufen realisiert werden konnte. Als Algorithmus wurde die in Abschnitt 2.4.8 vorgestellte Variante mit lokaler Parameterschätzung verwendet. Das Abbruchkriterium wurde identisch zur ILMON-Optimierung mit einem Trainingsfehler von $\epsilon_{train} = 0.4$ vorgegeben. Weiterhin wird

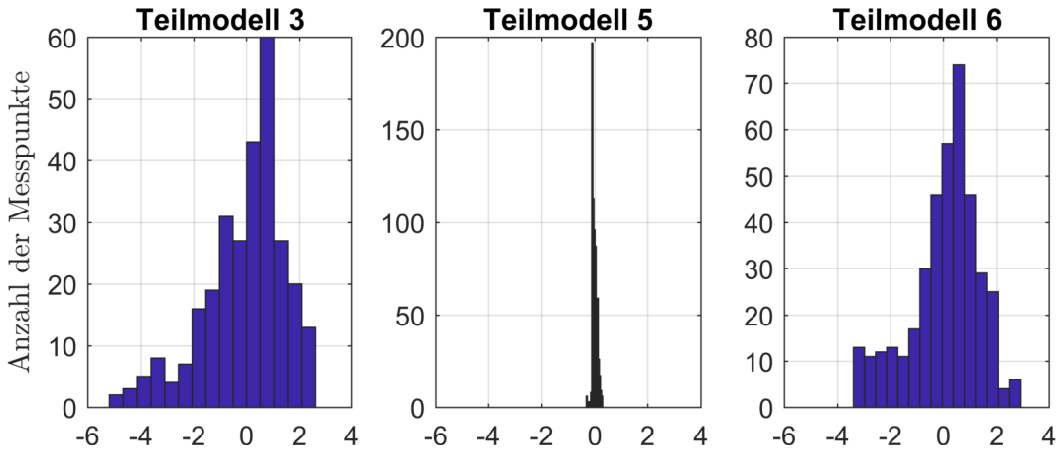

Abbildung 5.6.: Fehlerverteilung in ausgewählten Teilmodellen der ILOMN-Modellierung über $N = 700$ Messpunkte.

der Algorithmus abgebrochen sobald auf Grund zu weniger Messdaten in einem Teilmodell keine weiter Strukturoptimierung mehr vorgenommen werden kann. Dies war bei den Durchläufen mit $N = 100$ und $N = 200$ Datenpunkten der Fall. Der Verlauf des Trainings- und Validierungsfehlers über die Teilmodellanzahl eines LOLIMOT-Modells ist in Abbildung 5.7 dargestellt.

Im Vergleich kann die ILMON-Modellierung (siehe Abbildung 5.2) unabhängig von der verwendeten Messpunkteanzahl die gleichen Gütevorgaben mit weniger Teilmodellen erreichen. Die optimale Strukturoptimierung gestattet gegenüber dem festen heuristischen Partitionierungsalgorithmus der LOLIMOT-Modellierung eine bessere Approximation mit der gleichen Anzahl von Teilmodellen. Weiterhin kann durch die höhere Flexibilität der Basisfunktionen eine bessere Anpassung an die Messdaten stattfinden.

Ein Nachteil des ILMON-Optimierungsalgorithmus ist der deutlich höhere Berechnungsaufwand. Mit den über die Datensätze $N = 1000$ optimierten Modellen ergibt sich ein Verhältnis der Rechenzeit für die Optimierung von 250:1 zwischen der ILMON- und der LOLIMOT-Modellierung. Dagegen sinkt der Aufwand zur Berechnung der Ausgangsgleichung im ILMON-Modell deutlich und erreicht auf einem Desktop-PC mit FPU ein Verhältnis von 1:15 zwischen der ILMON- und der LOLIMOT-Modellierung. Auf Steuergeräten, die typischerweise ohne spezialisierte Gleitkommaeinheit (FPU) arbeiten, werden die in LOLIMOT verwendeten Exponentialfunktionen nochmals deutlich langsamer berechnet: je nach Prozessor kann man hier von einem Faktor von rund 100 bei einer Berechnung mit doppelter Genauigkeit (64 Bit) und rund 50 bei einfacher Genauigkeit (32 Bit) ausgehen. Damit ergibt sich eine ca. 750 bis 1500 mal schnellere Berechnung der ILMON-Ausgangsgleichung gegenüber LOLIMOT auf einem Motorsteuergerät, bei den in diesem Vergleich verwendeten Modellen.

5.1.5. Vergleich mit GMR-Modell

Die Gaussian-Mixture-Regression ist, wie in Abschnitt 2.4.7 beschrieben, kein iteratives Verfahren. Die Anzahl K der hier als Komponenten bezeichneten Teilmodelle muss zum Start des Algorithmus vorgegeben werden. Ebenso müssen für jede Komponente k die Wichtung

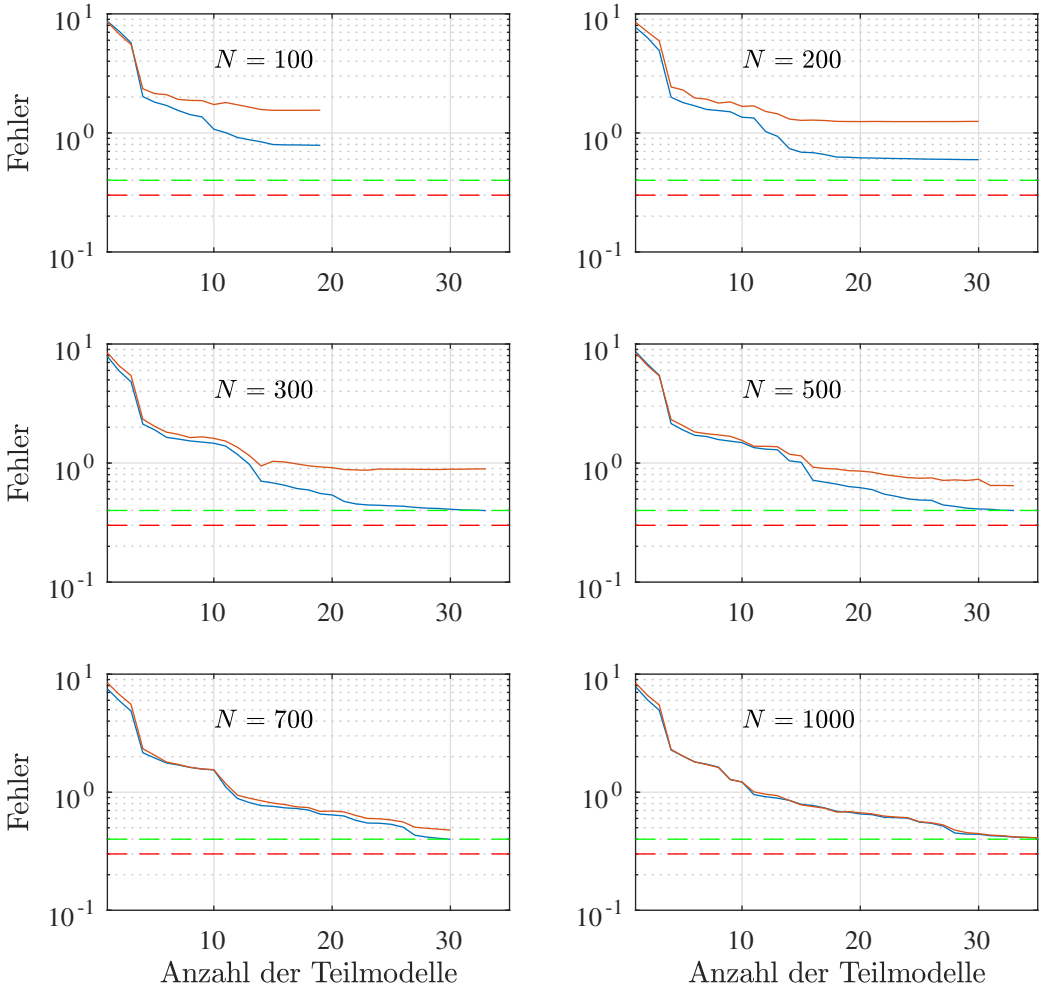

Abbildung 5.7.: Verlauf des Trainings- (blau) und Validierungsfehlers (rot) einer LOLIMOT-Modellierung für verschieden große Trainingsdatensätze. Als Abbruchkriterium wurde $RMSE = 0.4$ (grün-gestrichelte Linie) definiert. Die Standardabweichung des Messrauschens $\sigma = 0,3$ ist als rot-gestrichelte Linie eingezeichnet. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

w_k, die Mittelwertvektoren $\boldsymbol{\mu}_k$ und die Kovarianzmatrix $\boldsymbol{\Sigma}_k$ der Normalverteilung als Startparameter definiert werden.

Je nach Startparameter, Anzahl der Komponenten und Komplexität des Prozesses wird in der nichtlinearen Optimierung häufig nur ein lokales Optimum der Parameter gefunden. Um diese Gefahr zu verringern, wurden mit jeder Komponentenanzahl 400 Durchläufe mit unterschiedlichen Startparametern ausgeführt. Aus diesen Durchläufen wurde das Modell mit dem geringsten Trainingsfehler ausgewählt. Die Startparameter der Wichtungen, der Mittelwertvektoren und die Kovarianzmatrizen aller K Komponenten wurden zufällig gewählt und per „k-mean++“-Algorithmus [90] optimiert. Die Messdatensätze sind vor der

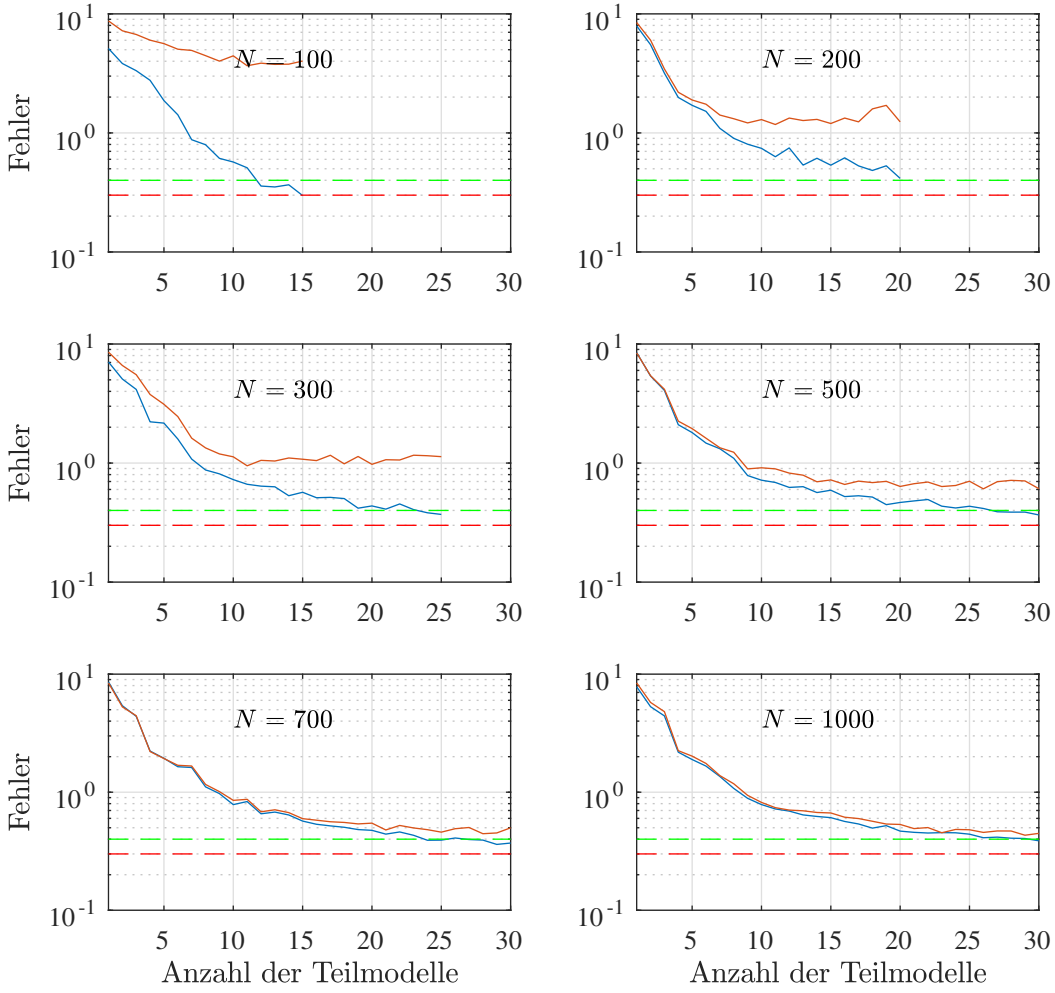

Abbildung 5.8.: Verlauf des Trainings- (blau) und Validierungsfehlers (rot) einer GMR-Modellierung für verschieden große Trainingsdatensätze. Als Abbruchkriterium wurde $RMSE \leq 0.4$ (grün-gestrichelte Linie) definiert. Die Standardabweichung des Messrauschens $\sigma = 0,3$ ist als rot-gestrichelte Linie eingezeichnet. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

Optimierung auf den Wertebereich von $[-1, 1]$ normiert worden.

Der Verlauf des Trainings- und Validierungsfehlers über die Komponentenanzahl des GMR-Modells ist in Abbildung 5.8 dargestellt. Auffällig sind die teilweise steigenden Fehlerwerte mit Erhöhung der Komponentenanzahl, welche trotz der 400 Durchläufe auf lokaloptimale Ergebnisse des Algorithmus' zurückzuführen sind.

Die hohe Parameteranzahl der GMR-Modellierung lässt eine hohe Flexibilität und damit verbunden eine starke Überanpassung an die Trainingsdaten auch mit wenigen Komponenten erwarten. Für die Approximation der Testfunktion werden jedoch deutlich mehr Teilmodelle als in der ILMON-Struktur benötigt. Dies lässt sich auf die nicht erfüllten statistischen Vor-

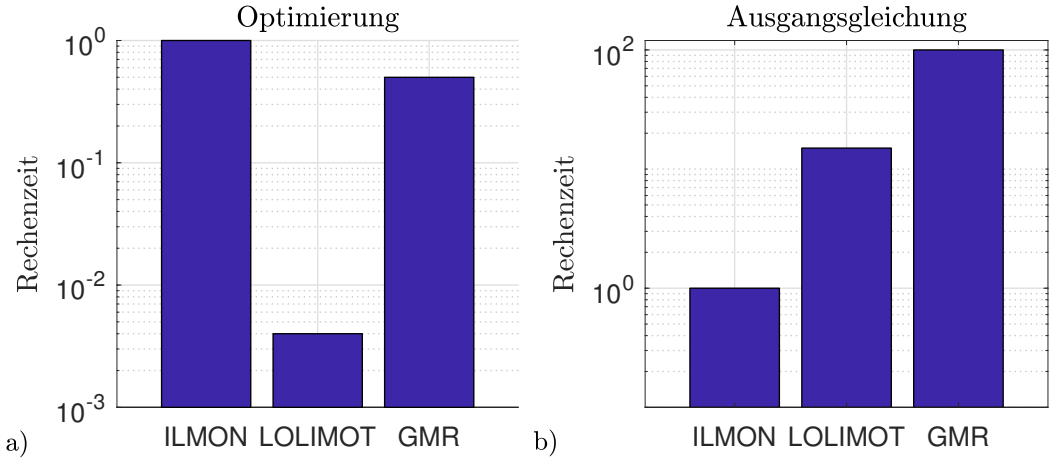

Abbildung 5.9.: Vergleich der Rechenzeiten jeweils für die ILMON-Modellierung, die LOLIMOT-Modellierung und die Gaussian-Mixture-Regression normiert auf die ILMON-Rechenzeit, a) für die Optimierung, b) für die Berechnung der Ausgangsgleichung

aussetzungen zur Anwendung der GMR zurückführen. Die Approximation über eine Summe gewichteter Normalverteilungen setzt auch die Normalverteilung der Messpunkte im Eingangsraum voraus. Bei einer gewünschten gleichmäßigen Verteilung der Messpunkte werden die gewichteten Normalverteilungen systematisch breiter geschätzt. Damit überlagern sich diese stärker, was einer Regularisierung entspricht und die effektive Parameteranzahl der Modellstruktur verringert. Die GMR benötigt somit für die Approximation starker Nichtlinearitäten mit einer gleichverteilten Anordnung der Messpunkte eine größere Anzahl an Komponenten.

Der Rechenbedarf des Optimierungsalgorithmus' ist hauptsächlich abhängig von der definierten Anzahl der Optimierungsdurchläufe zum sicheren Auffinden des globalen Optimums. Mit den genannten 400 Durchläufen und einer Komponentenanzahl von 30 liegt die Rechenzeit des GMR-Algorithmus im Verhältnis zur ILMON-Modellierung bei 1:2.

Die Berechnung der Ausgangsfunktion ist durch die notwendige Bestimmung aller Normalverteilungen bei voll besetzen Kovarianzmatrizen wesentlich aufwendiger als bei einem ILMON-Modell. Das Verhältnis der Rechenzeit der GMR-Ausgangsgleichung zur ILMON-Ausgangsgleichung liegt auf einem aktuellen Desktop-PC mit leistungsfähiger FPU bei ca. 100:1. Bei der Berechnung auf einem Steuergerät ohne FPU erhöht sich dieses Verhältnis auf ca. 5000:1 bei einfacher Genauigkeit und ca. 10000:1 bei doppelter Genauigkeit der Gleitkommazahlen und ist unter diesen Voraussetzungen nicht sinnvoll.

In Abbildung 5.9 sind die Rechenzeiten für die Optimierung und der Berechnung der Ausgangsgleichung auf einem Desktop-PC mit FPU grafisch dargestellt. Die Werte wurden auf die Rechenzeit der ILMON-Modellierung normiert.

5.1.6. Vergleich von ILMON-Modellierungen mit linearen und quadratischen lokalen Funktionen

In Kapitel 3.6 wurde die Möglichkeit der Erweiterung der lokalen Komponenten um quadratische Funktionsterme besprochen. Dies erlauben in vielen Fällen eine effektivere Modellierung mit einer geringeren Anzahl an Teilmodellen. Zum Vergleich beider Varianten

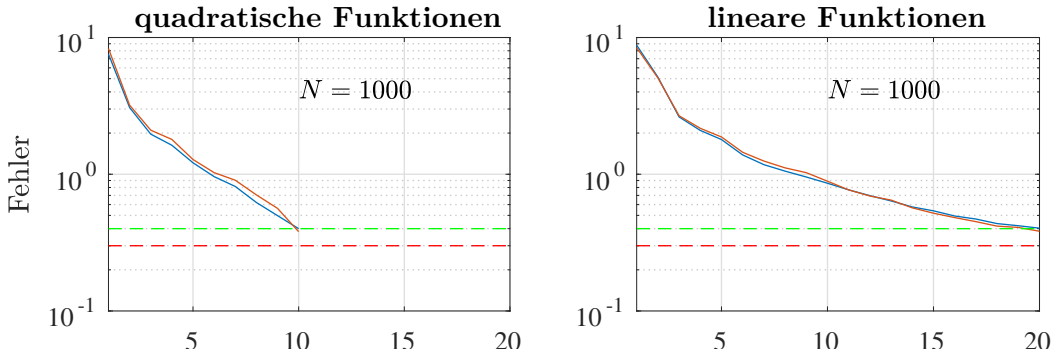

Abbildung 5.10.: Verlauf des Trainings- (blau) und Validierungsfehlers (rot) der ILMON-Modellierung mit a) lokal-quadratischen Funktionen und b) mit lokal-linearen Funktionen. Es wurde ein Trainingsdatensatz mit $N = 1000$ Messpunkten verwendet. Als Abbruchkriterium wurde ein $RMSE \leq 0.4$ (grün-gestrichelte Linie) definiert. Die Standardabweichung des Messrauschens $\sigma = 0,3$ ist als rot-gestrichelte Linie eingezeichnet. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

wurde die Modellierung über den Trainingsdatensatz von $N = 1000$ Messpunkten herangezogen. Die Rahmenbedingungen und das Abbruchkriterium blieben dabei gleich ($\sigma = 0,3$; $RMSE \leq 0,4$). Als quadratische Terme wurden u_1^2 und $u_1 u_2$ in die lokalen Komponenten aller Teilmodelle aufgenommen. Diese ergeben sich damit nach Gleichung (3.60) zu:

$$\alpha_k(\boldsymbol{u}) = \gamma_{0,k} + \gamma_{1,k} u_1 + \gamma_{2,k} u_2 + \gamma_{11,k} u_1^2 + \gamma_{12,k} u_1 u_2 \tag{5.3}$$

In Abbildung 5.10 ist der Verlauf des Trainings- und Validierungsfehlers mit zusätzlichen quadratischen Termen und im Vergleich dazu mit ausschließlich linearen Termen dargestellt. Die zusätzlichen Freiheitsgrade der erweiterten lokalen Komponenten bewirken eine wesentlich bessere Approximation des Prozesses mit weniger Teilmodellen. Der als Abbruchbedingung vorgegebene Trainingsfehler wird bereits mit 10 Teilmodellen erreicht. Es sei jedoch darauf hingewiesen, dass durch die Erhöhung der Freiheitsgerade auch die Gefahr der Überanpassung wächst. Da außerdem die Interpretierbarkeit des ILMON-Modells eingeschränkt wird, ist die Erweiterung mit quadratischen Termen nur bei physikalisch begründeten Zusammenhängen zwischen Eingangs- und Ausgangsgrößen zu empfehlen und bei der Validierung zu berücksichtigen. Insbesondere an den Teilmodellübergängen kann es bei kleinen Trainingsdatensätzen zu starken Anstiegsänderungen kommen, die oft keine Entsprechung im zu modellierenden Prozess haben. Werden diese Punkte beachtet, kann die Erweiterung der lokalen Komponenten um quadratische Terme die ILMON-Modellierung wesentlich verbessern.

5.1.7. ILMON-Modellierung mit iterativer Versuchsplanung

Der in Kapitel 4.3 beschriebene Algorithmus der iterativen Versuchsplanung wurde mit der Beispielfunktion nach Gleichung (5.1) umgesetzt. Als Designparameter muss im Vorfeld die Anzahl der Messpunkte pro Teilmodell N_k definiert werden. Dies erfolgt indirekt als Vielfaches der minimalen Messpunkteanzahl pro Teilmodell ν_k, siehe Kapitel 4.2.3.

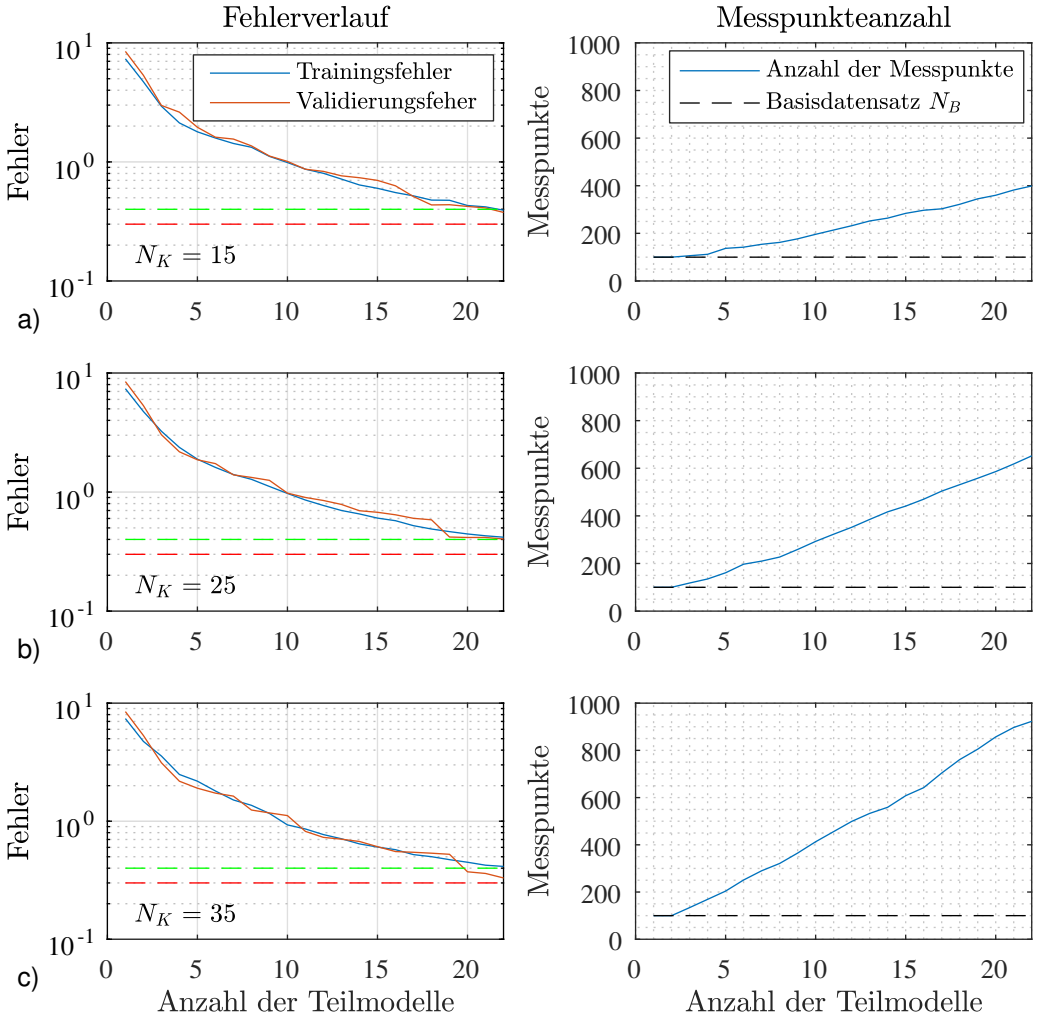

Abbildung 5.11.: Vergleich der ILMON-Modellierung mit iterativer Versuchsplanung für verschiedene Messpunkteanzahlen pro Teilmodell N_K und einem Basisdatensatz von $N_B = 100$ Messpunkten. Als Abbruchkriterium wurde ein $RMSE \leq 0.4$ (grün-gestrichelte Linie) definiert. Die Standardabweichung des Messrauschens $\sigma = 0,3$ ist als rot-gestrichelte Linie eingezeichnet. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

Zur Untersuchung der Auswirkungen unterschiedlicher Faktoren wurden Durchläufe mit $N_k = 3\nu_k$, $N_k = 5\nu_k$ und $N_k = 7\nu_k$ durchgeführt. Als Abbruchkriterium wurde wiederum ein $RMSE \leq 0,4$ vorgegeben. Der Start der Modellierung erfolgte mit einem Basisdatensatz von $N = 100$.

In Abbildung 5.11 sind die Verläufe der Fehlerwerte über die Teilmodellanzahl dargestellt. Neben den Trainings- und Validierungsfehlern ist die Messpunkteanzahl über die Teilmodellanzahl aufgetragen, welche zum jeweiligen Zeitpunkt durch die Versuchsplanung aufge-

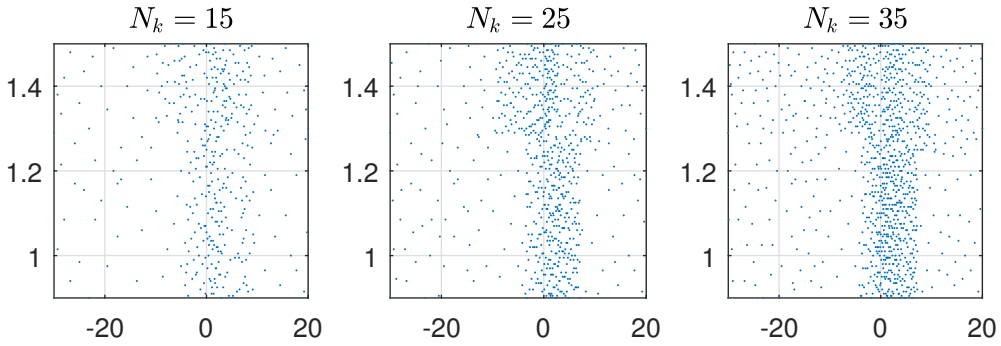

Abbildung 5.12.: Vergleich der ILMON-Modellierung mit iterativer Versuchsplanung für verschiedene Messpunkteanzahlen pro Teilmodell mit einem Basisdatensatz von $N = 100$ Messpunkten.

nommen wurde. Die dazugehörigen Verteilungen der Messpunkte im Eingangsraum sind in Abbildung 5.12 dargestellt.

Wie aus dem Vergleich hervorgeht, wird mit allen drei Vorgaben die geforderte Modellgüte mit jeweils 22 Teilmodellen erreicht. Eine Erhöhung der Anzahl an Messpunkten pro Teilmodell hat keinen Einfluss auf die erreichbare Modellgüte, solange sich alle Nichtlinearitäten in den Messungen widerspiegeln. Die höhere Messpunkteanzahl pro Teilmodell dient damit ausschließlich dem sicheren Erkennen der Nichtlinearitäten im Modell.

Hervorzuheben ist außerdem, dass der Trainings- und Validierungsfehler unabhängig von der Teilmodellanzahl weitgehend übereinstimmt. Die automatische Erhöhung der Messpunktedichte durch die iterative Versuchsplanung in den stark nichtlinearen Bereichen verhindert wirksam eine Überanpassung des Modells und ermöglicht somit auf Grundlage des Trainingsfehlers eine zuverlässige Aussage zur Modellgüte.

Im Vergleich mit einer ILMON-Modellierung bei gleichverteilten Messpunkten konnte die Datensatzgröße bei gleichem Validierungsfehler auf 40% reduziert werden, vergleiche Abbildung 5.11a und Abbildung 5.2. Die vorgestellte iterative Versuchsplanung ist somit ein effektives Mittel die für die geforderte Modellgüte notwendige Messpunkteanzahl und -verteilung ohne Prozesskenntnisse zu optimieren.

Beim Start der Versuchsplanung mit nur einem Teilmodell ist die Messpunktedichte abhängig von der definierten Anzahl pro Teilmodell sehr gering. Auf Grund dessen kann zu Beginn der Strukturoptimierung eine ungünstige Partitionierung des Eingangsraumes erfolgen oder lokal stark begrenzten Nichtlinearitäten könnten unerkannt bleiben. Ersteres kann zu einer Erhöhung der notwendigen Teilmodellanzahl führen, während der zweite Punkt die Modellgüte stark verringern kann.

Es ist daher empfehlenswert, zum Start einer Modellierung einen Basisdatensatz mit gleichverteilten Messpunkten zu erstellen, dessen Größe die Anzahl der Messpunkte pro Teilmodell deutlich übersteigt. In den obigen Durchläufen wurde dieser Basisdatensatz mit $N_B = 100$ festgelegt. Dies führt in der Regel kaum zu einer Erhöhung der Messpunkteanzahl im finalen Modell, da der Algorithmus die vorhandenen Messpunkte in der weiterführenden Optimierung mit einbezieht. Oft wird durch die günstigere Partitionierung sogar eine leichte Verringerung des Messaufwandes erreicht.

Beispielhaft wurde die Versuchsplanung und Modellierung unter den obigen Voraussetzungen mit $N_k = 15$ und ohne Basisdatensatz durchgeführt. Im Vergleich zur Versuchsplanung

mit $N_B = 100$ wurde die Güteforderung mit 23 statt 22 Teilmodellen bei einer Messpunkteanzahl von 406 statt 399 Messpunkten erreicht.

Die notwendige Größe des Basisdatensatzes ist stark vom zu modellierenden Ausgangsgrößenverlauf abhängig. Sind dort lokal stark eingegrenzte Nichtlinearitäten zu erwarten, sollte ein großer Basisdatensatz erzeugt werden. Liegen keine Prozesskenntnisse vor, ist $N_B = 10\nu_k \ldots 30\nu_k$ ein guter Startwert. Kann auf Grund von Prozesswissen eine Vorpartitionierung durchgeführt werden, ist die Aufnahme eines gesonderten Basisdatensatzes oft nicht notwendig, wenn die Nichtlinearitäten des Prozesses in initialen Modell ausreichend abgebildet sind.

5.2. Versuchsplanung und Modellierung der Füllungserfassung an einem 3,2l-Saugmotor mit variablem Ventiltrieb

Als Beispiel für eine praktische Umsetzung der Modellierung und Versuchsplanung mit einem ILMON-Modell wurde die Füllungserfassung eines 3,2l Benzinmotors mit Saugrohreinspritzung gewählt. Auf Grund der Reproduzierbarkeit der Messwerte wurde der Motor unter der Software GT-ISE von Gamma Technology simuliert. Ein weiterer Vorteil ist der reduzierte finanzielle und zeitliche Aufwand für die Umsetzung der gewünschten Referenz- und Vergleichsmessungen mit verschiedenen Parametern, welcher an einem realen Motorprüfstand enorm wäre. Als konkrete Modellierung wurde ein vollständig applizierter Motor aus der Beispielbibliothek des Programms entnommen. Ausgewählte technischen Daten des Motors sind in Tabelle 5.1 aufgeführt.

Tabelle 5.1.: Technische Daten des mit GT-ISE simulierten 4-Zylinder Ottomotors

Anzahl der Zylinder:	4
Typ der Einspritzung:	Saugrohreinspitzung
Zylinderbohrung:	100 mm
Hub:	100 mm
Hubraum:	3,236 dm^3
Kompression:	1:9,5

5.2.1. Prozessbeschreibung und Simulationskonfiguration

Zur Einhaltung der gewünschten Verbrennungsparameter und Ermittlung der einzuspritzenden Kraftstoffmenge ist die genaue Berechnung der im Brennraum eingeschlossenen Luftmasse notwendig. Dies ist neben den Öffnungs- und Schließzeiten der Ventile, der Drehzahl und der Öffnungsfläche der Drosselklappe von vielen anderen Parametern abhängig und vollständig nur über ein 3D-CFD-Modell abbildbar [91]. Zur Analyse und für vereinfachte Berechnungen wurden jedoch diverse Ansätze entwickelt, die eine gute Approximation des Prozesses möglich machen. Eine vereinfachte dynamische Beschreibung kann über eine nulldimensionale Modellierung erfolgen [92], [93], [94], [91]. Bei dieser hängen die Prozessgrößen nur von der Zeit aber nicht vom Ort ab. Die einzelnen Komponenten werden ersatzweise als Behälter oder Drosselstelle modelliert, wobei die Gasverteilung und Vermischung in den Behältern als ideal angenommen wird.

Auf Grund der hohen Dynamik des Gasaustausches erfordert dieser Ansatz allerdings auch sehr kleine Integrationsschrittweiten bei der Lösung der Differentialgleichungen. Für Berechnungen im Motorsteuergerät werden deshalb oft saugrohrdruckbasierte Ansätze genutzt [95],[96]. Diese Ansätze gehen von der Annahme aus, dass zum Zeitpunkt des Schließens des Einlassventils im Zylinder der gleiche Druck wie im Saugrohr herrscht, was bei neueren Motoren mit variablen Ventiltrieb nur eingeschränkt gültig ist.

Neben der Messung des Saugrohrdruckes und der einfachen Berechnung des Zylindervolumens zum Zeitpunkt des Schließens des Einlassventils ist die Berechnung oder Messung des Volumen des Restgases, der Gastemperatur im Zylinder und der spezifischen Gaskonstante des Gasgemisches sehr schwierig. Diese Variablen sind von sehr vielen Betriebsparametern des Motors abhängig und werden in aktuellen Motorsteuergeräten über sehr komplexe Strukturen mit vielen Kennfeldern bestimmt, die wiederum in der Entwicklungsphase des Motors aufwendig appliziert werden müssen [92], [95], [96]. Die Möglichkeit, diese Strukturen zur Berechnung der eingeschlossenen Luftmasse durch ein einzelnes datenbasiertes Modell zu ersetzen, kann somit erheblichen Entwicklungsaufwand sparen. Die in dieser Arbeit vorgestellte ILMON-Modellierung und iterative Versuchsplanung soll am Beispiel der Füllungserfassung des oben aufgeführten Motors demonstriert werden.

Es wurden 6 Eingangsgrößen für die Modellierung gewählt:

1. *Motordrehzahl:* Die Simulation wurde im Modus mit aufgeprägter Drehzahl betrieben, in welchem diese als Stellgröße vorgegeben werden kann und das zugehörige Drehmoment von der Simulationssoftware automatisch eingestellt wird. Konkret wurde der Drehzahlbereich mit $700\,\mathrm{min}^{-1} \leq n_m \leq 6500\,\mathrm{min}^{-1}$ definiert.

2. *Durchmesser der Drosselklappenöffnung:* Die Drosselklappe wird in der Simulation als Drosselstelle mit einem kreisförmigen Durchlass und variablen Lochdurchmesser simuliert. Der Durchmesser kann im Bereich $1\,\mathrm{mm} \leq d_{Dr} \leq 80\,\mathrm{mm}$ eingestellt werden.

3. *Position der Einlassnockenwelle:* Der Stellgrößenbereich der Einlassnockenwelle liegt bei $220°\mathrm{KW} \leq ENW \leq 280°\mathrm{KW}$. Daraus ergibt sich mit der definierten Hubkurve des Einlassventils der Zeitpunkt des Einlassöffnens im Bereich $333°\mathrm{KW} \leq EO \leq 393°\mathrm{KW}$ und des Einlassschließens im Bereich $605°\mathrm{KW} \leq ES \leq 665°\mathrm{KW}$.

4. *Position der Auslassnockenwelle:* Mit einem Stellbereich der Auslassnockenwelle von $95°\mathrm{KW} \leq ANW \leq 140°\mathrm{KW}$ ergibt sich der Zeitpunkt des Öffnens des Auslassventils im Bereich von $95°\mathrm{KW} \leq AO \leq 140°\mathrm{KW}$ und der des Auslassschließens zwischen $377°\mathrm{KW} \leq AS \leq 412°\mathrm{KW}$

5. *Umgebungsluftdruck:* Als Umgebungsluftdruck wurde ein Bereich von $600\,\mathrm{mbar} \leq p_U \leq 1100\,\mathrm{mbar}$ vorgegeben. Dies entspricht den möglichen Werten auf einer Höhe zwischen $-400\,\mathrm{m}$ und $4000\,\mathrm{m}$.

6. *Umgebungslufttemperatur:* Die Lufttemperatur der angesaugten Umgebungsluft wurde mit $-30\,°\mathrm{C} \leq T_U \leq 50\,°\mathrm{C}$ definiert.

Als Ausgangsgröße des Prozesses wurde die eingeschlossene Luftmasse m_l pro Arbeitsspiel und Zylinder definiert, welche von der Simulation direkt ausgegeben werden kann. In einer Umsetzung am realen Motorprüfstand kann diese über die Regelung des Lambda-Wertes

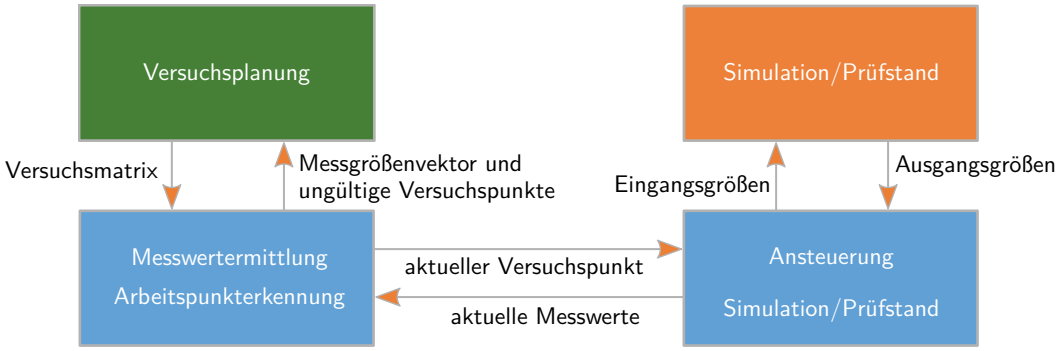

Abbildung 5.13.: Ablauf der Messwerterfassung in der Toolkette mit Matlab-Simulink

und Rückrechnung über die eingespritzte Kraftstoffmenge berechnet werden. Sind Zylinderdrucksensoren installiert, kann auch eine Berechnung über den Druckverlauf während der Kompressionsphase erfolgen.

Die Einbindung der Messwertermittlung in die Versuchsplanung wurde über Matlab-Simulink realisiert und ist weitestgehend universell gehalten, sodass die Verwendung sowohl an einem realen Motorprüfstand als auch an der GT-ISE-Simulation möglich ist. Wie in Abschnitt 4.3.1 ausgeführt, siehe auch Abbildung 4.7, nimmt die Messwertermittlung unter Simulink die Versuchsmatrix vom Planungsalgorithmus entgegen, fährt diese ab und gibt den entsprechenden Messwertvektor sowie die nicht anfahrbaren Versuchspunkte zurück. Die zeitliche Umschaltung zwischen den Messpunkten erfolgt über eine Arbeitspunkterkennung unter Vorgabe eines Toleranzbereiches und der Anzahl der Arbeitszyklen in denen die Zielgröße innerhalb dieses Bereiches verlaufen muss. Weiterhin können eine minimale und eine maximale Anzahl von Arbeitszyklen pro Versuchspunkt definiert werden. Neben der eigentlichen Zielgröße ist die Überwachung weiterer Messgrößen auf das Erreichen statischer Endwerte möglich. Dies ist oft bei Motoren mit Turboladern und den dort vorkommenden langsamen bzw. verzögerten Änderungen der Prozesszustände nötig, welche sich entsprechend langsam auf die Zielgröße auswirken können. Als letzter Aufgabenbereich der Messwertermittlung erfolgt eine Filterung der Eingangsgrößensignale während der Umschaltvorgänge, sodass praktische Restriktionen wie maximale Stellgeschwindigkeiten eingehalten werden.

Die in kontinuierliche Eingangsgrößenverläufe umgewandelten Versuchspunkte können nachfolgend der Steuerung eines Versuchsstands übergeben werden oder wie in dieser Arbeit der Simulink-Ansteuerung der GT-ISE-Simulation. Die resultierenden kontinuierlichen Messgrößen werden auf ihre stationären Endwerte reduziert und den jeweiligen Versuchspunkten zugeordnet. Die Struktur der Messwerterfassung ist in Abbildung 5.13 dargestellt.

5.2.2. Reduzierte Modellierung der Füllungserfassung mit zwei Eingangsgrößen

Für eine detaillierte Analyse wurde im ersten Schritt eine auf zwei Eingangsgrößen reduzierte Modellierung untersucht. Als variable Eingangsgrößen wurden die Drehzahl und die Drosselklappenöffnung verwendet. Die anderen 4 Eingangsgrößen wurden als feste Werte mit $ENW = 239°\text{KW}$, $ANW = 126°\text{KW}$, $p_U = 1000\,\text{mbar}$ und $T_U = 300\,\text{K}$ definiert. Die Validierung erfolgte über einem gleichverteilten Referenzdatensatz mit $N = 10.000$ Messdaten. Der Verlauf der eingeschlossenen Luftmasse in Abhängigkeit von der Drosselklappenöffnung

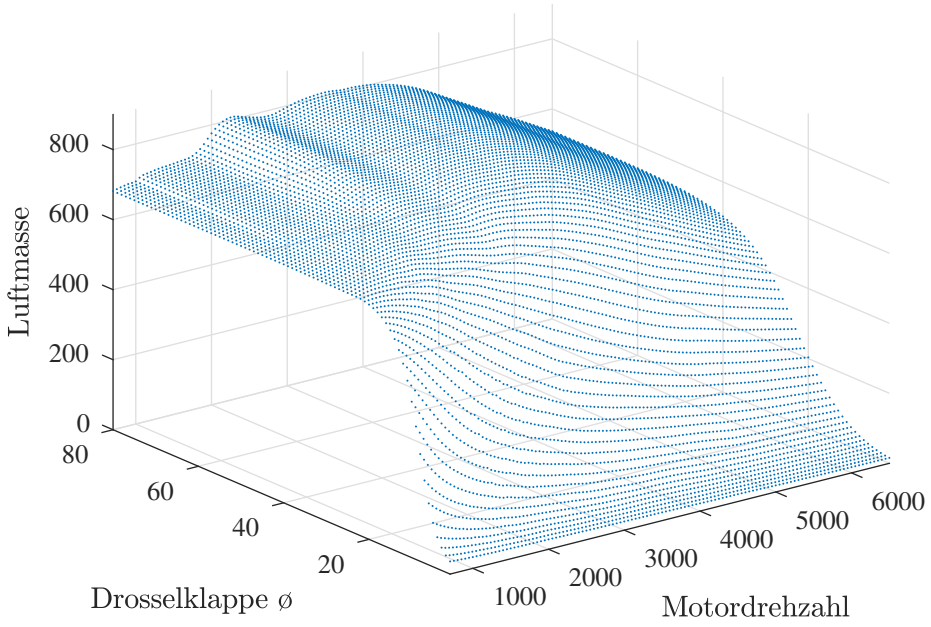

Abbildung 5.14.: Verlauf der eingeschlossenen Luftmasse pro Arbeitsspiel des Beispielmotors über die Motordrehzahl und dem Durchmesser der Drosselklappenöffnung. Die Messpunkte entsprechen dem Referenzdatensatz mit $N = 10000$.

und der Drehzahl ist in Abbildung 5.14 dargestellt.

Zur Demonstration der Möglichkeit dedizierte Optimierungskriterien zu definieren und in der ILMON-Modellierung zu nutzen, soll nachfolgend ein spezielles relatives Fehlerkriterium konstruiert werden. Da sich Abweichungen in der Füllungserfassung bei kleineren absoluten Werten stärker negativ auswirken, wurde als Ausgangspunkt das arithmetische Mittel des relativen Fehlers gewählt, welches folgendermaßen definiert wurde:

$$\epsilon_{rel} = \sqrt{\frac{\sum_{i=1}^{N} (\frac{\hat{y}-y}{y})^2}{N}}. \tag{5.4}$$

Die in einigen Bereichen des Prozesses schwierige Arbeitspunkterkennung resultiert in den Simulationsergebnissen in Fehlervarianzen von $\pm 2\,\mathrm{mg}$. Bei kleinen Werten der Ausgangsgröße führt dies im Wertebereich von $10\,\mathrm{mg} \leq y \leq 900\,\mathrm{mg}$ zu sehr großen relativen Fehlern. Bei Messungen am realen Motorprüfstand würden zusätzliche Messfehlervarianzen die Problematik weiter verschärfen. Zur Berücksichtigung dieser systembedingten Varianz der Ausgangsgröße wurde die Gleichung (5.4) wie folgt erweitert:

$$\epsilon_{rel} = \sqrt{\frac{\sum_{i=1}^{N} (\frac{\hat{y}-y}{z})^2}{N}} \quad \text{mit} \quad z = m_\epsilon y + n_\epsilon. \tag{5.5}$$

Mit der Wahl eines relativen Fehlers von $\epsilon_{rel} = 2\%$ ergibt sich für den Maximalwert der Ausgangsgröße von $y = 900\,\mathrm{mg}$ ein Absolutwert des Fehlers von $\epsilon = 18\,\mathrm{mg}$. Für den kleinsten

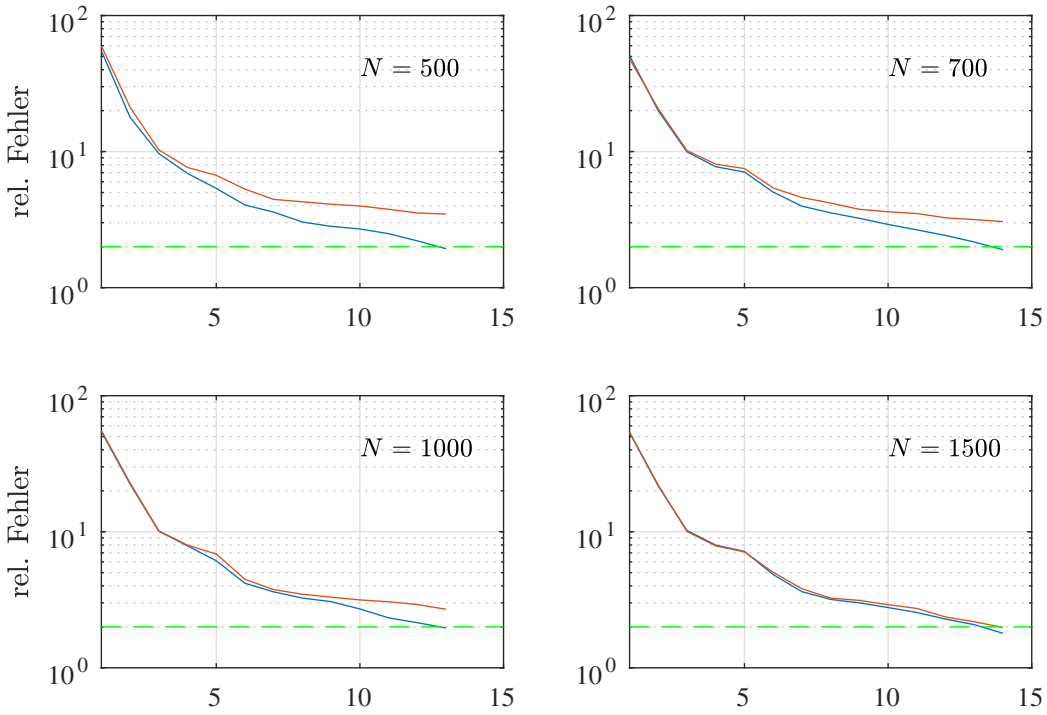

Abbildung 5.15.: Verlauf des Trainings-(blau) und Validierungsfehlers (rot) einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel für unterschiedlich große Trainingsdatensätze. Als Abbruchkriterium wurde $\epsilon_{rel} \leq 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

Wert der Ausgangsgröße $y = 10\,\mathrm{mg}$ soll ein durchschnittlicher absoluter Fehler von $2\,\mathrm{mg}$ eingehalten werden. Dieser soll im Fehlerkriterium mit der gleichen Wichtung berücksichtigt werden. Mit diesen Vorgaben ergibt sich als Gleichungssystem

$$\begin{aligned} 2\% &= \frac{18\,\mathrm{mg}}{m_\epsilon \cdot 900\,\mathrm{mg} + n_\epsilon} \\ 2\% &= \frac{2\,\mathrm{mg}}{m_\epsilon \cdot 10\,\mathrm{mg} + n_\epsilon} \end{aligned} \tag{5.6}$$

und als dessen Lösung die Werte für Gleichung 5.5 mit $m_\epsilon \approx 1,01$ und $n_\epsilon \approx 89{,}87\,\mathrm{mg}$.

Modellierung mit gleichverteilten Datensätzen

Zur Beurteilung der Modellgüte bei Modellierungen mit über dem Eingangsraum gleichverteilten Messpunkten wurden Durchläufe mit unterschiedlich großen Datensätzen durchgeführt. Die resultierenden Verläufe des Trainings- und Validierungsfehlers über die Teilmodellanzahl sind in Abbildung 5.15 dargestellt.

Der vorgegebene Trainingsfehler von $\epsilon_{rel} \leq 2\%$ wird mit allen Datensätzen erreicht, wobei bei niedrigen Messdatenanzahlen eine starke Überanpassung des Modells erfolgt. Ab $N =$

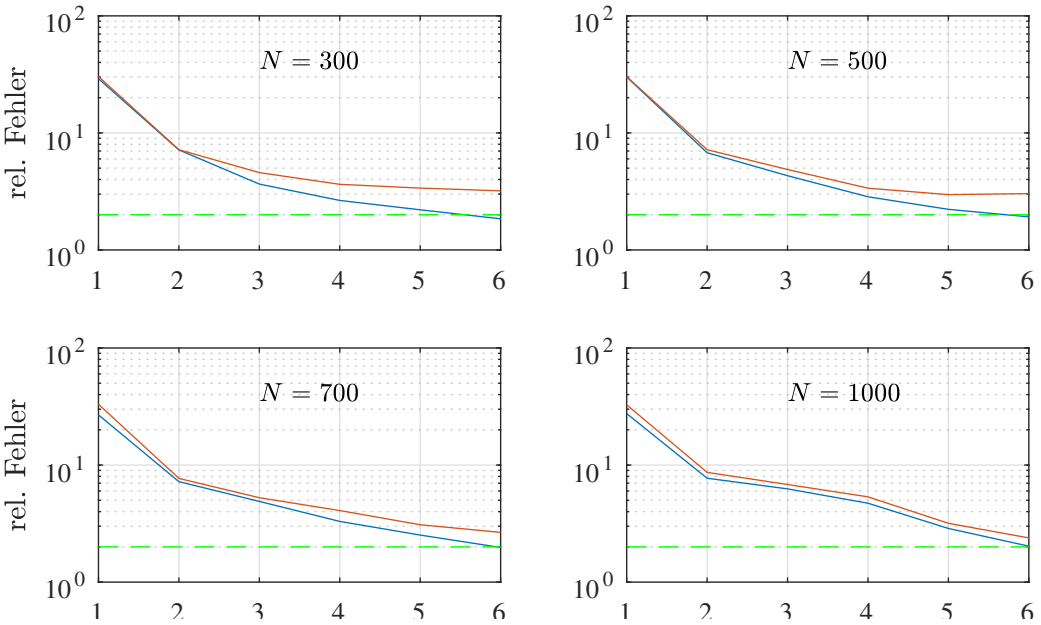

Abbildung 5.16.: Verlauf des Trainings- (blau) und Validierungsfehlers (rot) einer ILMON-Modellierung mit quadratischen Teilmodellen der eingeschlossenen Luftmasse pro Arbeitsspiel für unterschiedlich große Trainingsdatensätze. Als Abbruchkriterium wurde $\epsilon_{rel} \leq 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

1000 liegt der Validierungsfehler zwar höher als der Trainingsfehler, reduziert sich jedoch stetig. Erst ab einer Datensatzgröße von $N = 3000$ entspricht der Trainingsfehlerverlauf annähernd dem des Validierungsfehlers.

Ursache für diesen hohen Messdatenbedarf ist die Optimierung des Modells auf das relative Fehlerkriterium und damit auf die Bereiche mit niedrigen Luftmassewerten. Für eine höhere Modellgüte und zur Vermeidung von Überanpassungen müssten in diesen Bereichen mehr Messdaten aufgenommen werden, was jedoch a-priori Kenntnisse über den Verlauf der Ausgangsgröße notwendig macht.

Die stark abgerundeten Verläufe der Ausgangsgröße lassen eine wesentlich bessere Approximation mit quadratischen Komponenten $\alpha_k(\boldsymbol{u})$ erwarten, weshalb zum Vergleich eine Erweiterung der ILMON-Modellierung mit vollständigen quadratischen Termen vorgenommen wurde. Die Fehlerverläufe über vier Datensätze sind in Abbildung 5.16 dargestellt. Der vorgegebene Trainingsfehler konnte in dieser Konfiguration mit einer deutlich kleineren Teilmodellanzahl erreicht werden. Ein weitgehend übereinstimmender Verlauf des Trainings- und Validierungsfehler ist bereits bei einer Datensatzgröße von $N = 1000$ gegeben. Dies kann auf die höhere Messpunkteanzahl pro Teilmodell im Zusammenspiel mit der höheren Parameteranzahl pro Teilmodell zurückgeführt werden.

Modellierung mit iterativer Versuchsplanung

Die ILMON-Modellierung mit iterativer Versuchsplanung wurde in der Konfiguration mit rein linearen Komponenten als auch mit quadratischen Regressionstermen durchgeführt, wiederum mit der Fehlerdefinition nach Gleichung (5.5) und einem Abbruchkriterium von $\epsilon \leq 2\%$. In Abbildung 5.17 ist in der linken Spalte der Fehlerverlauf des Trainings- und Validierungsfehlers über der Teilmodellanzahl für verschiedene Messpunkteanzahlen pro Teilmodell dargestellt. Die zugehörige Gesamtzahl der Messpunkte in Abhängigkeit der erreichten Teilmodellanzahl ist in der rechten Spalte abgebildet.

Wie aus den dargestellten Verläufen hervorgeht, verhindert die optimale Messpunkteverteilung der Versuchsplanung effektiv eine Überanpassung an die Trainingsdaten und Trainings- und Validierungsfehler weisen auch mit einer geringen Anzahl an Messpunkten pro Teilmodell einen weitgehend deckungsgleichen Verlauf auf.

In Abbildung 5.18 ist der Trainings- und Validierungsfehlerverlauf für die Modellierung mit quadratischen Komponenten abgebildet. Im Vergleich zur Modellierung mit linearen Komponenten kommt diese Umsetzung mit einer wesentlich geringeren Teilmodellanzahl bei gleichem Modellfehler aus. Infolgedessen ist auch die Gesamtzahl der benötigten Messpunkte sehr gering. Der weitgehend identische Verlauf des Validierungsfehlers bestätigt die gute Approximation des Ausgangsgrößenverlaufs über die quadratischen Terme ohne eine Überanpassung des Modells an die Trainingsdaten.

Der Vergleich der Messpunkteverteilung beider Modellierungsvarianten in Abbildung 5.19 zeigt keinen prinzipiellen Unterschied in der Verteilung im Eingangsraum. Bedingt durch die relative Fehlerdefinition werden die Bereiche mit geringen Luftmassewerten stärker vermessen. Der größere Messpunktebedarf pro Teilmodell bei der Approximation mit quadratischen Teilmodellen wird in diesem Beispiel durch den geringeren Teilmodellbedarf ausgeglichen, sodass vom Versuchsplanungsalgorithmus insgesamt weniger Arbeitspunkte vermessen werden als bei der Modellierung mit linearen Termen.

5.2.3. Modellierung der Füllungserfassung mit 6 Eingangsgrößen

Zur Überprüfung und Bewertung der vollständigen Modellierung mit allen sechs Eingangsgrößen wurde ein Referenzdatensatz mit insgesamt 40.000 raumfüllend verteilten Versuchspunkten aufgenommen. Die Verteilung der Messdaten wurde über das Latin-Hypercube-Verfahren mit Maximin-Kriterium bestimmt. Die Luftmasse pro Arbeitsspiel liegt in den Messdaten im Bereich $1{,}22\,\mathrm{mg} \leq m_l \leq 1007{,}5\,\mathrm{mg}$.

In einigen Bereichen des Eingangsraumes oszillieren die Simulationswerte über mehrere Arbeitsspiele, was zu einer schwierigen Arbeitspunkterkennung führt und die Ergebnisse je nach Abbruchzeitpunkt um ca. $\pm 8\,\mathrm{mg}$ variieren lässt. Dieser Wert kann als Messfehler interpretiert werden und wurde daher in der Definition des Fehlerkriteriums nach Gleichung (5.5) als Minimalwert für die definierten 2% relativer Fehler eingesetzt. Mit dem oben genannten Wertebereich der Simulation ergeben sich die Parameter des Fehlerkriteriums aus dem Gleichungssystem

$$\begin{aligned} 2\% &= \frac{20\,\mathrm{mg}}{m_\epsilon \cdot 1000\,\mathrm{mg} + n_\epsilon} \\ 2\% &= \frac{8\,\mathrm{mg}}{m_\epsilon \cdot 1\,\mathrm{mg} + n_\epsilon} \end{aligned} \tag{5.7}$$

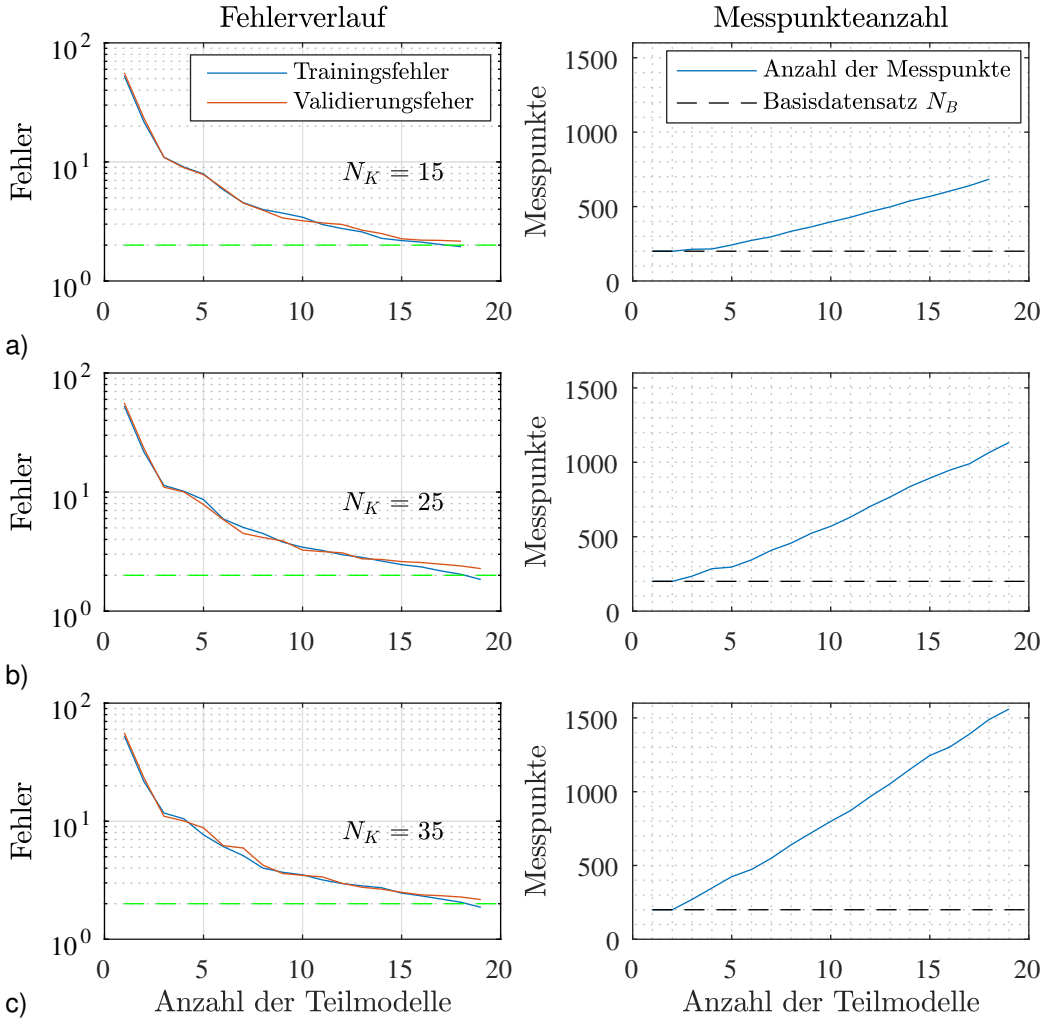

Abbildung 5.17.: Verlauf des Trainings- und Validierungsfehlers einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel mit linearen Teilmodellen und iterativer Versuchsplanung. Als Abbruchkriterium wurde $\epsilon_{rel} \leq 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

mit $m_\epsilon = 0,6$ und $n_\epsilon = 399{,}4\,\mathrm{mg}$.

Bedingt durch die 6 Eingangsgrößen des Prozesses ist für eine Modellierung mit gleichmäßig verteilten Messpunkten eine sehr hohe Messpunkteanzahl notwendig und in der praktischen Umsetzung nicht mehr sinnvoll. Die Messungen der Versuchsdaten der nachfolgenden Modellierungsbeispiele erfolgten daher über die iterative Versuchsplanung. Alle Modellierungsdurchläufe wurden ohne Vorpartitionierung mit einem einzelnen Teilmodell in den definierten Grenzen des Gesamtmodells gestartet.

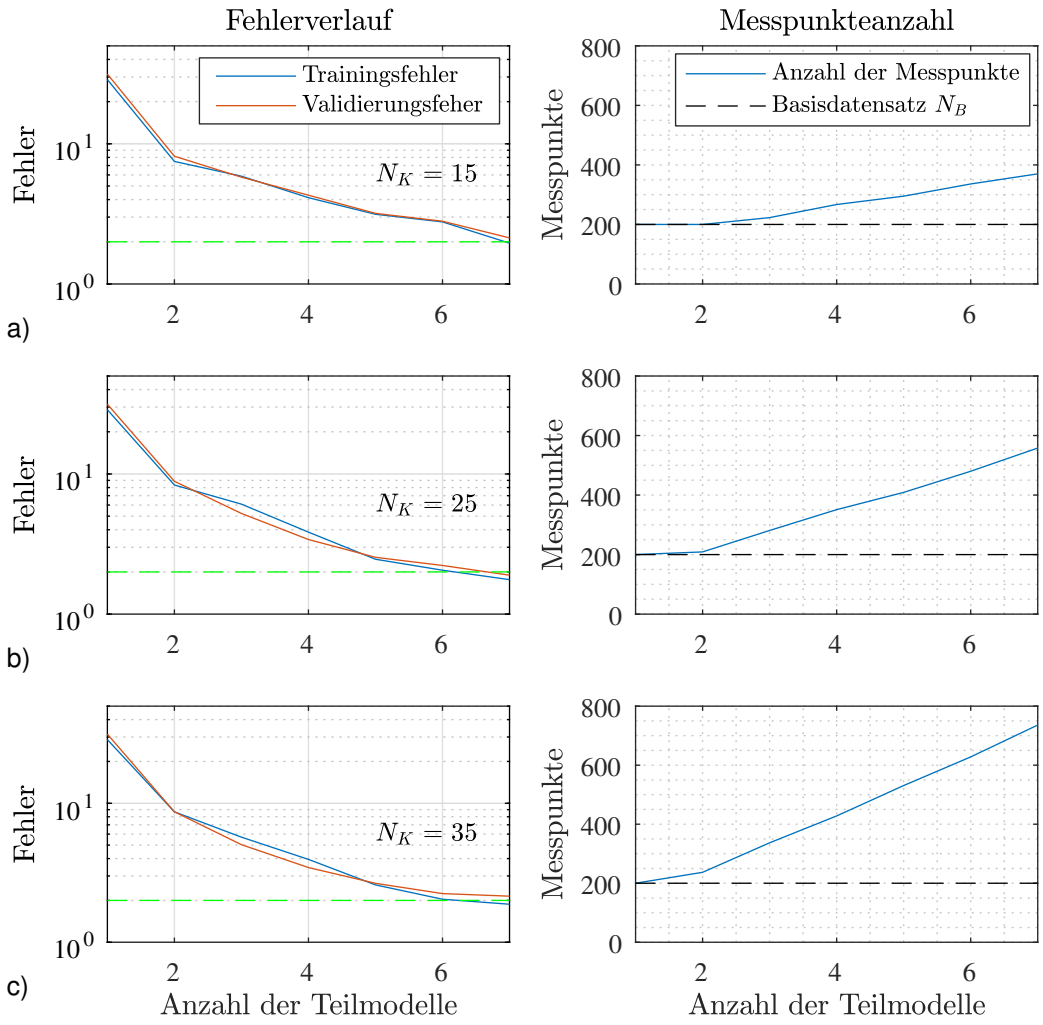

Abbildung 5.18.: Verlauf des Trainings- und Validierungsfehlers einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel mit linearen Teilmodellen und iterativer Versuchsplanung. Als Abbruchkriterium wurde $\epsilon_{rel} \leq 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 10.000$ bestimmt.

Modellierung mit vollständig quadratischen Komponenten Im ersten Beispiel wurde eine möglichst geringe Teilmodellanzahl im optimierten Modell angestrebt, weshalb eine ILMON-Modellierung mit vollständig aktivierten quadratischen Termen und damit der höchsten Regressoranzahl verwendet wurde. Nachteil dieser größeren Flexibilität ist die verringerte Interpretierbarkeit der einzelnen Teilmodelle.

Es wurde ein Basisdatensatz mit $N_B = 1000$ Messwerten verwendet, welche raumfüllend nach dem Latin-Hypercube-Verfahren mit Maximin-Kriterium aufgenommen wurden. Als Parameter für die Versuchsplanung wurde die Mindestanzahl an Messpunkten pro Teilmodell

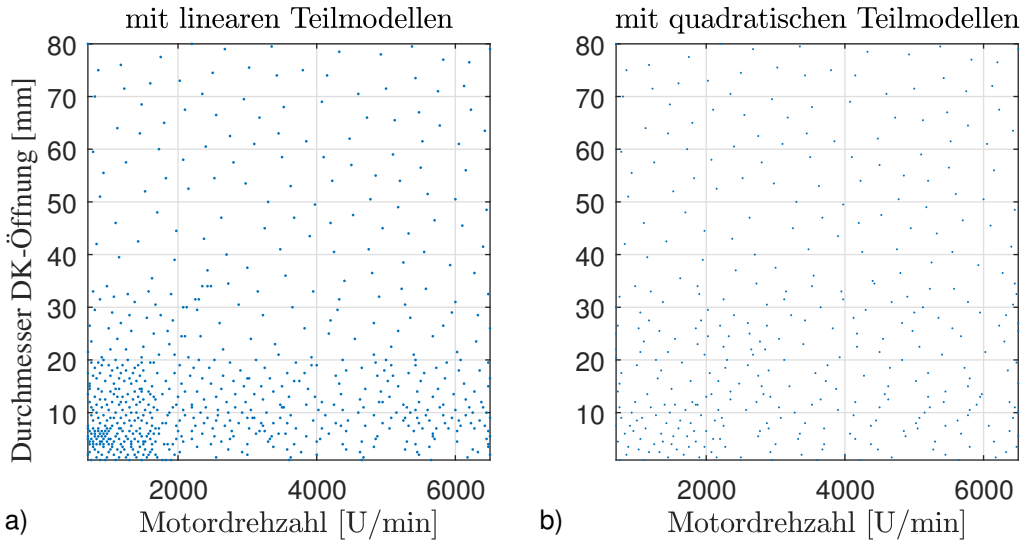

Abbildung 5.19.: Vergleich der Messpunkteverteilung bei einer ILMON-Modellierung mit iterativer Versuchsplanung mit linearen und quadratischen Teilmodellen.

nach Gleichung (4.12) mit dem Dreifachen sowie dem Fünffachen der minimal notwendigen Messpunkte pro Teilmodell ν_k definiert, was einer Messpunkteanzahl von $N_k = 3\nu_k = 156$ und $N_k = 5\nu_k = 260$ entspricht.

In Abbildung 5.20 sind die Verläufe des Trainings- und Validierungsfehlers sowie der Messpunkteanzahl über der Teilmodellanzahl dargestellt. Das angestrebte Ziel einer Approximation mit wenigen Teilmodellen konnte in dieser Konfiguration gut umgesetzt werden. Im Vergleich der Durchläufe mit verschiedenen Messpunkteanzahlen pro Teilmodell erfolgte jedoch die Partitionierung des Eingangsraumes in beiden Fällen mit der gleichen Effizienz, siehe Abbildung 5.20c. Der höhere Messaufwand führt in diesem Beispiel zu keiner Verringerung der Teilmodellanzahl aber erhöhte die Anzahl an Messdaten deutlich, vergleiche Abbildung 5.20b und 5.20d.

Auffallend in den Fehlerverläufen ist, dass der Validierungsfehler zwar dem Trainingsfehler folgt, dies jedoch in einigen Bereichen sprunghaft. Ursache hierfür ist die unterschiedliche Verteilung der Trainings- und Validierungsdaten über die Teilmodelle. Während der Versuchsplanungsalgorithmus eine annähernd gleiche Anzahl an Trainingsdaten in jedem Teilmodell erzeugt, richtet sich die Anzahl der Validierungsdaten in jedem Teilmodell, bedingt durch die gleichmäßige Verteilung im Eingangsraum, nach der Größe der einzelnen Teilmodelle. In dieser Konstellation kommt es zu einer sprunghaften Änderung des Validierungsfehlers, sobald eines der Teilmodelle mit einer großen Anzahl an Validierungsdaten geteilt wird und sich infolgedessen die Fehlerwerte sehr vieler Messpunkte ändern.

In Abbildung 5.21 ist die Verteilung der Trainings- und Validierungsdaten in den Teilmodellen für die in Abbildung 5.20a gezeigte Modellierung dargestellt. Die Teilmodelle 2, 6 und 9 enthalten hier mit einer Summe von ca. 20.000 Messpunkten rund 50 % der Validierungsdaten und dominieren damit das Fehlerkriterium. Teilmodelle mit einer sehr geringen Anzahl an Validierungsdaten (4, 11, 13 - 16, 18, 19) beeinflussen das gewählte Fehlerkriterium dagegen kaum. Im Unterschied dazu sind in den Trainingsdaten alle Teilmodelle annähernd gleich stark vertreten, wodurch der Trainingsfehler mit jeder Teilung gleichmäßig verringert

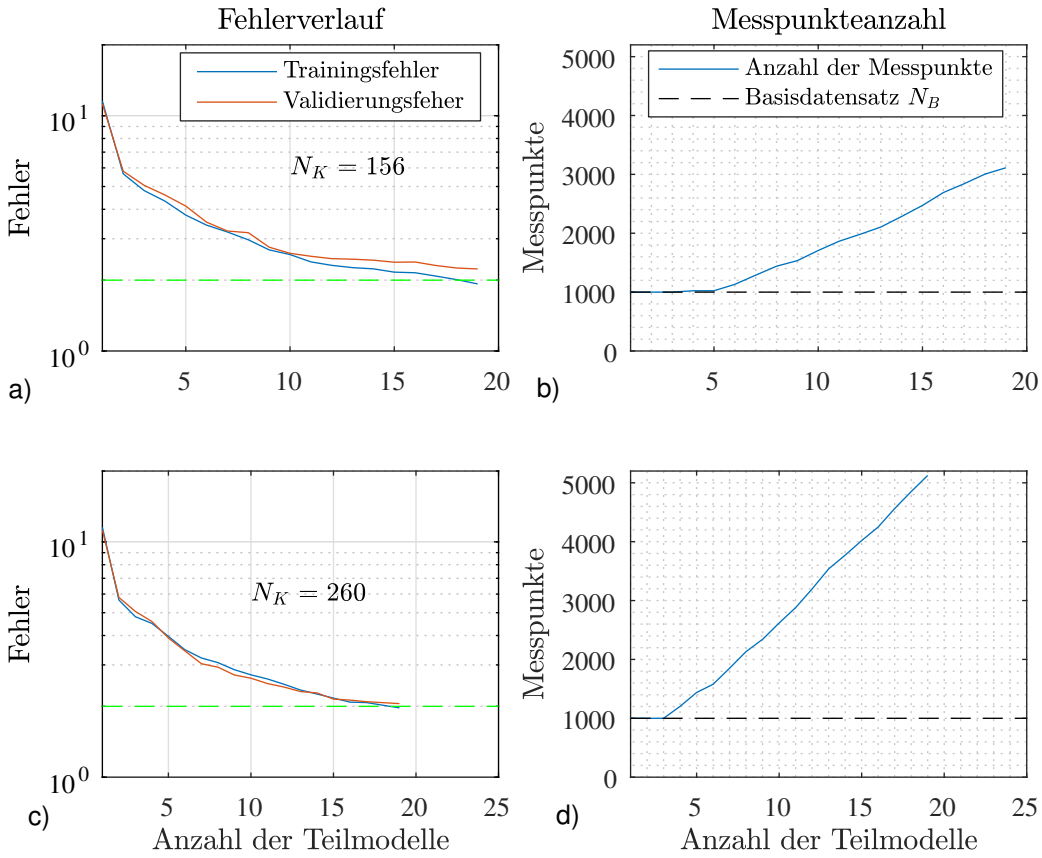

Abbildung 5.20.: Verlauf des Trainings- und Validierungsfehlers einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel mit quadratischen Teilmodellen und iterativer Versuchsplanung. Als Abbruchkriterium wurde $\epsilon_{rel} \leq 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 40.000$ bestimmt.

wird. Der Effekt reduziert sich, wie im Vergleich der Abbildungen 5.20a und c erkennbar, mit einer Erhöhung der Messpunkteanzahl pro Teilmodell in der iterativen Versuchsplanung.

Modellierung mit ausgewählten quadratischen Regressoren Der guten Approximation mit wenigen Teilmodellen und die damit verbundene einfachere Validierung des Modells steht als Nachteil die schlechtere Interpretierbarkeit der einzelnen Teilmodelle entgegen, da mit den vollständig aktivierten quadratischen Regressoren auch Kopplungen zweier Eingangsgrößen u_1u_2, u_1u_3 usw. in die lokalen Komponenten α_k des Modells einfließen. In der praktischen Umsetzung wird dieser Zielkonflikt häufig auftreten, weshalb sich als Strategie eine stufenweise Reduktion der Regressorenanzahl anbietet.

Im ersten Schritt wird dazu eine Modellierung mit vollständigen quadratischen Termen durchgeführt und das resultierende Modell zur Abschätzung der Signifikanz einzelner Regres-

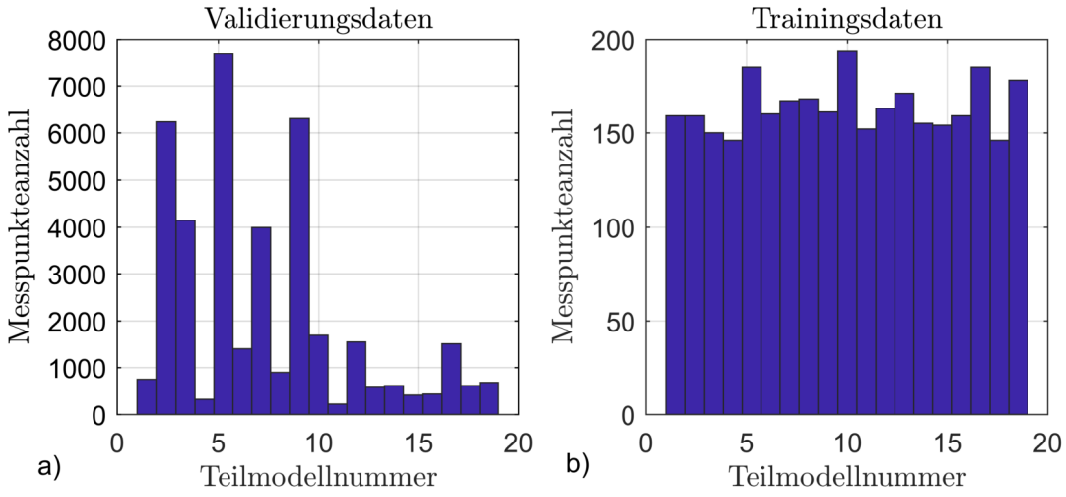

Abbildung 5.21.: Anzahl der Messdaten des Trainings- und Validierungsdatensatzes in den einzelnen Teilmodellen.

soren, der benötigten Teilmodellanzahl und der damit verbundenen Anzahl an Messdaten herangezogen. Zur Abschätzung der Signifikanz der Regressoren können Gütemaße, wie das Bestimmtheitsmaß oder die in Abschnitt 2.2.1 vorgestellten Informationskriterien genutzt werden oder automatische Verfahren, wie das Stepwise-Regression-Verfahren [97], [98], [99], welches sukzessiv die beste Kombination an Regressoren zur Berechnung der Zielgröße ermittelt. Die für die Anwendbarkeit der darin verwendeten statistischen Tests notwendigen Bedingungen sind in der Praxis jedoch oft nicht gegeben, weshalb diese Verfahren kontrovers diskutiert werden [98], [100], [101], [102]. Neben dieser statistischen Bestimmung der Abhängigkeiten ist die wissensbasierte und empirische Auswahl der Regressoren möglich und sinnvoll. Ein praktisch motivierter Kompromiss ist die Beschränkung auf die linearen und rein quadratischen Eingangsgrößen u_1^2, u_2^2 usw. als Regressoren. Diese wirken sich nur in der Achse der jeweiligen Eingangsgröße aus und lassen eine gute Abschätzung von Verläufen innerhalb der nicht von Messdaten abgedeckten Bereiche zu.

Nach der Auswahl der Regressoren kann im zweiten Schritt eine Modellierung mit reduzierter Regressorenanzahl und somit verbesserter Interpretierbarkeit durchgeführt werden. Je nach Einfluss der deaktivierten Regressoren erhöht sich in den meisten Anwendungen sowohl die Teilmodellanzahl als auch der Datenbedarf. Dabei lassen die Analyse des bestehenden Modells und die dafür benötigte Messpunkteanzahl eine grobe Abschätzung des Messaufwandes zu. Der im ersten Schritt entstandene Datensatz kann für den finalen Durchlauf als Basisdatensatz dienen, womit kein zusätzlicher Messaufwand entsteht.

Dieses Vorgehen wurde in einem zweiten Beispiel umgesetzt. Der Prozess wurde ohne Querkopplungen, ausschließlich mit den linearen Termen und den Quadraten der Eingangsgrößen modelliert. Die lineare Komponente jedes Teilmodells ergibt sich über die Gleichung

$$\begin{aligned}\alpha_k(\boldsymbol{u}) = \gamma_{0,k} + \gamma_{1,k} n_m + \gamma_{2,k} d_{Dr} + \gamma_{3,k} ENW + \gamma_{4,k} ANW + \gamma_{5,k} p_U + \gamma_{6,k} T_U \\ + \gamma_{7,k} n_m^2 + \gamma_{8,k} d_{Dr}^2 + \gamma_{9,k} ENW^2 + \gamma_{10,k} ANW^2 + \gamma_{11,k} p_U^2 + \gamma_{12,k} T_U^2.\end{aligned} \tag{5.8}$$

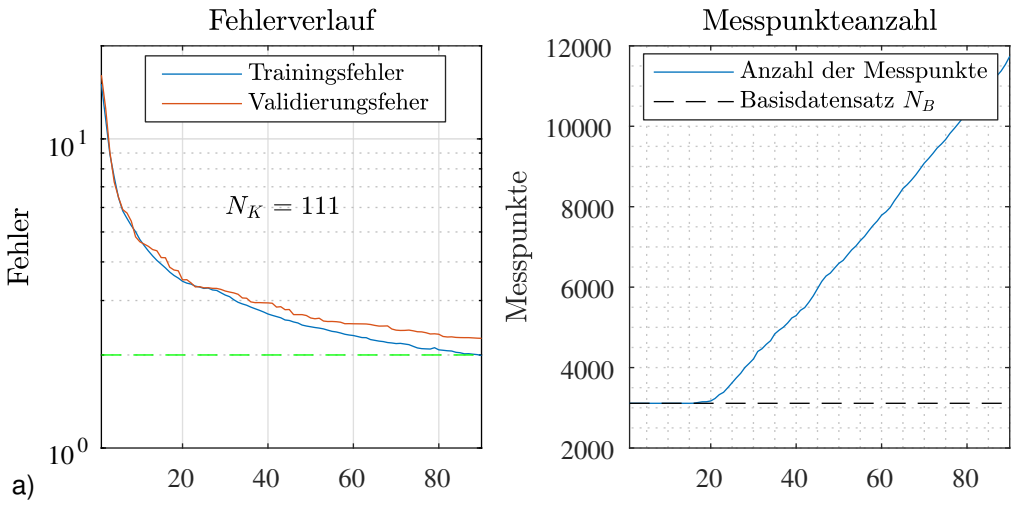

Abbildung 5.22.: Verlauf des Trainings- und Validierungsfehlers einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel mit quadratischen Teilmodellen und iterativer Versuchsplanung. Als Abbruchkriterium wurde $\epsilon_{rel} = 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 40.000$ bestimmt.

Als Basisdatensatz wurde der resultierende Trainingsdatensatz aus dem obigen Beispiel mit $N_B = 3112$ Messpunkten, siehe Abbildung 5.20b, gewählt.

Das Ergebnis dieses Modellierungsdurchlaufs ist in Abbildung 5.22 dargestellt. Wie erwartet, wird eine deutlich größere Anzahl an Teilmodellen für die Approximation des Prozesses mit der gleichen Modellgüte benötigt. Der Vorteil der verringerten Messdatenanzahl pro Teilmodell wird durch die höhere Teilmodellzahl negiert, sodass insgesamt ein wesentlich größerer Messdatensatz aufgenommen wurde. Zusammenfassend wird hier deutlich, dass bei dem vorliegenden Prozess die in diesem Beispiel deaktivierten Regressoren einen großen Einfluss auf die Zielgröße haben, der durch die höhere Anzahl an Teilmodellen kompensiert werden muss.

Modellierung mit wissensbasiert gewählten Regressoren Als abschließendes Beispiel und als Vergleich zu den obigen Parametrierungen soll eine Modellierung des Prozesses mit wissensbasiert gewählten Regressoren demonstriert werden. Ausgehend von früheren Untersuchungen [94], die zeigten, dass die quadratischen Terme des Umgebungsdrucks einen erheblichen Einfluss auf die im Zylinder eingeschlossene Luftmasse haben, wurden alle quadratischen Regressoren des Umgebungsdrucks aktiviert. Neben diesen aus Prozesswissen resultierenden Erweiterungen der lokalen Komponenten wurden zusätzlich die Quadrate von Drehzahl, Drosselklappenöffnung, Einlassnockenwellenwinkel und Auslassnockenwellenwinkel als Regressoren ausgewählt, um weitläufige Anstiegsänderungen der Zielgröße in diesen Dimension besser approximieren zu können. Zusammengefasst ergibt sich der Parametervektor $\boldsymbol{\gamma}_k$ eines Teilmodells somit aus 17 Parametern und eine lineare Komponente α_k berechnet

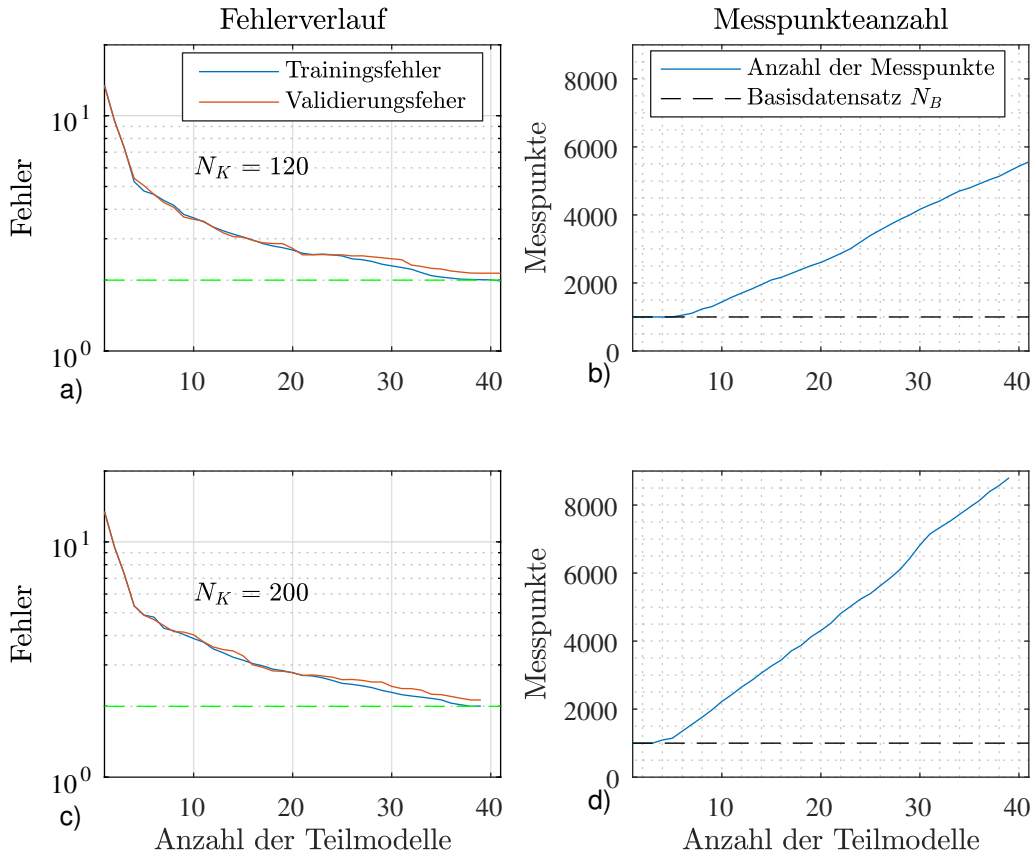

Abbildung 5.23.: Verlauf des Trainings- und Validierungsfehlers einer ILMON-Modellierung der eingeschlossenen Luftmasse pro Arbeitsspiel mit ausgewählten quadratischen Regressoren in den Teilmodellen sowie mit iterativer Versuchsplanung. Als Abbruchkriterium wurde $\epsilon_{rel} = 2\%$ (grün-gestrichelte Linie) definiert. Der Validierungsfehler wurde über dem Referenzdatensatz mit $N = 40.000$ bestimmt.

sich mit

$$\begin{aligned}\alpha_k(\boldsymbol{u}) = {} & \gamma_{0,k} + \gamma_{1,k} n_m + \gamma_{2,k} d_{Dr} + \gamma_{3,k} ENW + \gamma_{4,k} ANW + \gamma_{5,k} p_U + \gamma_{6,k} T_U \\ & + \gamma_{7,k} n_m^2 + \gamma_{8,k} d_{Dr}^2 + \gamma_{9,k} ENW^2 + \gamma_{10,k} ANW^2 + \gamma_{11,k} p_U^2 \\ & + \gamma_{12,k} n_m p_U + \gamma_{13,k} d_{Dr} p_U + \gamma_{14,k} ENW p_U + \gamma_{15,k} ANW p_U + \gamma_{16,k} T_U p_U .\end{aligned} \tag{5.9}$$

Als Basisdatensatz kam der $N_B = 1000$ Messwerte umfassende obige Datensatz zum Einsatz. Für die Versuchsplanung wurde die Mindestanzahl an Messpunkten pro Teilmodell nach Gleichung (4.12) mit dem Dreifachen der minimal notwendigen Messpunkte pro Teilmodell ν_k definiert, was einer Messpunkteanzahl von $N_k = 3\nu_k = 120$ entspricht. Zum Vergleich wurde weiterhin ein Durchlauf mit $N_k = 5\nu_k = 200$ Messpunkten durchgeführt.

In Abbildung 5.23 ist der Verlauf des Trainings- und Validierungsfehlers sowie der Messpunkteanzahl über die Teilmodellanzahl dargestellt. Es zeigt sich, dass die zusätzlichen Re-

gressoren des Umgebungsdrucks eine deutlich bessere Approximation in den einzelnen Teilmodellen ermöglichen und die Modellierung bei gleicher Modellfehlervorgabe mit deutlich weniger Teilmodellen auskommt. Die Vorgabe einer höheren Messpunkteanzahl pro Teilmodell ermöglicht im Ergebnis zwar einen minimal verringerten Validierungsfehler, bringt jedoch als deutlichen Nachteil einen stark erhöhten Messdatenbedarf mit sich.

Insgesamt bietet die wissensbasierte Auswahl der Regressoren einen guten Kompromiss zwischen Teilmodellanzahl und Interpretierbarkeit der einzelnen Teilmodelle. Die iterative Versuchsplanung ermöglicht somit, bei einer moderaten Anzahl von Messpunkten eine Modellierung mit geringem Varianzfehler und kann auch vorhandene Datensätze einbinden und effektiv nutzen. Insbesondere dieser Punkt ermöglicht ein mehrstufiges Vorgehen in der Modellierung ohne zusätzlichen Messaufwand.

5.3. Zusammenfassung

Im ersten Teil des vorangegangenen Kapitels wurden am Beispiel einer Testfunktion die Möglichkeiten der Parametrierung einer ILMON-Modellierung aufgezeigt und die Ergebnisse mit denen einer LOLIMOT- und GMR-Modellierung verglichen. Insbesondere konnte der im Vergleich zur LOLIMOT-Modellierung effektivere Partitionierungsalgorithmus und der daraus resultierende geringere Teilmodellbedarf dargestellt werden. Ebenso wurde die Integration von Vorwissen in den Modellierungsprozess und die Interpretierbarkeit der resultierenden Modellstruktur beispielhaft aufgezeigt. Mit der Erweiterung der lokalen Funktionen auf quadratische Regressoren konnte eine Möglichkeit demonstriert werden, die Flexibilität der lokalen Komponenten zu erhöhen, um die benötigte Teilmodellanzahl des resultierenden Modells zu reduzieren.

Mit dem Vergleich der ILMON-Modellierung mit gleichverteilten Datensätzen gegenüber der iterativen Versuchsplanung konnten die Vorteile der Versuchsplanung bezüglich der benötigten Messdatenanzahl und der Vermeidung von Überanpassungen des Modells aufgezeigt werden. Das Beispiel machte auch deutlich, dass mit der iterativen Versuchsplanung der Trainingsfehler die Modellgüte sehr gut abgebildet wird und gegebenenfalls auf eine Überprüfung mittels eines Validierungsdatensatz verzichtet werden kann.

Im zweiten Teil wurde der Modellierungs- und Versuchsplanungsalgorithmus in eine Toolkette zur Motorvermessung eingebunden und mit dieser die Füllung eines Verbrennungsmotors mit variablem Ventiltrieb in Abhängigkeit von 6 Eingangsgrößen modelliert. Als Versuchsmotor diente eine Motorsimulation unter der Software GT-ISE der Firma Gamma Technology. An Hand dieses praxisnahen Beispielprozesses sind mehrere Modellierungen mit unterschiedlichen Parametrierungen durchgeführt und verglichen worden. Dabei konnten insbesondere die Eigenschaften bei unterschiedlich definierten lokalen Komponenten dargestellt werden. Mit der stufenweise Modellierung unter Anpassung der Regressorenanzahl in den Teilmodellen ist eine praktisch orientierte Möglichkeit der Kompromissfindung zwischen Teilmodell- und Regressorenanzahl demonstriert worden.

6. Zusammenfassung

Ziel dieser Arbeit war die Entwicklung einer Methodik zur datenbasierten Modellierung von komplexen Prozessen für die Anwendung in der Motorenentwicklung. Ein besonderer Fokus lag hierbei auf dem Einsatz in aktuellen Motorsteuergeräten. Zu diesem Zweck wurden die typischen Beschränkungen und Problemfelder datenbasierter Modellierungsverfahren dargestellt sowie auf dieser Basis sowohl gewünschte als auch notwendige Kriterien für das genannte Anwendungsgebiet formuliert.

Es wurde herausgearbeitet, dass mit der erhöhten Validierungs- und Sicherheitsanforderung und dem durch die Komplexität der Prozesse begründeten hohen Datenbedarf, zwei in ihrer Auswirkung diametrale Gegebenheiten vorliegen. Dieser Widerspruch bedingt, dass eine datenbasierte Anwendung im Umfeld der Motorenentwicklung nur über eine gute Interpretierbarkeit der Modellstruktur darstellbar ist. Neben dieser Hauptforderung lag ein weiterer Entwicklungsschwerpunkt auf der ressourcenschonenden Berechnung der Ausgangsgleichung, die für eine Umsetzung auf aktuellen Motorsteuergeräten essentiell ist.

Auf diesen Anforderungen basierend wurde ein Kriterienkatalog definiert, der die genannten Forderungen auf die strukturellen und mathematischen Eigenschaften datenbasierter Methoden abbildet. Er diente weiterhin als Grundlage für die Bewertung verschiedener populärer Verfahren sowie dem Entwurf eines Modellierungsverfahren, das diesen Kriterien genügt.

ILMON-Modellstruktur und Optimierungsalgorithmus Im Rahmen dieser Arbeit wurde eine neuartige interpretierbare Modellstruktur entwickelt, die auf die allgemeine Basisfunktionsdefinition aufsetzt. Sie ist als lokal unabhängiges Modellnetz definiert, für das eine spezielle multivariate Basisfunktion entworfen wurde. Diese lässt eine einfache, intuitive und wissensbasierte Validierung zu und erlaubt lokale Anpassungen. Zusammen mit der Entwicklung eines optimalen Partitionierungsalgorithmus konnte so das ILMON-Modellierungsverfahren vorgestellt werden, dessen wichtigste Eigenschaften sich wie folgt zusammenfassen lassen:

- lokal-lineare Struktur mit einer optionalen Erweiterung auf einzelne, quadratische Regressoren und damit einer Erhöhung der Flexibilität der lokalen Komponenten
- iterative Strukturoptimierung mit einer optimalen Bestimmung der Teilung unter Vorgabe eines frei wählbaren Modellgütekriteriums
- Optimierung vorhandener Teilmodellgrenzen in jedem Iterationszyklus
- hierarchische, achsenorthogonale Partitionierung des Eingangsraumes mit einer möglichen Integration von Prozesswissen
- Sperren bestimmter Teilmodelle für eine weitere Strukturoptimierung
- Lokalität der Parameter und genau abgegrenzte Übergangszonen, die eine unabhängige Anpassung und Validierung der Teilmodelle zulassen

- einfaches Interpolationsverhalten
- variabel definierbares Extrapolationsverhalten
- sehr gute Interpretierbarkeit der optimierten Modellstruktur
- ressourcenschonende Berechnung der Ausgangsgleichung auf der typischen mikrocontrollerbasierten Hardware eines Motorsteuergerätes

Mit diesen Eigenschaften eignet sich der ILMON-Algorithmus gut für viele Modellierungsaufgaben auf dem Gebiet der Motorenentwicklung. Insbesondere an Prozessen, in denen sich stark nichtlineare Verläufe mit großen weitgehend linearen oder quadratischen Bereichen abwechseln, kann das Verfahren seine Stärken ausspielen. Nachteilig sind der hohe Rechenaufwand für die Optimierung der Modellstruktur als auch der große Teilmodellbedarf bei achsenschrägen Nichtlinearitäten.

Zur einfachen grafischen Validierung der optimierten Modellstrukturen wurde im Rahmen dieser Arbeit ein Werkzeug unter MATLAB erstellt, mit dem eine intuitive Anzeige einzelner Bereiche hochdimensionaler Modelle möglich ist und somit die guten Interpretationseigenschaften der Struktur ergänzt.

Iterative Versuchsplanung Die große Abhängigkeit datenbasierter Optimierungsverfahren von der Qualität der Messdaten macht es notwendig, besonderes Augenmerk auf die Ermittlung der Versuchsdaten zu legen. Dieser Tatsache Rechnung tragend wurde in dieser Arbeit eine optimal auf die ILMON-Struktur aufbauende iterative Versuchsplanung vorgestellt, deren sequentielle Ermittlung der Versuchspunkte den Vorteil bietet, dass mit der Modellierung wachsende Prozesswissen zur optimalen Platzierung der nächsten Messpunkte zu nutzen. Im Zusammenspiel mit der iterativen Partitionierung der ILMON-Modellierung realisiert der Algorithmus eine automatische Erhöhung der Messpunktedichte in den stark nichtlinearen Bereichen des Prozesses und erkennt lineare Bereiche, die keiner weiteren Vermessung bedürfen.

Eine wesentliche Eigenschaft des vorgestellten Planungsalgorithmus' ist die optimale Auswahl der Versuchspunkte, sowohl für die Schätzung der linearen Parameter der lokalen Komponenten als auch der Partitionierungsparameter zur iterativen Erhöhung der Modellkomplexität. Die resultierende Datenbasis ermöglicht eine sichere Bewertung der Modellgüte auf Grundlage der Trainingsdaten und gestattet einen weitgehenden Verzicht auf die Aufnahme eines Validierungsdatensatzes zu Überprüfung des Modells. Damit reduziert die hier vorgestellte Versuchsplanung den Messaufwand erheblich.

Die Verteilung der Messpunkte in den einzelnen Teilmodellen erfolgt auf Basis der Partitionierung raumfüllend nach dem Maximin-Kriterium mit Mahalanobis-Distanz. Der Nachteil der schlechten Projektionseigenschaften und die Realisierungsprobleme auf Grund der hohen theoretischen Messpunkteanzahl konnte durch eine geeignete Vorfilterung der möglichen Messpunkte beseitigt werden. Praktische Anforderungen, wie nichtäquidistante Rasterungen der Eingangsgrößen oder Ausschlüsse von komplexen, nicht anfahrbaren Bereichen im Eingangsraum können in der vorgestellten Versuchsplanung berücksichtigt werden. Geplante jedoch nicht ausführbare Messungen werden online als gesperrte Bereiche übernommen und nachfolgend in die weitere Versuchsplanung einbezogen.

Zusammengefasst weist der im Rahmen dieser Arbeit entwickelte Versuchsplanungsalgorithmus folgende wesentliche Eigenschaften auf:

- Die Versuchsplanung und -durchführung erfolgt unabhängig für jedes Teilmodell als gruppierter sequentieller Versuchsplan.
- Starke Nichtlinearitäten werden gegenüber weitgehend linearen Bereichen automatisch mit einer höheren Messpunktedichte vermessen. Dies erfolgt ohne Kenntnisse und Vorgaben zum Ausgangsgrößenverlauf.
- Die Versuchspunkte werden innerhalb jeden Teilmodells gleichverteilt und raumfüllend über dem Eingangsraum sowie gleichverteilt über die Rasterstellen der einzelnen Stellgrößen platziert.
- Rasterstellen in den einzelnen Stellgrößen können frei vorgegeben werden.
- Begrenzungen und Ausschlüsse im Versuchsraum können als Funktion der Eingangsgrößen frei definiert werden.
- Versuchspunkte, die in der Versuchsdurchführung nicht angefahren werden können, werden online gesperrt und in der weiteren Versuchsplanung berücksichtigt.

Realisierung Die Eigenschaften der ILMON-Struktur wurden beispielhaft an einer speziellen Testfunktion demonstriert. Neben den Einflüssen der Parameterwahl konnten so die Auswirkungen verschiedener Datensatzgrößen, die Integration von Prozesswissen, die Interpretierbarkeit der Modellstruktur sowie die Unterschiede in den Modellierungen mit linearen und quadratischen lokalen Komponenten dargestellt werden. Mit einem Vergleich der Modellierung per LOLIMOT-Algorithmus und GMR wurden die Ergebnisse des in dieser Arbeit entworfenen ILMON-Verfahrens denen populärer, datenbasierter Methoden gegenüber gestellt.

Mit der Modellierung der Füllungserfassung für einen 3,2l-Saugmotor mit variablem Ventiltrieb wurde ein praktisch relevantes Beispiel umgesetzt. Die Simulation des Motors unter der in der Motorenentwicklung häufig genutzten physikalisch-modellbasierten Modellierungsumgebung GT-ISE erlaubte eine umfangreiche Aufnahme von Validierungsdaten und den Vergleich mehrerer Parametervariationen in den Modellierungsdurchläufen. Neben der Darstellung der Eigenschaften bei unterschiedlich definierten lokalen Komponenten wurde mit der stufenweisen Modellierung unter Anpassung der Regressoren eine praktisch orientierte Möglichkeit der Kompromissfindung zwischen Teilmodell- und Regressorenanzahl aufgezeigt.

Die Umsetzung diente auch der Demonstration der entwickelten Toolkette. Zur Einbindung eines Versuchsstandes bzw. einer Motorsimulationssoftware in die Versuchsplanung wurde ein Interface unter MATLAB/Simulink erstellt. Dieses enthält neben einer Arbeitspunkterkennung unter Angabe wählbarer Konvergenzkriterien eine automatische Umschaltung der Versuchspunkte. Weiterhin wird hier die Zuordnung der stationären Endwerte der Zielgröße zu den jeweiligen Eingangsgrößen realisiert. Die Funktionen zum Erstellen und Optimieren eines ILMON-Modells wurden ebenfalls unter MATLAB programmiert.

A. Anhang

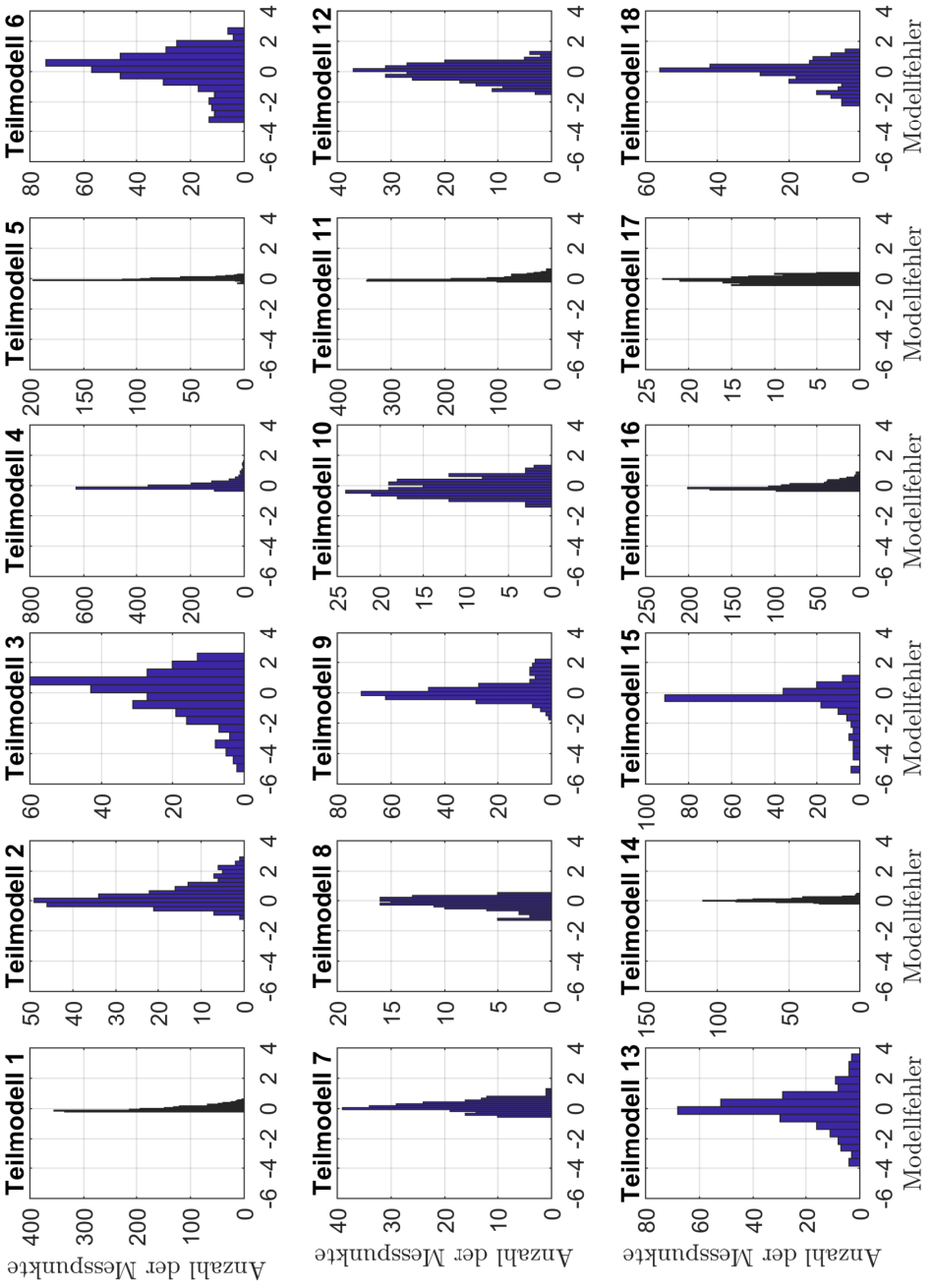

Abbildung A.1.: Histogramme aller Teilmodelle der ILMON-Modellierung über $N = 700$ Messpunkte.

Literatur

[1] S. Immen, „Jahresbericht des Kraftfahrtbundesamts 2013/2014“, Kraftfahrt Bundesamt, Flensburg, Techn. Ber., 2015.

[2] „World Energy Outlook 2016“, International Energy Agency, Techn. Ber., 2016.

[3] „Die Zukunft des Verbrennungsmotors / Bewertung der dieselmotorischen Situation“, Wissenschaftlichen Gesellschaft für Kraftfahrzeug und Motorentechnik e.V. (WKM), Berlin, Techn. Ber., 2017.

[4] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton university press, 1961.

[5] G. Seni und J. Elder, *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan und Claypool Publishers, 2010.

[6] G. James, D. Witten, T. Hastie und R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[7] C. Sammut und G. I. Webb, *Encyclopedia of Machine Learning*, 1st. Springer Publishing Company, Incorporated, 2011.

[8] O. Nelles, *Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models*, 2001. Aufl. Berlin, Heidelberg: Springer, 2001.

[9] S. Geman, E. Bienenstock und R. Doursat, „Neural Networks and the Bias/Variance Dilemma“, *Neural Comput.*, Jg. 4, Nr. 1, S. 1–58, Jan. 1992.

[10] R. Kohavi, „A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection“, S. 1137–1143, 1995.

[11] B. Efron, *Bootstrap methods: another look at the jackknife*. Springer, 1992, S. 569–593.

[12] B. Efron und R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[13] B. Efron und R. Tibshirani, „Improvements on Cross-Validation: The 632+ Bootstrap Method“, *Journal of the American Statistical Association*, Jg. 92, Nr. 438, S. 548–560, 1997.

[14] K. Burnham und D. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer New York, 2007.

[15] H. Akaike, „Information theory as an extension of the maximum likelihood principle“, Jg. Proceedings of the 2nd International Symposium on Information Theory, B. N. Petrov and F. Csaki, Eds.,pp. 267–281, Akademiai Kiado, Budapest, Hungary, 1973.

[16] H. Akaike, „A new look at the statistical model identification“, *IEEE transactions on automatic control*, Jg. 19, Nr. 6, S. 716–723, 1974.

[17] G. Schwarz u. a., „Estimating the dimension of a model“, *The annals of statistics*, Jg. 6, Nr. 2, S. 461–464, 1978.

[18] K. Aho, D. Derryberry und T. Peterson, „Model selection for ecologists: the worldviews of AIC and BIC“, *Ecology*, Jg. 95, Nr. 3, S. 631–636, 2014.

[19] Y. Yang, „Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation“, *Biometrika*, Jg. 92, Nr. 4, S. 937–950, 2005.

[20] Y. Fang, „Asymptotic equivalence between cross-validations and Akaike information criteria in mixed-effects models“, *Journal of data science*, Jg. 9, Nr. 1, S. 15–21, 2011.

[21] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson und A. Juditsky, „Nonlinear black-box modeling in system identification: a unified overview“, *Automatica*, Jg. 31, Nr. 12, S. 1691–1724, 1995.

[22] T. A. Johansen und R. Murray-Smith, „The operating regime approach to nonlinear modelling and control“, *Multiple model approaches to modelling and control*, Jg. 1, S. 3–72, 1997.

[23] M. Kortmann, „Die Identifikation nichtlinearer Ein- und Mehrgrössensysteme auf der Basis nichtlinearer Modellansätze.“, in *Fortschritt-Berichte VDI, Reihe 8, 177*.

[24] G. Birkhoff, „The algebra of multivariate interpolation“, *Constructive approaches to mathematical models*, S. 345–363, 1979.

[25] T. Gasca Mariano and Sauer, „Polynomial interpolation in several variables“, *Advances in Computational Mathematics*, Jg. 12, Nr. 4, S. 377, 2000.

[26] P. J. Olver, „On multivariate interpolation“, *Studies in Applied Mathematics*, Jg. 116, Nr. 2, S. 201–240, 2006.

[27] H. Sequenz, K. Keller und R. Isermann, „Zur Identifikation mehrdimensionaler Kennfelder für Verbrennungsmotoren“, *Automatisierungstechnik*, Jg. 60, S. 344–351, 2012.

[28] D. Kriesel, *A Brief Introduction to Neural Networks*. 2007. Adresse: `http://www.dkriesel.com`.

[29] K. Hornik, M. Stinchcombe und H. White, „Multilayer feedforward networks are universal approximators“, *Neural networks*, Jg. 2, Nr. 5, S. 359–366, 1989.

[30] D. Anguita, G. Parodi und R. Zunino, „Speed improvement of the backpropagation on current-generation workstations“, in *WCNN'93, Portland: World Congress on Neural Networks, July 11-15, 1993, Oregon Convention Center, Portland, Oregon*, Bd. 1, 1993.

[31] M. Buhmann, „Radial basis functions“, *Acta Numerica*, Jg. 9, S. 1–38, 2000.

[32] H. Wendland, *Scattered data approximation*. Cambridge university press, 2004, Bd. 17.

[33] D. S. Broomhead und D. Lowe, „Radial basis functions, multi-variable functional interpolation and adaptive networks“, DTIC Document, Techn. Ber., 1988.

[34] J. Park und I. W. Sandberg, „Universal approximation using radial-basis-function networks“, *Neural computation*, Jg. 3, Nr. 2, S. 246–257, 1991.

[35] S. Billings und W. Voon, „Piecewise linear identification of non-linear systems“, *International journal of control*, Jg. 46, Nr. 1, S. 215–235, 1987.

[36] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson und A. Juditsky, „Nonlinear black-box modeling in system identification: a unified overview“, *Automatica*, Jg. 31, Nr. 12, S. 1691–1724, 1995.

[37] T. Kavli und E. Weyer, „ASMOD (Adaptive Spline Modelling of Observation Data): some theoretical and experimental results“, in *IEE Colloquium on Advances in Neural Networks for Control and Systems*, Mai 1994, S. 3/1–3/7.

[38] J. H. Friedman, „Multivariate Adaptive Regression Splines“, *Ann. Statist.*, Jg. 19, Nr. 1, S. 1–67, März 1991.

[39] A. J. Smola und B. Schölkopf, „A tutorial on support vector regression“, *Statistics and computing*, Jg. 14, Nr. 3, S. 199–222, 2004.

[40] B. Schölkopf und C. J. Burges, *Advances in kernel methods: support vector learning*. MIT press, 1999.

[41] T. Howley und M. G. Madden, „An Evolutionary Approach to Automatic Kernel Construction“, S. 417–426, 2006.

[42] H. Sung, „Gaussian Mixture Regression and Classification“, phd thesis, RICE UNIVERSITY, Houston, Texas, 2004.

[43] G. McLachlan und D. Peel, *Finite Mixture Models*. Wiley, 2004.

[44] K. Mardia, J. Kent und J. Bibby, *Multivariate analysis*. Academic Press, 1979.

[45] A. Dempster, N. M. Laird und D. B. Rubin, „Maximum Likelihood From Incomplete Data Via The EM algorithm“, Jg. 39, S. 1–38, Jan. 1977.

[46] Z. Hu, „Initializing the EM Algorithm for Data Clustering and Sub-population Detection“, Dissertation, Ohio State University, 2015.

[47] J. Peña, J. Lozano und P. Larrañaga, „An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm“, *Pattern Recogn. Lett.*, Jg. 20, Nr. 10, S. 1027–1040, Okt. 1999.

[48] D. Steinley und M. J. Brusco, „Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques“, *J. Classif.*, Jg. 24, Nr. 1, S. 99–121, Juni 2007.

[49] M. E. Celebi, H. A. Kingravi und P. A. Vela, „A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm“, *CoRR*, Jg. abs/1209.1960, 2012.

[50] K. P. Burnham und D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Jan. 2004, Bd. 67.

[51] A. Cutler und M. P. Windham, *Information-Based Validity Functionals for Mixture Analysis*. Jan. 1994, S. 149–170.

[52] R. Murray-Smith und T. Johansen, „Local learning in local model networks“, in *Multiple Model Approaches to Modelling and Control*, Ser. Taylor and Francis systems and control book series, R. Murray-Smith und T. Johansen, Hrsg., London, UK: Taylor und Francis, 1997, S. 185–210.

[53] O. Nelles und R. Isermann, „Basis function networks for interpolation of local linear models“, in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, Bd. 1, Nov. 1996, 470–475 vol.1.

[54] R. Murray-Smith und T. A. Johansen, „Local learning in local model networks“, 1995.

[55] L. Breiman, „Hinging hyperplanes for regression, classification, and function approximation“, *IEEE Transactions on Information Theory*, Jg. 39, Nr. 3, S. 999–1013, 1993.

[56] R. Shorten und R. Murray-Smith, „Side-effects of normalising basis functions in local model networks“, Taylor and Francis systems and control book series, R. Murray-Smith und T. Johansen, Hrsg., S. 211–229, 1997.

[57] S. M. Savaresi und D. L. Boley, „On the performance of bisecting K-means and PDDP“, in *Proceedings of the 2001 SIAM International Conference on Data Mining*, S. 1–14.

[58] S. Jakubek und N. Keuth, „A local neuro-fuzzy network for high-dimensional models and optimization“, *Engineering applications of artificial intelligence*, Jg. 19, Nr. 6, S. 705–717, 2006.

[59] V. Estivill-Castro, „Why So Many Clustering Algorithms: A Position Paper“, *SIGKDD Explor. Newsl.*, Jg. 4, Nr. 1, S. 65–75, Juni 2002.

[60] J. Han, J. Pei und M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[61] A. K. Jain, M. N. Murty und P. J. Flynn, „Data Clustering: A Review“, *ACM Comput. Surv.*, Jg. 31, Nr. 3, S. 264–323, Sep. 1999.

[62] J. A. Nelder und R. Mead, „A simplex method for function minimization“, *The computer journal*, Jg. 7, Nr. 4, S. 308–313, 1965.

[63] J. C. Lagarias, J. A. Reeds, M. H. Wright und P. E. Wright, „Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions“, *SIAM Journal on Optimization*, Jg. 9, Nr. 1, S. 112–147, 1998.

[64] R. Murray-Smith, „A local model network approach to nonlinear modelling“, Diss., University of Strathclyde, 1994.

[65] W. S. Cleveland und S. J. Devlin, „Locally weighted regression: an approach to regression analysis by local fitting“, *Journal of the American statistical association*, Jg. 83, Nr. 403, S. 596–610, 1988.

[66] L. Breiman, J. Friedman, R. Olshen und C. Stone, *Classification and Regression Trees*. Chapman & Hall, 1984.

[67] T. Runkler und J. Bezdek, „Polynomial membership functions for smooth first order Takagi-Sugeno systems“, *GI-Workshop Fuzzy-Neuro-Systeme: Computaional Intelligence*, S. 382–387, 1997.

[68] R. Babuska, C. Fantuzzi, U. Kaymak und H. Verbruggen, „Improved inference for Takagi-Sugeno models“, in *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, IEEE, Bd. 1, 1996, S. 701–706.

[69] O. Nelles und M. Fischer, „Lokale Linearisierung von Fuzzy-Modellen.“, *Automatisierungstechnik*, S. 217–223, 1999.

[70] F. Harrell, *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Springer, 2001.

[71] R. Tibshirani, „Regression shrinkage and selection via the lasso: a retrospective“, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Jg. 73, Nr. 3, S. 273–282, 2011.

[72] K. Siebertz, D. van Bebber und T. Hochkirchen, *Statistische Versuchsplanung, Design of Experiments*. Springer Verlag, 2010.

[73] W. Kleppmann, *Versuchsplanung: Produkte und Prozesse optimieren*, 9. Auflage. Carl Hanser Verlag GmbH & Co. KG, 2016.

[74] M. Toyoda und T. Shen, „D-optimization based mapping calibration of air mass flow in combustion engines“, in *2016 European Control Conference (ECC)*, 2016, S. 1259–1264.

[75] A. Schreiber, „Elektronisches Management motorischer Fahrzeugantriebe: Elektronik, Modellbildung, Regelung und Diagnose für Verbrennungsmotoren, Getriebe und Elektroantriebe“, in. Wiesbaden: Vieweg+Teubner, 2010.

[76] Fischer, Michael and Röpke, Karsten, „Effiziente Applikation von Motorsteuerungsfunktionen für Ottomotoren“, *MTZ - Motortechnische Zeitschrift*, Jg. 61, Nr. 9, S. 562–570, 2000.

[77] C. Grundlach, „Entwicklung eines ganzheitlichen Vorgehensmodells zur problemorientierten Anwendung der statistischen Versuchsplanung.“, Dissertation, Universität Kassel, 2004.

[78] S. Boyd und L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004, 384 ff.

[79] M. D. McKay, R. J. Beckman und W. J. Conover, „A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code“, *Technometrics*, Jg. 21, Nr. 2, S. 239–245, 1979.

[80] M. Johnson, L. Moore und D. Ylvisaker, „Minimax and maximin distance designs“, *Journal of Statistical Planning and Inference*, Jg. 26, Nr. 2, S. 131–148, 1990.

[81] C. Ma und K.-T. Fang, „A new approach to construction of nearly uniform designs“, *International Journal of Materials and Product Technology*, Jg. 20, Nr. 1-3, S. 115–126, 2004.

[82] K.-T. Fang, R. Li und A. Sudjianto, *Design and modeling for computer experiments*. CRC Press, 2005.

[83] K.-T. Fang, „Theory, method and applications of the uniform design“, *International journal of reliability, quality and safety engineering*, Jg. 9, Nr. 04, S. 305–315, 2002.

[84] P. Bauer, V. Scheiber und F. X. Wohlzogen, *Sequentielle statistische Verfahren*. Gustav Fischer Verlag, 1986.

[85] J. O. Berger, „Sequential analysis“, in *The New Palgrave Dictionary of Economics*, S. N. Durlauf und L. E. Blume, Hrsg., Basingstoke: Palgrave Macmillan, 2008.

[86] A. C. Atkinson und V. Fedorov, „The design of experiments for discriminating between two rival models“, *Biometrika*, Jg. 62, Nr. 1, S. 57–70, 1975.

[87] B. Settles, „Active learning literature survey“, *University of Wisconsin, Madison*, Jg. 52, Nr. 55-66, S. 11, 2010.

[88] D. A. Cohn, „Neural network exploration using optimal experiment design“, *Advances in neural information processing systems*, S. 679–679, 1994.

[89] G. Schohn und D. Cohn, „Less is More: Active Learning with Support Vector Machines“, in *Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, S. 839–846.

[90] D. Arthur und S. Vassilvitskii, „K-means++: The Advantages of Careful Seeding“, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Ser. SODA '07, New Orleans, Louisiana: Society for Industrial und Applied Mathematics, 2007, S. 1027–1035.

[91] S. Bernhardt, „Ladungswechselrechnung“, in *Handbuch Verbrennungsmotor*, R. van Basshuysen und F. Schäfer, Hrsg., Springer Vieweg, 2015, Kap. 10.2, S. 488–491.

[92] G. P. Merker, C. Schwarz, G. Stiesch und F. Otto, *Simulating Combustion*. Berlin, Heidelberg: Springer DE, 2006.

[93] R. Beckmann, „Beitrag zur exakten Füllungssteuerung am aufgeladenen Ottomotor“, Dissertation, Universität Rostock, 2015.

[94] B. Kolewe, A. Haghani, R. Beckmann, R. Noack und T. Jeinsch, „Data-driven estimation of air mass using Gaussian mixture regression“, in *Proc. of IEEE International Symposium on Industrial Electronics Istanbul 1014*, 2014.

[95] J. B. Heywood, *Internal combustion engine fundamentals* -. New York: McGraw-Hill, 1988.

[96] J. Grizzle, J. Cook und W. Milam, „Improved Cylinder Air Charge Estimation for Transient Air Fuel Ratio Control“, in *Proceedings of the American Control Conference*, 1994.

[97] N. R. Draper und H. Smith, *Applied regression analysis*. John Wiley & Son, 1998.

[98] T. Johnsson, „A procedure for stepwise regression analysis“, *Statistical Papers*, Jg. 33, Nr. 1, S. 21–29, Nov. 1992.

[99] B. Kolewe, A. Haghani, R. Beckmann und T. Jeinsch, „Gaussian mixture regression and local linear network model for data-driven estimation of air mass“, *IET Control Theory Applications*, Jg. 9, Nr. 7, S. 1083–1092, 2015.

[100] A. C. Rencher und F. C. Pun, „Inflation of R2 in Best Subset Regression“, *Technometrics*, Jg. 22, Nr. 1, S. 49–53, 1980.

[101] C. M. Hurvich und C. Tsai, „The Impact of Model Selection on Inference in Linear Regression“, *The American Statistician*, Jg. 44, Nr. 3, S. 214–217, 1990.

[102] E. B. Roecker, „Prediction Error and Its Estimation for Subset-Selected Models“, *Technometrics*, Jg. 33, Nr. 4, S. 459–468, 1991.