

Lehrstuhl für Elektrische Antriebssysteme
Technische Universität München

Identifikation von nichtlinearen mechatronischen Systemen auf der Basis von Volterra–Reihen

Stefan Hofmann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und
Informationstechnik der Technischen Universität München zur Erlangung
des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Georg Färber

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Dr.-Ing. h.c. Dierk Schröder
2. Univ.-Prof. Dr.-Ing. Heinz Unbehauen, em.
Ruhr-Universität Bochum

Die Dissertation wurde am 23. April 2003 bei der Technischen Universität München
eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik
am 29. Juli 2003 angenommen.

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der Identifikation von nichtlinearen mechatronischen Systemen. Die Modellbildung mechatronischer Systeme ist zeitaufwändig, da diese sehr komplex und in ihrer Struktur in der Regel nur teilweise bekannt sind. Charakteristisch für mechatronische Systeme ist, dass einerseits die Struktur des mechanischen Teils aufgrund der Wirkungskette Moment – Geschwindigkeit – Position gegeben ist, andererseits jedoch die Struktur des elektrischen Teils nur mit hohem Aufwand bestimmt werden kann.

Ausgehend vom Neuronalen Beobachter wird ein Identifikationsverfahren entwickelt, das es ermöglicht unbekannte dynamische Teilsysteme innerhalb eines bekannten Teilsystems zu identifizieren. Die Identifikation der dynamischen Teilsysteme erfolgt auf Basis von Volterra-Reihen, die die Einbringung von Vorwissen in Form einer blockorientierten Modellstruktur erlauben. Die Verbindung mit den strukturierten rekurrenten Neuronalen Netzen ermöglicht zusätzlich die Identifikation von unbekannten linearen Parametern des in seiner Struktur bekannten Teilsystems.

Das neue Identifikationsverfahren baut somit eine Brücke zwischen einer exakten White-Box-Modellierung und der Modellierung des Ein-/Ausgangsverhaltens durch Black-Box-Verfahren.

Abstract

The objective of this thesis is the identification of nonlinear mechatronic systems. Due to the fact, that mechatronic systems are very complex and their structures are normally only known in parts, modeling such systems is time-consuming. Typical for mechatronic systems is, that on the one hand the structure of the mechanical subsystem is completely available due to the functional chain torque – velocity – position and on the other hand the structure of the electrical subsystem can be determined however not without high effort.

Based on the neural observer an identification method is developed, which allows the determination of unknown dynamic subsystems within a well-known subsystem. The identification algorithms are derived from the Volterra series and enable the contribution of prior knowledge in the form of a block-oriented structure. Additionally unknown linear parameters of the subsystem with the well-known structure can be identified by the combination with structured recurrent neural networks.

Consequently the new identification method builds a bridge between accurate white-box-modeling techniques and modeling of the input/output behavior by black-box methods.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als Wissenschaftlicher Assistent am Lehrstuhl für Elektrische Antriebssysteme der Technischen Universität München.

Mein besonderer Dank gilt dem Leiter des Lehrstuhls, Herrn Prof. Dr.-Ing. Dr.-Ing. h.c. Dierk Schröder, für die fortwährende und anregende Unterstützung, die wesentlich zum Gelingen dieser Arbeit beigetragen hat. Er gab mir die Möglichkeit, auf einem innovativen Forschungsgebiet unter hervorragenden Arbeitsbedingungen tätig zu sein.

Für die Übernahme des Koreferats und das entgegengebrachte Interesse an dieser Arbeit danke ich Herrn Prof. em. Dr.-Ing. Heinz Unbehauen; für die Übernahme des Prüfungsvorsitzes gebührt mein Dank Herrn Prof. Dr.-Ing. Georg Färber.

Danken möchte ich ebenso allen Kolleginnen und Kollegen des Lehrstuhls für die freundschaftliche Zusammenarbeit. Besonderer Dank gilt den Kollegen der „Laterne“ sowie Herrn Dr.-Ing. Peter Havel. Die gute Atmosphäre und die zahlreichen offenen Diskussionen waren für mich fachlich wie menschlich eine Bereicherung. Für die kompetente Hilfe bei der Textkorrektur bedanke ich mich insbesondere bei Frau Anne-Kathrin Knapp, Herrn Dipl.-Ing. Christian Hintz und Herrn Dipl.-Ing. Thomas Treichl.

Nicht vergessen möchte ich alle Studenten und Diplomanden, die stets engagiert zum Gelingen dieser Arbeit beigetragen haben.

Meinen Eltern gebührt mein besonderer Dank für die langjährige Förderung und Ermöglichung des Studiums. Ihnen widme ich diese Arbeit.

München, im Juli 2003

Stefan Hofmann

Inhaltsverzeichnis

1	Einführung	1
1.1	Stand der Technik	2
1.2	Motivation und Zielsetzung	3
1.3	Gliederung der Arbeit	5
2	Approximation statischer Nichtlinearitäten	7
2.1	Aufbau von Neuronalen Netzen	7
2.2	Multi-Layer-Perceptron-Network	11
2.3	Radial-Basis-Function-Network	13
2.4	General-Regression-Neural-Network	15
2.5	Harmonic-Activation-Neural-Network	16
2.6	LOcal-LInear-MODEl-Tree	18
2.7	Parameteradaptionsverfahren	20
2.7.1	Least-Squares-Algorithmus	22
2.7.2	Gradientenabstiegsverfahren	23
2.8	Kurzzusammenfassung	24
3	Identifikation nichtlinearer Systeme	25
3.1	Modellbildung und Identifikation	25
3.2	Identifikation linearer dynamischer Systeme	29
3.2.1	Modelle mit Ausgangsrückkopplung	31
3.2.1.1	Autoregressive with Exogenous Input Model	31
3.2.1.2	Output Error Model	32
3.2.2	Modelle ohne Ausgangsrückkopplung	33
3.2.2.1	Finite Impulse Response Model	34
3.2.2.2	Orthonormal Basis Function Model	35
3.3	Identifikation nichtlinearer dynamischer Systeme	39
3.3.1	Klassifikation nichtlinearer dynamischer Systeme	39
3.3.1.1	Nichtlineare Zustandsdarstellung	39
3.3.1.2	Blockorientierte nichtlineare Modelle	40
3.3.1.3	Allgemeine nichtlineare Systembeschreibung	40
3.3.2	Neuronaler Beobachter	41

3.3.2.1	Beobachterentwurf bei messbarem Eingangsraum	42
3.3.2.2	Beobachterentwurf bei nicht messbarem Eingangsraum	46
3.3.3	Nichtlineare Modelle mit Ausgangsrückkopplung	47
3.3.3.1	Nonlinear Autoregressive with Exogenous Input Model	48
3.3.3.2	Nonlinear Output Error Model	49
3.3.4	Nichtlineare Modelle ohne Ausgangsrückkopplung	49
3.3.4.1	Volterra–Funktionalpotenzreihe	50
3.3.4.2	Hammerstein–Modell und Wiener–Modell im Ansatz der Volterra–Funktionalpotenzreihe	52
3.3.4.3	Volterra–Funktionalpotenzreihe mit Basisfunktionen	54
3.4	Kurzzusammenfassung	57
4	Identifikation von nichtlinearen Mehrgrößensystemen	61
4.1	Mehrgrößenerweiterung des Hammerstein–Modells	61
4.1.1	Linearer Eingriff einer Eingangsgröße in das Hammerstein–Modell . .	62
4.1.2	Nichtlinearer Eingriff einer Eingangsgröße in das Hammerstein–Modell	67
4.2	Mehrgrößenerweiterung des Wiener–Modells	72
4.2.1	Linearer Eingriff einer Eingangsgröße in das Wiener–Modell	72
4.2.2	Nichtlinearer Eingriff einer Eingangsgröße in das Wiener–Modell . . .	76
4.3	Identifikation von Mehrgrößensystemen	78
4.3.1	Parameterreduktion durch Einführung orthonormaler Basisfunktio- nen bei Mehrgrößensystemen	79
4.3.2	Stabilitätsbeweis nach Lyapunov	86
4.3.3	Identifikationsbeispiele	91
4.3.3.1	Beispiel 1 — Linear verkoppelte Hammerstein–Modelle	91
4.3.3.2	Beispiel 2 — Nichtlinear verkoppelte Hammerstein–Modelle	96
4.4	Rekonstruktion der blockorientierten Modellstruktur	102
4.4.1	Rekonstruktion von SISO–Modellstrukturen	103
4.4.1.1	Rekonstruktion der Hammerstein–Modellstruktur	103
4.4.1.2	Rekonstruktion der Wiener–Modellstruktur	104
4.4.2	Rekonstruktion von MISO–Modellstrukturen	106
4.4.2.1	Rekonstruktion von linear verkoppelten Modellstrukturen	106
4.4.2.2	Rekonstruktion von nichtlinear verkoppelten Modellstrukturen	107
4.5	Kurzzusammenfassung	108
5	Erweiterung des Verfahrens mit Funktionsapproximatoren	111
5.1	Motivation und Prinzip des erweiterten Verfahrens	111
5.2	Erweiterung am Beispiel des Hammerstein–Modells	112
5.2.1	Beschreibung des Hammerstein–Modells mit Funktionsapproxima- toransatz	112
5.2.2	Vergleich zwischen Polynom– und Funktionsapproximatoransatz . . .	114
5.3	Erweiterung am Beispiel des Wiener–Modells	118
5.3.1	Beschreibung des Wiener–Modells mit Funktionsapproximatoransatz	118

5.3.1.1	Berechnung der Gradienten für das Lerngesetz	120
5.3.1.2	Berechnung der Gradienten bei Einführung von Basisfunktionen . . .	122
5.3.2	Beispiel — Wiener-Modell mit Funktionsapproximatoransatz	123
5.4	Erweiterung auf Mehrgrößensysteme	126
5.4.1	Identifikation von nichtlinear verkoppelten Hammerstein-Modellen .	126
5.4.2	Identifikation von nichtlinear verkoppelten Wiener-Modellen	130
5.5	Kurzzusammenfassung	135
6	Erweiterung des Neuronalen Beobachters	137
6.1	Systeme mit dynamischen Nichtlinearitäten	139
6.2	Beobachterentwurf	141
6.2.1	Beobachterentwurf bei messbarem Eingangsraum der dynamischen Nichtlinearität	142
6.2.2	Beobachterentwurf bei nicht messbarem Eingangsraum der dynami- schen Nichtlinearität	148
6.3	Identifikation von global integrierenden Systemen	151
6.3.1	Identifikation eines global integrierenden Hammerstein-Modells . . .	151
6.3.2	Identifikation eines global integrierenden Wiener-Modells	155
6.4	Simulationsbeispiel für Beobachterentwurf	161
6.5	Kurzzusammenfassung	167
7	Anwendung des Identifikationsverfahrens	169
7.1	Identifikation eines mechatronischen Antriebssystems	169
7.1.1	Identifikation in der Simulationsumgebung	171
7.1.2	Validierung am realen System	176
7.2	Identifikation eines biomechanischen Systems	179
7.2.1	Motivation und Aufgabenstellung	179
7.2.2	Modellbildung des biomechanischen Systems	180
7.2.3	Identifikation in der Simulationsumgebung	186
7.2.3.1	Identifikation der Rekrutierung und der Aktivierungsdynamik	187
7.2.3.2	Identifikation des Hill'schen Modells in Beobachterstruktur	189
7.2.4	Versuchsdurchführung am Klinikum Rechts der Isar	195
7.2.4.1	Versuchsaufbau zur Messung der Kraft	195
7.2.4.2	Versuchsdurchführung	197
7.2.4.3	Identifikationsergebnisse	198
7.3	Kurzzusammenfassung	202
8	Identifikation in strukturierten rekurrenten Netzen	205
8.1	Prinzip der strukturierten rekurrenten Netze	205
8.2	Identifikation mit strukturierten rekurrenten Netzen	207
8.2.1	Identifikation in der Simulationsumgebung	209
8.2.2	Validierung am realen System	214

8.3	Kurzzusammenfassung	217
9	Zusammenfassung und Ausblick	219
A	Definitionen	223
A.1	Definition des $*$ -Operators	223
A.2	Definition des \otimes -Operators	223
B	Technische Daten der Testanlage	224
C	Bilder des Versuchsaufbaus am Klinikum	226
	Bezeichnungen	229
	Literaturverzeichnis	235

1 Einführung

Mechatronische Systeme enthalten mechanische, elektrische und elektronische Komponenten, die komplex miteinander verknüpft sind. Die zunehmende Integration dieser Komponenten erfordert eine ganzheitliche Betrachtungsweise – die Mechatronik. Der Begriff Mechatronik bringt das enge Zusammenwirken von Mechanik, Elektronik und Software zum Ausdruck. Ziel der Mechatronik ist es, das Verhalten eines technischen Systems oder Produktes zu verbessern, indem mit Sensoren Informationen über die Umgebung bzw. über das System selbst erfasst, diese Informationen in Mikroprozessoren verarbeitet und mit Hilfe von Aktoren in ein optimales Verhalten des Systems umgesetzt werden. Der Steuerungs- und Regelungstechnik kommt in diesem Innovationsprozess eine zentrale Rolle zu.

Moderne leistungsfähige Regelungsverfahren basieren zunehmend auf nichtlinearen Prozessmodellen. Die Gewinnung dieser nichtlinearen Prozessmodelle stellt eine große Herausforderung dar. Der Mehraufwand einer nichtlinearen Modellbildung ist oftmals notwendig, da viele technische Prozesse ein stark nichtlineares Verhalten aufweisen, das sich nicht mehr um einen Arbeitspunkt linearisieren lässt.

Bei der Modellbildung kann prinzipiell zwischen der Modellstruktur und den Modellparametern unterschieden werden. Die Modellstruktur bestimmt das generelle Verhalten und die Komplexität des Modells. Das spezielle Verhalten eines Modells mit gegebener Struktur wird durch die Modellparameter festgelegt. Das Ziel einer Identifikation ist es, mit Hilfe gemessener Ein- und Ausgangssignale des Systems ein Modell zu bestimmen, welches das statische und dynamische Verhalten des Prozesses möglichst gut nachbildet. Jede Identifikation setzt sich dabei aus den zwei grundlegenden Schritten Strukturauswahl und Adaption der Parameter zusammen. Dabei ist klar, dass die Adaption der Parameter nur so gut sein kann, wie die ausgewählte Modellstruktur die funktionalen Zusammenhänge des Systems beschreiben kann. So können beispielsweise durch eine lineare Modellstruktur auch nur lineare Prozesseigenschaften beschrieben werden. Somit kommt der Modellstruktur hinsichtlich der Qualität des Modells eine zentrale Rolle zu.

Die Struktur eines Modells resultiert in der Regel aus einer Analyse der physikalischen und geometrischen Zusammenhänge eines Systems. Hierzu muss das Systemverhalten sehr genau untersucht werden, was in der Praxis sehr häufig zu Problemen führt, da die funktionalen Zusammenhänge eines Systems nicht immer offensichtlich sind. Aus diesem Grund wurden in der Vergangenheit Black-Box-Identifikationsverfahren entwickelt, die mit minimalem strukturellem Vorwissen das Ein-/Ausgangsverhalten eines Systems beschreiben. Aufgrund der Tatsache, dass

es sich hierbei um abstrakte mathematische Verfahren handelt, ist keine physikalische Interpretation der identifizierten Modellparameter möglich, was die Akzeptanz solcher Verfahren beim Anwender erschwert. Darüber hinaus liefern Black-Box-Verfahren keine Informationen über interne Systemzustände, die vor allem hinsichtlich eines Reglerentwurfes sehr hilfreich sein können.

In der vorliegenden Arbeit wird ein Identifikationsverfahren vorgestellt, das es einerseits ermöglicht strukturelles Wissen in vorhandenem Umfang einzubringen und es andererseits dem Anwender überlässt, wie umfangreich die exakte physikalische Modellierung des Gesamtsystems sein soll, indem Teilsysteme auch bei unzureichender Strukturkenntnis beschrieben werden können.

1.1 Stand der Technik

Die Identifikation linearer dynamischer Systeme wird seit den sechziger Jahren untersucht und kann als abgeschlossen betrachtet werden. Als Literatur können beispielsweise die Bücher *System Identification* von Soederstroem und Stoica [55], *Theory and Practice of Recursive Identification* von Ljung und Soederstroem [33] sowie *Identifikation dynamischer Systeme 1/2* von Isermann [24, 25] genannt werden.

Mit der Weiterentwicklung in der Rechnertechnik wurden auch die Forschungsaktivitäten auf dem Gebiet der Identifikation von nichtlinearen dynamischen Systemen in den letzten Jahrzehnten intensiviert. Die mathematischen Ansätze zur Beschreibung nichtlinearer dynamischer Systeme reichen teilweise bis in die Anfänge des letzten Jahrhunderts zurück. So veröffentlichte Volterra, die nach ihm benannte Volterra-Funktionalpotenzreihe [66] im Jahre 1930. Auch die Anfänge der Neuronalen Netze gehen bis in die vierziger Jahre zurück. McCulloch und Pitts beschrieben beispielsweise in ihrem Aufsatz *A logical calculus of the ideas immanent in nervous activity* [35] bereits 1943 neurologische Netzwerke und zeigten, dass Neuronale Netze prinzipiell jede arithmetische oder logische Funktion berechnen können.

Seit dem Beginn der achtziger Jahre finden statische Neuronale Netze verstärkt Beachtung in der Literatur. Als die wichtigsten Vertreter sind hier das Radial-Basis-Function-Network bzw. das General-Regression-Neural-Network [56] als dessen Weiterentwicklung und das Multi-Layer-Perceptron-Network [72] zu nennen. Die Erweiterung der statischen Neuronalen Netze mit Speicherelementen führte schließlich zu den dynamischen Neuronalen Netzen, die zur Identifikation des Ein-/Ausgangsverhaltens nichtlinearer dynamischer Systeme eingesetzt werden können. In Abb. 1.1 sind die verschiedenen Verfahren zur Identifikation nichtlinearer dynamischer Systeme im Überblick dargestellt. Dynamische Neuronale Netze benötigen kein oder nur wenig Vorwissen. Sie lassen sich in Netze mit interner und externer Dynamik unterteilen. Zu den Netzen mit interner Dynamik gehören z.B. voll rekurrente Netze (VRN) und partiell rekurrente Netze (PRN). In [71] wurde ein voll rekurrentes Netz vorgestellt, bei dem jede Verbindung zwischen zwei Neuronen einen internen Zustand darstellt. Die Nachteile von voll rekurrenten Netzen

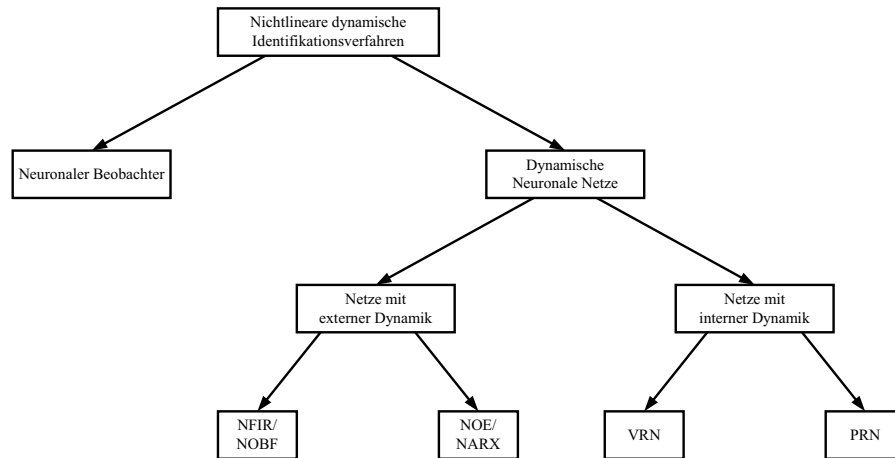


Abb. 1.1: *Nichtlineare Identifikationsverfahren im Überblick*

sind langsame Konvergenz und Stabilitätsprobleme. Partiiell rekurrente Netze besitzen eine strenge Feedforward-Architektur. Sie enthalten eine Kontextschicht, deren Neuronen als interne Zustände dienen. Typische partiiell rekurrente Netze sind das Elman- oder Jordan-Netz [72]. Bei den Netzen mit externer Dynamik handelt es sich prinzipiell um statische Funktionsapproximatoren, die mit externen Filterketten erweitert sind. Grundsätzlich sind dabei Netze mit und ohne Ausgangsrückkopplung zu unterscheiden. Zu den Netzen mit Ausgangsrückkopplung zählt beispielsweise das Time-Delay-Neural-Network [72] in den Grundstrukturen NOE (Nonlinear Output Error) und NARX (Nonlinear AutoRegressive with eXogenous input). Zu den Netzen ohne Ausgangsrückkopplung kann die Volterra-Funktionalpotenzreihe [48] gezählt werden, die eine NFIR-Struktur (Nonlinear Finite Impulse Response) aufweist. Erfolgt eine Komprimierung des Eingangsraumes durch die Einführung orthonormaler Basisfunktionen [30] kann von einer NOBF-Struktur (Nonlinear Orthonormal Basis Function) gesprochen werden. Im Gegensatz zu den dynamischen Neuronalen Netzen werden beim Neuronalen Beobachter [52] statische Neuronale Netze in einem Beobachter verwendet und somit statische Nichtlinearitäten innerhalb eines bekannten dynamischen Systems identifiziert. Dieser Ansatz setzt deutlich mehr Strukturwissen voraus als dynamische Neuronale Netze.

Ein ausführlicherer Überblick zu diesem Forschungsgebiet kann beispielsweise in *Nonlinear System Identification* [41] von Nelles oder in *Simulation Neuronaler Netze* [72] von Zell nachgelesen werden.

1.2 Motivation und Zielsetzung

Zu Beginn der Modellbildung eines nichtlinearen dynamischen Systems stellt sich die Frage nach der Modellstruktur. Die Modellstrukturierung ist für die spätere Qua-

lität des Modells von zentraler Bedeutung und verlangt sehr viel ingenieurmäßiges Einfühlungsvermögen. Prinzipiell können dabei zwei unterschiedliche Wege eingeschlagen werden. Einerseits führt die exakte theoretische Analyse eines Systems zu einem sog. White-Box-Modell. Die Struktur dieses Modells ist genau bekannt, was bedeutet, dass die Modellparameter eine physikalische Bedeutung haben und auch auf interne Systemzustände, die in der Regel nicht messbar sind, zugegriffen werden kann. Der Preis für die hohe Transparenz von diesen Modellen ist der enorme Zeitaufwand, der für diese Art der Modellbildung aufgebracht werden muss. Andererseits führt die sog. experimentelle Analyse eines Systems zu einem Black-Box-Modell, das lediglich das Ein-/Ausgangsverhalten eines Systems beschreibt. Der Vorteil dieser Art der Modellbildung ist, dass sehr wenig Vorkenntnisse über die Struktur eines Systems erforderlich sind und somit der Zeitaufwand für die Modellstrukturierung deutlich geringer ist. Jedoch haben die Modellparameter keinerlei physikalische Bedeutung mehr und der Zugriff auf interne Systemzustände ist nicht möglich.

Im Umfeld der Modellierung von mechatronischen Systemen ist der Neuronale Beobachter entstanden [32, 57]. Der Neuronale Beobachter basiert auf einer nichtlinearen Zustandsdarstellung des Systems und ist somit zur Klasse der White-Box-Modelle zu zählen. Die nichtlineare Zustandsdarstellung erlaubt die Beschreibung statischer Nichtlinearitäten, wie z.B. Reib- und Federkennlinien, wenn ihre strukturelle Wirkung im System bekannt ist. Ist dies der Fall, ermöglicht der Neuronale Beobachter durch die Einbindung von statischen Funktionsapproximatoren die Identifikation dieser statischen Nichtlinearitäten, in dem ansonsten bekannten dynamischen Modell. Der Einsatz des Neuronalen Beobachters hat sich in vielen Anwendungen bewährt [11, 58], nicht zuletzt deswegen, weil die Struktur von mechatronischen Systemen aufgrund der mechanischen Wirkungskette Moment – Geschwindigkeit – Position in der Regel bekannt ist. Dennoch ist aufgrund der Komplexität von mechatronischen Systemen ein Bedarf zur Erweiterung des Neuronalen Beobachters vorhanden.

Ziel der vorliegenden Arbeit ist es, den Neuronalen Beobachter in der Art zu erweitern, dass neben statischen Nichtlinearitäten auch dynamische Nichtlinearitäten und unbekannte lineare Parameter im System identifiziert werden können. Eine dynamische Nichtlinearität kann dabei als ein blockorientiertes nichtlineares dynamisches Teilsystem betrachtet werden, dessen Struktur lediglich in Form von statischen Nichtlinearitäten und dynamischen Übertragungsfunktionen sowie linearen und nichtlinearen Verkopplungen bekannt ist. Die Identifikation von dynamischen Nichtlinearitäten benötigt somit ein deutlich geringeres Maß an strukturellem Vorwissen und gewährt aufgrund der blockorientierten Struktur trotzdem eine gewisse Transparenz der Identifikationsergebnisse. Grundlage für die Identifikation von dynamischen Nichtlinearitäten ist die Volterra-Funktionalpotenzreihe bzw. deren Erweiterung aufgrund der Einführung von orthonormalen Basisfunktionen. Die Identifikation von dynamischen Nichtlinearitäten kann somit als die Integration von NFIR- (Nonlinear Finite Impulse Response) bzw. NOBF-Modellstrukturen (Nonlinear Orthonormal Basis Function) in den Neuronalen Beobachter betrachtet werden. In einem weiteren Schritt soll in dieser Arbeit die Verbindung zu den strukturierten

rekurrenten Neuronalen Netzen aus [16] hergestellt werden. Ein strukturiertes rekurrentes Netz kann als partiell rekurrentes Netz (PRN) betrachtet werden, bei dem das gesamte Vorwissen über die Struktur des Systems eingebracht wird und das es ermöglicht, neben den statischen Nichtlinearitäten im System, auch unbekannte lineare Parameter, wie z.B. Trägheitsmomente, zu identifizieren. Erste vielversprechende Ideen zu den strukturierten rekurrenten Netzen sind in [8] zu finden, jedoch brachte erst die Verbindung mit dem Neuronalen Beobachter die gewünschten Ergebnisse [16].

Durch die Identifikation dynamischer Nichtlinearitäten innerhalb eines strukturierten rekurrenten Neuronalen Netzes können Teilsysteme, die nur sehr schwierig oder extrem aufwändig physikalisch zu modellieren sind, in das strukturierte rekurrente Netz integriert und identifiziert werden. In Abb. 1.2 sind die Erweiterungen des Neuronalen Beobachters noch einmal veranschaulicht.

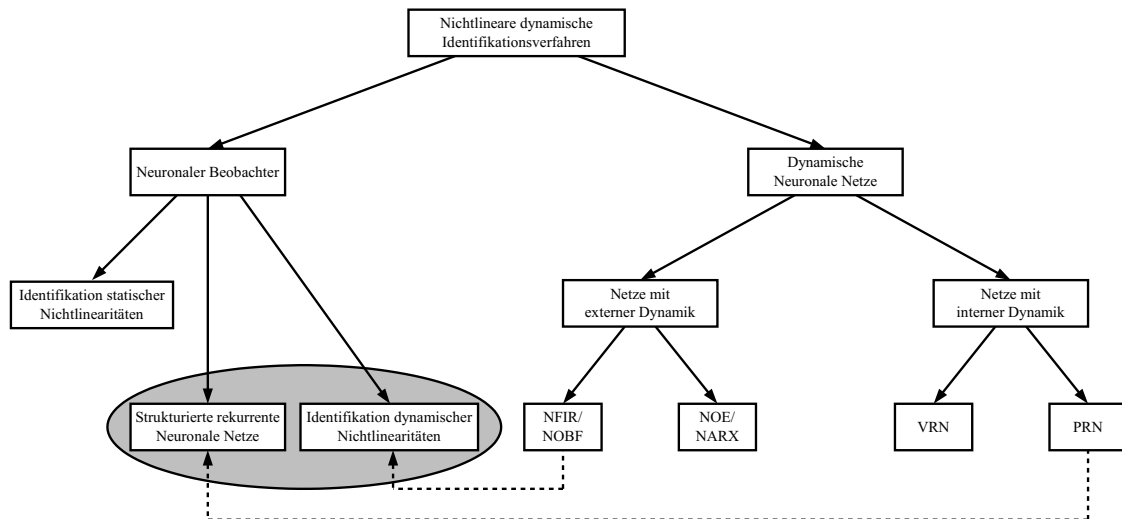


Abb. 1.2: Erweiterungen des Neuronalen Beobachters

Das Identifikationsverfahren, das im Folgenden vorgestellt wird, baut somit eine Brücke zwischen einer exakten White-Box-Modellierung und der Modellierung des Ein-/Ausgangsverhaltens durch Black-Box-Verfahren.

1.3 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich in folgende Abschnitte:

In **Kapitel 2** werden die Grundlagen zur statischen Funktionsapproximation mit Neuronalen Netzen beschrieben. Hierbei werden die in der Regelungstechnik wichtigsten Netztypen hinsichtlich ihrer Topologie, Funktionsweise und ihrer Eigenschaften

vorgestellt. Eine Diskussion der verschiedenen Adaptionsverfahren schließt das Kapitel ab.

Darauf aufbauend werden in **Kapitel 3** Verfahren zur Identifikation dynamischer Systeme vorgestellt. Dabei wird zunächst auf die Identifikation linearer dynamischer Systeme eingegangen, da sie die Grundlage für die Identifikation nichtlinearer dynamischer Systeme bilden. Diese werden im Anschluss ausführlich hinsichtlich ihrer Funktionsweise und Eigenschaften diskutiert. Ein wesentliches Unterscheidungsmerkmal wird hierbei der Grad an strukturellem Vorwissen sein, der in das Identifikationsverfahren eingebracht werden kann.

Die Volterra–Funktionalpotenzreihe, als ein Identifikationsverfahren für nichtlineare dynamische Systeme mit einem Eingang und einem Ausgang, wird in **Kapitel 4** in der Art erweitert, dass auch Mehrgrößensysteme identifiziert werden können. Die grundsätzliche Untersuchung von linear und nichtlinear verkoppelten Hammerstein– und Wiener–Modellen liefert ein umfangreiches Systemverständnis, das es letztendlich ermöglicht, beliebige blockorientierte Systeme mittels der Volterra–Funktionalpotenzreihe zu identifizieren. Die Vorteile einer Identifikation mittels der Volterra–Funktionalpotenzreihe, wie z.B. garantierte Stabilität des Lernens, werden ausführlich diskutiert. Das Problem der hohen Parameteranzahl wird durch die Einführung orthonormaler Basisfunktionen entschärft.

In **Kapitel 5** wird die Volterra–Funktionalpotenzreihe durch die Einführung statischer Funktionsapproximatoren erweitert. Die Vorteile dieser Erweiterung ergeben sich aus den verbesserten Interpolations– und Extrapolationseigenschaften der statischen Funktionsapproximatoren gegenüber den bisher eingesetzten Polynomen.

Die Identifikation von Systemen mit dynamischen Nichtlinearitäten wird in **Kapitel 6** behandelt. Hierzu werden die auf den Volterra–Reihen basierenden Identifikationsverfahren in einen Beobachter integriert, so dass es möglich wird, blockorientierte Systeme innerhalb eines komplexeren Gesamtsystems zu identifizieren.

Die Praxisrelevanz der vorgestellten Theorie wird in **Kapitel 7** veranschaulicht. Am Beispiel eines mechatronischen Antriebssystems und eines biomechanischen Systems wird das weite Anwendungsgebiet des Identifikationsverfahrens verdeutlicht.

In **Kapitel 8** wird die Theorie zu den dynamischen Nichtlinearitäten in ein strukturiertes rekurrentes Neuronales Netz integriert. Damit wird es möglich, Systeme mit unbekanntem dynamischen Nichtlinearitäten und unbekanntem linearen Parametern zu identifizieren. Anhand eines Beispiels wird das Potenzial des neuen Verfahrens zur Identifikation nichtlinearer mechatronischer Systeme verdeutlicht.

Abgeschlossen wird die vorliegende Arbeit mit einer Zusammenfassung der Ergebnisse und einem Ausblick in **Kapitel 9**.

2 Neuronale Netze zur Approximation statischer Nichtlinearitäten

Die Komplexität mechatronischer Systeme erfordert zunehmend den Einsatz von nichtlinearen Approximationsverfahren zur Lösung von Aufgabenstellungen im Bereich der Prädiktion, Simulation, modellbasierten Regelung und Fehlerdiagnose. Die unbekanntenen nichtlinearen Funktionen sind in der Regel nicht oder nur mit großem Aufwand analytisch beschreibbar.

Neuronale Netze bieten eine Alternative zu den in der Mathematik seit langem bekannten Funktionsapproximationsmethoden, wie z.B. eine algebraische Repräsentation in Form von Polynomen oder eine tabellarische Repräsentation durch eine Menge diskreter Stützstellen [51]. Der große Vorteil von Neuronalen Netzen liegt in ihrer Fähigkeit eine Funktion anhand von Trainingsdaten zu lernen, anschließend zu reproduzieren sowie das gelernte Wissen auf ungelernete Eingangsdaten zu verallgemeinern.

Die Neuronalen Netze, die in diesem Kapitel vorgestellt werden, stellen statische Funktionsapproximatoren dar. Aus diesem Grund soll zunächst der Begriff der statischen Nichtlinearität erläutert werden. Wird ein N -dimensionaler Eingangsvektor $\underline{u}^T = [u_1, u_2, \dots, u_N]$ durch eine kontinuierliche, begrenzte und zeitinvariante Funktion \mathcal{NL} auf einen skalaren Ausgangswert $y_{\mathcal{NL}}$ abgebildet, so wird dies mathematisch wie folgt formuliert:

$$y_{\mathcal{NL}} = \mathcal{NL}(\underline{u}) \quad (2.1)$$

Die Funktion $\mathcal{NL} : \mathbb{R}^N \rightarrow \mathbb{R}$ wird als statische Nichtlinearität bezeichnet.

Im Folgenden werden einige Grundlagen zu Neuronalen Netzen erklärt. Darauf aufbauend werden verschiedene Netztypen hinsichtlich ihrer Topologie, Funktionsweise und ihrer Eigenschaften beschrieben. Abschließend werden die wichtigsten linearen und nichtlinearen Parameteradaptionsverfahren im Überblick vorgestellt und die für diese Arbeit relevanten Algorithmen kurz erläutert.

2.1 Aufbau von Neuronalen Netzen

Künstliche Neuronale Netze setzen sich analog zu den Nervensystemen von Lebewesen aus vielen miteinander verknüpften Nervenzellen oder Neuronen zusammen.

Als Vorbild für künstliche Neuronen, die auch als Perzeptronen bezeichnet werden, dienen biologische Nervenzellen, wie sie in Abb. 2.1 dargestellt sind.

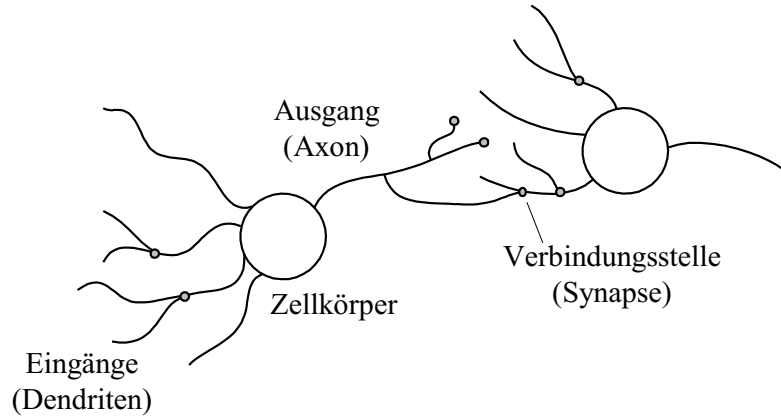


Abb. 2.1: Prinzipdarstellung biologischer Neuronen

Künstliche Neuronen bestehen entsprechend dem biologischen Vorbild aus einem Zellkörper, der den Aktivierungszustand beschreibt, den Dendriten, d.h. den Eingängen in den Zellkörper sowie dem Axon, der den Ausgang des Neurons darstellt. Der Ausgang verzweigt sich und ist mit den Dendriten anderer Neuronen über Synapsen verbunden. Die technische Abstraktion eines biologischen Neurons bzw. Perzeptrons zeigt Abb. 2.2.

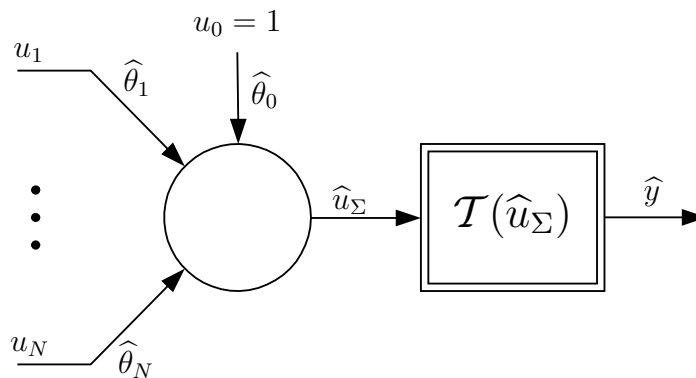


Abb. 2.2: Funktionsweise eines Perzeptrons

Die Eingänge u_i werden zunächst mit den zugehörigen Verbindungsgewichten $\hat{\theta}_i$ gewichtet und anschließend aufsummiert. Zusätzlich zu dieser Summe wird noch der sog. Bias oder Offset $\hat{\theta}_0$ addiert. Die anschließende Transferfunktion $\mathcal{T}(\hat{u}_\Sigma)$ (oder Entscheidungs- bzw. Aktivierungsfunktion) bildet den Summenausgang \hat{u}_Σ nichtlinear auf den Perzeptronausgang \hat{y} ab¹. Mathematisch lässt sich ein einzelnes Per-

¹Der Begriff der *Transferfunktion* darf an dieser Stelle nicht mit dem englischen Begriff *transfer function* für Übertragungsfunktion verwechselt werden.

zeptron wie folgt beschreiben:

$$\hat{y} = \mathcal{T}(\hat{u}_\Sigma) \quad \text{mit} \quad \hat{u}_\Sigma = \hat{\underline{\theta}}^T \cdot \underline{u} = \sum_{i=0}^N \hat{\theta}_i \cdot u_i \quad (2.2)$$

In Gl. (2.2) stellt $\underline{u}^T = [1, u_1, \dots, u_N]$ den Eingangsvektor des Perzeptrons und $\hat{\underline{\theta}}^T = [\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_N]$ den Vektor der trainierbaren Verbindungsgewichte dar. Der Transferfunktion $\mathcal{T}(\hat{u}_\Sigma)$ kommt im Perzeptron eine besondere Bedeutung zu. Sie bildet den Summenausgang \hat{u}_Σ in der Regel nichtlinear auf den Ausgang ab. In Tab. 2.1 sind Beispiele für typische Transferfunktionen dargestellt. Als Transferfunktionen werden in der Regel sigmoide Funktionen verwendet. Dabei wird zwischen weichen und harten Transferfunktionen unterschieden. Weiche Transferfunktionen sind stetig und im gesamten Eingangsraum beliebig oft differenzierbar. Im Gegensatz dazu sind harte Transferfunktionen unstetig und nur stückweise differenzierbar.

Durch seine einfache Struktur ist die Approximationsfähigkeit eines einzelnen Neurons relativ eingeschränkt. Aus diesem Grund werden mehrere Neuronen miteinander vernetzt. In Abb. 2.3 sind einige typische Netztopologien dargestellt.

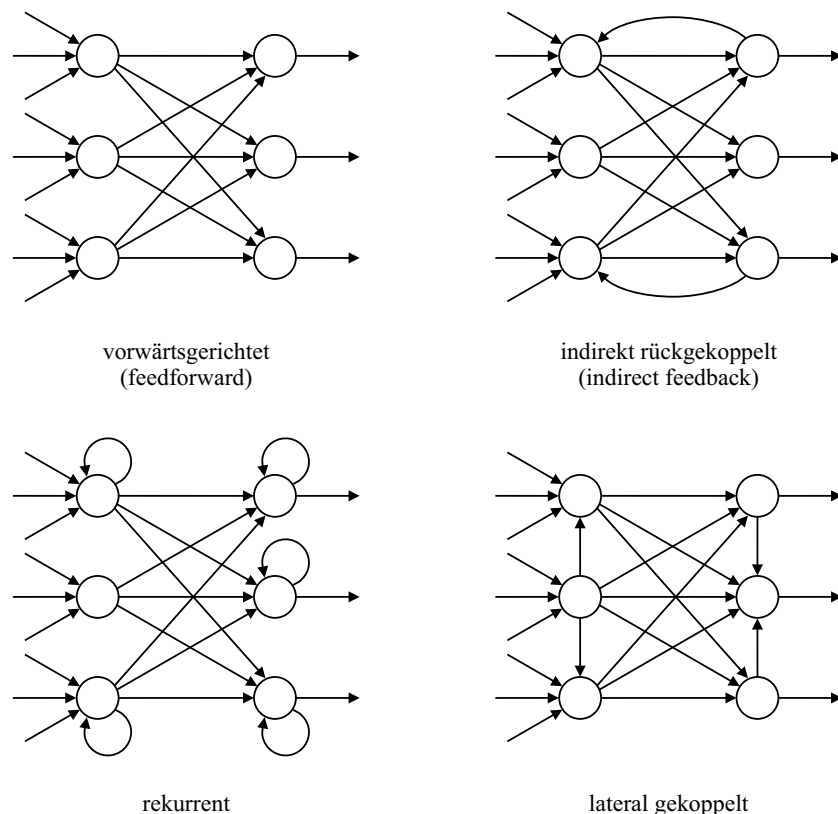


Abb. 2.3: *Typische Netztopologien*

Vorwärtsgerichtete Netze werden auch als Feedforward-Netze bezeichnet. Sie enthalten keine Rückkopplungen und interne Zeitverzögerungen und besitzen deshalb

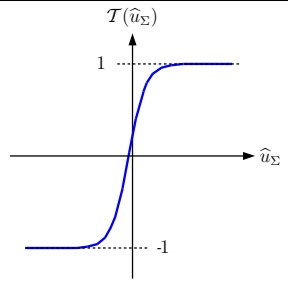
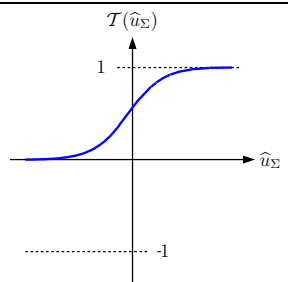
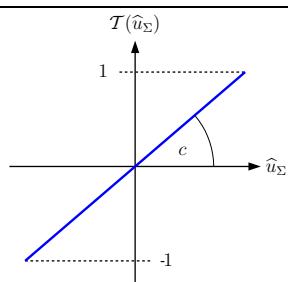
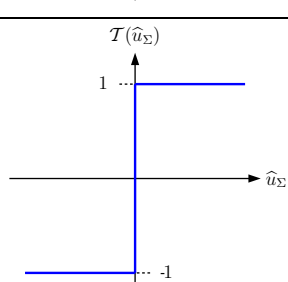
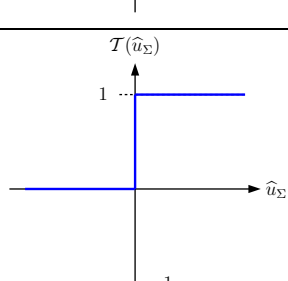
weiche Transferfunktionen	<p>Tangenshyperbolicusfunktion</p> $\mathcal{T}(\hat{u}_\Sigma) = \tanh(\hat{u}_\Sigma) = \frac{1 - e^{-2\hat{u}_\Sigma}}{1 + e^{-2\hat{u}_\Sigma}}$ <p>sigmoid</p>	
	<p>Logistikfunktion</p> $\mathcal{T}(\hat{u}_\Sigma) = \frac{1}{1 + e^{-\hat{u}_\Sigma}}$ <p>sigmoid</p>	
	<p>Lineare Funktion</p> $\mathcal{T}(\hat{u}_\Sigma) = c \cdot \hat{u}_\Sigma$	
harte Transferfunktionen	<p>Signumfunktion</p> $\mathcal{T}(\hat{u}_\Sigma) = \text{sign}(\hat{u}_\Sigma)$ <p>sigmoid</p>	
	<p>Stufenfunktion</p> $\mathcal{T}(\hat{u}_\Sigma) = \begin{cases} 1 & \hat{u}_\Sigma \geq 0 \\ 0 & \text{sonst} \end{cases}$ <p>sigmoid</p>	

Tabelle 2.1: Beispiele für Transferfunktionen $\mathcal{T}(\hat{u}_\Sigma)$

keine dynamischen Eigenschaften. Im Gegensatz dazu weisen indirekt rückgekoppelte Netze Rückkopplungen von Neuronen höherer Schichten zu Neuronen niedrigerer Schichten auf. Solche Netze werden auch als Indirect-Feedback-Netze bezeichnet. Sie besitzen aufgrund von internen Zeitverzögerungen dynamische Eigenschaften. In rekurrenten Netzen kann ein Neuron über eine direkte Verbindung von seinem Ausgang zu seinem Eingang seine eigene Aktivierung beeinflussen. Diese Art von Netzen zählt somit ebenfalls zu den rückgekoppelten Netzen und besitzt dynamische Eigenschaften. Auch lateral gekoppelte Netze zählen zu den rückgekoppelten Netzen. Bei ihnen existieren Verbindungen zwischen Neuronen derselben Schicht. Neben den vorgestellten Netztopologien gibt es noch weitere Untervarianten, die z.B. in [72] beschrieben sind.

In diesem Kapitel werden ausschließlich statische Neuronale Netze betrachtet, die alle eine Feedforward-Struktur besitzen. Feedback-Strukturen werden in Kapitel 3 noch einmal aufgegriffen, da sie in der Regel zur Identifikation nichtlinearer dynamischer Systeme eingesetzt werden. Bei den statischen Funktionsapproximatoren, die im Folgenden genauer vorgestellt werden, handelt es sich lediglich um eine Auswahl, die keinen Anspruch auf Vollständigkeit erhebt.

2.2 Multi-Layer-Perceptron-Network

Einen klassischen statischen Funktionsapproximator stellen vorwärtsgerichtete MLP-Netze² dar. In Abb. 2.4 ist das Beispiel eines einschichtigen MLP-Netzes zur Approximation einer zweidimensionalen statischen Nichtlinearität dargestellt.

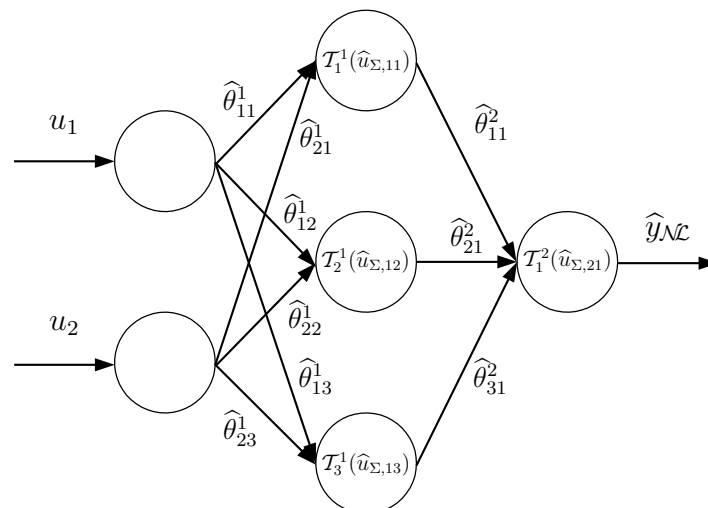


Abb. 2.4: Multi-Layer-Perceptron-Network mit einer Zwischenschicht

²MLP ... Multi-Layer-Perceptron

Die im Allgemeinen N Perzeptronen der Eingangsschicht (input layer) leiten die Eingabe in das Netz weiter und dienen zur Auffächerung der Eingangssignale. Die Eingangsschicht hat somit keine Verarbeitungsfunktion, so dass lineare Transferfunktionen verwendet werden können. Die einzelnen Neuronen in der Zwischenschicht (hidden layer) und die in der Ausgabeschicht (output layer) dienen zur Informationsverarbeitung und entsprechen im Einzelnen Abb. 2.2³. Für das MLP-Netz aus Abb. 2.4 berechnet sich der Netzausgang zu:

$$\hat{y}_{\mathcal{N}} = \mathcal{T}^2 \left(\hat{\theta}_2^T \cdot \underline{\mathcal{T}}^1 \left(\hat{\Theta}_1^T \cdot \underline{u} \right) \right) \quad (2.3)$$

In Gl. (2.3) bezeichnet \underline{u} den Eingangsvektor und $\hat{y}_{\mathcal{N}}$ den Netzausgang. Der Vektor $\underline{\mathcal{T}}^1$ enthält die Transferfunktionen der Neuronen der Zwischenschicht und \mathcal{T}^2 stellt die Transferfunktion des Neurons in der Ausgabeschicht dar. Die Matrix $\hat{\Theta}_1$ beinhaltet die Verbindungsgewichte der ersten trainierbaren Schicht und der Vektor $\hat{\theta}_2$ die Verbindungsgewichte der zweiten trainierbaren Schicht. Die Verbindungsgewichte der zweiten Schicht bilden in diesem Fall keine Matrix, da die Ausgabeschicht nur aus einem Neuron besteht.

Für die Auslegung von MLP-Netzen, d.h. wie viele trainierbare Schichten und wie viele Neuronen in den jeweiligen Schichten eingesetzt werden, gibt es keine allgemein gültigen Regeln. Ein wichtiger Satz aus der Approximationstheorie, der auf Weierstrass [51] zurück geht, besagt, dass es mit zwei- bzw. mehrschichtigen vorwärtsgerichteten Neuronalen Netzen mit monoton steigenden Transferfunktionen, die Squashing-Charakter aufweisen, möglich ist, jede kontinuierliche Funktion beliebig genau zu approximieren. In der Praxis muss die optimale Anzahl an Schichten und Neuronen in der Regel durch gezieltes Ausprobieren ermittelt werden.

Charakteristisch für MLP-Netze ist, dass die unbekanntes Netzwerkgewichte nicht-linear in das Ausgangssignal eingehen und somit ein nichtlineares Adaptionsverfahren zum Einsatz kommen muss. Als Adaptionsverfahren wird bei MLP-Netzen der Backpropagation-Algorithmus [72] verwendet. Der Backpropagation-Algorithmus ist eine Verallgemeinerung des Gradientenabstiegsverfahrens (siehe Kapitel 2.7.2). Die Gradienten der versteckten Schichten werden dabei durch mehrfaches Anwenden der Kettenregel bei der Differentiation berechnet.

Das MLP-Netz eignet sich vor allem zur Approximation von Funktionen der Dimension $N \geq 3$, da bei MLP-Netzen eine Komprimierung des Eingangsraumes stattfindet⁴. Der Nachteil dieser Eingangsraumkomprimierung ist, dass die Netzwerkgewichte keine lokale Wirkung mehr haben, d.h. es ist keine physikalische Zuordnung der Netzwerkgewichte und ihrer Wirkung zum Eingangsraum des Netzes möglich.

³Der Bias der einzelnen Perzeptronen wurde aus Gründen der einfacheren Darstellung vernachlässigt.

⁴Dies ist auch der Grund, weshalb das MLP-Netz zur Approximation nichtlinearer dynamischer Systeme eingesetzt werden kann (vgl. Kapitel 3.3.3).

2.3 Radial-Basis-Function-Network

Neuronale Netze mit radialsymmetrischen Basisfunktionen gehören ebenfalls zu den mehrschichtigen vorwärtsgerichteten Netzen. Aufgrund ihres stützwertebasierten Approximationsverhaltens eignen sie sich besonders für regelungstechnische Aufgabenstellungen [47]. In [52] werden verschiedene Funktionen vorgestellt, die sich als Basisfunktionen eignen, wie z.B. Gauß'sche Glockenkurven oder die Manhattan-Distanz. Neuronale Netze mit radialsymmetrischen Basisfunktionen weisen nach [72] gegenüber anderen Netztypen, wie z.B. MLP-Netzen, zwei Vorteile auf. Erstens liefert die Aktivierungsfunktion eines Neurons nur in der Nähe ihres Zentrums einen großen Wert. Für Eingangswerte, die außerhalb des durch die Stützwerte beschriebenen Eingangsbereiches liegen, liefert das Netz nur sehr geringe Aktivierungen und somit nur geringe Ausgangswerte. Im Gegensatz dazu liefern Neuronale Netze mit sigmoiden Transferfunktionen außerhalb des gelernten Bereiches oft unvorhersehbare Aussagen. Zweitens ermöglicht die einfache Struktur des RBF-Netzes mit nur einer verdeckten Schicht auch eine einfache Berechnung der Netzwerkgewichte, da diese den Netzausgang linear beeinflussen. Das bedeutet, dass in der Regel keine aufwändigen, iterativen Lernverfahren wie bei anderen Netztypen notwendig sind.

Die Struktur des RBF-Netzes ist in Abb. 2.5 veranschaulicht.

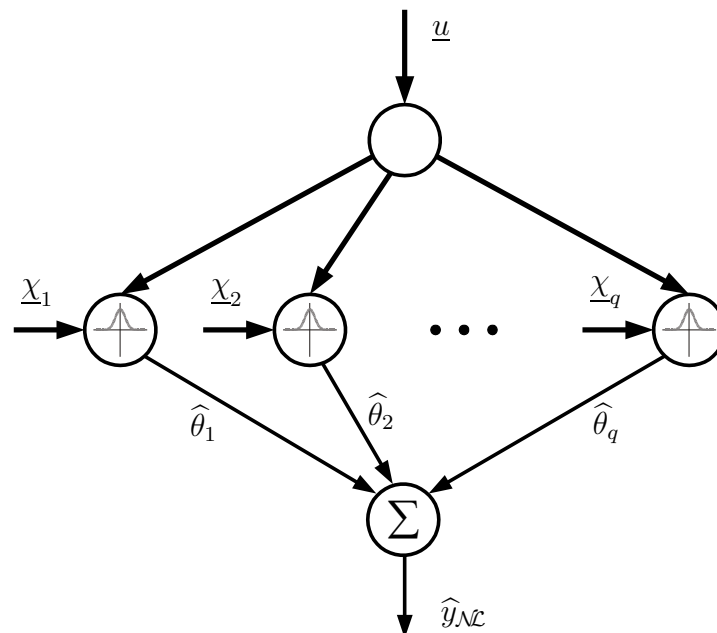


Abb. 2.5: Struktur des RBF-Netzes

Der Ausgang eines RBF-Netzes mit q Stützpunkten berechnet sich allgemein zu:

$$\hat{y}_{\mathcal{N}} = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{u}) \quad (2.4)$$

Der Vektor $\hat{\underline{\theta}}^T = [\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_q]$ enthält die geschätzten Gewichte. $\underline{\mathcal{A}}(\underline{u})$ bezeichnet den Aktivierungsvektor mit den radialsymmetrischen Basisfunktionen. Üblicherweise werden als Aktivierungsfunktionen für RBF-Netze Gauß'sche Glockenkurven verwendet [51]. Die Aktivierung des i -ten Neurons berechnet sich in diesem Fall zu:

$$\mathcal{A}_i(\underline{u}) = \exp \left[-\frac{1}{2} \left(\frac{(u_1 - \chi_{1,i})^2}{\sigma_1^2} + \dots + \frac{(u_N - \chi_{N,i})^2}{\sigma_N^2} \right) \right] \quad \text{mit } i = 1 \dots q \quad (2.5)$$

Die Zentrumskoordinaten des i -ten Stützpunktes werden durch $\chi_{1,i} \dots \chi_{N,i}$ beschrieben. Mit $\sigma_1 \dots \sigma_N$ werden die Glättungsfaktoren bezeichnet, die den Grad der Überlappung von benachbarten Aktivierungsfunktionen festlegen. Ein Nachteil des RBF-Netzes ist die fehlende Monotonieerhaltung zwischen den aufeinander folgenden Stützpunkten. Das bedeutet, dass der Ausgangswert des Netzes außerhalb des durch die benachbarten Stützstellen begrenzten Bereiches liegen kann. In ungelerten Bereichen fällt der Netzausgang auf den Wert Null ab. Dies ist zwar ein Vorteil gegenüber Netztypen mit sigmoiden Transferfunktionen, aus regelungstechnischer Sicht ist jedoch auch dieses Extrapolationsverhalten nicht optimal. In Abb. 2.6 ist das beschriebene Verhalten noch einmal verdeutlicht.

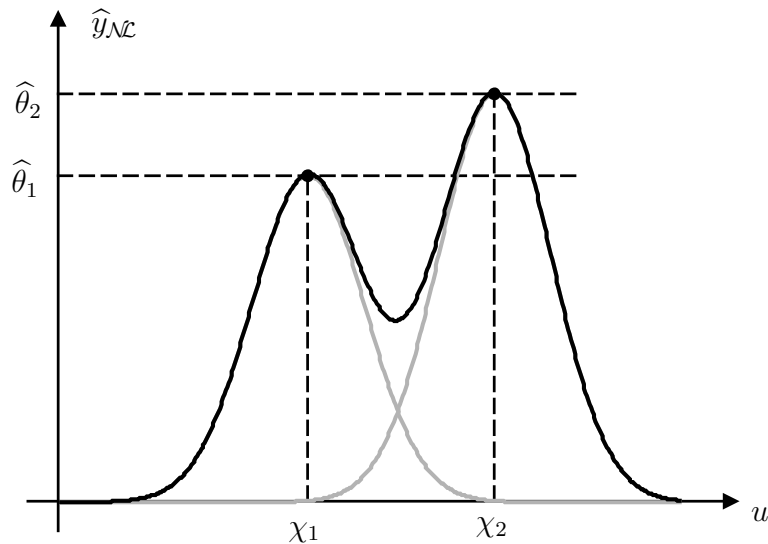


Abb. 2.6: Inter- und Extrapolationsverhalten des RBF-Netzes

Die nachfolgende Modifikation des RBF-Netzes beseitigt diese Nachteile.

2.4 General-Regression-Neural-Network

Das General-Regression-Neural-Network (GRNN) stellt eine Weiterentwicklung des RBF-Netzes dar. Durch die Normierung der einzelnen Aktivierungsfunktionen auf die Summe aller Aktivierungsfunktionen kann der Nachteil der fehlenden Monotonieerhaltung des RBF-Netzes beseitigt werden. Ausführliche Untersuchungen zum GRNN können in [47, 56] nachgelesen werden. Die Aktivierung des i -ten Neurons berechnet sich beim GRNN zu:

$$\mathcal{A}_i(\underline{u}) = \frac{\exp \left[-\frac{1}{2} \left(\frac{(u_1 - \chi_{1,i})^2}{\sigma_1^2} + \dots + \frac{(u_N - \chi_{N,i})^2}{\sigma_N^2} \right) \right]}{\sum_{j=1}^q \exp \left[-\frac{1}{2} \left(\frac{(u_1 - \chi_{1,j})^2}{\sigma_1^2} + \dots + \frac{(u_N - \chi_{N,j})^2}{\sigma_N^2} \right) \right]} \quad \text{mit } i = 1 \dots q \quad (2.6)$$

Für die Summe aller Aktivierungsfunktionen gilt folglich:

$$\sum_{j=1}^q \mathcal{A}_j(\underline{u}) = 1 \quad (2.7)$$

Die Struktur des GRNN ist in Abb. 2.7 veranschaulicht.

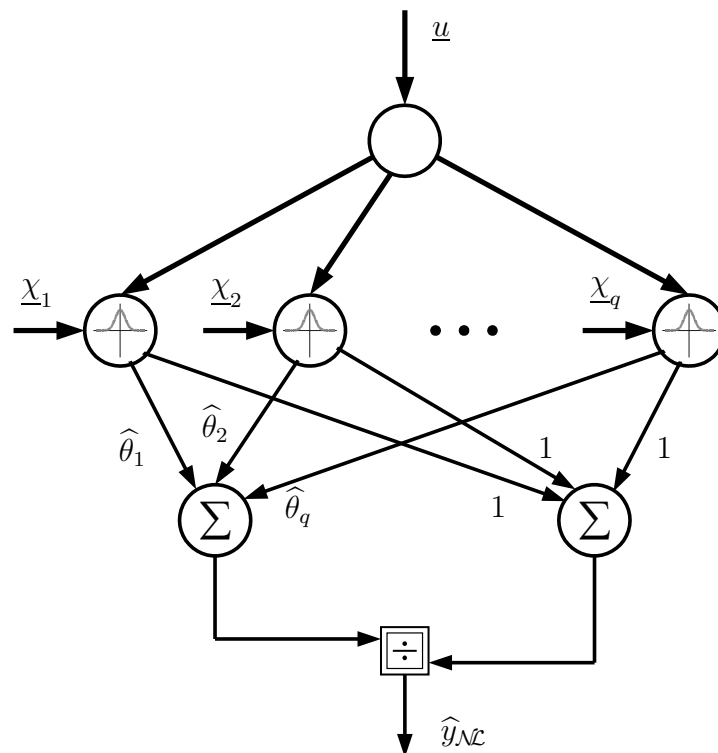


Abb. 2.7: Struktur des GRNN

Aufgrund der Normierung der Aktivierungsfunktionen kann das GRNN nur Ausgangswerte zwischen dem kleinsten und dem größten Stützwert annehmen. Zwischen den einzelnen Stützwerten ist im Gegensatz zum RBF-Netz Monotonieerhaltung gegeben. In ungelerten Bereichen extrapoliert das GRNN konstant auf den Wert des nächstgelegenen Stützwertes. Einen direkten Vergleich zwischen dem GRNN und dem RBF-Netz zeigt Abb. 2.8.

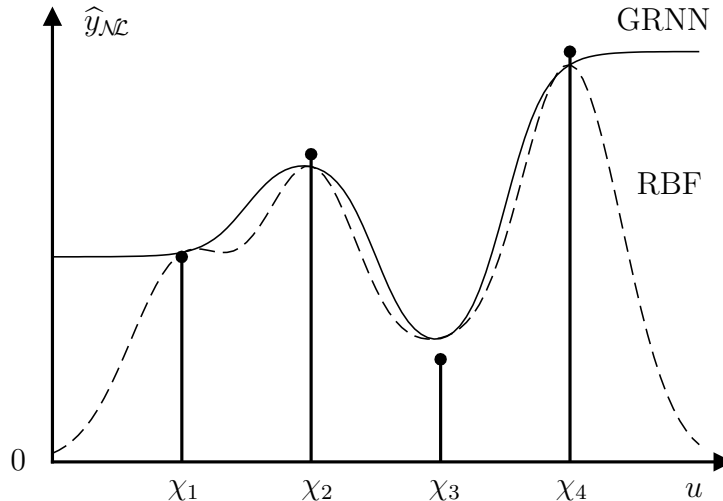


Abb. 2.8: Vergleich zwischen GRNN und RBF-Netz

Das GRNN eignet sich vor allem zur Approximation von statischen Nichtlinearitäten der Dimension $N \leq 2$. Bei einer höheren Eingangsdimension steigt die Parameteranzahl stark an, so dass das GRNN nur noch bedingt einsetzbar ist. Aufgrund der Tatsache, dass die unbekannt Parameter linear in den Ausgang des GRNN eingehen, können sowohl lineare Adaptionungsverfahren, wie z.B. der rekursive Least-Squares-Algorithmus, als auch nichtlineare Adaptionungsverfahren, wie z.B. das Gradientenabstiegsverfahren, eingesetzt werden. Beide Lernverfahren werden in Kapitel 2.7 noch eingehender behandelt.

2.5 Harmonic-Activation-Neural-Network

Die Approximation periodischer Signale stellt ein weiteres Anwendungsgebiet für Neuronale Netze dar. In [6] wurde ein Neuronales Netz mit harmonischen Basisfunktionen vorgestellt und zur Identifikation von Drehmomentschwingungen am Verbrennungsmotor eingesetzt. Eine weitere Einsatzmöglichkeit ist in [12] beschrieben, wo das Harmonic-Activation-Neural-Network (HANN) zur On-Board-Diagnose von Verbrennungsmotoren verwendet wurde.

Gegenüber den bereits vorgestellten statischen Funktionsapproximatoren bietet das HANN den Vorteil, dass für periodische Nichtlinearitäten eine hohe Approximati-

ongenauigkeit bei geringer Parameteranzahl erreicht wird. Ebenso ist die wahlweise Ausblendung bestimmter Frequenzanteile des periodischen Signals möglich. Eine im Winkel φ periodische Nichtlinearität kann als reelle Fourierreihe dargestellt werden. Werden die Sinus- und Cosinusterme als Basisfunktionen eines Neuronalen Netzes interpretiert, ergibt sich der Ausgang dieses Neuronalen Netzes zu:

$$\hat{y}_{\mathcal{N}} = \sum_{k=0}^K \hat{\theta}_{A,k} \cdot \mathcal{A}_{A,k}(\varphi) + \sum_{k=1}^K \hat{\theta}_{B,k} \cdot \mathcal{A}_{B,k}(\varphi) = \hat{\underline{\theta}}_A^T \cdot \underline{\mathcal{A}}_A(\varphi) + \hat{\underline{\theta}}_B^T \cdot \underline{\mathcal{A}}_B(\varphi) \quad (2.8)$$

Aufgrund der harmonischen Aktivierungsfunktionen wird Gl. (2.8) als Harmonic-Activation-Neural-Network bezeichnet. Der Index A kennzeichnet die geraden, der Index B die ungeraden Spektralanteile der nichtlinearen Funktion. Die unbekanntenen Gewichte $\hat{\theta}_{A,k}$ und $\hat{\theta}_{B,k}$ in Gl. (2.8) entsprechen den Koeffizienten einer reellen Fourierreihe. Für die Gewichts- und Aktivierungsvektoren gilt:

$$\hat{\underline{\theta}}_A^T = [\hat{\theta}_{A,0}, \hat{\theta}_{A,1}, \hat{\theta}_{A,2}, \dots, \hat{\theta}_{A,K}] \quad (2.9)$$

$$\hat{\underline{\theta}}_B^T = [\hat{\theta}_{B,1}, \hat{\theta}_{B,2}, \dots, \hat{\theta}_{B,K}] \quad (2.10)$$

$$\underline{\mathcal{A}}_A^T(\varphi) = [1, \cos(\varphi), \cos(2\varphi), \dots, \cos(K\varphi)] \quad (2.11)$$

$$\underline{\mathcal{A}}_B^T(\varphi) = [\sin(\varphi), \sin(2\varphi), \dots, \sin(K\varphi)] \quad (2.12)$$

Die Struktur des HANN ist in Abb. 2.9 veranschaulicht.

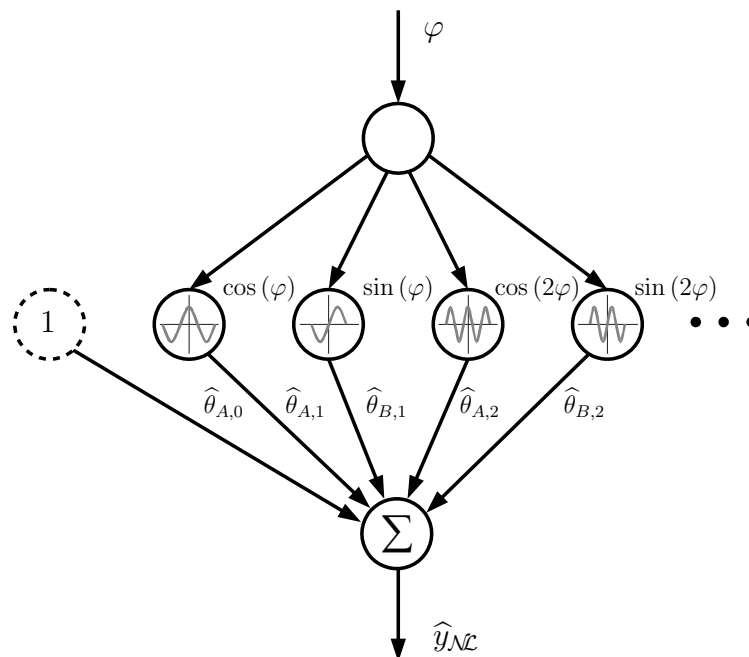


Abb. 2.9: Struktur des HANN

Das Neuron mit der Aktivierungsfunktion $\mathcal{A}_{A,0}(\varphi) = 1$ und dem zugehörigen Parameter $\widehat{\theta}_{A,0}$ charakterisiert den Gleichanteil, der in [6] beispielsweise ausgeblendet wurde.

Im Gegensatz zu Neuronalen Netzen mit lokalen Basisfunktionen können die einzelnen Stützwerte des HANN nicht mehr einem bestimmten Eingangsbereich lokal zugeordnet werden. Stattdessen lassen die Gewichte eine Aussage über die spektrale Zusammensetzung einer periodischen Funktion zu. Aufgrund der Tatsache, dass die unbekannt Parameter linear in den Ausgang des HANN eingehen, können analog zu den Neuronalen Netzen mit radialsymmetrischen Basisfunktionen sowohl lineare als auch nichtlineare Adaptionsverfahren eingesetzt werden.

2.6 Local-Linear-Model-Tree

Der Local-Linear-Model-Tree (LOLIMOT) approximiert komplexe nichtlineare Funktionen durch einfachere lokale lineare Modelle, die mittels Aktivierungsfunktionen überlagert werden. Die lokalen linearen Modelle können als Neuronen mit einer linearen Transferfunktion der Steigung Eins (Identitätsfunktion — vgl. Kapitel 2.1) interpretiert werden. Mittels eines Konstruktionsverfahrens wird schrittweise eine Netzstruktur aufgebaut, wobei sowohl die Anzahl der Neuronen als auch deren Gültigkeitsbereich im Eingangsraum optimiert wird. LOLIMOT komprimiert den Eingangsraum und eignet sich deswegen, neben dem Einsatz als statischer Funktionsapproximator⁵, vor allem zur Identifikation nichtlinearer dynamischer Systeme, worauf in Kapitel 3.3.3 kurz eingegangen wird.

LOLIMOT basiert auf der Idee, eine nichtlineare Funktion aus mehreren linearen Modellen stückweise nachzubilden. Die Bereiche, in denen die einzelnen Teilmodelle Gültigkeit haben, werden mittels eines Strukturselektionsalgorithmus bestimmt. Jedem Bereich ist eine Zugehörigkeitsfunktion oder Aktivierungsfunktion zugeordnet, welche die Gültigkeit eines Teilmodells innerhalb des Eingangsraumes festlegt, wobei der Übergang zwischen den einzelnen Gültigkeitsbereichen unscharf ist. Als Aktivierungsfunktionen werden in der Regel normierte Gauß'sche Glockenkurven verwendet. Durch die Normierung ist die Summe der Aktivierungsfunktionen an jeder Stelle des Eingangsraumes exakt gleich Eins. Der Ausgang von LOLIMOT berechnet sich für M Teilmodelle zu [36, 37, 40, 41, 39]:

$$\widehat{y}_{\mathcal{N}} = \sum_{i=1}^M \underbrace{\left(\widehat{\theta}_{0,i} + \widehat{\theta}_{1,i} \cdot u_1 + \dots + \widehat{\theta}_{N,i} \cdot u_N \right)}_{\text{lineares Teilmodell}} \cdot \underbrace{\mathcal{A}_i(\underline{u}, \underline{\chi}_i, \underline{\sigma}_i)}_{\text{Aktivierungsfunktion}} \quad (2.13)$$

$\widehat{\theta}_{0,i} \dots \widehat{\theta}_{N,i}$ bezeichnet die unbekannt Netzwerkgewichte und \mathcal{A}_i die Aktivierungsfunktion des i -ten Teilmodells mit den Zentrumskoordinaten $\underline{\chi}_i$ und den Standard-

⁵Hier besonders bei einer Eingangsdimension von $N \geq 3$.

abweichungen $\underline{\sigma}_i$. Die Aktivierungsfunktion des i -ten Teilmodells kann nach folgender Gleichung berechnet werden:

$$\mathcal{A}_i(\underline{u}, \underline{\chi}_i, \underline{\sigma}_i) = \frac{\exp \left[-\frac{1}{2} \left(\frac{(u_1 - \chi_{1,i})^2}{\sigma_{1,i}^2} + \dots + \frac{(u_N - \chi_{N,i})^2}{\sigma_{N,i}^2} \right) \right]}{\sum_{j=1}^M \exp \left[-\frac{1}{2} \left(\frac{(u_1 - \chi_{1,j})^2}{\sigma_{1,j}^2} + \dots + \frac{(u_N - \chi_{N,j})^2}{\sigma_{N,j}^2} \right) \right]} \quad \text{mit } i = 1 \dots M \quad (2.14)$$

Die Netzstruktur von LOLIMOT zur Approximation statischer Nichtlinearitäten ist in Abb. 2.10 veranschaulicht. Im eindimensionalen Fall können die Teilmodelle als

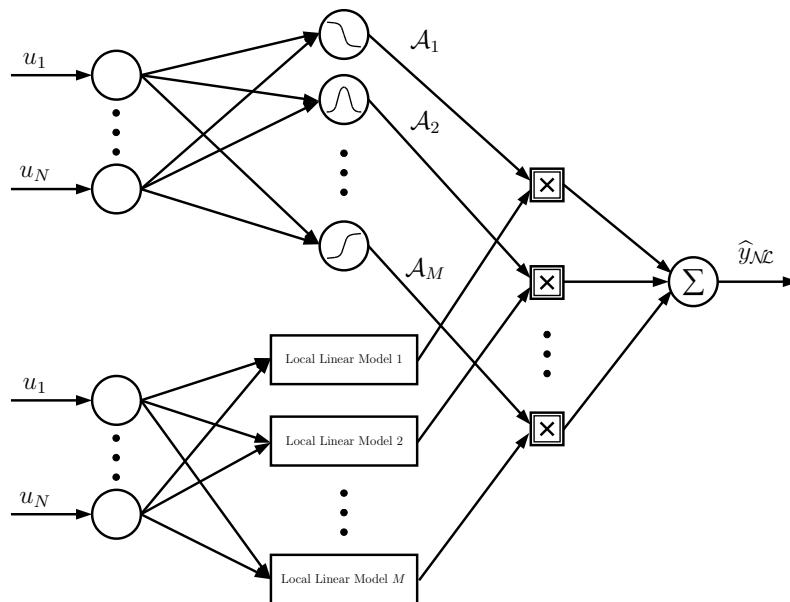


Abb. 2.10: Struktur von LOLIMOT

Geraden interpretiert werden, mittels derer die nichtlineare Funktion approximiert wird. Im zweidimensionalen bzw. N -dimensionalen Eingangsraum können die Teilmodelle als Ebenen bzw. N -dimensionale Hyperebenen angesehen werden. Die Parameteradaptation erfolgt mittels eines gewichteten Least-Squares-Verfahrens [24]⁶. Dabei wird die Parameterberechnung lokal durchgeführt, d.h. für jedes Teilmodell einzeln, was den Rechenaufwand verringert. Durch den Algorithmus zur Strukturoptimierung können tendenziell lineare Bereiche der zu approximierenden Funktion von nichtlinearen Bereichen unterschieden werden und so die Parameteranzahl optimiert werden. Die Berechnung der optimalen Struktur ist nur offline möglich.

⁶Dies ist aufgrund der Linearität der Parameter im Ausgang von LOLIMOT möglich.

2.7 Parameteradaptionsverfahren

Die Adaption der unbekanntenen Netzwerkgewichte wird im Zusammenhang mit Neuronalen Netzen auch als Lernen bezeichnet. In [72] werden drei verschiedene Arten des Lernens unterschieden: das überwachte Lernen, das bestärkende Lernen und das unüberwachte Lernen. In dieser Arbeit wird ausschließlich das überwachte Lernen betrachtet. Beim überwachten Lernen wird davon ausgegangen, dass zu jedem Eingangsmuster das gewünschte Ausgangsmuster bekannt ist. Zwischen dem gemessenen Systemausgang y und dem Modellausgang \hat{y} wird ein Fehler e gebildet, der für die Adaption der Netzwerkgewichte eingesetzt wird. Der gemessene Systemausgang ist in der Regel von einem Rauschsignal z überlagert. In Abb. 2.11 ist die beschriebene Anordnung, die auch als Vorwärtslernstruktur bekannt ist, veranschaulicht.

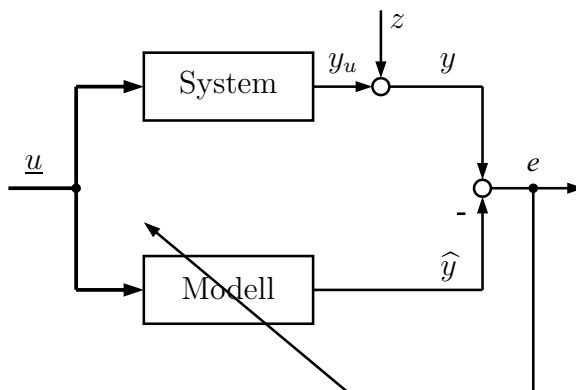


Abb. 2.11: Struktur des überwachten Lernens

Ziel des Adaptionsverfahrens ist es, die Gewichte des Neuronalen Netzes so zu adaptieren, dass der Netzausgang bei gleichem Eingangsmuster das gewünschte Ausgangsmuster möglichst gut reproduziert. Darüber hinaus soll das trainierte Neuronale Netz auch bei ähnlichen Eingangsmustern die entsprechenden Ausgangsmuster berechnen können. Die einzelnen Kriterien zur Beurteilung der Güte von trainierten Neuronalen Netzen werden in Kapitel 3.1 noch einmal diskutiert.

Das überwachte Lernen kann nach [41] in drei Klassen eingeteilt werden: lineare, nichtlineare lokale und nichtlineare globale Adaptionsverfahren. In Abb. 2.12 sind die verschiedenen Adaptionsverfahren im Überblick dargestellt. Lineare Adaptionsverfahren können immer dann angewendet werden, wenn der Modellausgang linear in den Parametern ist. Als Gütefunktional⁷ wird die Summe des quadratischen Fehlers minimiert. Lineare Adaptionsverfahren führen immer zum globalen Minimum des Gütefunktionals [5]. Prinzipiell können bei den linearen Verfahren, direkte, rekursive und iterative Ansätze unterschieden werden. Beispiele hierfür sind der Least-Squares-Algorithmus (LS), der rekursive Least-Squares-Algorithmus (RLS)

⁷Es wird auch von der Verlustfunktion gesprochen.

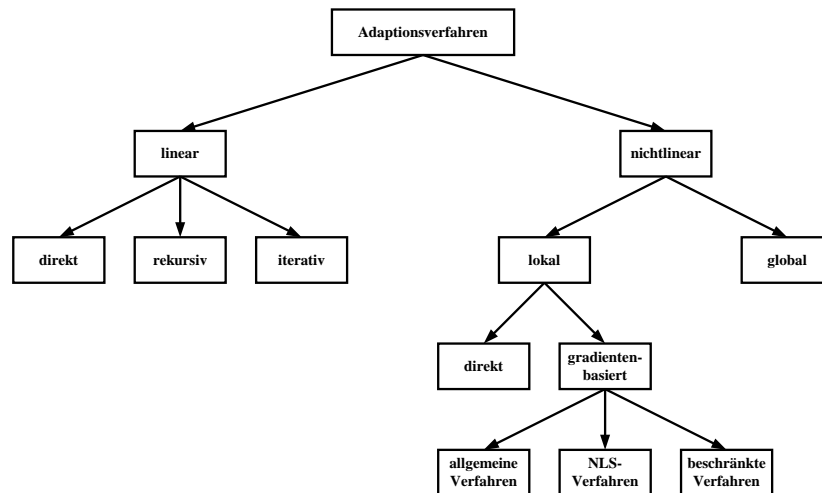


Abb. 2.12: Parameteradaptionsverfahren im Überblick

und der Least-Mean-Squares-Algorithmus (LMS) [41]. Nichtlineare Adaptionsverfahren müssen in den Fällen angewendet werden, in denen die unbekannt Parameter nichtlinear in den Modellausgang eingehen. Nichtlineare lokale Verfahren versuchen das globale Minimum des Gütefunktional, ausgehend von einem Startpunkt, aufgrund von Informationen aus der unmittelbaren Umgebung des Startpunktes, zu finden. Häufig führen diese Verfahren zu einem Minimum in der Umgebung des Startpunktes, was ein lokales und nicht das globale Minimum ist. Direkte Methoden, wie z.B. die Simplex-Search-Methode oder der Hooke-Jeeves-Algorithmus, basieren ausschließlich auf der Auswertung des Gütefunktional [46]. Die partiellen Ableitungen des Gütefunktional nach den Parametern werden nicht benötigt, da das globale Minimum des Gütefunktional entlang vordefinierter Richtungen gesucht wird. Direkte Verfahren sind einfach zu implementieren, jedoch haben sie eine langsame Konvergenz. Können die partiellen Ableitungen bestimmt werden, ist es sinnvoller ein gradientenbasiertes Adaptionsverfahren zu verwenden. Zu den allgemeinen Verfahren zählen beispielsweise das einfache Gradientenabstiegsverfahren oder das Newton-Verfahren, welches zusätzlich die Ableitungen zweiter Ordnung des Gütefunktional benötigt. Die Anwendung von nichtlinearen Least-Squares-Verfahren (NLS), wie z.B. dem Gauß-Newton- oder Levenberg-Marquardt-Verfahren, setzt voraus, dass das Gütefunktional eine quadratische Struktur aufweist. Das Vorwissen über die Struktur des Gütefunktional ermöglicht effizientere Algorithmen als bei den allgemeinen Verfahren. Die Berücksichtigung von Beschränkungen führt zu deutlich komplexeren Adaptionsalgorithmen. Als Beispiel kann die quadratische Programmierung genannt werden. Nichtlineare globale Verfahren beinhalten in der Regel stochastische Elemente um dem Algorithmus die Möglichkeit zu geben, aus lokalen Minima zu entkommen und das globale Minimum zu finden. Die Fähigkeit aus lokalen Minima entkommen zu können wird mit einem hohen Rechenaufwand und einer langsamen Konvergenz bezahlt. Nach [41] eignen sich globale Methoden um Regionen mit guten Minima im Gütefunktional zu finden, während sich lokale Me-

thoden dafür eignen in diesen Regionen das lokale Minimum, bzw. unter Umständen auch das globale Minimum zu finden. Typische nichtlineare globale Adaptionsverfahren sind das Simulated Annealing sowie evolutionäre und genetische Algorithmen. Einen tieferen Einblick in die Parameteradaptionsverfahren geben [24, 25, 33, 41, 42, 55]. In den folgenden Kapiteln werden die Adaptionsverfahren, die im Rahmen dieser Arbeit verwendet werden, kurz erläutert.

2.7.1 Least-Squares-Algorithmus

Ausgangspunkt eines jeden Adaptionsverfahrens ist ein Gütefunktional, das die Optimierungsaufgabe mathematisch formuliert. Im Folgenden wird angenommen, dass der Modellausgang in der Form

$$\hat{y} = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{u}) \quad (2.15)$$

dargestellt werden kann. Dies entspricht beispielsweise der formalen Beschreibung des GRNN⁸. Die Berechnung der unbekannt Parameter erfolgt durch die Minimierung der Summe der quadrierten Gleichungsfehler. Liegt ein Datensatz mit P Trainingspaaren von \underline{u} und y vor, so ergibt sich das Gütefunktional zu:

$$E(\hat{\underline{\theta}}) = \frac{1}{2} \cdot \sum_{k=1}^P e[k] = \frac{1}{2} \cdot \left(\underline{y} - \mathbf{X} \cdot \hat{\underline{\theta}} \right)^T \cdot \left(\underline{y} - \mathbf{X} \cdot \hat{\underline{\theta}} \right) \longrightarrow \min_{\hat{\underline{\theta}}} \quad (2.16)$$

mit

$$\mathbf{X} = \begin{bmatrix} \mathcal{A}_1(\underline{u}[1]) & \mathcal{A}_2(\underline{u}[1]) & \dots & \mathcal{A}_q(\underline{u}[1]) \\ \mathcal{A}_1(\underline{u}[2]) & \mathcal{A}_2(\underline{u}[2]) & \dots & \mathcal{A}_q(\underline{u}[2]) \\ \vdots & & & \vdots \\ \mathcal{A}_1(\underline{u}[P]) & \mathcal{A}_2(\underline{u}[P]) & \dots & \mathcal{A}_q(\underline{u}[P]) \end{bmatrix} \quad \text{und} \quad \underline{y} = \begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[P] \end{bmatrix} \quad (2.17)$$

Normalerweise ist die Zahl der Trainingspaare wesentlich größer als die Stützwertezahl ($P \gg q$), so dass sich ein überbestimmtes Gleichungssystem ergibt. Durch zu Null setzen der Ableitung von Gl. (2.16) nach dem Parametervektor $\hat{\underline{\theta}}$ ergibt sich der Parametervektor zu:

$$\frac{\partial E(\hat{\underline{\theta}})}{\partial \hat{\underline{\theta}}} = -\mathbf{X}^T \cdot \left(\underline{y} - \mathbf{X} \cdot \hat{\underline{\theta}} \right) \stackrel{!}{=} \underline{0} \quad (2.18)$$

$$\implies \hat{\underline{\theta}} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \underline{y}$$

⁸Auch die Identifikationsansätze auf der Basis von Volterra-Reihen, die in den folgenden Kapiteln eingehend behandelt werden, lassen sich in dieser Form darstellen, wenn der Aktivierungsvektor $\underline{\mathcal{A}}$ durch den dynamischen Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}$ ersetzt wird.

Die Gleichungen (2.18) beschreiben den nicht rekursiven Least–Squares–Algorithmus, der ausschließlich offline einsetzbar ist, da alle Trainingspaare gleichzeitig in einem Datensatz vorliegen müssen.

Im Gegensatz dazu kann der rekursive Least–Squares–Algorithmus (RLS) online eingesetzt werden. Die Trainingspaare werden sukzessive zur Berechnung der unbekannt Parameter verarbeitet. Der Rechenaufwand ist zu jedem Abtastschritt gleich. Für die Rekursionsformeln ergibt sich nach [24]:

$$\hat{\underline{\theta}}[k] = \hat{\underline{\theta}}[k-1] + \underline{h}[k] \cdot \left(y[k] - \hat{\underline{\theta}}^T[k-1] \cdot \underline{\mathcal{A}}[k] \right) \quad (2.19)$$

$$\underline{h}[k] = \frac{\mathbf{P}[k-1] \cdot \underline{\mathcal{A}}[k]}{1 + \underline{\mathcal{A}}^T[k] \cdot \mathbf{P}[k-1] \cdot \underline{\mathcal{A}}[k]} \quad (2.20)$$

$$\mathbf{P}[k] = \mathbf{P}[k-1] - \underline{h}[k] \cdot \underline{\mathcal{A}}^T[k] \cdot \mathbf{P}[k-1] \quad (2.21)$$

Die Matrix $\mathbf{P} \in \mathbb{R}^{q \times q}$ ist ein normierter Schätzwert der Kovarianzmatrix bezüglich der Parameterschätzfehler, $\underline{h}[k] \in \mathbb{R}^{q \times 1}$ ist der sog. Korrekturvektor. Der Vektor $\underline{\mathcal{A}}^T[k] = [\mathcal{A}_1(\underline{u}[k]), \dots, \mathcal{A}_q(\underline{u}[k])]$ bezeichnet die aktuellen Aktivierungsfunktionen. Als Startwert für die Matrix \mathbf{P} eignet sich eine obere Dreiecksmatrix, deren Werte zwischen 100 und 1000 liegen. Der Parametervektor kann mit Null initialisiert werden.

2.7.2 Gradientenabstiegsverfahren

Das Gradientenabstiegsverfahren ist eines der einfachsten nichtlinearen Adaptionenverfahren. Als Gütefunktional dient der quadratische Fehler:

$$E(\hat{\underline{\theta}}) = \frac{1}{2} \cdot e^2 = \frac{1}{2} \cdot \left(\sum_{i=1}^q \hat{\theta}_i \cdot \mathcal{A}_i(\underline{u}) - y \right)^2 \quad (2.22)$$

Die notwendige Änderung jedes Gewichtes $\hat{\theta}_i$ wird durch das Gradientenabstiegsverfahren⁹ festgelegt. Dazu wird der quadratische Fehler nach dem jeweiligen Gewicht abgeleitet:

$$\frac{\partial E(\hat{\underline{\theta}})}{\partial \hat{\theta}_i} = \left(\sum_{k=1}^q \hat{\theta}_k \cdot \mathcal{A}_k(\underline{u}) - y \right) \cdot \mathcal{A}_i(\underline{u}) = e \cdot \mathcal{A}_i(\underline{u}) \quad \text{mit} \quad i = 1 \dots q \quad (2.23)$$

Mit Gl. (2.23) ergibt sich das zeitkontinuierliche Gradientenabstiegsverfahren zu:

$$\dot{\hat{\theta}}_i = -\eta \cdot \frac{\partial E(\hat{\underline{\theta}})}{\partial \hat{\theta}_i} = -\eta \cdot e \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}_i} = -\eta \cdot e \cdot \mathcal{A}_i(\underline{u}) \quad \text{mit} \quad i = 1 \dots q \quad (2.24)$$

⁹Das Gradientenabstiegsverfahren wird auch als Delta–Lernregel [51] bezeichnet.

Die Skalierung mit der Lernschrittweite η dient der Einstellung einer gewünschten Lerngeschwindigkeit bzw. einer gewissen Glättungswirkung bei der Adaption. Das negative Vorzeichen stellt eine Anpassung der Gewichte in Richtung kleinerer Fehler sicher. Für die zeitdiskrete Realisierung des Gradientenabstiegsverfahrens gilt:

$$\hat{\theta}_i[k] = \hat{\theta}_i[k-1] - \eta' \cdot e[k] \cdot \mathcal{A}_i[k] \quad \text{mit} \quad i = 1 \dots q \quad (2.25)$$

Die Lernschrittweite η' ergibt sich aus der Lernschrittweite η durch Verrechnung der Abtastzeit. Der Vorteil des Gradientenabstiegsverfahrens ist die einfache Implementierung, insbesondere dann, wenn die partiellen Ableitungen nach den unbekannt Parametern deutlich komplizierter sind als in obigem Beispiel. Aufgrund der Tatsache, dass es sich um ein lokales nichtlineares Verfahren handelt, besteht jedoch immer die Gefahr, dass der Algorithmus in einem lokalen Minimum endet und nicht das globale Minimum des Gütefunktional findet. In [72] werden die Probleme des Gradientenabstiegsverfahrens ausführlich diskutiert und wirkungsvolle Verbesserungen, wie z.B. die Einführung eines Momentum-Terms, vorgeschlagen. Die Diskussion soll jedoch an dieser Stelle nicht weiter vertieft werden.

2.8 Kurzzusammenfassung

In diesem Kapitel wurden zunächst einige Grundlagen zu Neuronalen Netzen und ihrer Analogie zu biologischen Nervensystemen erklärt. Anschließend wurden verschiedene statische Funktionsapproximatoren genauer erläutert. Es zeigte sich, dass sich das Multi-Layer-Perzeptron-Netz vor allem zur Approximation von Funktionen mit einer hohen Eingangsraumdimension eignet. Als Nachteile des MLP-Netzes sind das aufwändige nichtlineare Lernverfahren sowie die globale Wirkung der Parameter zu nennen. Im Gegensatz dazu zeigte sich bei Netzen mit radialsymmetrischen Basisfunktionen eine lokale Wirkung der Parameter, was im Falle des GRNN, der Weiterentwicklung des RBF-Netzes, zu guten Interpolations- und Extrapolationseigenschaften führt. Die Linearität der Parameter im Netzausgang erlaubt zudem die Anwendung eines einfachen Adaptionsverfahrens. Es wurde jedoch deutlich, dass das Einsatzgebiet des GRNN aufgrund der mit der Eingangsraumdimension stark ansteigenden Parameteranzahl auf ein- und zweidimensionale Funktionen begrenzt ist. Mit LOLIMOT wurde ein Neuronales Netz vorgestellt, das sich sowohl zur Approximation statischer Nichtlinearitäten als auch nichtlinearer dynamischer Systeme eignet. Allerdings handelt es sich hierbei um ein Neuronales Netz, das ausschließlich offline eingesetzt werden kann. Das HANN wurde speziell zur Approximation von periodischen Signalen entwickelt. Die Parameter des HANN entsprechen den Fourierkoeffizienten eines periodischen Signals und erlauben eine Aussage über dessen spektrale Zusammensetzung. Abschließend wurden verschiedene Parameteradaptionsverfahren im Überblick vorgestellt. Der rekursive Least-Squares-Algorithmus sowie das Gradientenabstiegsverfahren wurden formal erläutert.

3 Identifikation nichtlinearer Systeme

Technische Prozesse weisen im Allgemeinen ein nichtlineares dynamisches Verhalten auf. Aufgrund der Vielfalt und Komplexität nichtlinearer dynamischer Systeme war und ist deren Identifikation das Thema zahlreicher Forschungsarbeiten.

In diesem Kapitel werden zunächst einige Begriffe und Grundlagen zur Identifikation erläutert. Im Anschluss an die Identifikation linearer dynamischer Systeme wird eine Klassifikation der unterschiedlichen nichtlinearen dynamischen Systeme vorgestellt, die sich durch den Grad des strukturellen Vorwissens ergibt. Aufbauend auf dieser Klassifikation werden verschiedene Identifikationsverfahren für nichtlineare dynamische Systeme genauer erklärt und analysiert.

3.1 Modellbildung und Identifikation

Grundlage für eine Identifikation ist immer eine modellhafte Vorstellung der physikalischen Realität. Im Allgemeinen können folgende Ansätze zur Modellbildung, abhängig vom Vorwissen über das zu modellierende System unterschieden werden [4, 40]:

- **White-Box-Modelle**

White-Box-Modelle resultieren aus einer genauen theoretischen Analyse des Systemverhaltens. Diese Analyse erfolgt durch das Aufstellen von physikalischen und geometrischen Gleichungen. Charakteristisch für White-Box-Modelle ist, dass die Modellstruktur genau bekannt ist und die Modellparameter physikalischen Parametern entsprechen. Die Modellparameter können durch Messungen abgeglichen werden. White-Box-Modelle weisen eine hohe Genauigkeit auf, setzen aber voraus, dass das Systemverhalten sehr genau analysiert wurde, was in der Regel sehr zeitaufwändig ist.

- **Black-Box-Modelle**

Soll eine aufwändige theoretische Analyse vermieden werden oder sind nur unzureichende Kenntnisse über das Systemverhalten vorhanden, kann sich der experimentellen Analyse bedient werden. Das Resultat ist ein Black-Box-Modell. Charakteristisch für Black-Box-Modelle ist, dass keine oder nur sehr

wenige Vorkenntnisse über das Systemverhalten bzw. die Systemstruktur bekannt sind und die Modellparameter keine physikalische Bedeutung haben. Es wird in diesem Fall auch von einem nichtparametrischen Modell gesprochen, da das Modell nur das Ein-/Ausgangsverhalten abbildet und die physikalischen Parameter z.B. implizit in Form von Gewichtungsfunktionen oder Tabellen enthalten sind.

- **Grey-Box-Modelle**

Grey-Box-Modelle sind eine Mischung aus White-Box- und Black-Box-Modellen. Sie beinhalten im Allgemeinen Informationen aus physikalischen Gleichungen und Messdaten sowie qualitative Informationen in Form von Regeln. Grey-Box-Modellen liegt häufig eine Annahme über die Struktur des Prozesses zu Grunde. In diesem Fall wird auch von einem parametrischen Modell gesprochen, da die Parameter einer gewissen Modellvorstellung zugeordnet werden können, was nicht gleichzeitig bedeutet, dass sie den physikalischen Parametern entsprechen.

In Abb. 3.1 sind die erläuterten Begriffe noch einmal grafisch gegeneinander abgegrenzt [4, 61].

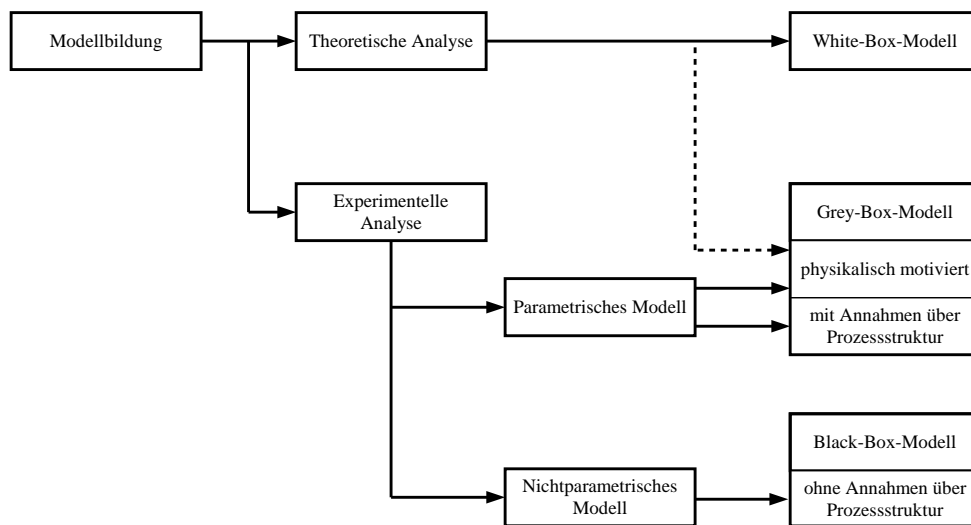


Abb. 3.1: Überblick zur Systemmodellierung

Die Wahl der Systemanalyse (theoretisch oder experimentell) hängt in erster Linie vom Grad des Vorwissens über das System und vom Verwendungszweck des Modells ab. Werden interne Systemzustände zum Aufbau einer Zustandsregelung benötigt, ist im Allgemeinen die theoretische Analyse vorzuziehen (vgl. Kapitel 3.3.2). Wird lediglich das Ein-/Ausgangsverhalten eines Modells zur Prädiktion, Simulation oder Diagnose benötigt, kann auf die experimentelle Analyse zurückgegriffen werden.

Ausgehend von einer modellhaften Vorstellung der physikalischen Realität erfolgt die Identifikation. Ziel einer Identifikation ist es, mit Hilfe gemessener Ein- und Ausgangssignale des Prozesses ein Modell zu bestimmen, welches das statische und dynamische Verhalten des Prozesses möglichst gut nachbildet. Dabei wird angenommen, dass ein eindeutiger Zusammenhang zwischen den Eingangssignalen \underline{u} , der Anregung des Prozesses, und dem Ausgangssignal y_u existiert. Aufgrund der Tatsache, dass auf jeden Prozess Störungen, wie z.B. Messrauschen, einwirken, kann nur das gestörte Ausgangssignal y zur Identifikation genutzt werden, welches als Überlagerung des ungestörten Prozessausgangs y_u mit einem Störsignal z angesehen werden kann. Abbildung 3.2 zeigt die grundsätzliche Struktur einer Identifikation.

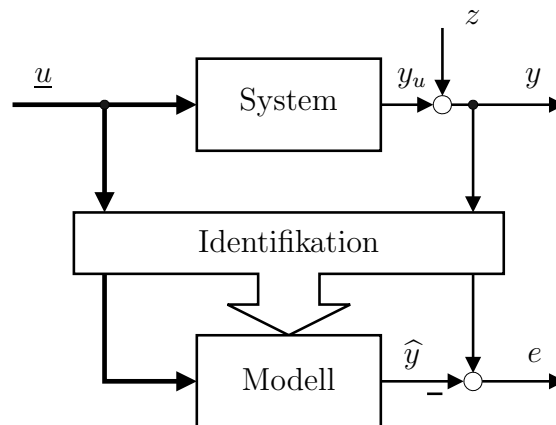


Abb. 3.2: *Prinzipielle Struktur einer Identifikation*

Jede Identifikation setzt sich aus zwei grundlegenden Schritten zusammen:

1. Strukturauswahl bzw. Strukturbestimmung
2. Adaption der Parameter

Zunächst muss für das Modell eine Struktur festgelegt bzw. bestimmt [26, 29] werden. Die Struktur gibt vor, welche funktionalen Zusammenhänge beschrieben werden können. So können z.B. durch eine lineare Modellstruktur auch nur lineare Prozesseigenschaften abgebildet werden. Prinzipiell kann festgehalten werden, dass bei der Strukturauswahl so viel Vorwissen wie möglich berücksichtigt werden sollte, um die Anzahl der Parameter klein und die Konvergenzzeiten kurz zu halten. Auf die unterschiedlichen Strukturen wird in den folgenden Kapiteln noch genauer eingegangen.

Im zweiten Schritt müssen die Parameter der gewählten Modellstruktur so angepasst werden, dass ein zwischen Prozess und Modell gebildetes Fehlersignal e möglichst klein wird. Dieser Teil wird deshalb als Parameteradaption bezeichnet und wurde in Kapitel 2.7 schon genauer erläutert. Prinzipiell besteht bei der Parameteradaption

von nichtlinearen statischen Funktionen und nichtlinearen dynamischen Systemen kein Unterschied, da es sich um ein mathematisches Optimierungsproblem handelt, dessen Lösung ausschließlich von der Form des Gleichungssystems abhängt bzw. ob die unbekannt Parameter linear oder nichtlinear in den Ausgang eingehen. Allerdings ist bei der Adaption von White-Box-Modellen in der Regel der Fehler nicht direkt messbar. Die Problematik, die sich daraus ergibt, wird in Kapitel 3.3.2 noch genauer behandelt.

Bei der Parameteradaption hat das Anregesignal einen entscheidenden Einfluss auf die Modellgüte. Die einfache Vorstellung einer Messung, bei der der Prozess während der gesamten Messzeit in Ruhe verweilt, macht deutlich, dass aus einer solchen Messung keine Informationen über das dynamische Verhalten des Systems gewonnen werden können [30]. Bei der Identifikation muss also stets darauf geachtet werden, dass der gesamte Eingangsraum, wie auch die Dynamik des Prozesses durch eine entsprechende Wahl des Eingangssignals bzw. der Eingangssignale angeregt wird. Dies wird auch als Persistent Excitation [38] bezeichnet. In der Literatur werden stochastische Signale wie z.B. das amplitudenmodulierte Pseudo-Rausch-Binärsignal [24, 25, 41] als besonders geeignet angesehen, da sie viele unterschiedliche Frequenzen und Amplituden beinhalten. Mit derartigen Eingangssignalen wird das Auffinden des globalen Minimums des Gütefunktional begünstigt.

Nach einer Identifikation stellt sich immer die Frage nach der Güte des identifizierten Modells. Hierzu muss das Modell validiert werden. Im Einzelnen sind folgende Kriterien zur Beurteilung der Modellgüte von Bedeutung [17, 51]:

- **Repräsentationsfähigkeit**

Repräsentationsfähigkeit bedeutet, dass das gewählte Neuronale Netz bzw. die gewählte Modellstruktur in der Lage ist, die Abbildungsaufgabe zu erfüllen.

- **Lernfähigkeit**

Lernfähigkeit bedeutet, dass die gewünschte Abbildung mittels eines geeigneten Lernverfahrens in das Neuronale Netz bzw. die Modellstruktur eintrainiert werden kann. In der Praxis spielen dabei Kriterien, wie Lerngeschwindigkeit (Konvergenz der Parameter), Stabilität des Lernverfahrens und Online-Fähigkeit des Lernverfahrens, eine wichtige Rolle.

- **Verallgemeinerungsfähigkeit**

Die Verallgemeinerungsfähigkeit beschreibt die Fähigkeit des trainierten Netzes bzw. Modells während der Reproduktionsphase auch Wertepaare, die nicht in der Trainingsmenge enthalten sind, hinreichend genau zu repräsentieren. Hierbei ist speziell zwischen dem Interpolations- bzw. Extrapolationsverhalten zu unterscheiden, welche die Approximationsfähigkeit des Netzes innerhalb bzw. außerhalb des Eingangsbereiches der Trainingsdaten charakterisieren.

Das vorrangige Ziel ist es, eine gute Verallgemeinerungsfähigkeit zu erreichen. Die Überprüfung der Verallgemeinerungsfähigkeit darf in der Regel nicht mit dem Trainingsdatensatz erfolgen, sondern es muss ein zweiter separater Datensatz verwendet werden. Der Grund dafür ist, dass bei zu langem Training mit dem gleichen Datensatz der Effekt des Übertrainierens (Overfitting) [41] eintritt, d.h. das Neuronale Netz bzw. das Modell spezialisiert sich durch „Auswendiglernen“ auf die optimale Reproduktion des Trainingsdatensatzes, zeigt aber eine schlechte Verallgemeinerungsfähigkeit. Im Gegensatz dazu reicht bei zu kurzem Training das angeeignete Wissen nicht aus, um das gewünschte Verhalten zu erzielen.

3.2 Identifikation linearer dynamischer Systeme

Die Identifikation linearer dynamischer Systeme ist ein umfassend erforschtes Gebiet. Dennoch sollen in diesem Kapitel einige Grundlagen erklärt werden, die zum Verständnis dieser Arbeit im Folgenden erforderlich sind. Detaillierte Ausführungen zu diesem Thema können z.B. in [33, 41, 55, 61] gefunden werden.

Ausgangspunkt für die Identifikation von linearen dynamischen Systemen im Zeitbereich ist eine allgemeine modellhafte Vorstellung des linearen Systems, wie sie in Abb. 3.3 dargestellt ist.

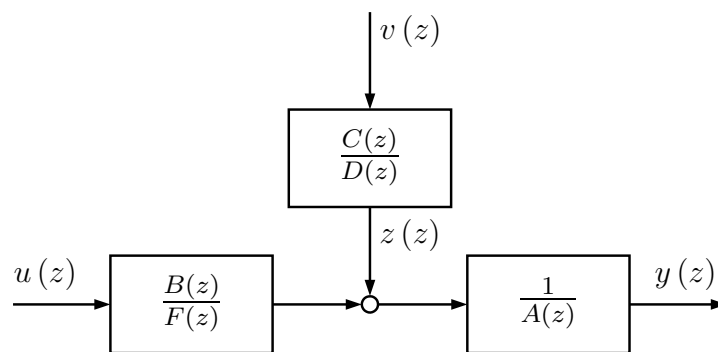


Abb. 3.3: Allgemeine lineare Modellstruktur

Dieses Modell besteht aus einem deterministischen und einem stochastischen Anteil, wie in Gl. (3.1) zu erkennen ist:

$$y(z) = \underbrace{\frac{B(z)}{F(z) \cdot A(z)}}_{\text{Eingang-übertragungsfunktion}} \cdot u(z) + \underbrace{\frac{C(z)}{D(z) \cdot A(z)}}_{\text{Rausch-übertragungsfunktion}} \cdot v(z) \quad (3.1)$$

In Gl. (3.1) bezeichnet $u(z)$ das Eingangssignal und $y(z)$ das Ausgangssignal des linearen Systems. Das Ausgangssignal wird zusätzlich durch das weiße Rauschen $v(z)$, welches durch die Rauschübertragungsfunktion gefiltert wird, beeinflusst. Die

Eingangs- und die Rauschübertragungsfunktion können einen gemeinsamen Anteil $A(z)$ haben, der in Abb. 3.3 separat eingezeichnet wurde. Diese allgemeine lineare Modellstruktur wird normalerweise in der Praxis nicht verwendet, aus ihr können durch Vereinfachung jedoch alle in der Praxis üblichen Modellstrukturen abgeleitet werden. Tabelle 3.1 zeigt einige dieser Modellstrukturen.

Modellstruktur	Modellgleichung
ARX	$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z)$
ARMAX	$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{C(z)}{A(z)} \cdot v(z)$
OE	$y(z) = \frac{B(z)}{F(z)} \cdot u(z) + v(z)$
FIR	$y(z) = B(z) \cdot u(z) + v(z)$

Tabelle 3.1: *Lineare Modellstrukturen im Überblick*

Das ARX-Modell (AutoRegressive with eXogenous input) besitzt, ebenso wie das ARMAX-Modell (AutoRegressive Moving Average with eXogenous input), ein gemeinsames Nennerpolynom $A(z)$ im deterministischen und stochastischen Anteil. Beide Modelle gehören zur Klasse der Gleichungsfehlermodelle. Im Gegensatz dazu ist beim OE-Modell (Output Error) sowie beim FIR-Modell (Finite Impulse Response) der stochastische Anteil unabhängig vom deterministischen Anteil. Diese Modelle gehören zur Klasse der Ausgangsfehlermodelle. In Abb. 3.4 ist diese Art der Klassifikation noch einmal verdeutlicht.

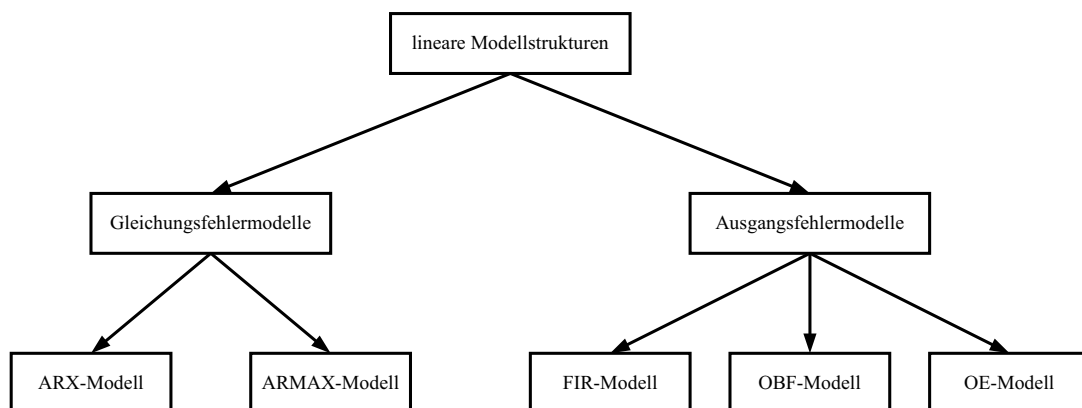


Abb. 3.4: *Klassifikation der linearen Modellstrukturen*

In Abb. 3.4 wurde zusätzlich das OBF-Modell (Orthonormal Basis Function) aufgeführt, das eine Erweiterung des FIR-Modells darstellt und in Kapitel 3.2.2.2 noch eingehender behandelt wird. Darüber hinaus gibt es noch weitere lineare Modellstrukturen, auf die jedoch nicht weiter eingegangen wird.

3.2.1 Modelle mit Ausgangsrückkopplung

In diesem Kapitel werden lineare Modelle mit Ausgangsrückkopplung, so wie es das ARX-, ARMAX- und OE-Modell sind, genauer betrachtet. Am Beispiel des ARX- und des OE-Modells soll der Unterschied zwischen einem Gleichungsfehler- und einem Ausgangsfehlermodell erläutert werden.

3.2.1.1 Autoregressive with Exogenous Input Model

Das ARX-Modell wird sehr häufig zur Identifikation linearer Systeme verwendet. Das liegt in erster Linie daran, dass der Modellausgang linear in den Parametern ist und deshalb auch lineare Adaptionsverfahren (vgl. Kapitel 2.7.1) eingesetzt werden können. In Abb. 3.5 ist die Gleichungsfehlerstruktur des ARX-Modells dargestellt.

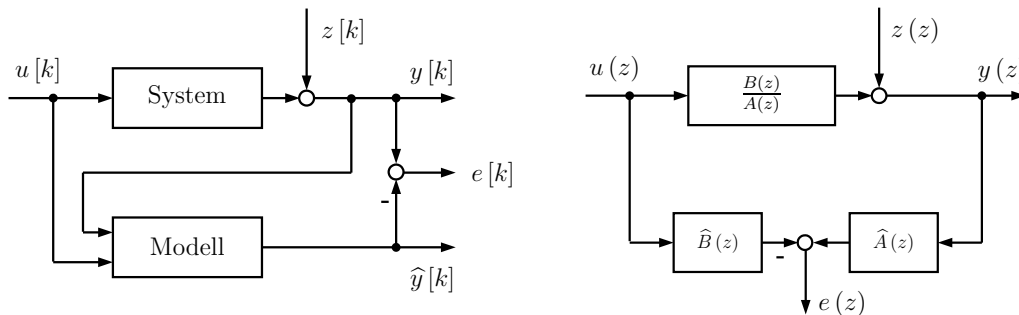


Abb. 3.5: Gleichungsfehlerstruktur des ARX-Modells

Das ARX-Modell ist definiert durch die Gleichung:

$$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z) \quad (3.2)$$

Die optimale Identifikationsgleichung ergibt sich, wenn das Fehlersignal $e(z)$ gleich dem weißen Rauschen $v(z)$ ist, d.h. es gilt:

$$e(z) = y(z) - \hat{y}(z) = v(z) \quad (3.3)$$

Daraus folgt:

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) + [1 - \hat{A}(z)] \cdot y(z) \quad (3.4)$$

Wird Gl. (3.4) in eine Differenzgleichung umgewandelt, ergibt sich¹:

$$\hat{y}[k] = b_1 \cdot u[k-1] + \dots + b_n \cdot u[k-n] - a_1 \cdot y[k-1] - \dots - a_n \cdot y[k-n] \quad (3.5)$$

Aus Gl. (3.5) wird deutlich, dass die unbekanntenen Gewichte $b_1 \dots b_n$ und $a_1 \dots a_n$ linear in den Ausgang eingehen. Außerdem ist die ARX-Identifikationsstruktur garantiert stabil, da die Identifikationsgleichung keine Rückkopplungen enthält. Der Nachteil dieser Modellstruktur ist, dass mit der ARX-Identifikationsgleichung kein paralleles Modell, sondern ein seriell-paralleles Modell bestimmt wird. In der Praxis wird jedoch oft ein paralleles Modell benötigt. Die ARX-Modellstruktur ist somit nur bedingt zur Identifikation von parallelen Modellen geeignet, da mit zunehmenden Prozessrauschen, das Identifikationsergebnis mit einem zunehmenden systematischen Fehler behaftet ist [24].

3.2.1.2 Output Error Model

Das OE-Modell gehört, wie der Name schon andeutet, zur Klasse der Ausgangsfehlermodelle. Der Vorteil der OE-Modellstruktur ist, dass mit ihr ein paralleles Modell bestimmt werden kann. Allerdings ist das OE-Modell nichtlinear in den Parametern und deswegen mit nichtlinearen Verfahren (vgl. Kapitel 2.7.2) zu adaptieren. In Abb. 3.6 ist die Ausgangsfehlerstruktur des OE-Modells dargestellt.

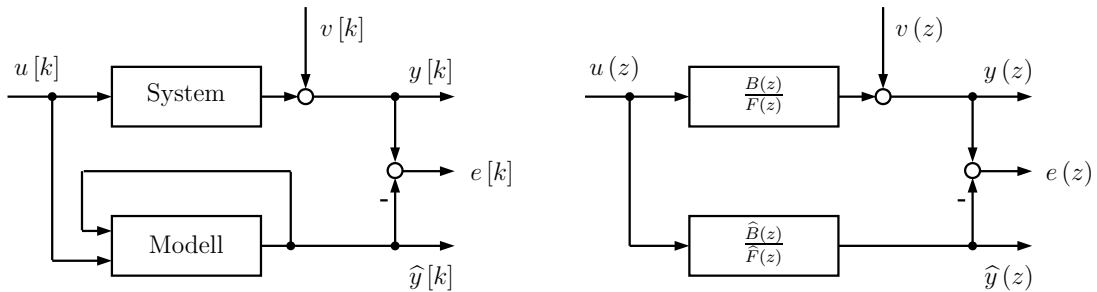


Abb. 3.6: Ausgangsfehlerstruktur des OE-Modells

Das OE-Modell aus Abb. 3.6 ist durch die folgende Gleichung definiert:

$$y(z) = \frac{B(z)}{F(z)} \cdot u(z) + v(z) \quad (3.6)$$

Die optimale Identifikationsgleichung für ein OE-Modell ergibt sich analog zu Kapitel 3.2.1.1 zu:

$$\hat{y}(z) = \frac{\hat{B}(z)}{\hat{F}(z)} \cdot u(z) \quad (3.7)$$

¹In Gl. (3.5) wird angenommen, dass der Prozess nicht sprungfähig ist und der Grad von $A(z)$ sowie $B(z)$ gleich n ist.

Die Umwandlung von Gl. (3.7) in eine Differenzgleichung verdeutlicht, warum das OE-Modell nichtlinear in den Parametern ist:

$$\hat{y}[k] = b_1 \cdot u[k-1] + \dots + b_n \cdot u[k-n] - a_1 \cdot \hat{y}[k-1] - \dots - a_n \cdot \hat{y}[k-n] \quad (3.8)$$

Im Vergleich zum ARX-Modell wurde in Gl. (3.8) der gemessene Ausgang durch den Modellausgang ersetzt. Hier liegt auch der Grund dafür, dass die OE-Modellstruktur nichtlinear in den Parametern ist, da die Vergangenheitswerte $\hat{y}[k-i]$ des Modellausgangs selbst von den zu optimierenden Parametern abhängen. Dem Vorteil, dass mit der OE-Modellstruktur ein echt paralleles Modell identifiziert werden kann, steht somit der Nachteil gegenüber, dass die Adaption der Parameter deutlich aufwändiger wird. Ein weiterer Nachteil ist, dass die Stabilität des OE-Modells aufgrund der Modellrückkopplungen nicht mehr garantiert werden kann.

3.2.2 Modelle ohne Ausgangsrückkopplung

Lineare Modelle ohne Ausgangsrückkopplung, wie z.B. die FIR- und die OBF-Modellstruktur, gehören generell zur Klasse der Ausgangsfehlermodelle. Modelle ohne Ausgangsrückkopplung beruhen prinzipiell auf der Faltungssumme, während Modelle mit Ausgangsrückkopplung auf der Differenzgleichung basieren. Daraus resultieren unterschiedliche Vor- und Nachteile für Modelle ohne Ausgangsrückkopplung.

Wie schon erwähnt gehören Modellstrukturen ohne Ausgangsrückkopplung prinzipiell zur Klasse der Ausgangsfehlermodelle, d.h. das Ergebnis einer Identifikation ist immer ein echt paralleles Modell. Im Gegensatz zum OE-Modell sind Modelle ohne Ausgangsrückkopplung aber auch linear in den Parametern, so dass lineare Adaptionsverfahren eingesetzt werden können. Ein weiterer Vorteil ist die garantierte Stabilität, da Modelle ohne Ausgangsrückkopplung nur von Eingangssignalen abhängen. Allerdings kann diese Stabilität auch ein Nachteil sein, wenn es darum geht ein instabiles oder grenzstabiles System, z.B. ein global integrierendes System², zu identifizieren. Diese Problematik wird in Kapitel 6.3 noch genauer behandelt. Die Tatsache, dass Modelle ohne Ausgangsrückkopplung nur von Eingangssignalen abhängig sind, führt zu einer weiteren angenehmen Eigenschaft. Rauschen am Prozessausgang hat keinen Einfluss auf die Eingangssignale des Identifikationsalgorithmus, so dass die Parameteradaption nur aufgrund des Fehlersignals, das Rauschen enthält, beeinträchtigt wird. Diesen Vorteilen steht wohl als gravierendster Nachteil, die hohe Anzahl an unbekanntem Parametern gegenüber. Diese ist in der Regel deutlich höher als bei Modellen mit Ausgangsrückkopplung. Im Falle von linearen Systemen kann die hohe Parameteranzahl noch als akzeptabel angesehen werden, dies ändert sich jedoch bei der Identifikation von nichtlinearen dynamischen Systemen auf der Basis von Modellen ohne Ausgangsrückkopplung.

²Unter einem global integrierenden System wird ein System mit einem Integrator am Systemausgang verstanden.

3.2.2.1 Finite Impulse Response Model

Allgemein kann ein lineares dynamisches System zeitdiskret durch die Faltungssumme beschrieben werden [62]. Der Systemausgang berechnet sich entsprechend Gl. (3.9) aus der Faltung der Impulsantwort mit vergangenen Eingangssignalen³:

$$y[k] = \sum_{i=0}^k h[i] u[k-i] \quad (3.9)$$

Mit fortschreitender Zeit wird die Anzahl der Abtastzeitpunkte k und somit der Rechenaufwand zur Berechnung der Faltungssumme immer größer. Um einen konstanten Rechenaufwand zu gewährleisten, wird die Faltungssumme bei einer oberen Grenze m unter Vernachlässigung eines Restfehlers abgeschnitten. Dies ist möglich, da für stabile Systeme gilt⁴:

$$\lim_{i \rightarrow \infty} h[i] = 0 \quad (3.10)$$

Auf die Wahl der oberen Grenze m , die auch als Antwortlänge bezeichnet wird, wird später noch genauer eingegangen. An dieser Stelle ist nur wichtig, dass die Impulsantwort durch das Abschneiden endlich wird. Ein derart motiviertes Modell wird somit als FIR-Modell bezeichnet. Abbildung 3.7 zeigt das FIR-Modell in seiner Struktur.

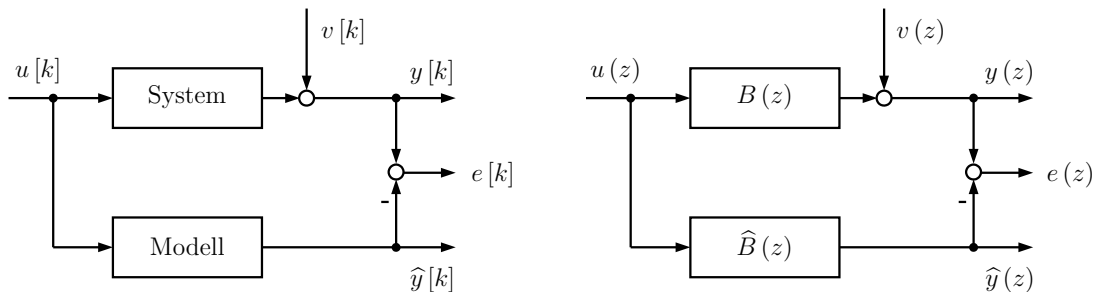


Abb. 3.7: Ausgangsfehlerstruktur des FIR-Modells

FIR-Modelle gehören grundsätzlich zur Klasse der Ausgangsfehlermodelle und sind linear in den Parametern, jedoch können prinzipbedingt nur Systeme mit abklingender Impulsantwort, d.h. stabile Systeme beschrieben werden. Das FIR-Modell aus Abb. 3.7 ist durch die folgende Gleichung definiert:

$$y(z) = B(z) \cdot u(z) + v(z) \quad (3.11)$$

Die optimale Identifikationsgleichung für ein FIR-Modell ergibt sich somit zu:

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) \quad (3.12)$$

³Der Term $h[0] \cdot u[k]$ kann vernachlässigt werden, wenn das System nicht sprunghaft ist.

⁴Die Gleichung (3.10) gilt nicht für grenzstabile Systeme.

Die Umwandlung von Gl. (3.12) führt zu:

$$\hat{y}[k] = b_1 \cdot u[k-1] + b_2 \cdot u[k-2] + \dots + b_m \cdot u[k-m] \quad (3.13)$$

Gleichung (3.13) verdeutlicht noch einmal, dass das FIR-Modell aufgrund der fehlenden Ausgangsrückkopplung sowohl linear in den Parametern ist als auch zur Klasse der Ausgangsfehlermodelle gehört.

Den oben genannten Vorteilen steht jedoch der gravierende Nachteil gegenüber, dass die Anzahl der unbekannt Parameter sehr hoch ist. Die Anzahl der Parameter ist nach Gl. (3.13) identisch mit der Antwortlänge m , d.h. $p_{FIR} = m$. Die Antwortlänge hängt wiederum von der Systemdynamik und der Abtastzeit h ab. Als Faustformel wird in [30] $m \simeq \frac{T_{99.9}}{h}$ angegeben⁵. Im Vergleich dazu ist die Anzahl unbekannter Parameter bei Modellen mit Ausgangsrückkopplung durch $p_{ARX/OE} = 2 \cdot n$ gegeben. Ein Beispiel soll die Größenordnung der Anzahl unbekannter Parameter verdeutlichen. Betrachtet wird ein System zweiter Ordnung mit den auf die Abtastzeit $h = 1s$ normierten Zeitkonstanten $T_1 = 5$ und $T_2 = 7$. Für dieses System ergeben sich die Parameteranzahlen $p_{ARX/OE} = 4$ und $p_{FIR} = 54$. Diese hohe Anzahl an Parametern stellt bereits bei linearen Modellen ein Problem dar, was sich bei der Erweiterung des FIR-Modells auf nichtlineare Systeme noch verstärkt. Deshalb wurde bereits beim FIR-Modell nach Möglichkeiten gesucht, die Parameteranzahl zu reduzieren, woraus das OBF-Modell entstand.

3.2.2.2 Orthonormal Basis Function Model

Durch die Einführung von orthonormalen Basisfunktionen lässt sich die Anzahl unbekannter Parameter von FIR-Modellen deutlich reduzieren. Die Idee von OBF-Modellen ist, die Impulsantwort durch eine gewichtete Überlagerung von orthonormalen Basisfunktionen zu beschreiben⁶. In Abb. 3.8 ist dieses Prinzip verdeutlicht. Mit der gewichteten Überlagerung orthonormaler Basisfunktionen kann die Impulsantwort (gestrichelte Kurve) eines Systems nachgebildet werden. Die Herausforderung dabei ist, Basisfunktionen zu finden, mit denen es möglich ist, die Impulsantwort eines Systems zu rekonstruieren. Hierzu ist ein gewisses Vorwissen über die Dynamik des Systems erforderlich, das jedoch durch die Analyse der Sprungantwort leicht gewonnen werden kann.

Das OBF-Modell kann wie folgt definiert werden:

$$y(z) = b_1 \cdot L_1(z) \cdot u(z) + b_2 \cdot L_2(z) \cdot u(z) + \dots + b_{m_r} \cdot L_{m_r}(z) \cdot u(z) + v(z) \quad (3.14)$$

$L_1(z) \dots L_{m_r}(z)$ kennzeichnet die Übertragungsfunktionen der orthonormalen Filter. $m_r \in \mathbb{N}$ ist die Anzahl der orthonormalen Filter bzw. der orthonormalen Basis-

⁵ $T_{99.9}$ ist die Zeit, bis 99.9% des Endwertes der Sprungantwort eines Systems erreicht sind.

⁶Das FIR-Modell kann auch als gewichtete Überlagerung von Basisfunktionen angesehen werden. Die Basisfunktionen sind in diesem Fall Dirac-Impulse, die sich nicht überlappen.

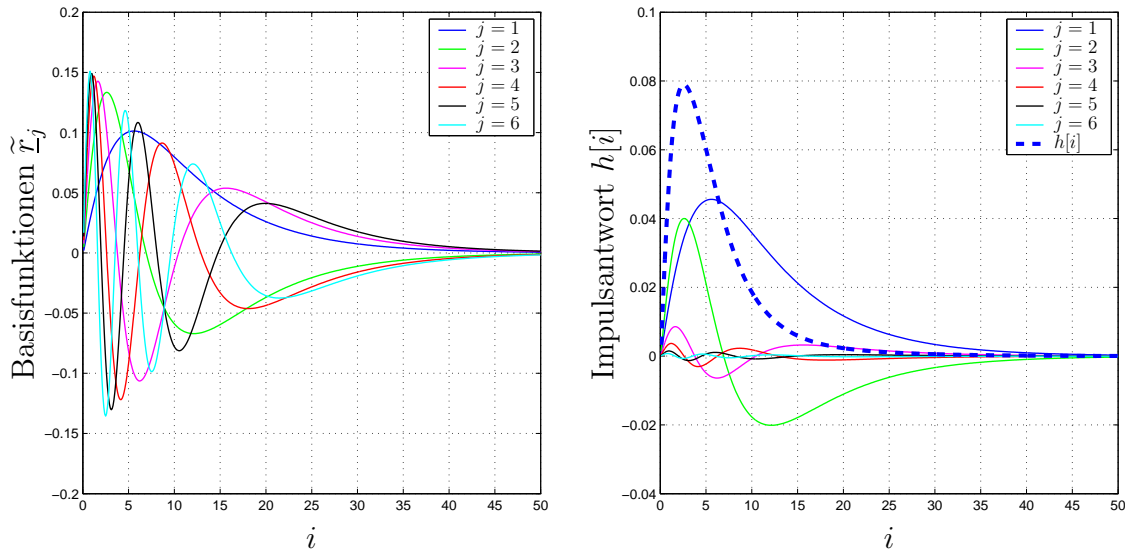


Abb. 3.8: Orthonormale Basisfunktionen (links) und Überlagerung gewichteter Basisfunktionen zur Impulsantwort (rechts)

funktionen, für die die Beziehung $m_r \leq m$ gilt. Die Impulsantworten der orthonormalen Filter $L_j(z)$ stellen die orthonormalen Basisfunktionen dar. Es gilt:

$$L_j(z) = \sum_{i=1}^{\infty} \tilde{r}_j[i] \cdot z^{-i} \quad \text{mit} \quad j = 1 \dots m_r \quad (3.15)$$

Das OBF-Modell nach Gl. (3.14) kann in ein FIR-Modell überführt werden, wenn für die orthonormalen Filter $L_1(z) = z^{-1} \dots L_m(z) = z^{-m}$ mit $m_r = m$ eingesetzt wird. Die Wahl der orthonormalen Filter $L_j(z)$ kann als das Einbringen von Vorwissen über die Dynamik des Systems betrachtet werden. In der Literatur sind verschiedene Filter bekannt. Laguerre-Filter [67] eignen sich für stark gedämpfte Systeme, da sie auf Vorwissen über einen reellen Pol des Systems beruhen. Umgekehrt eignen sich Kautz-Filter [68] für schwach gedämpfte, oszillierende Systeme, da sie Vorwissen über ein konjugiert komplexes Polpaar beinhalten. In [14, 41] werden sog. verallgemeinerte Filter vorgestellt, die es erlauben eine beliebige Anzahl von reellen Polen und konjugiert komplexen Polpaaren zu berücksichtigen. Laguerre- und Kautz-Filter sind als Spezialfälle in diesen verallgemeinerten Filtern enthalten.

In [27] werden als Basisfunktionen orthonormalisierte verzerrte Sinusfunktionen vorgeschlagen. Nach [30] eignen sich diese Basisfunktionen sowohl für schwach als auch stark gedämpfte Prozesse, weshalb sie im Folgenden genauer betrachtet werden. Die noch nicht orthonormierten verzerrten Sinusfunktionen lassen sich im Zeitbereich mit $i = 1 \dots m$ und $j = 1 \dots m_r$ berechnen durch:

$$r_j[i] = \frac{1}{\sqrt{\frac{m}{2}}} \cdot \sin \left[j \cdot \pi \cdot \left(1 - e^{-\frac{i-0,5}{\zeta}} \right) \right] \quad (3.16)$$

In Gl. (3.16) bezeichnet $m \in \mathbb{N}$ die Antwortlänge und $m_r \in \mathbb{N}$ die Anzahl der Basisfunktionen. Mit dem Formfaktor $\zeta \in \mathbb{R}^+$ ist es möglich den Grad der Verzerrung

der Basisfunktionen festzulegen und diese auf die Prozessdynamik anzupassen. Die Basisfunktionen können wie folgt zu einer Matrix zusammengefasst werden:

$$\mathbf{R} = \begin{bmatrix} \underline{r}_1^T \\ \underline{r}_2^T \\ \vdots \\ \underline{r}_{m_r}^T \end{bmatrix} \quad \text{mit} \quad \underline{r}_j^T = \left[r_j[1], r_j[2], \dots, r_j[m] \right] \quad (3.17)$$

$\mathbf{R} \in \mathbb{R}^{m_r \times m}$ wird als Rekonstruktionsmatrix bezeichnet und enthält die Basisfunktionen zeilenweise. Die Rekonstruktionsmatrix ist nicht orthogonal und nicht normiert, dies kann gezeigt werden durch $\mathbf{R}^T \mathbf{R} \neq \mathbf{E}$. Diese Orthonormalität ist jedoch wichtig, da jede Basisfunktion ihren eigenen Beitrag zur Rekonstruktion der Impulsantwort leisten soll. Durch Orthonormalisierung ergibt sich das tatsächliche Orthonormalsystem, die orthonormierte Rekonstruktionsmatrix $\tilde{\mathbf{R}}$:

$$\begin{aligned} \mathbf{R} &= \mathbf{C}^T \tilde{\mathbf{R}} & \mathbf{C} &\in \mathbb{R}^{m_r \times m_r} \\ \mathbf{R} \mathbf{R}^T &= \mathbf{C}^T \tilde{\mathbf{R}} \tilde{\mathbf{R}}^T \mathbf{C} = \mathbf{C}^T \mathbf{C} & & (3.18) \\ \Rightarrow \tilde{\mathbf{R}} &= (\mathbf{C}^T)^{-1} \mathbf{R} & \tilde{\mathbf{R}} &\in \mathbb{R}^{m_r \times m} \end{aligned}$$

Die Berechnung der quadratischen Matrix \mathbf{C} ist in der Literatur auch als Cholesky-Zerlegung bekannt.

Die in Gl. (3.16) definierten Basisfunktionen eignen sich gut für die Identifikation von Prozessen der Ordnungen $n \geq 2$. Bei Systemen mit der Ordnung $n = 1$ entspricht die Impulsantwort einer abklingenden Exponentialfunktion. Durch die Einführung einer zusätzlichen Grundbasisfunktion [30] kann die Impulsantwort für den Fall $n = 1$ besser rekonstruiert werden. Die erweiterte Rekonstruktionsmatrix ergibt sich zu:

$$r_0[i] = \frac{1}{\sqrt{\frac{m}{2}}} \cdot e^{-\frac{j-0.5}{\zeta}} \quad \mathbf{R} = \begin{bmatrix} \underline{r}_0^T \\ \underline{r}_1^T \\ \vdots \\ \underline{r}_{m_r}^T \end{bmatrix} \quad \mathbf{R} \in \mathbb{R}^{(m_r+1) \times m} \quad (3.19)$$

Die orthonormierten Sinusfunktionen sind mit und ohne die abklingende Exponentialfunktion für die Einstellwerte $m = 50$, $\zeta = 9$, $m_r = 5$ in Abb. 3.9 dargestellt.

Für die richtige Wahl des Formfaktors ζ sowie der Anzahl orthonormalisierter verzerrter Basisfunktionen m_r wurden in [30] die Faustformeln nach Tab. 3.2 festgelegt. Die Zeitkonstanten T_{63} und T_{95} bezeichnen die Zeit bis 63 % bzw. 95 % des Endwertes der Sprungantwort eines Systems erreicht sind. Diese Zeitkonstanten müssen noch auf die Abtastzeit h bezogen werden.

In Tab. 3.2 ist für die Basisfunktionenanzahl m_r ein Wert empfohlen. In der Regel führen mehr Basisfunktionen auch zu einem besseren Modell. Hier ist im Einzelfall zwischen Aufwand und Nutzen abzuwägen. Die Identifikationsgleichung für das OBF-Modell ergibt sich somit nach der Einführung orthonormaler Basisfunktionen zu:

$$\hat{y}[k] = b_1 \cdot \tilde{\underline{r}}_1^T \cdot \underline{u}[k] + b_2 \cdot \tilde{\underline{r}}_2^T \cdot \underline{u}[k] + \dots + b_{m_r} \cdot \tilde{\underline{r}}_{m_r}^T \cdot \underline{u}[k] \quad (3.20)$$

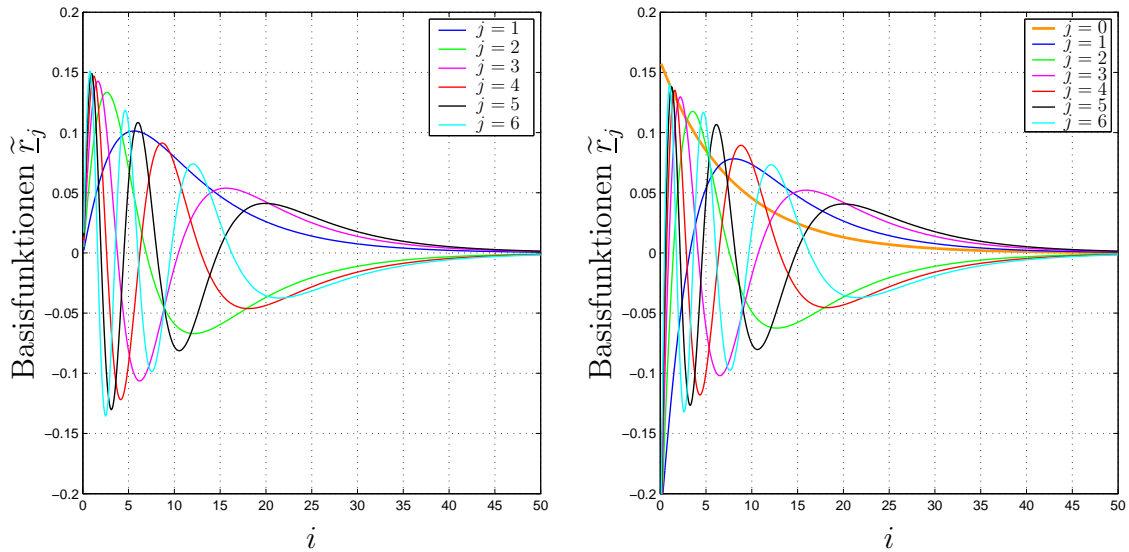


Abb. 3.9: Orthonormale verzerrte Sinusfunktionen ohne (links) und mit (rechts) abklingender Exponentialfunktion

Dämpfung	Formfaktor	Basisfunktionsanzahl
$D > 0.7$	$\zeta \simeq \frac{T_{63}}{h}$	$m_r \simeq 6$
$D < 0.7$	$\zeta \simeq \frac{T_{95}}{h}$	$m_r \simeq \frac{\zeta}{2}$

Tabelle 3.2: Faustformeln zur Wahl des Formfaktors ζ und der Basisfunktionsanzahl m_r abhängig vom Dämpfungsgrad D

Die Vektoren $\tilde{\underline{r}}_j^T \in \mathbb{R}^{1 \times m}$ bezeichnen die orthonormierten Basisfunktionen. Der Vektor $\underline{u}[k] \in \mathbb{R}^{m \times 1}$ enthält m Vergangenheitswerte von $u[k]$ und ergibt sich zu:

$$\underline{u}^T[k] = \left[u[k-1], u[k-2], \dots, u[k-m] \right] \quad (3.21)$$

Werden die unbekannt Parameter b_i zu einem Parametervektor $\underline{\theta}$ zusammengefasst, vereinfacht sich Gl. (3.20) mit Hilfe der orthonormierten Basisfunktionsmatrix $\tilde{\mathbf{R}}$ und das optimale OBF-Modell ergibt sich zu:

$$\hat{y}[k] = \underline{\theta}^T \cdot \tilde{\mathbf{R}} \cdot \underline{u}[k] \quad (3.22)$$

Das vorgestellte OBF-Modell überwindet somit den Nachteil der hohen Parameteranzahl des FIR-Modells. Anstatt $p_{FIR} = m$ müssen nur noch $p_{OBF} = m_r$ Parameter bestimmt werden. Das OBF-Modell vereinigt als einziges Modell alle positiven Eigenschaften von Gleichungsfehlermodellen und Ausgangsfehlermodellen, d.h. es gehört zur Klasse der Ausgangsfehlermodelle, es ist linear in den Parametern, die Stabilität bleibt garantiert, da die Basisfunktionen vorab berechnet werden und die

Parameteranzahl bleibt gering. Diese Eigenschaften werden bei der Erweiterung des OBF-Modells auf nichtlineare dynamische Systeme in Kapitel 3.3.4 eine entscheidende Rolle spielen.

3.3 Identifikation nichtlinearer dynamischer Systeme

Die Identifikationsverfahren für nichtlineare dynamische Systeme stellen in der Regel eine Erweiterung linearer Verfahren dar. Im Gegensatz zu den linearen Systemen, wo ausschließlich SISO-Systeme (Single Input – Single Output) betrachtet werden, gewinnen bei den nichtlinearen Systemen die MISO-Systeme (Multiple Input – Single Output) oder allgemein Mehrgrößensysteme an Bedeutung.

3.3.1 Klassifikation nichtlinearer dynamischer Systeme

Zur Klassifikation und Beschreibung von nichtlinearen dynamischen Systemen ist der Grad des Vorwissens über die Struktur von entscheidender Bedeutung. Im Folgenden soll ein Überblick über die Darstellungsformen nichtlinearer dynamischer Systeme gegeben werden. Diese stellen die Grundlage für die Ableitung nichtlinearer Identifikationsverfahren dar.

3.3.1.1 Nichtlineare Zustandsdarstellung

Ist die Struktur sowie der nichtlineare Einfluss einer Strecke genau bekannt, kann die Strecke in eine nichtlineare Zustandsdarstellung der folgenden Form überführt werden [52]:

$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A} \cdot \underline{x} + \underline{b} \cdot u + \underline{k} \cdot \mathcal{NL}(\underline{x}, u) \\ y &= \underline{c}^T \cdot \underline{x} + d \cdot u\end{aligned}\tag{3.23}$$

Die Nichtlinearität $\mathcal{NL}(\underline{x}, u)$ ist statisch und kann von der Eingangsgröße u sowie den Systemzuständen \underline{x} abhängen. Der Eingriff der statischen Nichtlinearität kann mit dem Vektor \underline{k} beschrieben werden.

Die exakte Systemkenntnis erlaubt den Entwurf eines nichtlinearen Beobachters, mit dem alle Streckenzustände beobachtet werden können. Gleichzeitig kann die unbekannt statische Nichtlinearität mit einem statischen Funktionsapproximator (vgl. Kapitel 2) identifiziert werden. Zum Beobachterentwurf muss das System durch \mathbf{A} , \underline{b} , \underline{c}^T , d sowie dem Einkopplungsvektor \underline{k} der Nichtlinearität bekannt sein. Die Thematik des nichtlinearen Beobachterentwurfs wird ausführlich in Kapitel 3.3.2 behandelt.

3.3.1.2 Blockorientierte nichtlineare Modelle

Blockorientierte nichtlineare Modelle werden dann verwendet, wenn nur ein grobes Strukturvorwissen des Systems vorhanden ist. In Abb. 3.10 sind einige blockorientierte nichtlineare Modelle dargestellt.

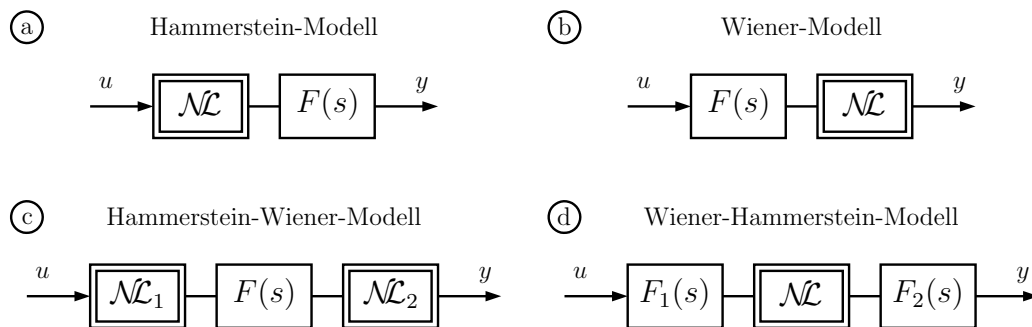


Abb. 3.10: Blockorientierte nichtlineare Modelle

Sie alle entstehen durch die Kombination aus statischen Nichtlinearitäten \mathcal{NL}_i mit dynamischen Übertragungsfunktionen $F_i(s)$. Diese nichtlinearen Modellstrukturen, besonders das Hammerstein- und das Wiener-Modell, stellen anerkannte nichtlineare Prozesse dar. Das Hammerstein-Modell (vgl. Abb. 3.10 a) setzt sich aus einer statischen Nichtlinearität \mathcal{NL} , gefolgt von einer dynamischen Übertragungsfunktion $F(s)$, zusammen. Im Gegensatz dazu wird beim Wiener-Modell (vgl. Abb. 3.10 b) die statische Nichtlinearität \mathcal{NL} durch die vorgeschaltete Übertragungsfunktion angeregt. Beide Modelle werden in Kapitel 3.3.4.2 noch genauer behandelt. Weiterhin sind auch Mischmodelle beider Ansätze denkbar, wie z.B. das in Abb. 3.10 c gezeigte Hammerstein-Wiener-Modell oder das in Abb. 3.10 d dargestellte Wiener-Hammerstein-Modell. Identifikationsverfahren auf Basis von blockorientierten nichtlinearen Modellen werden in Kapitel 3.3.4 genauer betrachtet.

3.3.1.3 Allgemeine nichtlineare Systembeschreibung

In Abb. 3.11 ist die Beschreibung eines nichtlinearen dynamischen Systems in seiner allgemeinsten Form dargestellt.

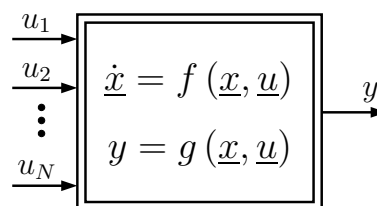


Abb. 3.11: Allgemeine nichtlineare Systembeschreibung

Das nichtlineare dynamische System lässt sich wie folgt beschreiben:

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad y = g(\underline{x}, \underline{u}) \quad (3.24)$$

In den Gleichungen (3.24) stellen $f(\underline{x}, \underline{u})$ und $g(\underline{x}, \underline{u})$ unbekannte nichtlineare Funktionen dar, die von dem Eingangsvektor \underline{u} und dem Zustandsvektor \underline{x} abhängig sind. Sind diese Funktionen nicht bekannt bedeutet das, dass kein Strukturwissen vorhanden ist. In diesem Fall muss das System mit einem Black-Box-Verfahren identifiziert werden. In Kapitel 3.3.3 wird kurz das Prinzip solcher Verfahren vorgestellt.

3.3.2 Neuronaler Beobachter

Am Lehrstuhl für Elektrische Antriebssysteme der TU München wurde in den vergangenen Jahren eine umfangreiche Theorie zur Identifikation nichtlinearer dynamischer Systeme mittels Neuronaler Beobachter entwickelt [32, 47, 52, 57, 58]. Der Neuronale Beobachter hat die Aufgabe alle Streckenzustände zu schätzen sowie die auftretende statische Nichtlinearität \mathcal{NL} mit einem Funktionsapproximator zu lernen. Die identifizierte Nichtlinearität $\widehat{\mathcal{NL}}$, die das dynamische Verhalten des Systems beeinflusst, kann in einem nachfolgenden Reglerentwurf berücksichtigt werden. Der Neuronale Beobachter kann auf eine bestimmte Klasse von nichtlinearen dynamischen Systemen angewendet werden. Diese Systeme werden als Systeme mit isolierter statischer Nichtlinearität bezeichnet und sind durch Gl. (3.23) definiert. In Abb. 3.12 ist das Blockschaltbild eines Systems mit isolierter statischer Nichtlinearität in Zustandsdarstellung gezeigt.

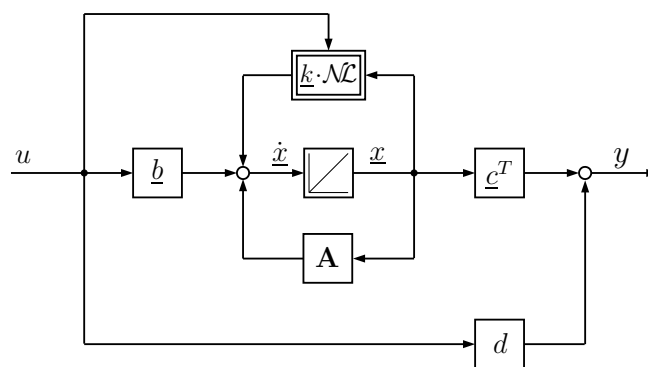


Abb. 3.12: Blockschaltbild einer Strecke mit statischer Nichtlinearität

Zum Beobachterentwurf muss \mathbf{A} , \underline{b} , \underline{c}^T , d und der Einkopplungsvektor \underline{k} der Nichtlinearität konstant und bekannt sein. Das bedeutet, dass das System in seiner Struktur genau gegeben ist. Außerdem muss die Nichtlinearität statischer Art sein. Für den Entwurf eines Neuronalen Beobachters stellt sich die Aufgabe, diese statische Nicht-

linearität durch einen statischen Funktionsapproximator⁷ zu identifizieren. Dabei ist zu unterscheiden, ob der Eingangsraum des Beobachters messbar ist oder nicht.

3.3.2.1 Beobachterentwurf bei messbarem Eingangsraum

Der Beobachterentwurf bei messbarem Eingangsraum geht davon aus, dass alle Zustände, von denen die statische Nichtlinearität $\mathcal{NL}(\underline{x}, u)$ abhängig ist, messbar sind. In der Regel hängt die Nichtlinearität nicht von allen Zuständen \underline{x} ab⁸. Zur Unterscheidung sollen die Zustände von denen die Nichtlinearität abhängt mit $\underline{x}_{\mathcal{NL}} \in \underline{x}$ bezeichnet werden. Das Neuronale Netz wird in einen Zustandsbeobachter eingebettet und hat die Aufgabe, die Nichtlinearität nachzubilden. Die statische Nichtlinearität kann nur identifiziert werden, wenn ihre Auswirkungen am Streckenausgang sichtbar sind. Die Sichtbarkeit einer Nichtlinearität am Ausgang einer SISO–Strecke ist in [52] definiert und genau dann gegeben, wenn die Übertragungsfunktion $H_S(s)$

$$H_S(s) = \frac{y(s)}{y_{\mathcal{NL}}(s)} = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot \underline{k} \quad (3.25)$$

vom Angriffspunkt der Nichtlinearität zum Streckenausgang ungleich Null ist⁹. Der Zustandsbeobachter kann nun wie folgt angesetzt werden [52]:

$$\begin{aligned} \hat{\underline{x}} &= \mathbf{A} \cdot \hat{\underline{x}} + \underline{b} \cdot u + \underline{k} \cdot \widehat{\mathcal{NL}}(\underline{x}_{\mathcal{NL}}, u) - \underline{l} \cdot e \\ \hat{y} &= \underline{c}^T \cdot \hat{\underline{x}} + d \cdot u \end{aligned} \quad (3.26)$$

Der Beobachterfehler ist definiert zu¹⁰:

$$e(s) = \hat{y}(s) - y(s) \quad (3.27)$$

In Abb. 3.13 ist die Beobachterstruktur in Zustandsdarstellung gezeigt, wobei der Vektor $\underline{x}_{\mathcal{NL}}$ den Eingangsraum der Nichtlinearität kennzeichnet. Der Rückführvektor \underline{l} muss so dimensioniert werden, dass die Beobachterpole der Matrix $(\mathbf{A} - \underline{l} \cdot \underline{c}^T)$ negativen Realteil besitzen und links der Streckenpole liegen. Die Dimensionierung kann wie im linearen Fall mittels LQ–Optimierung oder durch Polvorgabe erfolgen [34]. Der Beobachter dient zur Schätzung aller nicht messbaren Systemzustände und der Identifikation der statischen Nichtlinearität. Zur Adaption der Parameter wird der Beobachterfehler e verwendet, da der Fehler $e_{\mathcal{NL}}(s) = \hat{y}_{\mathcal{NL}}(s) - y_{\mathcal{NL}}(s)$ ebenfalls nicht messbar ist. Aus diesem Grund muss zur Ableitung eines stabilen Lerngesetzes die Fehlerübertragungsfunktion $H(s)$ berechnet werden. $H(s)$ beschreibt das Übertragungsverhalten des Beobachters vom Angriffspunkt der Nichtlinearität auf das Fehlersignal e . Die Fehlerübertragungsfunktion $H(s)$ hängt sowohl von den Streckenparametern als auch vom Rückführvektor \underline{l} ab und wird nach [32] wie folgt berechnet:

⁷Für diese Aufgabenstellung hat sich das GRNN besonders bewährt.

⁸Wenn dies der Fall wäre und alle Zustände \underline{x} messbar sind, wäre ein Beobachter überflüssig.

⁹Es gilt $y_{\mathcal{NL}}(s) = \mathcal{L}\{\mathcal{NL}(\underline{x}_{\mathcal{NL}}, u)\}$ bzw. $\hat{y}_{\mathcal{NL}}(s) = \mathcal{L}\{\widehat{\mathcal{NL}}(\underline{x}_{\mathcal{NL}}, u)\}$.

¹⁰Das Fehlersignal ist in diesem Kapitel analog zur bestehenden Literatur mit umgekehrtem Vorzeichen definiert. Desweiteren gilt $e(s) = \mathcal{L}\{e\}$.

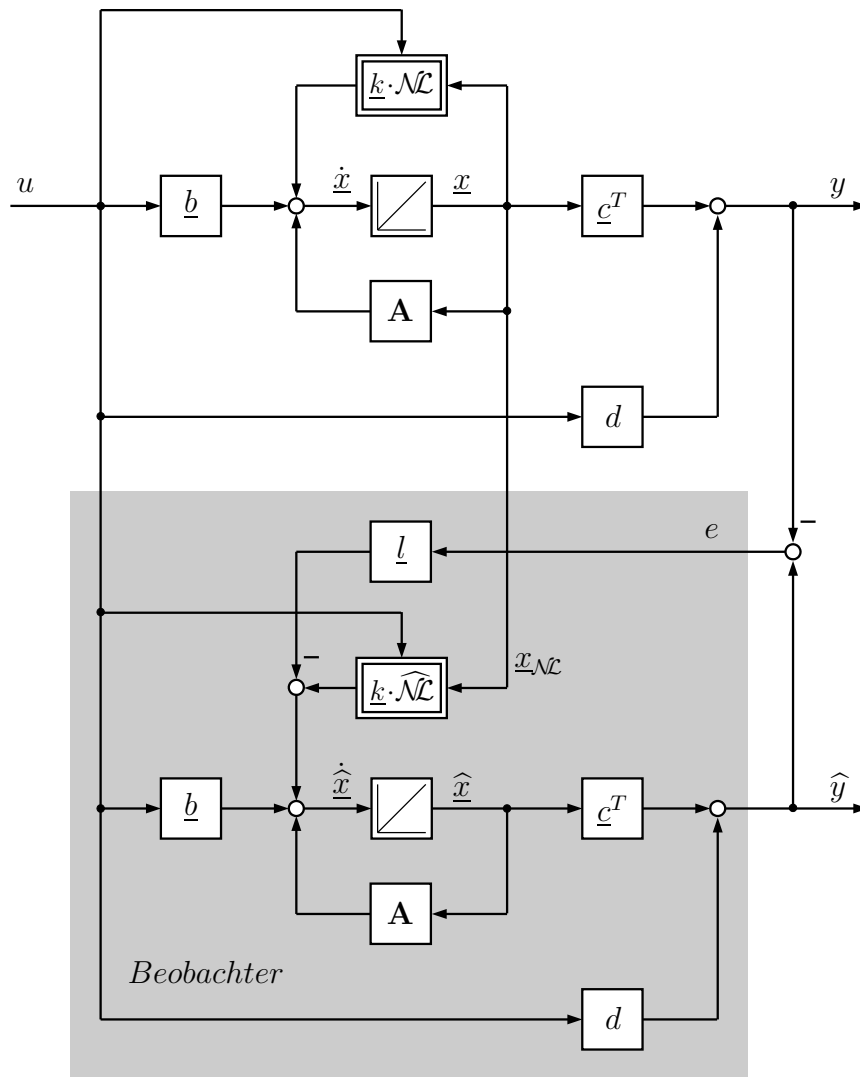


Abb. 3.13: *Neuronaler Zustandsbeobachter bei messbarem Eingangsraum der statischen Nichtlinearität*

$$H(s) = \frac{e(s)}{e_{\mathcal{NL}}(s)} = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)^{-1} \cdot \underline{k} = \frac{\det \begin{bmatrix} s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T & \vdots & -\underline{k} \\ \dots & \dots & \dots \\ \underline{c}^T & \vdots & 0 \end{bmatrix}}{\det(s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)} \quad (3.28)$$

Bei der Ableitung eines stabilen Lerngesetzes müssen zwei Fälle unterschieden werden. Abhängig davon, ob die Fehlerübertragungsfunktion $H(s)$ die SPR-Bedingung¹¹ erfüllt, muss das Fehlermodell 3 bzw. 4 angewandt werden. Eine Übertragungsfunktion $H(s)$ ist eine streng positive reelle Funktion, wenn gilt [38, 64]:

¹¹SPR ... strictly positive real

1. Alle Pol- und Nullstellen von $H(s)$ besitzen einen negativen Realteil.
2. $\operatorname{Re}\{H(j\omega)\} > 0 \quad \forall \omega \geq 0$

Anschaulich bedeutet die SPR-Bedingung, dass eine SPR-Übertragungsfunktion im Bodediagramm keine Phasendrehung um mehr als 90° erzeugt. Dies ist z.B. bei einer asymptotisch stabilen PT_1 -Übertragungsfunktion immer gegeben.

Erfüllt die Fehlerübertragungsfunktion die SPR-Bedingung kann unter der Voraussetzung, dass der verwendete Funktionsapproximator linear in den Parametern ist, das Fehlermodell 3 angewendet werden. In diesem Fall errechnet sich der Systemausgang y mit dem optimalen und konstanten Parametervektor $\underline{\theta}$ zu:

$$y(s) = H(s) \cdot y_{\mathcal{NL}}(s) = H(s) \cdot \mathcal{L} \{ \underline{\theta}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u) \} \quad (3.29)$$

Für den Beobachteraussgang ergibt sich unter der Annahme, dass sich der Parametervektor $\hat{\underline{\theta}}$ nur langsam im Vergleich zum System ändert:

$$\hat{y}(s) = H(s) \cdot \hat{y}_{\mathcal{NL}}(s) = H(s) \cdot \mathcal{L} \{ \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u) \} \quad (3.30)$$

Damit folgt für den Lernfehler mit dem Parameterfehlervektor $\underline{\Phi}$:

$$e(s) = H(s) \cdot \mathcal{L} \{ (\hat{\underline{\theta}}^T - \underline{\theta}^T) \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u) \} = H(s) \cdot \mathcal{L} \{ \underline{\Phi}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u) \} \quad (3.31)$$

In diesem Fall ergibt sich das Lerngesetz für die Gewichte des Neuronalen Netzes, wie beim Gradientenabstiegsverfahren nach Gl. (2.24) zu:

$$\hat{\underline{\theta}} = -\eta \cdot e \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)$$

In Abb. 3.14 ist das Fehlermodell 3 noch einmal grafisch veranschaulicht.

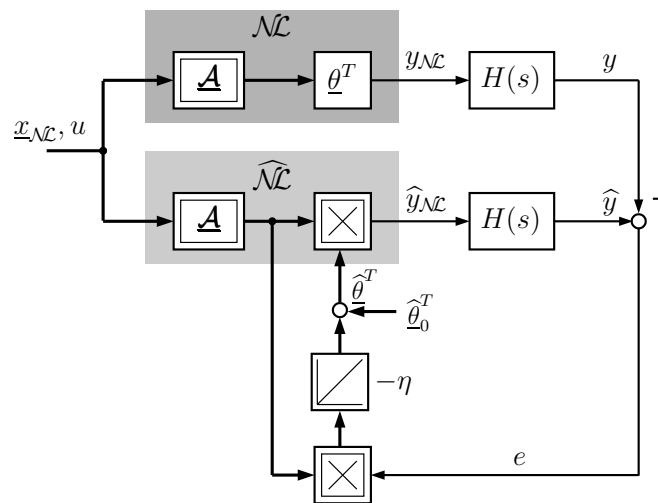


Abb. 3.14: Lerngesetz nach Fehlermodell 3

Im allgemeinen Fall einer Fehlerübertragungsfunktion $H(s)$, die die SPR-Bedingung nicht erfüllt, muss das Fehlermodell 4 aus [38] angewendet werden. Das Fehlermodell 4 ist in Abb. 3.15 dargestellt.

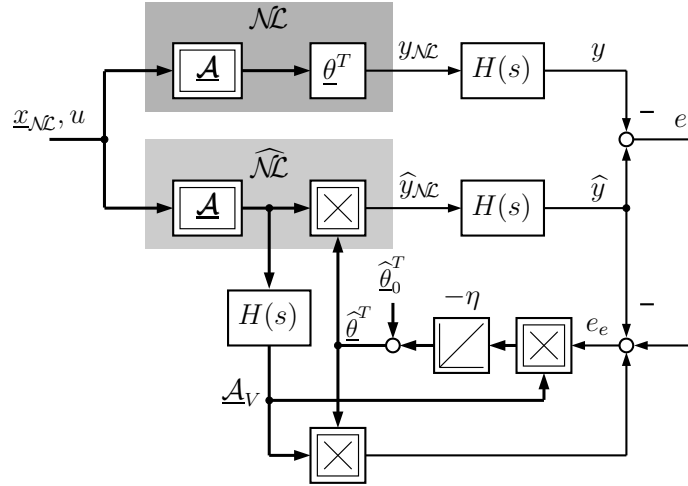


Abb. 3.15: Lerngesetz nach Fehlermodell 4

Zur Adaption der unbekannt Parameter wird der erweiterte Lernfehler $e_e(s)$ eingeführt:

$$e_e(s) = e(s) + e_z(s) \quad (3.32)$$

mit dem zusätzlichen Fehler

$$e_z(s) = \hat{\underline{\theta}}^T \cdot H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} - H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} \quad (3.33)$$

Bei der Definition des zusätzlichen Fehlers wird wiederum angenommen, dass sich der Parametervektor $\hat{\underline{\theta}}$ nur sehr langsam ändert. Für den erweiterten Fehler folgt:

$$\begin{aligned} e_e(s) &= e(s) + \hat{\underline{\theta}}^T \cdot H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} - H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} \\ &= \left[H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} - H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} \right] + \\ &+ \hat{\underline{\theta}}^T \cdot H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} - H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} \\ &= \hat{\underline{\theta}}^T \cdot H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} - H(s) \cdot \mathcal{L}\{\hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} \end{aligned} \quad (3.34)$$

Da der optimale Parametervektor $\underline{\theta}$ der zu approximierenden nichtlinearen Funktion konstant ist, kann Gl. (3.34) umgeformt werden:

$$e_e(s) = \left(\hat{\underline{\theta}}^T - \underline{\theta}^T \right) \cdot H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\} = \underline{\Phi}^T \cdot \underbrace{H(s) \cdot \mathcal{L}\{\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)\}}_{\text{verzögerte Aktivierung}} = \underline{\Phi}^T \cdot \underline{\mathcal{A}}_V(s) \quad (3.35)$$

Im Zeitbereich lautet das Lerngesetz somit¹²:

$$\hat{\underline{\theta}} = -\eta \cdot e_e \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_V(s) \} \quad (3.36)$$

Dieser Ansatz wird auch als Verfahren der verzögerten Aktivierung bezeichnet [32, 57]. Durch die Filterung der Aktivierungsfunktionen mit der Fehlerübertragungsfunktion $H(s)$ bewirken auch Signalanteile, die eine Phasendrehung um mehr als 90° verursachen, eine Adaption der Gewichte in die richtige Richtung.

Für die beschriebenen Lernstrukturen und Fehlermodelle ist in [32] ein Stabilitätsbeweis nach Lyapunov geführt, d.h. die Stabilität des Lerngesetzes kann mathematisch bewiesen werden. Dies ist ein wichtiges Merkmal des Neuronalen Beobachters und soll deshalb an dieser Stelle noch einmal betont werden.

3.3.2.2 Beobachterentwurf bei nicht messbarem Eingangsraum

Im Gegensatz zu Kapitel 3.3.2.1 ist nun die Messbarkeit des gesamten Eingangsraumes der Nichtlinearität $\mathcal{NL}(\underline{x}_{\mathcal{NL}}, u)$ nicht mehr gefordert. Wenn der Eingangsraum der Nichtlinearität nicht messbar ist, müssen die Eingangssignale beobachtet werden. Dies setzt jedoch die vollständige Zustandsbeobachtbarkeit des linearen Teils der Strecke voraus. Ein System nach Gl. (3.23) der Ordnung n ist im linearen Teil genau dann vollständig zustandsbeobachtbar, wenn für die Beobachtbarkeitsmatrix \mathbf{Q}_{obs} gilt:

$$\det(\mathbf{Q}_{obs}) \neq 0 \quad (3.37)$$

mit

$$\mathbf{Q}_{obs} = \left[\underline{c}, \mathbf{A}^T \underline{c}, \dots, (\mathbf{A}^T)^{(n-1)} \underline{c} \right] \quad (3.38)$$

Mit dieser Voraussetzung kann analog zu Kapitel 3.3.2.1 ein Neuronaler Beobachter angesetzt werden. Im Vergleich zum Beobachterentwurf mit messbarem Eingangsraum besteht der Unterschied darin, dass die Nichtlinearität im Beobachter abhängig vom beobachteten Eingangsraum $\hat{\underline{x}}_{\mathcal{NL}}$ angesetzt wird. Es gilt:

$$\begin{aligned} \hat{\underline{x}} &= \mathbf{A} \cdot \hat{\underline{x}} + \underline{b} \cdot u + \underline{k} \cdot \widehat{\mathcal{NL}}(\hat{\underline{x}}_{\mathcal{NL}}, u) - \underline{l} \cdot e \\ \hat{y} &= \underline{c}^T \cdot \hat{\underline{x}} + d \cdot u \end{aligned} \quad (3.39)$$

In Abb. 3.16 ist das Blockschaltbild eines Neuronalen Zustandsbeobachters mit nicht messbarem Eingangsraum dargestellt.

Die Berechnung der Fehlerübertragungsfunktion erfolgt analog zu Gl. (3.28). In [32] wurde durch die Einführung von sog. virtuellen Stützwerten gezeigt, dass die Lerngesetze nach Fehlermodell 3 und 4 für den Fall eines nicht messbaren Eingangsraumes¹³ unter der Voraussetzung, dass die Lernschrittweite in einem bestimmten Bereich liegt, garantiert stabil sind.

¹²Es gilt $e_e(s) = \mathcal{L}\{e_e\}$.

¹³In diesem Fall muss $\underline{\mathcal{A}}(\underline{x}_{\mathcal{NL}}, u)$ durch $\underline{\mathcal{A}}(\hat{\underline{x}}_{\mathcal{NL}}, u)$ ersetzt werden.

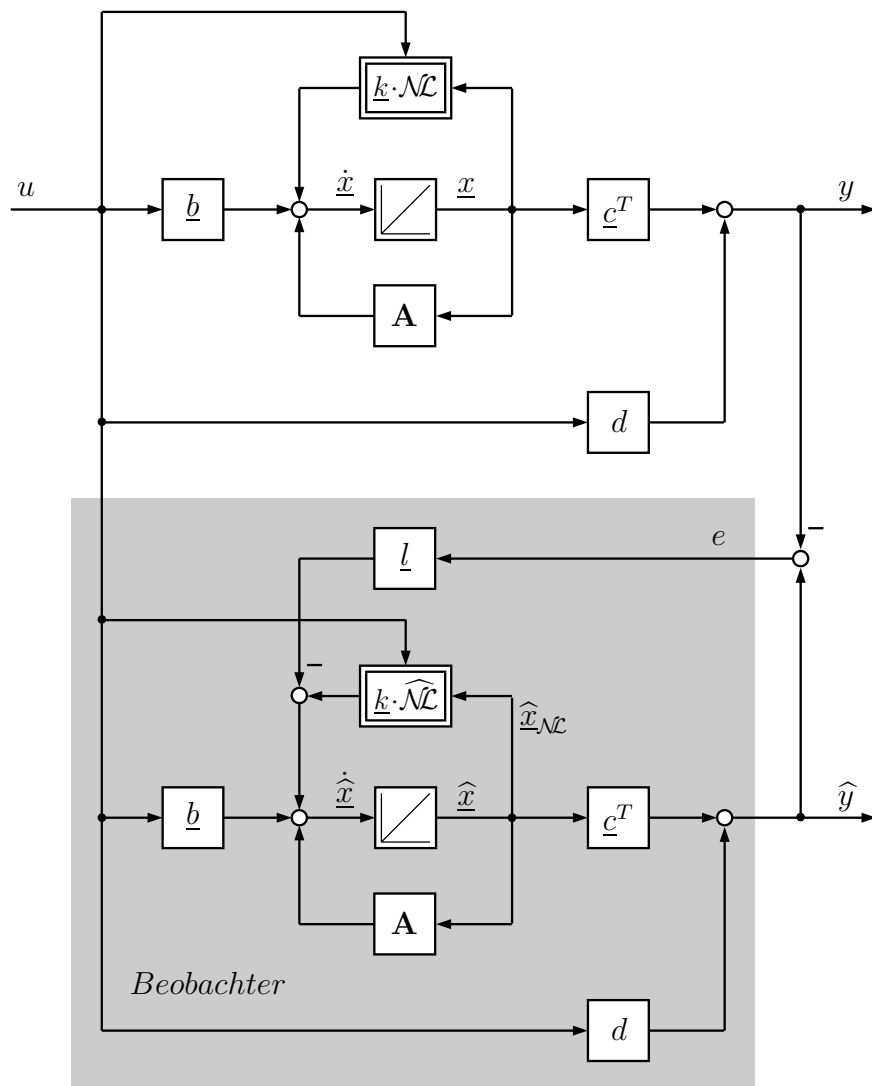


Abb. 3.16: Neuronaler Zustandsbeobachter bei nicht messbarem Eingangsraum der statischen Nichtlinearität

3.3.3 Nichtlineare Modelle mit Ausgangsrückkopplung

Bei den nichtlinearen Modellen mit Ausgangsrückkopplung können, abhängig von der Art der Rückkopplung, das nichtlineare Ausgangsfehlermodell (NOE) und das nichtlineare Gleichungsfehlermodell (NARX) unterschieden werden. Analog zu den linearen Systemen sind auch Modelle mit komplexeren Störsignalmodellen möglich, wie z.B. das NARMAX-Modell [41]. Bei den nichtlinearen Modellen mit Ausgangsrückkopplung wird ein statischer Funktionsapproximator mit externen dynamischen Filterketten am Netzeingang erweitert. Diese sind zur Modellierung der Dynamik erforderlich.

3.3.3.1 Nonlinear Autoregressive with Exogenous Input Model

Das NARX-Modell ist dadurch gekennzeichnet, dass der Systemausgang über externe Filterketten zurückgekoppelt wird und dem statischen Funktionsapproximator als Eingang zur Verfügung gestellt wird. In Abb. 3.17 ist die Struktur des NARX-Modells dargestellt.

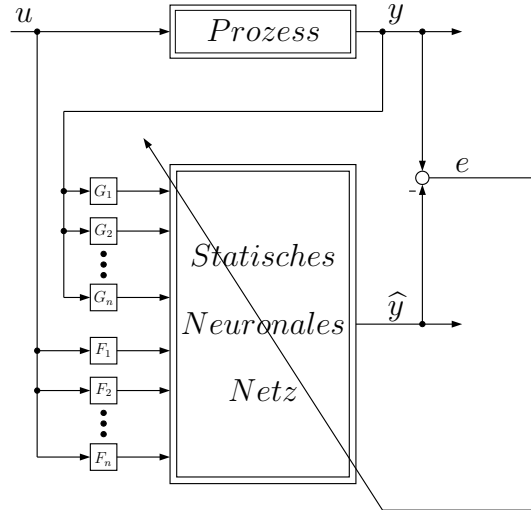


Abb. 3.17: NARX-Modell

Bei den Filtern $F_1 \dots F_n$ und $G_1 \dots G_n$ handelt es sich in der Regel um zeitliche Schiebeoperatoren, für die gilt:

$$F_i(z) = z^{-i} \quad G_i(z) = z^{-i} \quad \text{mit} \quad i = 1 \dots n \quad (3.40)$$

Das NARX-Modell kann somit durch die Gleichung

$$\hat{y}[k] = f(u[k-1], \dots, u[k-n], y[k-1], \dots, y[k-n]) \quad (3.41)$$

beschrieben werden. Die maximale Verzögerung n entspricht der dynamischen Ordnung des zu identifizierenden Prozesses. Die Ordnung des Prozesses muss entweder durch Vorüberlegungen oder durch Vorversuche bestimmt werden. Als statischer Funktionsapproximator kann im Prinzip jeder der in Kapitel 2 vorgestellten Netztypen verwendet werden. Allerdings ist zu beachten, dass bei der Identifikation nichtlinearer dynamischer Systeme die Anzahl der Eingangssignale stark ansteigt. So hat der Eingangsraum bei der Identifikation eines nicht sprungfähigen Prozesses 2. Ordnung bereits die Dimension $N = 4$. Im Falle des RBF-Netzes und des GRNN, bei denen die Parameteranzahl potenziell mit der Eingangsdimension ansteigt, würde dies bedeuten, dass z.B. bei 10 Stützwerten je Dimension 10000 Parameter zu adaptieren wären. Dies macht deutlich, warum das RBF-Netz bzw. das GRNN in der dargestellten Weise nur mit einer Strukturelektion [26] zur Identifikation dynamischer Systeme geeignet ist. Im Gegensatz dazu findet beim MLP-Netz und bei LOLIMOT

eine Komprimierung des Eingangsraumes statt, so dass diese beiden Netztypen zur Identifikation nichtlinearer dynamischer Systeme geeignet sind [39, 72].

3.3.3.2 Nonlinear Output Error Model

Beim nichtlinearen Ausgangsfehlermodell wird der Modellausgang über externe Filterketten zurückgekoppelt und dem statischen Funktionsapproximator als Eingang zur Verfügung gestellt. In Abb. 3.17 ist die Struktur des NOE-Modells dargestellt.

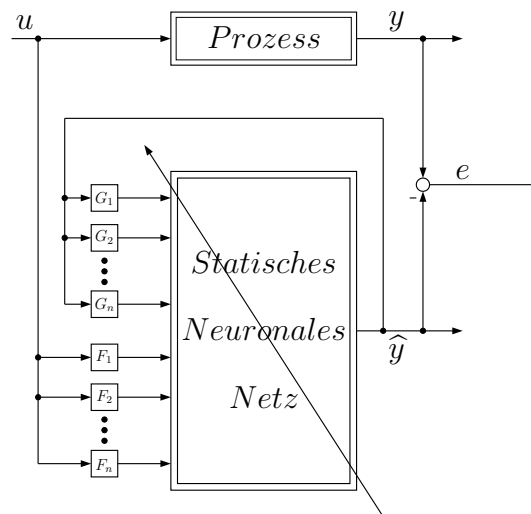


Abb. 3.18: NOE-Modell

Analog zu Gl. (3.41) gilt für das NOE-Modell:

$$\hat{y}[k] = f(u[k-1], \dots, u[k-n], \hat{y}[k-1], \dots, \hat{y}[k-n]) \quad (3.42)$$

Aus Gl. (3.42) wird wie im Falle des OE-Modells bei der Identifikation linearer dynamischer Systeme deutlich, dass die Parameter nichtlinear in den Modellausgang eingehen, selbst wenn der statische Funktionsapproximator linear in den Parametern ist. Alle Eigenschaften des ARX-Modells sowie des OE-Modells aus Kapitel 3.2 lassen sich auf deren nichtlineare Erweiterungen übertragen. Die Vor- und Nachteile der einzelnen Strukturen sollen an dieser Stelle nicht erneut erörtert werden.

3.3.4 Nichtlineare Modelle ohne Ausgangsrückkopplung

Nichtlineare Modelle ohne Ausgangsrückkopplung basieren ausschließlich auf zeitverzögerten oder gefilterten Eingangssignalen. Die Vor- und Nachteile von linearen Modellen ohne Ausgangsrückkopplung wurden bereits in Kapitel 3.2.2 ausführlich diskutiert. Sie lassen sich ohne Ausnahme auf die nichtlinearen Erweiterungen dieser

Modelle übertragen. So ist der Hauptnachteil die hohe Anzahl an Parametern, die bereits im linearen Fall ein großes Problem darstellte, was sich im nichtlinearen Fall drastisch verschärft.

3.3.4.1 Volterra–Funktionalpotenzreihe

Die Volterra–Funktionalpotenzreihe zählt zu den ersten allgemeinen Modellansätzen für nichtlineare Systeme mit einem Eingang und einem Ausgang [66]. Die diskrete Volterra–Funktionalpotenzreihe stellt eine Erweiterung der zeitdiskreten Faltungssumme dar. Das Identifikationsergebnis wird deshalb auch als NFIR–Modell bezeichnet. Die Volterra–Funktionalpotenzreihe dient als Grundlage zahlreicher aktueller Forschungsarbeiten im Bereich der Regelungs– und Steuerungstechnik, insbesondere zur Identifikation nichtlinearer dynamischer Systeme [9, 27, 30, 48, 70]. Die allgemeinste Form der Volterra–Funktionalpotenzreihe ist in Gl. (3.43) dargestellt.

$$\begin{aligned}
 y[k] = & g_0 + \sum_{i_1=0}^{\infty} g[i_1] u[k-i_1] + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} g[i_1, i_2] u[k-i_1] u[k-i_2] + \dots + \\
 & + \sum_{i_1=0}^{\infty} \dots \sum_{i_q=0}^{\infty} g[i_1, \dots, i_q] u[k-i_1] \dots u[k-i_q]
 \end{aligned}
 \tag{3.43}$$

Die Parameter $g[i_1]$, $g[i_1, i_2]$, \dots , $g[i_1, \dots, i_q]$ werden als Volterra–Kerne ersten, zweiten und q -ten Grades bezeichnet. Die Konstante $g_0 \in \mathbb{R}$ ist der sog. Beharrungswert, der sich bei eingangsseitig verschwindender Anregung am Ausgang einstellt. Die Volterra–Kerne ersten und zweiten Grades sind beispielhaft in Abb. 3.19 veranschaulicht.

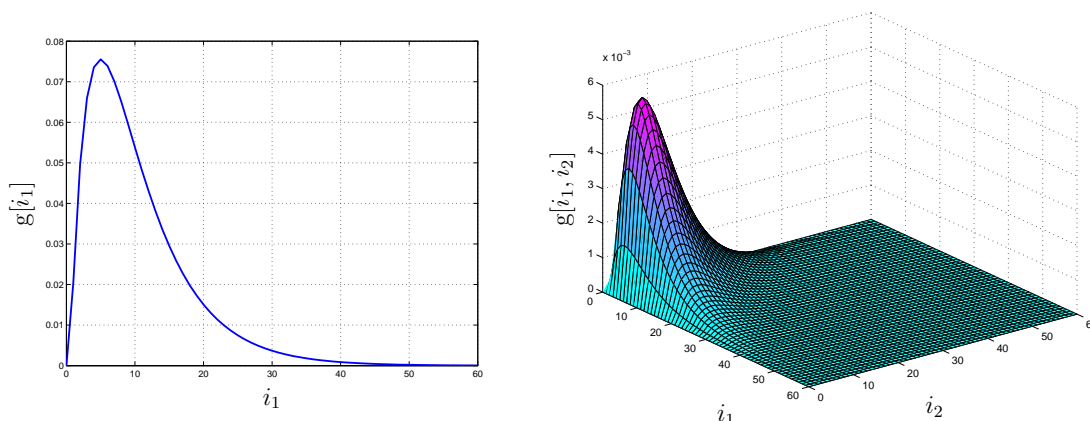


Abb. 3.19: Volterra–Kern ersten (links) und zweiten Grades (rechts)

Die erste Summe in Gl. (3.43) stellt die Faltungssumme dar. Analog zur Faltungssumme können für stabile Systeme die Volterra–Kerne bei einer oberen Grenze m abgeschnitten werden. Bei den blockorientierten Modellen aus Kapitel 3.3.1.2, wie z.B. dem Hammerstein– und dem Wiener–Modell, wird sich in Kapitel 3.3.4.2 zeigen, dass die vollbesetzten Kerne für $q \geq 2$ symmetrisch sind. Es gilt z.B. für die

Volterra–Kerne zweiten und dritten Grades:

$$\begin{aligned} g[i_1, i_2] &= g[i_2, i_1] \\ g[i_1, i_2, i_3] &= g[i_1, i_3, i_2] = g[i_2, i_1, i_3] = \dots \end{aligned} \quad (3.44)$$

Diese Eigenschaft reduziert den Aufwand erheblich, da z.B. bei dem Volterra–Kern zweiter Ordnung aus Abb. 3.19 nur die Kernelemente der Hauptdiagonale und die Kernelemente links oder rechts der Hauptdiagonale zu identifizieren sind. Allgemein reduziert sich für den Kern q -ten Grades der Rechenaufwand auf den $q!$ -ten Teil. Diese Symmetrie wirkt sich in Gl. (3.43) auf die Indizierung der Summen aus. Eine weitere Vereinfachung der allgemeinen Volterra–Funktionalpotenzreihe ergibt sich, wenn sprungfähige Systeme ausgeschlossen werden. Dies ist für eine Vielzahl realer Systeme zulässig und stellt keine Einschränkung der Allgemeingültigkeit dar, da jederzeit durch eine Veränderung des unteren Summenindex von $i = 1$ auf $i = 0$ eine vorhandene Sprungfähigkeit korrigiert werden kann. Werden die aufgeführten Vereinfachungen bei der allgemeinen Volterra–Funktionalpotenzreihe berücksichtigt, ergibt sich der vereinfachte Ansatz entsprechend zu:

$$\begin{aligned} y[k] &= g_0 + \sum_{i_1=1}^m g[i_1] u[k - i_1] + \sum_{i_1=1}^m \sum_{i_2=i_1}^m g[i_1, i_2] u[k - i_1] u[k - i_2] + \dots + \\ &+ \sum_{i_1=1}^m \dots \sum_{i_q=i_{q-1}}^m g[i_1, \dots, i_q] u[k - i_1] \dots u[k - i_q] \end{aligned} \quad (3.45)$$

Dabei gilt für den Beharrungswert und den Volterra–Kern ersten Grades:

$$g_0 = g_0 \quad g[i_1] = g[i_1] \quad (3.46)$$

Für den Volterra–Kern zweiten Grades gilt beispielsweise, dass die Elemente auf der Hauptdiagonale gleich sind, während alle anderen Elemente doppelt so groß sind (vgl. Abb. 3.19 rechts):

$$g[i_1, i_2] = g[i_1, i_2] \quad \forall \quad i_1 = i_2 \quad g[i_1, i_2] = 2 \cdot g[i_1, i_2] \quad \forall \quad i_1 \neq i_2 \quad (3.47)$$

Für die Volterra–Kerne höheren Grades können entsprechende Beziehungen gefunden werden. Wird Gl. (3.45) in Vektorschreibweise dargestellt, ergibt sich eine klare Trennung zwischen den unbekanntem Parametern, d.h. den Volterra–Kernen und den bekannten Eingangssignalen.

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (3.48)$$

$\underline{\mathcal{A}}_{dyn}[k]$ wird dabei als dynamischer Aktivierungsvektor bezeichnet und enthält nur Eingangssignale. $\underline{\theta}$ ist der optimale Parametervektor und beinhaltet den Beharrungswert g_0 und die einzelnen Elemente der Volterra–Kerne. Die Anzahl der Parameter p des Parametervektors ist abhängig von der Antwortlänge m und dem sog. Grad q

und ergibt sich zu $p = \binom{m+q}{q}$. Der dynamische Aktivierungsvektor $\underline{A}_{dyn}[k]$ und der Parametervektor $\underline{\theta}$ setzen sich wie folgt zusammen:

$$\begin{aligned} \underline{A}_{dyn}^T[k] &= \left[1, u[k-1], \dots, u[k-m], u^2[k-1], u[k-1]u[k-2], \dots, \right. \\ &\quad \left. u^2[k-m], \dots, u^q[k-m] \right] \\ \underline{\theta}^T &= \left[g_0, g[1], \dots, g[m], g[1,1], g[1,2], \dots, g[m,m], \dots, g[m, \dots, m] \right] \end{aligned} \quad (3.49)$$

Auf die Parameteranzahl wird in Kapitel 3.3.4.3 noch genauer eingegangen. Zunächst soll gezeigt werden, wie das Hammerstein- und das Wiener-Modell, als blockorientierte Modelle, in der Volterra-Funktionalpotenzreihe enthalten sind.

3.3.4.2 Hammerstein-Modell und Wiener-Modell im Ansatz der Volterra-Funktionalpotenzreihe

Das Hammerstein- und das Wiener-Modell sind blockorientierte Modelle, die aus einer Reihenschaltung einer zeitinvarianten Übertragungsfunktion und einer statischen Nichtlinearität bestehen. In Abb. 3.20 sind beide Modelle dargestellt.



Abb. 3.20: Hammerstein-Modell (links) und Wiener-Modell (rechts)

Beim Hammerstein-Modell, befindet sich der nichtlineare Teil der Strecke vor dem linearen Teil. Die Eingangssignale werden durch die statische Nichtlinearität auf einen Ausgangswertebereich abgebildet, diese Ausgangssignale werden wiederum an den Eingang der zeitinvarianten Übertragungsfunktion angelegt. Wird die statische Nichtlinearität durch ein Polynom q -ten Grades beschrieben, lässt sich das Hammerstein-Modell mathematisch beschreiben durch:

$$\begin{aligned} v(u) = \mathcal{NL}(u) &= a_0 + a_1 u + \dots + a_q u^q & y[k] &= \sum_{i=1}^m h[i] v[k-i] \\ y[k] &= a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] u[k-i] + \dots + a_q \sum_{i=1}^m h[i] u^q[k-i] \end{aligned} \quad (3.50)$$

Wird Gl. (3.50) dem vereinfachten Ansatz der Volterra-Funktionalpotenzreihe gegenübergestellt, kann festgestellt werden, dass nur Kernelemente mit gleichen Indizes ungleich Null sind:

$$\begin{aligned} g_0 &= a_0 \sum_{i=1}^m h[i] & g[i] &= a_1 h[i] & \dots & & g[i, \dots, i] &= a_q h[i] \\ y[k] &= g_0 + \sum_{i=1}^m g[i] u[k-i] + \dots + \sum_{i=1}^m g[i, \dots, i] u^q[k-i] \end{aligned} \quad (3.51)$$

Anschaulich bedeutet das, dass alle Volterra-Kerne der Ordnung $q \geq 2$ ausschließlich auf ihrer Hauptdiagonale besetzt sind. Dies ist charakteristisch für das Hammerstein-Modell. Die einzelnen Elemente der Volterra-Kerne ergeben sich mathematisch aus den Elementen der Impulsantwort und den Polynomkoeffizienten.

Das Hammerstein-Modell im Ansatz der Volterra-Funktionalpotenzreihe kann auch in vektorieller Schreibweise dargestellt werden, wie in den Gleichungen (3.52) gezeigt ist. Die Anzahl der zu bestimmenden Parameter ergibt sich zu $p = q \cdot m + 1$.

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (3.52)$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left[1, u[k-1], \dots, u[k-m], u^2[k-1], \dots, u^2[k-m], \dots, u^q[k-m] \right]$$

$$\underline{\theta}^T = \left[g_0, g[1], \dots, g[m], g[1,1], \dots, g[m,m], \dots, g[m, \dots, m] \right]$$

Im Gegensatz zum Hammerstein-Modell befindet sich beim Wiener-Modell die statische Nichtlinearität hinter dem linearen Streckenteil (vgl. Abb. 3.20 rechts). Die Ausgangswerte der zeitdiskreten Faltungssumme dienen als Eingangswerte des nichtlinearen Streckenteils. Wird die Nichtlinearität wieder durch ein Polynom q -ten Grades beschrieben, lässt sich das Wiener-Modell mathematisch wie folgt formulieren:

$$v[k] = \sum_{i=1}^m h[i] u[k-i] \quad y(v) = \mathcal{NL}(v) = a_0 + a_1 v + \dots + a_q v^q$$

$$y[k] = a_0 + a_1 \sum_{i_1=1}^m h[i_1] u[k-i_1] + \dots +$$

$$+ a_q \sum_{i_1=1}^m \sum_{i_2=1}^m \dots \sum_{i_q=1}^m h[i_1] \dots h[i_q] u[k-i_1] u[k-i_2] \dots u[k-i_q] \quad (3.53)$$

Wird ein Koeffizientenvergleich zwischen der obigen Gleichung und Gl. (3.45) durchgeführt, kann festgestellt werden, dass das Wiener-Modell genau mit der Volterra-Funktionalpotenzreihe übereinstimmt. Die Kerne der vereinfachten Volterra-Funktionalpotenzreihe aus Gleichung (3.45) können beschrieben werden durch:

$$g_0 = a_0 \quad g[i_1] = a_1 h[i_1]$$

$$g[i_1, i_2] = a_2 h[i_1] h[i_2] \quad \forall \quad i_1 = i_2 \quad g[i_1, i_2] = 2 \cdot a_2 h[i_1] h[i_2] \quad \forall \quad i_1 \neq i_2$$

$$\vdots$$

$$y[k] = g_0 + \sum_{i_1=1}^m g[i_1] u[k-i_1] + \dots +$$

$$+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \dots \sum_{i_q=i_{q-1}}^m g[i_1, i_2, \dots, i_q] u[k-i_1] u[k-i_2] \dots u[k-i_q] \quad (3.54)$$

Wird das Wiener-Modell in vektorieller Schreibweise dargestellt, ergibt sich:

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (3.55)$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left[1, u[k-1], \dots, u[k-m], \dots, u[k-m] \cdots u[k-m] \right]$$

$$\underline{\theta}^T = \left[g_0, g[1], \dots, g[m], \dots, g[m, \dots, m] \right]$$

Die Anzahl der zu bestimmenden Parameter ist beim Wiener-Modell deutlich höher als beim Hammerstein-Modell und beträgt $p = \binom{m+q}{q}$.

Die hohe Anzahl an unbekanntem Parametern macht einen praktischen Einsatz des NFIR-Modells unmöglich. Deshalb muss analog zu dem FIR-Modell bei der Identifikation linearer dynamischer Systeme eine Parameterreduktion erfolgen.

3.3.4.3 Volterra-Funktionalpotenzreihe mit Basisfunktionen

Durch die Einführung von orthonormalen Basisfunktionen lässt sich die Parameteranzahl deutlich verringern. Das identifizierte Modell wird als NOBF-Modell (Non-linear Orthonormal Basis Function) bezeichnet. Das Prinzip der Parameterreduktion wurde bereits in Kapitel 3.2.2.2 erläutert und kann in analoger Weise zur Parameterreduktion der Volterra-Kerne beitragen. Dies wird am Beispiel des Hammerstein- sowie des Wiener-Modells genauer betrachtet. In Abb. 3.21 sind zwei typische Volterra-Kerne veranschaulicht.

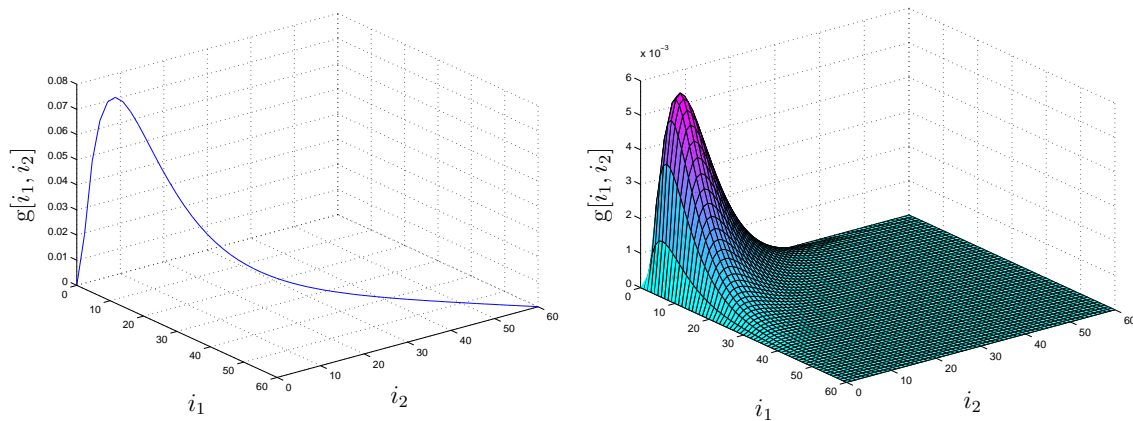


Abb. 3.21: Volterra-Kern zweiten Grades eines Hammerstein-Modells (links) und eines Wiener-Modells (rechts)

Abbildung 3.21 links zeigt einen Volterra-Kern zweiten Grades, der ausschließlich auf der Hauptdiagonale besetzt ist. Dieser Kerntyp ist charakteristisch für Hammerstein-Modelle. Im Gegensatz dazu zeigt Abb. 3.21 rechts einen vollbesetzten symmetrischen Volterra-Kern zweiten Grades, wie er charakteristisch für Wiener-Modelle

ist. Im Folgenden wird die Parameterreduktion für das Hammerstein- und Wiener-Modell mit ihren charakteristischen Kerneigenschaften genauer untersucht¹⁴.

Bei genauer Analyse von Gl. (3.51) wird eine Besonderheit des Hammerstein-Modells deutlich. Die Volterra-Kerne des Hammerstein-Modells sind ausschließlich auf der Hauptdiagonale besetzt. Jeder Volterra-Kern kann durch eine gewichtete Überlagerung von Basisfunktionen beschrieben werden. Durch die Einführung von sog. reduzierten Gewichtsfolgevektoren $\underline{g}_r \in \mathbb{R}^{m_r \times 1}$ können die einzelnen Kerne mit Hilfe der Basisfunktionen \tilde{r}_j bzw. der Basisfunktionenmatrix $\tilde{\mathbf{R}}$ wie folgt approximiert werden:

$$\begin{aligned} g[i] &= \sum_{j=1}^{m_r} g_r[j] \tilde{r}_j[i] \\ g[i, i] &= \sum_{j=1}^{m_r} g_r[j, j] \tilde{r}_j[i] && \text{mit} \quad m_r < m \\ &\vdots \\ g[i, \dots, i] &= \sum_{j=1}^{m_r} g_r[j, \dots, j] \tilde{r}_j[i] \end{aligned} \quad (3.56)$$

Die Parameterreduktion durch die Einführung von Basisfunktionen führt zu:

$$\begin{aligned} y[k] &= g_0 + \sum_{i=1}^m \left(\sum_{j=1}^{m_r} g_r[j] \tilde{r}_j[i] \right) u[k-i] + \dots + \\ &+ \sum_{i=1}^m \left(\sum_{j=1}^{m_r} g_r[j, \dots, j] \tilde{r}_j[i] \right) u^q[k-i] \end{aligned} \quad (3.57)$$

Wird Gl. (3.57) in vektorielle Schreibweise überführt, ergibt sich:

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (3.58)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[1, \underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T, \underline{u}^{2T}[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{u}^{qT}[k] \cdot \tilde{\mathbf{R}}^T \right] \\ \underline{u}^{iT}[k] &= \left[u^i[k-1], u^i[k-2], \dots, u^i[k-m] \right] \\ \underline{\theta}^T &= \left[g_0, g_r[1], \dots, g_r[m_r], g_r[1, 1], \dots, g_r[m_r, m_r], \dots, g_r[m_r, \dots, m_r] \right] \end{aligned}$$

¹⁴An dieser Stelle sei angemerkt, dass auch Modellansätze für das Hammerstein- und Wiener-Modell existieren, bei denen die Übertragungsfunktion durch eine Differenzgleichung beschrieben wird [28, 29]. Dies führt zu einer deutlich geringeren Anzahl an unbekanntem Parametern, allerdings treten bei diesen Ansätzen wieder die bereits beschriebenen Probleme von NARX- und NOE-Modellen auf.

Die Anzahl der Parameter hat sich beim Hammerstein-Modell durch die Einführung orthonormaler Basisfunktionen auf $p = q \cdot m_r + 1$ reduziert. Als Zwischengröße wurde $\underline{u}^i[k]$ eingeführt, womit die Multiplikation der orthonormalen Basisfunktionen mit den Vergangenheitswerten des Eingangssignals kompakter darstellbar ist. Für die orthonormalen Basisfunktionen gelten dieselben Einstellregeln, wie für die Identifikation linearer dynamischer Systeme.

Im Gegensatz zum Hammerstein-Modell sind beim Wiener-Modell die Volterra-Kernelemente ungleicher Indizes ungleich Null. In Abb. 3.21 rechts sowie in Gl. (3.53) ist die Symmetrie der Volterra-Kerne bezüglich der Hauptdiagonale zu erkennen. Durch die Einführung von sog. reduzierten Gewichtsfolgevektoren können die einzelnen Volterra-Kerne des Wiener-Modells mit Hilfe der Basisfunktionen \tilde{r}_j bzw. der Basisfunktionenmatrix $\tilde{\mathbf{R}}$ wie folgt approximiert werden:

$$\begin{aligned}
 g[i_1] &= \sum_{j_1=1}^{m_r} g_r[j_1] \tilde{r}_{j_1}[i_1] \\
 g[i_1, i_2] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} g_r[j_1, j_2] \tilde{r}_{j_2}[i_2] \tilde{r}_{j_1}[i_1] && \text{mit } m_r < m \\
 &\vdots \\
 g[i_1, \dots, i_q] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \cdots \sum_{j_q=j_{q-1}}^{m_r} g_r[j_1, \dots, j_q] \tilde{r}_{j_q}[i_q] \cdots \tilde{r}_{j_1}[i_1]
 \end{aligned} \tag{3.59}$$

Die Parameterreduktion durch die Einführung der orthonormalen Basisfunktionen ergibt sich beim Wiener-Modell wie folgt:

$$\begin{aligned}
 y[k] &= g_0 + \sum_{i_1=1}^m \left(\sum_{j_1=1}^{m_r} g_r[j_1] \tilde{r}_{j_1}[i_1] \right) u[k - i_1] + \\
 &+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \left(\sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} g_r[j_1, j_2] \tilde{r}_{j_2}[i_2] \tilde{r}_{j_1}[i_1] \right) u[k - i_1] u[k - i_2] + \dots + \\
 &+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \cdots \sum_{i_q=i_{q-1}}^m \left(\sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \cdots \sum_{j_q=j_{q-1}}^{m_r} g_r[j_1, \dots, j_q] \tilde{r}_{j_q}[i_q] \cdots \tilde{r}_{j_1}[i_1] \right) u[k - i_1] \cdots u[k - i_q]
 \end{aligned} \tag{3.60}$$

Um Gl. (3.60) in eine vektorielle Schreibweise überführen zu können, wurde in [30] ein spezieller Rechenoperator definiert, der sämtliche Kombinationen von Verknüpfungen zwischen den Vergangenheitswerten des Eingangs multipliziert mit den orthonormalen Basisfunktionen erzeugt. Dieser Rechenoperator wird als sog. $*$ -Operator bezeichnet und ist im Anhang A definiert. Mit Hilfe dieses $*$ -Operators ergibt sich für Gl. (3.60):

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (3.61)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[1, \underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T, \left(\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*2}, \dots, \left(\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*q} \right] \\ \underline{u}^T[k] &= \left[u[k-1], u[k-2], \dots, u[k-m] \right] \\ \underline{\theta}^T &= \left[g_0, g_r[1], \dots, g_r[m_r], g_r[1,1], g_r[1,2], \dots, g_r[m_r, m_r], \dots, g_r[m_r, \dots, m_r] \right] \end{aligned}$$

Die Parameterreduktion beim Wiener-Modell ist beträchtlich. Ursprünglich waren insgesamt $p = \binom{m+q}{q}$ Parameter nötig. Durch die Einführung von Basisfunktionen wird die Parameteranzahl auf $p = \binom{m_r+q}{q}$ reduziert. Als Beispiel wird ein Wiener-Modell mit einer Antwortlänge von $m = 50$ und einer Nichtlinearität vom Grad $q = 3$ betrachtet. Ohne die Einführung von orthonormalen Basisfunktionen sind 23426 Parameter nötig. Werden die Volterra-Kerne durch z.B. sieben Basisfunktionen ($m_r = 7$) approximiert, so ergibt sich die Anzahl der unbekannt Parameter zu $p = \binom{7+3}{3} = 120$. Diese deutliche Parameterreduktion ermöglicht erst den Einsatz der Volterra-Funktionalpotenzreihe zur Identifikation nichtlinearer dynamischer Systeme.

3.4 Kurzzusammenfassung

In diesem Kapitel wurde ein umfassender Einblick in die Identifikation nichtlinearer dynamischer Systeme gegeben. Ausgehend von der Identifikation linearer dynamischer Systeme wurden zunächst einige Begriffe und grundlegende Ansätze zur Identifikation erläutert. Dabei zeigte sich, dass ausschließlich bei Ausgangsfehlerstrukturen ein echt paralleles Prozessmodell identifiziert werden kann. Bei Ausgangsfehlermodellen mit Ausgangsrückkopplung (OE-Modell) kann jedoch keine Stabilität bewiesen werden und die unbekannt Parameter gehen nichtlinear in den Modellausgang ein. Im Gegensatz dazu gehen bei Ausgangsfehlermodellen ohne Ausgangsrückkopplung (FIR- und OBF-Modell) die Parameter linear in den Modellausgang ein und die Stabilität dieser Modelle kann bewiesen werden. Jedoch hat das FIR-Modell eine hohe Anzahl an unbekannt Parametern, die eine praktische Anwendung verhindert. Durch die Einführung orthonormaler Basisfunktionen kann dieser Nachteil beim OBF-Modell beseitigt werden. So kann an dieser Stelle festgehalten werden, dass allein das OBF-Modell alle wesentlichen Vorteile in sich vereinigt. Garantierte Stabilität, Linearität in den Parametern, Identifikation eines echt parallelen Modells sowie Robustheit gegenüber Prozessrauschen gehen einher mit einer geringen Anzahl an unbekannt Parametern.

Bei der Identifikation nichtlinearer dynamischer Systeme handelt es sich im Wesentlichen um Erweiterungen linearer dynamischer Identifikationsverfahren. Aufgrund

der Vielzahl an unterschiedlichen nichtlinearen dynamischen Systemen, gibt es auch eine Vielzahl an Identifikationsverfahren, die sich in erster Linie durch den Grad des strukturellen Vorwissens unterscheiden. Der Neuronale Beobachter setzt ein hohes Maß an strukturellem Vorwissen voraus und ist in der Lage statische Nichtlinearitäten innerhalb eines bekannten dynamischen Systems zu identifizieren. Blockorientierte nichtlineare dynamische Modelle setzen ein geringes Maß an strukturellem Vorwissen voraus und können durch die Volterra-Funktionalpotenzreihe identifiziert werden. Allerdings muss hierzu erst eine Parameterreduktion mittels orthonormaler Basisfunktionen durchgeführt werden. Dieses sog. NOBF-Modell ermöglicht es somit nichtlineare blockorientierte Modelle, wie z.B. das Hammerstein- oder das Wiener-Modell zu identifizieren. Dabei bleiben alle Vorteile, die vom OBF-Modell bekannt sind, erhalten. Desweiteren wurden noch Identifikationsverfahren vorgestellt, mit denen es möglich ist Black-Box-Systeme zu identifizieren, d.h. Systeme bei denen kein strukturelles Vorwissen vorhanden ist.

An dieser Stelle soll noch einmal auf die Motivation dieser Arbeit eingegangen werden. Wie bereits erwähnt wurde, ist die Wahl eines geeigneten Verfahrens zur Identifikation nichtlinearer dynamischer Systeme stark vom strukturellen Vorwissen abhängig. Am Lehrstuhl für Elektrische Antriebssysteme wurde für verschiedene Problemstellungen in der Mechatronik der Neuronale Beobachter entwickelt. Er erfordert jedoch ein hohes Maß an strukturellem Vorwissen, das bei komplexeren Systemen nicht oder nur unter hohem Aufwand erlangt werden kann. Aus diesem Grund soll der Neuronale Beobachter in der vorliegenden Arbeit in der Richtung weiterentwickelt werden, dass es möglich ist, unbekannte nichtlineare Teilsysteme mit wenig strukturellem Vorwissen innerhalb eines bekannten dynamischen Systems zu identifizieren. Die beweisbare Stabilität des Neuronalen Beobachters soll dabei unter allen Umständen erhalten bleiben. In Abb. 3.22 ist die Erweiterung des Neuronalen Beobachters veranschaulicht.

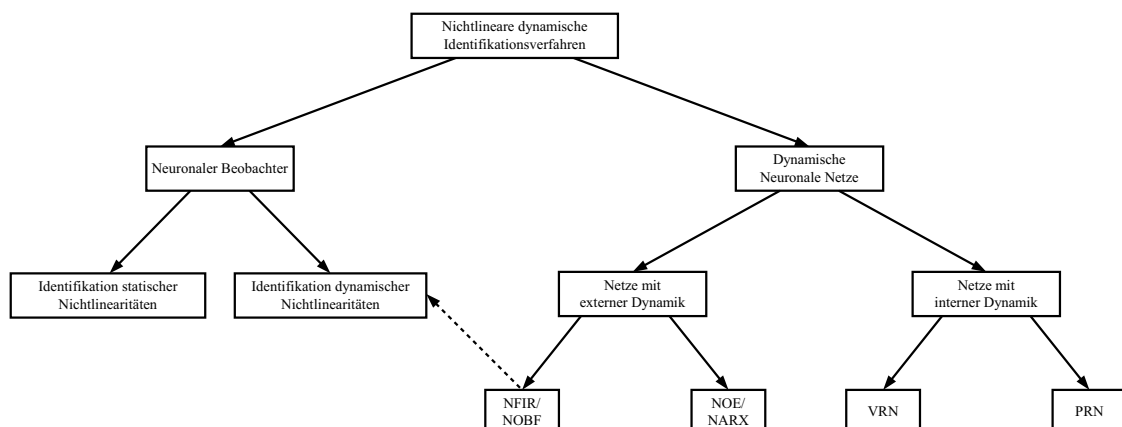


Abb. 3.22: Erweiterung des Neuronalen Beobachters

Bei der Analyse verschiedener Identifikationsverfahren für nichtlineare dynamische Systeme zeigte sich, dass sich das NOBF-Modell zur Identifikation blockorientierter

Systeme sehr gut eignet. Die Identifikation auf der Basis von blockorientierten Modellen erfordert wenig strukturelles Vorwissen. Tabelle 3.3 zeigt das NOBF-Modell im Vergleich zu anderen nichtlinearen Modellansätzen bezüglich wichtiger Eigenschaften.

	NFIR	NOBF	NARX	NOE	VRN
Minimierung des Ausgangsfehlers	ja	ja	nein	ja	ja
Linearität der Parameter	ja	ja	ja	nein	nein
Stabilität beweisbar	ja	ja	ja	nein	nein
niedrige Parameteranzahl	nein	ja	ja	ja	ja

Tabelle 3.3: *Dynamische Neuronale Netze im Vergleich*

Das Ziel der vorliegenden Arbeit ist es, das NOBF-Identifikationsverfahren in den Neuronalen Beobachter zu integrieren und somit die Anwendbarkeit des Neuronalen Beobachters auf eine größere Klasse von nichtlinearen dynamischen Systemen zu ermöglichen¹⁵.

Zur Veranschaulichung dieser Motivation ist in Abb. 3.23 das Blockschaltbild eines nichtlinearen dynamischen Systems mit einem unterschiedlichen Grad an strukturellem Vorwissen dargestellt.

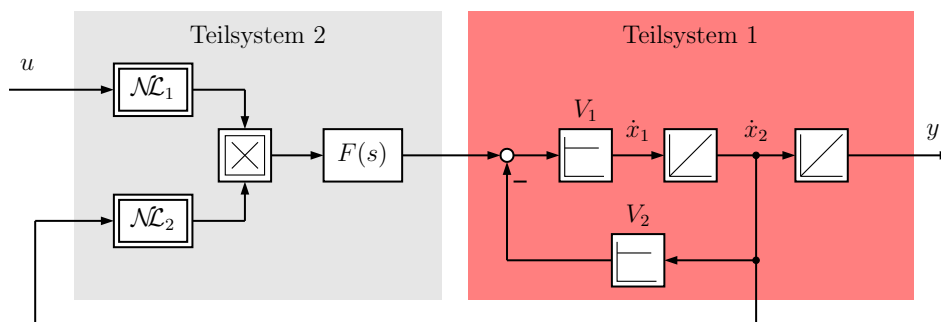


Abb. 3.23: *Nichtlineares dynamisches System mit unterschiedlichem Grad an strukturellem Vorwissen*

Das Teilsystem 1 zeigt ein hohes Maß an strukturellem Vorwissen, da alle Zustandsgrößen des Systems bekannt sind. Im Gegensatz dazu weist das Teilsystem 2 nur eine geringe Strukturvorkenntnis auf. Es ist z.B. nicht bekannt wie viele Zustände in der Übertragungsfunktion enthalten sind. Das Strukturvorwissen dieses Teilsystems beschränkt sich auf die Unterscheidung von statischen Nichtlinearitäten und dynamischen Übertragungsfunktionen, die beispielsweise additiv oder multiplikativ miteinander verknüpft sein können. Diese Art von Strukturvorwissen kann sehr leicht durch einfache physikalische Überlegungen gewonnen werden. Es soll nun ein Verfahren entwickelt werden, das es ermöglicht Systeme, von denen die Struktur nur teilweise bekannt ist (vgl. Abb. 3.23), zu identifizieren. Zu diesem Zweck wird

¹⁵Im weiteren Verlauf der Arbeit wird in diesem Zusammenhang von der Identifikation dynamischer Nichtlinearitäten gesprochen.

aufbauend auf den Kapiteln 2 und 3 im Verlauf der Arbeit in folgenden Schritten vorangegangen:

- In Kapitel 4 wird die Identifikation von blockorientierten Systemen auf Basis der Volterra–Funktionalpotenzreihe auf mehrere Eingangsgrößen erweitert. Dabei wird angenommen, dass statische Nichtlinearitäten als Polynome approximiert werden können und der Systemausgang gemessen werden kann. In dem Beispiel aus Abb. 3.23 bedeutet das, dass der Ausgang von Teilsystem 2 vorliegt.
- In Kapitel 5 wird die Beschränkung aufgehoben, dass die statischen Nichtlinearitäten durch Polynome approximiert werden müssen. Hierzu wird das GRNN als statischer Funktionsapproximator in die Identifikationsverfahren aus Kapitel 4 integriert.
- In Kapitel 6 wird die Annahme aufgehoben, dass der Systemausgang direkt gemessen werden kann. In Abb. 3.23 bedeutet das, dass der Ausgang von Teilsystem 2 nicht vorliegt, sondern nur der Ausgang von Teilsystem 1 gemessen werden kann. Die Identifikationsverfahren aus Kapitel 4 und 5 werden in den Neuronalen Beobachter integriert. Hierbei wird angenommen, dass die Struktur und die Parameter des Teilsystems 1 bekannt sind.
- Die Anwendung der Theorie wird in Kapitel 7 vorgestellt. Am Beispiel eines mechatronischen Antriebssystems und eines biomechanischen Systems wird die Praxisrelevanz dieser Arbeit verdeutlicht.
- In Kapitel 8 wird schließlich auch die Annahme aufgehoben, dass das Teilsystem 1 in seinen Parametern bekannt ist.

4 Identifikation von nichtlinearen Mehrgrößensystemen auf Basis der Volterra–Funktionalpotenzreihe

In Kapitel 3.3.4 wurde die Identifikation des Hammerstein– und Wiener–Modells auf Basis der Volterra–Funktionalpotenzreihe vorgestellt. Bei beiden Systemen handelt es sich um SISO–Systeme. Häufig besitzen nichtlineare dynamische Systeme mehr als eine Eingangsgröße. Aus diesem Grund werden im Folgenden blockorientierte nichtlineare dynamische Systeme mit mehreren Eingangsgrößen und einer Ausgangsgröße betrachtet¹. Solche Systeme werden allgemein als MISO–Systeme² bezeichnet. Zunächst wird ein Hammerstein–Modell herangezogen und durch gezielte Überlegungen erweitert. Im Anschluss daran erfolgen entsprechende Überlegungen zum Wiener–Modell. Zur Parameterreduktion werden für die betrachteten MISO–Systeme orthonormale Basisfunktionen eingeführt. Zum Abschluss dieses Kapitels wird anhand konkreter Beispiele die Identifikation von nichtlinearen Mehrgrößensystemen veranschaulicht.

4.1 MISO–Systembeschreibungen durch Erweiterung des Hammerstein–Modells

Als Ausgangspunkt der folgenden Untersuchungen wird ein Hammerstein–Modell betrachtet. In Abb. 4.1 sind die Möglichkeiten eines linearen und eines nichtlinearen Eingriffs in das Modell dargestellt.

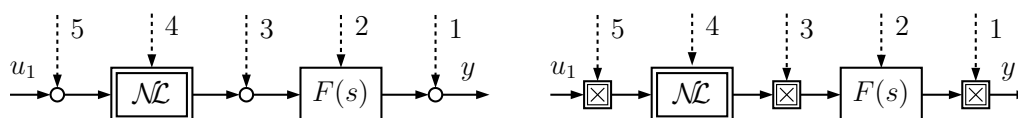


Abb. 4.1: Lineare und nichtlineare Eingriffspunkte einer weiteren Eingangsgröße beim Hammerstein–Modell

¹In [29] wurden bereits Identifikationsansätze für Mehrgrößensysteme vorgestellt, jedoch mit dem Unterschied, dass Übertragungsfunktionen durch Differenzgleichungen beschrieben werden.

²MISO ... Multiple Inputs Single Output

Prinzipiell sind fünf mögliche Eingriffspunkte einer weiteren Eingangsgröße denkbar. Die gestrichelten Linien deuten darauf hin, dass an diesen Stellen in das Modell eingegriffen werden kann. Beispielhaft werden in den folgenden Abschnitten die Systembeschreibungen für eine Anzahl von $r = 2$ Eingangssignalen abgeleitet und auf den allgemeineren Fall, d.h. eine beliebige Anzahl an Eingangssignalen r , überführt. Die Systembeschreibungen sind Ausgangspunkt für die Identifikation.

Bei den nachfolgenden Untersuchungen wird davon ausgegangen, dass die Systeme nicht sprungfähig sind. Der Eingangsraum wird durch $\underline{u} \in \mathbb{R}^r$ charakterisiert. Die statischen Nichtlinearitäten werden als Polynome q -ten Grades approximiert. Die zeitinvarianten Übertragungsfunktionen werden durch die zeitdiskrete Faltungssumme beschrieben.

4.1.1 Linearer Eingriff einer Eingangsgröße in das Hammerstein-Modell

Die Möglichkeit, dass eine weitere Eingangsgröße $u_2 \in \mathbb{R}$ additiv in Punkt 1 eingreift, wird ausgeschlossen, da nur nicht sprungfähige Systeme betrachtet werden sollen. Durch Erweiterung der Systembeschreibung eines Hammerstein-Modells nach Gl. (3.50) wird der direkte Einfluss der Eingangsgröße u_2 auf den Systemausgang y sichtbar. Dies lässt sich zeigen durch:

$$y[k] = a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] u_1[k-i] + \dots + a_q \sum_{i=1}^m h[i] u_1^q[k-i] + u_2[k] \quad (4.1)$$

Eine weitere Möglichkeit ist das Eingreifen von u_2 in Punkt 3. Durch die geeignete Wahl einer Hilfsübertragungsfunktion $F_2(s)$ kann dieser Eingriffspunkt zum Systemausgang verschoben werden. Wird $F_2(s) = F_1(s) = F(s)$ gewählt, lässt sich das System nach Abb. 4.2 grafisch darstellen. Das System lässt sich wiederum durch

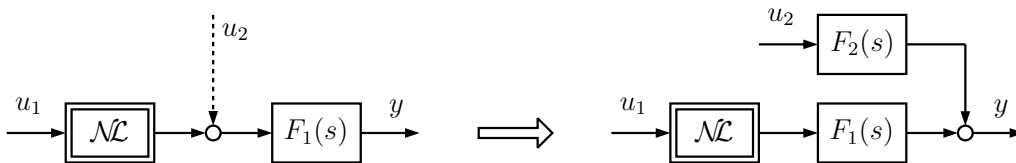


Abb. 4.2: Linearer Eingriff im Punkt 3

Erweiterung von Gl. (3.50) mathematisch beschreiben durch:

$$y[k] = a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] u_1[k-i] + \dots + a_q \sum_{i=1}^m h[i] u_1^q[k-i] + \sum_{i=1}^m h[i] u_2[k-i] \quad (4.2)$$

Im Gegensatz zum vorherigen Fall handelt es sich hier um ein nicht sprungfähiges System. Die Eingangsgröße u_2 greift nicht direkt am Systemausgang an. Dies ist in obiger Gleichung dadurch zu erkennen, dass nur Vergangenheitswerte des Eingangssignals $u_2[k-i]$ vorkommen. Außerdem wird deutlich, dass die Verschiebung des

Eingriffspunktes 3 nach Punkt 1 den Fall beinhaltet, dass die Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ nicht gleich sein müssen. Durch diese Systembeschreibung können folglich zwei Systeme beschrieben werden, die eine unterschiedliche Struktur besitzen. Einerseits kann eine weitere Eingangsgröße u_2 direkt an Punkt 3 angreifen und andererseits ist es möglich, dass die Eingangsgröße u_2 über die Übertragungsfunktion $F_2(s)$ auf den Systemausgang wirkt.

Ein direktes Eingreifen einer zweiten Eingangsgröße in die Übertragungsfunktion $F(s)$ am Punkt 2 nach Abb. 4.1 links, setzt voraus, dass die Übertragungsfunktion $F(s)$ in weitere zeitinvariante Teilübertragungsfunktionen aufgespalten werden kann, d.h. dass die Übertragungsfunktion $F(s)$ in lineare Grundblöcke zerlegt werden kann. Das Überführen dieses Ansatzes ist in gleicher Weise wie beim vorherigen Fall möglich. Durch das Einführen einer Hilfsübertragungsfunktion $F_2(s)$, die den Teil von Eingriffspunkt 2 zum Ausgang der Übertragungsfunktion enthält, kann die Strecke entsprechend Abb. 4.3 transformiert werden.

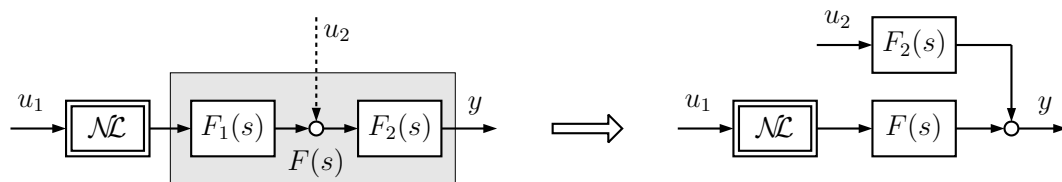


Abb. 4.3: Linearer Eingriff im Punkt 2

Die vierte Möglichkeit eines additiven Eingreifens in das Hammerstein-Modell ist Punkt 4 (vgl. Abb. 4.1 links). Innerhalb der statischen Nichtlinearität sind Verkopplungen in unterschiedlichster Form möglich. Vorerst werden jedoch statische Nichtlinearitäten betrachtet, die sich bei einem Eingreifen einer weiteren Eingangsgröße im Punkt 4 durch ein Polynom der Form

$$v(u_1, u_2) = a_{10} + a_{11}u_1 + a_{12}u_1^2 + \dots + a_{1q}u_1^q + a_{20} + a_{21}u_2 + a_{22}u_2^2 + \dots + a_{2q}u_2^q \quad (4.3)$$

beschreiben lassen. Wird diese Gleichung näher betrachtet, kann festgestellt werden, dass sich dieses Polynom aus einer linearen Verkopplung zweier statischer Nichtlinearitäten zusammensetzt. In Abb. 4.4 ist die transformierte Struktur dieser Strecke dargestellt.

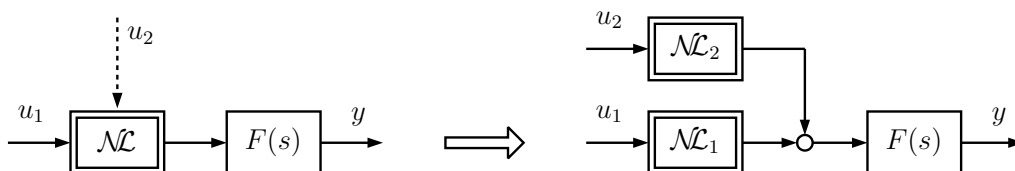


Abb. 4.4: Linearer Eingriff im Punkt 4

Die Ausgänge der beiden Nichtlinearitäten treffen in Eingriffspunkt 3 nach Abb. 4.1 aufeinander. Diese Struktur kann jedoch mit den zuletzt vorgestellten Ansätzen zu

Eingriffspunkt 1 umgeformt werden. Das Ergebnis ist eine Parallelschaltung zweier Hammerstein-Modelle, bei denen sowohl die statische Nichtlinearität als auch die lineare Dynamik unterschiedlich sein dürfen. Mathematisch lässt sich ein aus zwei linear verkoppelten Hammerstein-Modellen beschreibbares System wie folgt darstellen:

$$\begin{aligned}
 y[k] = & a_{10} \sum_{i=1}^m h_1[i] + a_{11} \sum_{i=1}^m h_1[i] u_1[k-i] + \dots + a_{1q} \sum_{i=1}^m h_1[i] u_1^q[k-i] + \\
 & + a_{20} \sum_{i=1}^m h_2[i] + a_{21} \sum_{i=1}^m h_2[i] u_2[k-i] + \dots + a_{2q} \sum_{i=1}^m h_2[i] u_2^q[k-i]
 \end{aligned} \tag{4.4}$$

Die letzte Möglichkeit für das lineare Eingreifen einer weiteren Eingangsgröße in ein Hammerstein-Modell ist Punkt 5 (vgl. Abb. 4.1 links). Systeme, die eine solche Struktur besitzen, können mit der Systembeschreibung eines Hammerstein-Modells (vgl. Kapitel 3.3.4.2) mathematisch dargestellt werden. Werden zwei Eingangssignale miteinander addiert und anschließend an den Eingang der Nichtlinearität angelegt, so ergibt sich folgende Gleichung:

$$\begin{aligned}
 y = & a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] (u_1[k-i] + u_2[k-i]) + a_2 \sum_{i=1}^m h[i] (u_1[k-i] + u_2[k-i])^2 + \dots + \\
 & + a_q \sum_{i=1}^m h[i] (u_1[k-i] + u_2[k-i])^q
 \end{aligned} \tag{4.5}$$

Wird ein Vergleich zwischen dieser Beschreibung und den Überlegungen resultierend in Gl. (4.4) durchgeführt, kann festgestellt werden, dass die linearen Anteile und die potenzierten Vergangenheitswerte der Eingangssignale u_1 und u_2 bereits in Gl. (4.4) vorkommen. Jedoch enthält Gl. (4.5) weitere Anteile, die aus Kombinationen der Eingangssignale u_1 und u_2 zu unterschiedlichen Potenzen bestehen. Für die Summe von u_1 und u_2 zur dritten Potenz soll dies noch einmal veranschaulicht werden, wobei die bereits bekannten Anteile in der ersten Zeile und die zusätzlichen Anteile in der zweiten Zeile dargestellt sind.

$$\begin{aligned}
 a_3 \sum_{i=1}^m h[i] (u_1[k-i] + u_2[k-i])^3 = & a_3 \sum_{i=1}^m h[i] u_1^3[k-i] + a_3 \sum_{i=1}^m h[i] u_2^3[k-i] + \\
 & + 3a_3 \sum_{i=1}^m h[i] u_1^2[k-i] u_2[k-i] + 3a_3 \sum_{i=1}^m h[i] u_1[k-i] u_2^2[k-i]
 \end{aligned} \tag{4.6}$$

Um an dieser Stelle eine Systembeschreibung zu erhalten, in der alle bisher betrachteten Fälle berücksichtigt werden, müssen die Beschreibungen, die für die Eingriffspunkte 4 und 5 aufgestellt wurden, so miteinander kombiniert werden, dass jeder Term genau einmal in der neuen Beschreibung erscheint. Diese Gesamtbeschreibung

kann wie folgt formuliert werden³:

$$\begin{aligned}
y[k] = & a_{10} \sum_{i=1}^m h_1[i] + a_{11} \sum_{i=1}^m h_1[i] (u_1[k-i] + u_2[k-i]) + a_{12} \sum_{i=1}^m h_1[i] (u_1[k-i] + u_2[k-i])^2 + \\
& + \dots + a_{1q} \sum_{i=1}^m h_1[i] (u_1[k-i] + u_2[k-i])^q + a_{20} \sum_{i=1}^m h_2[i] + a_{21} \sum_{i=1}^m h_2[i] u_2[k-i] + \\
& + a_{22} \sum_{i=1}^m h_2[i] u_2^2[k-i] + \dots + a_{2q} \sum_{i=1}^m h_2[i] u_2^q[k-i]
\end{aligned} \quad (4.7)$$

Mit Gl. (4.7) ist es möglich, Systeme mathematisch zu beschreiben, bei denen eine weitere Eingangsgröße u_2 an einer beliebigen Stelle in ein Hammerstein-Modell linear eingreift⁴.

Um Gl. (4.7) in eine Volterra-Funktionalpotenzreihe⁵ überführen zu können, müssen vorab noch einige wichtige Punkte betrachtet werden. Dafür wird Gl. (4.7) nach den Eingangssignalen und deren Potenzen sortiert. Es ergibt sich somit folgende Gleichung:

$$\begin{aligned}
y[k] = & a_{10} \sum_{i=1}^m h_1[i] + a_{20} \sum_{i=1}^m h_2[i] + a_{11} \sum_{i=1}^m h_1[i] u_1[k-1] + \\
& + \sum_{i=1}^m (a_{11} h_1[i] + a_{21} h_2[i]) u_2[k-i] + a_{12} \sum_{i=1}^m h_1[i] u_1^2[k-1] + \\
& + \sum_{i=1}^m (a_{12} h_1[i] + a_{22} h_2[i]) u_2^2[k-i] + 2a_{12} \sum_{i=1}^m h_1[i] u_1[k-i] u_2[k-i] + \dots
\end{aligned} \quad (4.8)$$

Die Volterra-Funktionalpotenzreihe lässt sich für Gl. (4.8) wie folgt definieren:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i=1}^m g_1[i] u_1[k-i] + \sum_{i=1}^m g_2[i] u_2[k-i] + \sum_{i=1}^m g_1[i, i] u_1^2[k-i] + \\
& + \sum_{i=1}^m g_2[i, i] u_2^2[k-i] + \sum_{i=1}^m g_3[i, i] u_1[k-i] u_2[k-i] + \dots
\end{aligned} \quad (4.9)$$

Durch einen Vergleich von Gl. (4.8) mit Gl. (4.9) ergeben sich der Beharrungswert und die Elemente der Volterra-Kerne zu:

$$\begin{aligned}
g_0 &= a_{10} \sum_{i=1}^m h_1[i] + a_{20} \sum_{i=1}^m h_2[i] \\
g_1[i] &= a_{11} h_1[i] & g_2[i] &= a_{11} h_1[i] + a_{21} h_2[i] \\
g_1[i, i] &= a_{12} h_1[i] & g_2[i, i] &= a_{12} h_1[i] + a_{22} h_2[i] & g_3[i, i] &= 2a_{12} h_1[i] \\
& & & & & \vdots
\end{aligned}$$

³Die Indizes von Gl. (4.5) sind so angepasst worden, dass zwischen verschiedenen statischen Nichtlinearitäten und linearen Dynamiken unterschieden werden kann.

⁴Außer am Eingriffspunkt 1, der aus genannten Gründen ausgeschlossen wurde.

⁵Der Begriff *Volterra-Funktionalpotenzreihe* soll auch im MISO-Fall beibehalten werden.

Die Definitionen des Beharrungswertes und der Volterra–Kerne erlauben die folgenden Aussagen:

- Der Beharrungswert g_0 beinhaltet sowohl den Gleichanteil, der durch die Impulsantwort $h_1[i]$ hervorgerufen wird, als auch den Gleichanteil, der durch $h_2[i]$ hervorgerufen wird, d.h. dass zwischen ursprünglich zwei Gleichanteilen ausgangsseitig nicht mehr unterschieden werden kann.
- Die Volterra–Funktionalpotenzreihe für ein Hammerstein–Modell beinhaltet nur einen Volterra–Kern jeder Ordnung. In Gl. (4.9) ist jedoch zu erkennen, dass durch einen additiven Eingriff in ein Hammerstein–Modell mehrere Volterra–Kerne gleicher Ordnung entstehen.
- Wie auch beim Beharrungswert kann bei den Kernen gleicher Ordnung der Einfluss der unterschiedlichen Impulsantworten nicht mehr unterschieden werden. Die auftretenden Volterra–Kerne sind analog zum SISO–Fall ausschließlich auf der Hauptdiagonale besetzt.

In [60] wurde gezeigt, dass für eine beliebige Anzahl an Eingangsgrößen u_1, u_2, \dots, u_r und einem beliebigen Grad q der Nichtlinearitäten, die Volterra–Funktionalpotenzreihe für linear verkoppelte Hammerstein–Modelle wie folgt formuliert werden kann:

$$\begin{aligned}
 y[k] = & g_0 + \sum_{i=1}^m g_1[i] u_1[k-i] + \sum_{i=1}^m g_2[i] u_2[k-i] + \dots + \sum_{i=1}^m g_r[i] u_r[k-i] + \\
 & + \sum_{i=1}^m g_1[i, i] u_1^2[k-i] + \sum_{i=1}^m g_2[i, i] u_2^2[k-i] + \sum_{i=1}^m g_3[i, i] u_1[k-i] u_2[k-i] + \dots + \\
 & + \sum_{i=1}^m g_{\binom{r+1}{r}}[i, i] u_r^2[k-i] + \dots + \sum_{i=1}^m g_1[i, \dots, i] u_1^q[k-i] + \sum_{i=1}^m g_2[i, \dots, i] u_2^q[k-i] + \\
 & + \sum_{i=1}^m g_3[i, \dots, i] u_1[k-i] u_2^{q-1}[k-i] + \dots + \sum_{i=1}^m g_{\binom{q+r-1}{r}}[i, \dots, i] u_r^q[k-i] \quad (4.10)
 \end{aligned}$$

Mit Gl. (4.10) können Systeme, wie sie in Abb. 4.5 beispielhaft für eine Anzahl von $r = 2$ Eingängen dargestellt sind, mathematisch beschrieben werden.

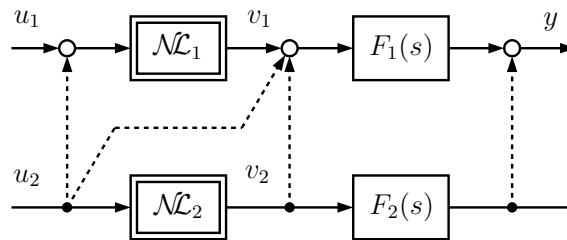


Abb. 4.5: Zwei linear verkoppelte Hammerstein–Modelle nach Gl. (4.10)

In Abb. 4.5 muss beachtet werden, dass in der Regel jeweils nur eine Verkopplung

(gestrichelte Linie) und keine Mehrfachverkopplungen durch Gl. (4.10) darstellbar sind. Ausnahmefälle von dieser Regel wurden in [60] untersucht. Diese sollen jedoch nicht weiter betrachtet werden.

Bisher wurde davon ausgegangen, dass die linear verkoppelten Hammerstein-Modelle sowohl denselben Grad q der Nichtlinearitäten als auch dieselbe Antwortlänge m besitzen. Die folgenden Untersuchungen sollen verdeutlichen, dass unterschiedliche Antwortlängen parallel angeordneter Modelle, sowie unterschiedliche Grade der Nichtlinearitäten keinen Einfluss auf die Allgemeingültigkeit von Gl. (4.10) haben. Es wird ein Modell mit zwei Eingängen betrachtet, bei dem der Grad des ersten Hammerstein-Modells $q_1 > q_2$ dem Grad des zweiten Hammerstein-Modells ist. Wird der Grad q in Gl. (4.10) gleich dem höchsten Grad im System, d.h. gleich q_1 in dem betrachteten Beispiel, gesetzt, kann durch zu Null setzen jener Terme, die keinen Einfluss auf das Ausgangsverhalten haben, die Allgemeingültigkeit von Gl. (4.10) aufrecht erhalten werden. Zukünftig wird daher der Grad q in Gl. (4.10) immer gleich dem höchsten Grad aller Nichtlinearitäten im System gesetzt.

Zwei linear verkoppelte Hammerstein-Modelle besitzen in der Regel nicht dieselben Antwortlängen. Trotzdem existiert im Falle eines MISO-Systems nur eine Antwortlänge m , da ein MISO-System nur einen Ausgang besitzt. Wenn angenommen wird, dass die Antwortlänge des ersten Hammerstein-Modells $m_1 > m_2$ der Antwortlänge des zweiten Hammerstein-Modells ist, dominiert die längere Antwortlänge m_1 das Einschwingverhalten des Gesamtsystems. Die Bestimmung der benötigten Antwortlänge erfolgt durch die Auswertung der Systemsprungantwort, wobei im Falle eines zu identifizierenden MISO-Systems ein Einheitssprung als Anregesignal an beide Eingänge gleichzeitig angelegt wird. Zukünftig wird deshalb für die Herleitung der Volterra-Funktionalpotenzreihen auch nur eine Antwortlänge m für die Beschreibung von MISO-Systemen verwendet, sowie es auch in Gl. (4.10) erfolgte.

4.1.2 Nichtlinearer Eingriff einer Eingangsgröße in das Hammerstein-Modell

Analog zu Kapitel 4.1.1 wird im Folgenden ein Hammerstein-Modell durch einen nichtlinearen Eingriff erweitert. Die möglichen Eingriffspunkte einer weiteren Eingangsgröße wurden schematisch bereits in Abb. 4.1 rechts dargestellt.

Der einfachste Fall ergibt sich, wenn eine weitere Eingangsgröße u_2 am Punkt 5 eingreift. Die beiden Eingangssignale werden miteinander multipliziert und an den Eingang der Nichtlinearität angelegt. Die mathematische Beschreibung ergibt sich wie folgt:

$$\begin{aligned}
 y[k] = & a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] u_1[k-i] u_2[k-i] + a_2 \sum_{i=1}^m h[i] u_1^2[k-i] u_2^2[k-i] + \dots + \\
 & + a_q \sum_{i=1}^m h[i] u_1^q[k-i] u_2^q[k-i]
 \end{aligned} \tag{4.11}$$

Bei einem nichtlinearen Eingriff in das Hammerstein-Modell am Punkt 4 muss eine mehrdimensionale Nichtlinearität angesetzt werden. Nach [43] ist eine mehrdimensionale Nichtlinearität für $r = 2$ Eingänge durch die folgende Gleichung gegeben:

$$v(u_1, u_2) = \sum_{i=0}^q \sum_{j=0}^q a_{ij} \cdot u_1^i \cdot u_2^j = a_{00} + a_{01}u_2 + a_{11}u_1u_2 + a_{12}u_1u_2^2 + \dots + a_{qq}u_1^q u_2^q \quad (4.12)$$

Wird die statische Nichtlinearität um weitere Eingänge erweitert, kommt für jeden zusätzlichen Eingang ein Summenelement hinzu. Die allgemeinste Form einer mehrdimensionalen Nichtlinearität lässt sich formulieren durch:

$$v(u_1, u_2, \dots, u_r) = \sum_{n_1=0}^q \sum_{n_2=0}^q \dots \sum_{n_r=0}^q a_{n_1 n_2 \dots n_r} u_1^{n_1} u_2^{n_2} \dots u_r^{n_r} \quad (4.13)$$

Wird ein zweidimensionales Polynom mit einem beliebigen Grad q der Nichtlinearitäten zur Ableitung der Systembeschreibung eines Hammerstein-Modells mit einer zusätzlichen Eingangsgröße im Punkt 4 eingesetzt, ergibt sich:

$$\begin{aligned} y[k] = & a_{00} \sum_{i=1}^m h[i] + a_{10} \sum_{i=1}^m h[i] u_1[k-i] + \dots + a_{q0} \sum_{i=1}^m h[i] u_1^q[k-i] + \\ & + a_{11} \sum_{i=1}^m h[i] u_1[k-i] u_2[k-i] + a_{21} \sum_{i=1}^m h[i] u_1^2[k-i] u_2[k-i] + \dots + \\ & + a_{q(q-1)} \sum_{i=1}^m h[i] u_1^q[k-i] u_2^{q-1}[k-i] + a_{qq} \sum_{i=1}^m h[i] u_1^q[k-i] u_2^q[k-i] \end{aligned} \quad (4.14)$$

Für einen multiplikativen Eingriff einer weiteren Eingangsgröße im Punkt 3 (vgl. Abb. 4.1 rechts) folgt:

$$\begin{aligned} y[k] = & a_0 \sum_{i=1}^m h[i] u_2[k-i] + a_1 \sum_{i=1}^m h[i] u_1[k-i] u_2[k-i] + \\ & + a_2 \sum_{i=1}^m h[i] u_1^2[k-i] u_2[k-i] + \dots + a_q \sum_{i=1}^m h[i] u_1^q[k-i] u_2[k-i] \end{aligned} \quad (4.15)$$

Die Eingangsgröße u_2 tritt in obiger Gleichung nur zur ersten Potenz auf. Gegenüber den bisherigen Systembeschreibungen existiert in diesem Fall kein Beharrungswert mehr. Wird ein Vergleich zwischen den Systembeschreibungen für Eingriffspunkt 4 und 3 durchgeführt, kann festgestellt werden, dass die Systemgleichung für Eingriffspunkt 3 bereits vollständig in der Gleichung für Eingriffspunkt 4 enthalten ist.

Im Gegensatz zu einem linearen Eingriff darf in Punkt 2 nicht multiplikativ in ein Hammerstein-Modell eingegriffen werden. Ein nichtlinearer Eingriff würde die Übertragungsfunktion zu einer Nichtlinearität überführen. Deshalb wird dieser Fall eines nichtlinearen Eingreifens ausgeschlossen.

Ein linearer Eingriff in ein Hammerstein-Modell in Punkt 1 führte zu einem sog. Durchgriff und war daher nicht erlaubt. Wird die Eingangsgröße u_2 mit allen Termen der Systemgleichung eines Hammerstein-Modells multipliziert, wird deutlich, dass es sich auch bei einem direkten nichtlinearen Eingriff in Punkt 1 um einen Durchgriff handelt. Die Eingangsgröße u_2 wirkt um den Beharrungswert verstärkt ohne Zeitverzögerung auf den Ausgang, so dass auch dieser Fall nicht weiter untersucht wird.

In [60] wurde gezeigt, dass die Systembeschreibungen zweier Hammerstein-Modelle miteinander multipliziert werden müssen, um alle bisher betrachteten nichtlinearen Eingriffe in einer Gleichung zusammenfassen zu können. Beispielhaft soll dies für den Fall gezeigt werden, dass die Nichtlinearitäten den Grad $q = 2$ besitzen. In Abb. 4.6 ist das betrachtete System veranschaulicht.

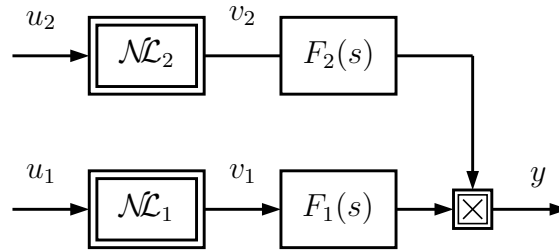


Abb. 4.6: MISO-System aus zwei ausgangsseitig multiplikativ verkoppelten Hammerstein-Modellen

Die Systemgleichung für zwei multiplikativ verkoppelte Hammerstein-Modelle mit dem Grad $q = 2$ lautet wie folgt:

$$\begin{aligned}
 y[k] = & a_{10}a_{20} \sum_{i_1=1}^m h_1[i_1] \sum_{i_2=1}^m h_2[i_2] + a_{11}a_{20} \sum_{i_2=1}^m h_2[i_2] \sum_{i_1=1}^m h_1[i_1] u_1[k - i_1] + \\
 & + a_{10}a_{21} \sum_{i_1=1}^m h_1[i_1] \sum_{i_2=1}^m h_2[i_2] u_2[k - i_2] + a_{11}a_{21} \sum_{i_1=1}^m \sum_{i_2=1}^m h_1[i_1] h_2[i_2] u_1[k - i_1] u_2[k - i_2] + \\
 & + a_{12}a_{20} \sum_{i_2=1}^m h_2[i_2] \sum_{i_1=1}^m h_1[i_1] u_1^2[k - i_1] + a_{12}a_{21} \sum_{i_1=1}^m \sum_{i_2=1}^m h_1[i_1] h_2[i_2] u_1^2[k - i_1] u_2[k - i_2] + \\
 & + a_{10}a_{22} \sum_{i_1=1}^m h_1[i_1] \sum_{i_2=1}^m h_2[i_2] u_2^2[k - i_2] + a_{11}a_{22} \sum_{i_1=1}^m \sum_{i_2=1}^m h_1[i_1] h_2[i_2] u_1[k - i_1] u_2^2[k - i_2] + \\
 & + a_{12}a_{22} \sum_{i_1=1}^m \sum_{i_2=1}^m h_1[i_1] h_2[i_2] u_1^2[k - i_1] u_2^2[k - i_2] \tag{4.16}
 \end{aligned}$$

Bezüglich einer Überführung in eine Volterra-Funktionalpotenzreihe können die folgenden Aussagen getroffen werden:

- Der erste Term entspricht in der Volterra-Funktionalpotenzreihe dem Beharrungswert g_0 . Wie auch im Falle eines linearen Eingriffs in das Hammerstein-

Modell kann zwischen ursprünglich zwei Gleichanteilen nicht mehr unterschieden werden. Damit folgt für den Beharrungswert g_0 :

$$g_0 = a_{10}a_{20} \sum_{i_1=1}^m h_1[i_1] \sum_{i_2=1}^m h_2[i_2]$$

- Im zweiten und dritten Term sind die Eingänge u_1 und u_2 nur in linearer Form enthalten. Innerhalb dieser Terme gibt es keine Möglichkeit zu unterscheiden, wie viel jede Impulsantwort multipliziert mit einem Eingangssignal zum Systemausgang beiträgt. Die Gewichte der beiden linearen Volterra-Kerne können mathematisch beschrieben werden durch:

$$g_1[i_1] = a_{11}a_{20} \sum_{i_2=1}^m h_2[i_2] h_1[i_1] \quad g_2[i_2] = a_{10}a_{21} \sum_{i_1=1}^m h_1[i_1] h_2[i_2]$$

- Der fünfte und der siebte Term stellt jeweils einen Volterra-Kern zweiter Ordnung dar, der ausschließlich auf der Hauptdiagonale besetzt ist. Die Elemente dieser beiden Volterra-Kerne lassen sich formulieren durch:

$$g_1[i_1, i_1] = a_{12}a_{20} \sum_{i_2=1}^m h_2[i_2] h_1[i_1] \quad g_2[i_2, i_2] = a_{10}a_{22} \sum_{i_1=1}^m h_1[i_1] h_2[i_2]$$

- Bei den übrigen Doppelsummen handelt es sich um voll besetzte Volterra-Kerne zweiter Ordnung. In jeder Doppelsumme sind sowohl Vergangenheitswerte des ersten als auch des zweiten Eingangs zu finden. Zu beachten ist, dass die Volterra-Kerne höherer Ordnung nicht mehr symmetrisch sind, wie es beim Wiener-Modell der Fall war, d.h. die Symmetrieeigenschaft aus Kapitel 3.3.4.1 ist nicht länger gültig. Für unsymmetrische Volterra-Kerne zweiter Ordnung gilt zum Beispiel:

$$g[i_1, i_2] \neq g[i_2, i_1]$$

Unsymmetrische Volterra-Kerne höherer Ordnungen werden, wie alle anderen Volterra-Kerne, in Kapitel 4.3.1 noch genauer behandelt. Die Elemente der unsymmetrischen Volterra-Kerne aus Gl. (4.16) lassen sich wie folgt formulieren:

$$\begin{aligned} g_3[i_1, i_2] &= a_{11}a_{21}h_1[i_1] h_2[i_2] & g_4[i_1, i_2] &= a_{12}a_{21}h_1[i_1] h_2[i_2] \\ g_5[i_1, i_2] &= a_{11}a_{22}h_1[i_1] h_2[i_2] & g_6[i_1, i_2] &= a_{12}a_{22}h_1[i_1] h_2[i_2] \end{aligned}$$

Somit lässt sich die Volterra-Funktionalpotenzreihe für Gl. (4.16) wie folgt definieren:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i_1=1}^m g_1[i_1] u_1[k - i_1] + \sum_{i_2=1}^m g_2[i_2] u_2[k - i_2] + \\
& + \sum_{i_1=1}^m g_1[i_1, i_1] u_1^2[k - i_1] + \sum_{i_2=1}^m g_2[i_2, i_2] u_2^2[k - i_2] + \\
& + \sum_{i_1=1}^m \sum_{i_2=1}^m g_3[i_1, i_2] u_1[k - i_1] u_2[k - i_2] + \sum_{i_1=1}^m \sum_{i_2=1}^m g_4[i_1, i_2] u_1^2[k - i_1] u_2[k - i_2] + \\
& + \sum_{i_1=1}^m \sum_{i_2=1}^m g_5[i_1, i_2] u_1[k - i_1] u_2^2[k - i_2] + \sum_{i_1=1}^m \sum_{i_2=1}^m g_6[i_1, i_2] u_1^2[k - i_1] u_2^2[k - i_2]
\end{aligned} \tag{4.17}$$

Wird die Volterra-Funktionalpotenzreihe nach Gl. (4.17) auf eine beliebige Anzahl von r Eingängen erweitert, entstehen keine Doppelsummen, sondern Mehrfachsummen mit der entsprechenden Anzahl r an Einzelsummen. Daneben entstehen bei einer Erhöhung der Anzahl an Eingängen mehr Kombinationsmöglichkeiten zwischen den Eingängen und deren unterschiedlicher Potenzen, d.h. es entstehen noch mehr unsymmetrische Volterra-Kerne höherer Ordnungen. Wird der Grad der Nichtlinearitäten erhöht, müssen zusätzliche Verkopplungen der Eingangssignale berücksichtigt werden. Werden alle betrachteten nichtlinearen Eingriffe in ein Hammerstein-Modell in einer Volterra-Funktionalpotenzreihe zusammengefasst, lässt sich diese für eine beliebige Anzahl von r Eingängen und einem beliebigen Grad q der Nichtlinearitäten wie folgt beschreiben:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i_1=1}^m g_1[i_1] u_1[k - i_1] + \sum_{i_2=1}^m g_2[i_2] u_2[k - i_2] + \dots + \sum_{i_r=1}^m g_r[i_r] u_r[k - i_r] + \\
& + \sum_{i_1=1}^m g_1[i_1, i_1] u_1^2[k - i_1] + \sum_{i_2=1}^m g_2[i_2, i_2] u_2^2[k - i_2] + \dots + \sum_{i_r=1}^m g_r[i_r, i_r] u_r^2[k - i_r] + \\
& + \sum_{i_1=1}^m \sum_{i_2=1}^m g_{r+1}[i_1, i_2] u_1[k - i_1] u_2[k - i_2] + \dots + \sum_{i_1=1}^m g_1[i_1, \dots, i_1] u_1^q[k - i_1] + \dots + \\
& + \sum_{i_r=1}^m g_r[i_r, \dots, i_r] u_r^q[k - i_r] + \dots + \sum_{i_1=1}^m \dots \sum_{i_r=1}^m g_1[i_1, \dots, i_r] u_1^q[k - i_1] \dots u_r^q[k - i_r]
\end{aligned} \tag{4.18}$$

Zur Veranschaulichung von Gl. (4.18) sind in Abb. 4.7 noch einmal alle Systeme für den Fall $r = 2$ dargestellt, die durch Gl. (4.18) beschrieben werden können. Dabei darf in der Regel nur eine der gestrichelten Verkopplungen im System vorkommen. Ausnahmefälle von dieser Regel wurden ebenfalls in [60] untersucht, sollen jedoch nicht weiter betrachtet werden.

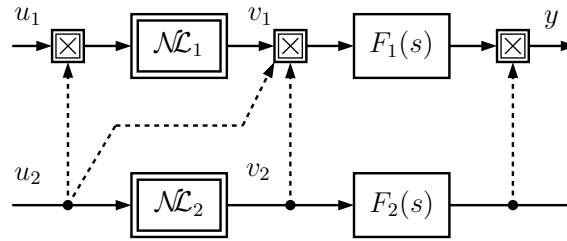


Abb. 4.7: Zwei nichtlinear verknüpfte Hammerstein-Modelle nach Gleichung (4.18)

4.2 MISO-Systembeschreibungen durch Erweiterung des Wiener-Modells

In analoger Vorgehensweise zu Kapitel 4.1, werden in diesem Kapitel systematisch die Möglichkeiten einer Verkopplung von Wiener-Modellen untersucht. In Abb. 4.8 sind alle Möglichkeiten eines linearen und eines nichtlinearen Eingriffs in das Wiener-Modell dargestellt.

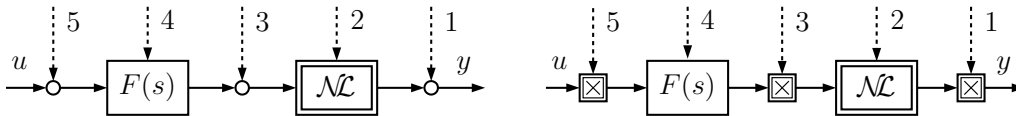


Abb. 4.8: Lineare und nichtlineare Eingriffspunkte einer weiteren Eingangsgröße beim Wiener-Modell

Die gestrichelten Linien deuten darauf hin, dass an diesen Stellen in das Modell eingegriffen werden kann. Beispielhaft werden in den folgenden Abschnitten die Systembeschreibungen für eine Anzahl von $r = 2$ Eingängen abgeleitet und soweit dies möglich⁶, auf den allgemeinen Fall einer beliebigen Anzahl r an Eingängen überführt. Die Systembeschreibungen sind Ausgangspunkt für die Ableitung der Volterra-Funktionalpotenzreihen, die wiederum für die Identifikation benötigt werden.

4.2.1 Linearer Eingriff einer Eingangsgröße in das Wiener-Modell

Im Unterschied zum Hammerstein-Modell befindet sich beim Wiener-Modell die statische Nichtlinearität hinter der zeitinvarianten Übertragungsfunktion. Es ist zu erkennen, dass es sich bei einem direkten, linearen Eingriff einer weiteren Eingangs-

⁶An dieser Stelle sei darauf hingewiesen, dass die Systembeschreibungen beim Wiener-Modell bereits bei $r = 2$ Eingangsgrößen eine solche Komplexität erreichen, dass eine anschauliche Verallgemeinerung auf r Eingangssignale nicht möglich ist.

größe u_2 in den Punkten 1, 2 und 3 um einen Durchgriff handelt, d.h. die Eingangsgröße u_2 wirkt ohne Zeitverzögerung auf den Systemausgang y . Strukturen, die einen Durchgriff aufweisen, werden nicht weiter untersucht.

Die Möglichkeit, dass eine weitere Eingangsgröße u_2 in Punkt 5 (vgl. Abb. 4.8 links) eingreift, kann in den Fall überführt werden, dass diese Eingangsgröße über die Übertragungsfunktion $F_2(s) = F_1(s) = F(s)$ in Punkt 3 eingreift. Ein allgemeinerer Fall ist jedoch, dass in dieser Struktur die Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ unterschiedlich sind. In Abb. 4.9 sind diese Zusammenhänge veranschaulicht.

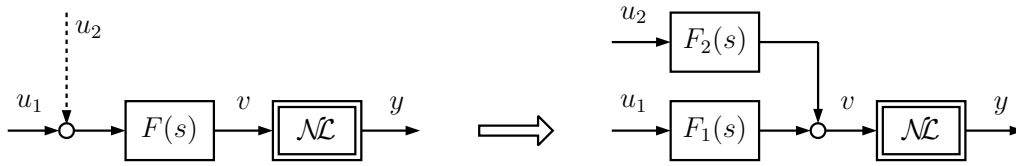


Abb. 4.9: Linearer Eingriff im Punkt 5

Die gleichen Überlegungen gelten auch für Eingriffspunkt 4 nach Abb. 4.8 links, wenn die Übertragungsfunktion $F(s)$ in zwei Teilübertragungsfunktionen $F_1(s)$ und $F_2(s)$ aufgespalten werden kann. Für die zuletzt betrachteten Möglichkeiten eines linearen Eingriffs in das Wiener-Modell lässt sich folgende Systembeschreibung formulieren:

$$\begin{aligned}
y[k] = & a_0 + a_1 \sum_{i_1=1}^m h_1[i_1] u_1[k - i_1] + a_2 \sum_{i_{11}=1}^m \sum_{i_{12}=i_{11}}^m h_1[i_{11}] h_1[i_{12}] u_1[k - i_{11}] u_1[k - i_{12}] + \dots + \\
& + a_q \sum_{i_{11}=1}^m \dots \sum_{i_{1q}=i_{1(q-1)}}^m h_1[i_{11}] \dots h_1[i_{1q}] u_1[k - i_{11}] \dots u_1[k - i_{1q}] + \\
& + a_1 \sum_{i_2=1}^m h_2[i_2] u_2[k - i_2] + a_2 \sum_{i_{21}=1}^m \sum_{i_{22}=i_{21}}^m h_2[i_{21}] h_2[i_{22}] u_2[k - i_{21}] u_2[k - i_{22}] + \dots + \\
& + a_q \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m h_2[i_{21}] \dots h_2[i_{2q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] + \tag{4.19} \\
& + 2a_2 \sum_{i_{11}=1}^m \sum_{i_{21}=1}^m h_1[i_{11}] h_2[i_{21}] u_1[k - i_{11}] u_2[k - i_{21}] + \\
& + 3a_3 \sum_{i_{11}=1}^m \sum_{i_{12}=i_{11}}^m \sum_{i_{21}=1}^m h_1[i_{11}] h_1[i_{12}] h_2[i_{21}] u_1[k - i_{11}] u_1[k - i_{12}] u_2[k - i_{21}] + \\
& + 3a_3 \sum_{i_{11}=1}^m \sum_{i_{21}=1}^m \sum_{i_{22}=i_{21}}^m h_1[i_{11}] h_2[i_{21}] h_2[i_{22}] u_1[k - i_{11}] u_2[k - i_{21}] u_2[k - i_{22}] + \dots
\end{aligned}$$

Auf Gl. 4.19 wird im Folgenden noch genauer eingegangen. Zunächst soll jedoch ein weiterer Fall betrachtet werden. In Abb. 4.10 ist die ausgangsseitige Addition zweier Wiener-Modelle veranschaulicht.

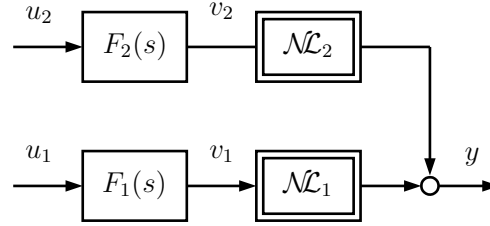


Abb. 4.10: Ausgangsseitige Addition zweier Wiener-Modelle

Die Addition der beiden statischen Nichtlinearitäten kann abhängig von den Signalen v_1 und v_2 durch

$$y(v_1, v_2) = a_{10} + a_{11}v_1 + a_{12}v_1^2 + \dots + a_{1q}v_1^q + a_{20} + a_{21}v_2 + a_{22}v_2^2 + \dots + a_{2q}v_2^q \quad (4.20)$$

beschrieben werden. Werden für v_1 und v_2 die Faltungssummen eingesetzt, kann die Systemgleichung wie folgt formuliert werden:

$$\begin{aligned} y[k] = & a_{10} + a_{11} \sum_{i_1=1}^m h_1[i_1] u_1[k - i_1] + a_{12} \sum_{i_{11}=1}^m \sum_{i_{12}=i_{11}}^m h_1[i_{11}] h_1[i_{12}] u_1[k - i_{11}] u_1[k - i_{12}] + \dots + \\ & + a_{1q} \sum_{i_{11}=1}^m \dots \sum_{i_{1q}=i_{1(q-1)}}^m h_1[i_{11}] \dots h_1[i_{1q}] u_1[k - i_{11}] \dots u_1[k - i_{1q}] + \quad (4.21) \\ & + a_{20} + a_{21} \sum_{i_2=1}^m h_2[i_2] u_2[k - i_2] + a_{22} \sum_{i_{21}=1}^m \sum_{i_{22}=i_{21}}^m h_2[i_{21}] h_2[i_{22}] u_2[k - i_{21}] u_2[k - i_{22}] + \dots + \\ & + a_{2q} \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m h_2[i_{21}] \dots h_2[i_{2q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] \end{aligned}$$

Ein Vergleich mit Gl. (4.19) zeigt, dass in obiger Systembeschreibung keine weiteren Verkopplungen zwischen den Eingangsgrößen existieren, die berücksichtigt werden müssten. Die ausgangsseitige Addition von zwei Wiener-Modellen, kann auch gleichungsmäßig als Addition zweier einzelner Wiener-Modelle beschrieben werden. Es treten in diesem Fall neben den linearen Volterra-Kernen ausschließlich symmetrische Volterra-Kerne höherer Ordnungen auf. Im Gegensatz dazu treten in Gl. (4.19) weitere Volterra-Kerne auf, die entweder vollkommen unsymmetrisch sind oder teilweise symmetrisch und teilweise unsymmetrisch sind. So ist z.B. der letzte explizit angeschriebene Summand in Gl. (4.19) in den Richtungen i_{21} und i_{22} symmetrisch, während er in den Richtungen i_{11} und i_{21} unsymmetrisch ist.

Aufgrund der Komplexität von verkoppelten Wiener-Modellen wird auf die Formulierung einer allgemeinen Volterra-Funktionalpotenzreihe für r Eingangssignale verzichtet. In Gl. (4.22) ist die aus Gl. (4.19) resultierende Volterra-Funktionalpotenzreihe für $r = 2$ Eingangssignale und einen allgemeinen Grad q der Nichtlinearität

angeschrieben:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i_1=1}^m g_1[i_1] u_1[k - i_1] + \sum_{i_2=1}^m g_2[i_2] u_2[k - i_2] + \\
& + \sum_{i_{11}=1}^m \sum_{i_{12}=i_{11}}^m g_1[i_{11}, i_{12}] u_1[k - i_{11}] u_1[k - i_{12}] + \sum_{i_{21}=1}^m \sum_{i_{22}=i_{21}}^m g_2[i_{21}, i_{22}] u_2[k - i_{21}] u_2[k - i_{22}] + \\
& + \sum_{i_{11}=1}^m \sum_{i_{21}=1}^m g_3[i_{11}, i_{21}] u_1[k - i_{11}] u_2[k - i_{21}] + \dots + \\
& + \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m g_2[i_{21}, \dots, i_{2q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] + \dots
\end{aligned} \tag{4.22}$$

In Gl. (4.22) treten alle Kombinationsmöglichkeiten zwischen den Eingangsvariablen auf, d.h. sowohl eine Vermischung unterschiedlicher Potenzen als auch eine Vermischung unterschiedlicher Zeitpunkte der Eingangsvariablen. Daraus ergeben sich eine Vielzahl von Volterra-Kernen gleicher Ordnungen, die teilweise symmetrisch und zum Teil auch unsymmetrisch sind.

Wird Gl. (4.22) auf r Eingangsgrößen erweitert, kann die Annahme getroffen werden, dass eine noch komplexere Verschachtelungsmöglichkeit von unterschiedlichen Potenzen und Zeitpunkten nicht existiert, d.h. alle denkbaren linearen und nichtlinearen Strukturen durch eine Volterra-Funktionalpotenzreihe dieser Form beschrieben werden können. Diese Annahme für einen beliebigen Grad q der Nichtlinearitäten und eine beliebige Anzahl r von Eingangssignalen mathematisch zu beweisen, wird an dieser Stelle nicht weiter verfolgt und deshalb die Untersuchungen abgebrochen. Der Vollständigkeit halber werden jedoch im nachfolgenden Kapitel 4.2.2 noch die einfachen Fälle eines nichtlinearen Eingriffs in ein Wiener-Modell beschrieben.

In Abb. 4.11 sind noch einmal die linearen Verkopplungen zweier Wiener-Modelle dargestellt, die durch Gl. (4.22) beschrieben werden können.

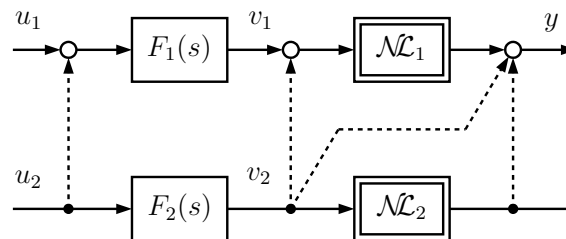


Abb. 4.11: Zwei linear verkoppelte Wiener-Modelle nach Gl. (4.22)

In Abb. 4.11 muss wiederum beachtet werden, dass in der Regel jeweils nur eine Verkopplung (gestrichelte Linie) und keine Mehrfachverkopplungen durch Gl. (4.22) darstellbar sind.

4.2.2 Nichtlinearer Eingriff einer Eingangsgröße in das Wiener-Modell

In Abb. 4.8 rechts wurden bereits die Möglichkeiten eines nichtlinearen Eingriffs in ein Wiener-Modell dargestellt. Wie bei einem linearen Eingriff in das Wiener-Modell, handelt es sich bei einem direkten Einwirken einer Eingangsgröße u_2 in den Punkten 1, 2 und 3 um einen Durchgriff. Solche Strukturen werden nicht weiter betrachtet. In Punkt 4 darf beim Wiener-Modell nicht multiplikativ eingegriffen werden, da ansonsten die zeitinvariante Übertragungsfunktion in eine nichtlineare Funktion übergeht. Eingriffspunkt 5 ist der einzige Punkt im Wiener-Modell, in dem ein nichtlinearer Eingriff erfolgen kann. Die Systembeschreibung hierfür lässt sich wie folgt formulieren:

$$\begin{aligned}
 y[k] = & a_0 + a_1 \sum_{i_1=1}^m h[i_1] u_1[k-i_1] u_2[k-i_1] + \\
 & + a_2 \sum_{i_1=1}^m \sum_{i_2=i_1}^m h[i_1] h[i_2] u_1[k-i_1] u_1[k-i_2] u_2[k-i_1] u_2[k-i_2] + \\
 & + \dots + a_q \sum_{i_1=1}^m \dots \sum_{i_q=i_{q-1}}^m h[i_1] \dots h[i_q] u_1[k-i_1] \dots u_1[k-i_q] u_2[k-i_1] \dots u_2[k-i_q]
 \end{aligned} \tag{4.23}$$

Es ist zu erkennen, dass die beiden Eingangsvariablen u_1 und u_2 immer gemeinsam in einer Summe vorkommen. Dieses System kann durch die Gleichung eines Wiener-Modells mit einer Eingangsgröße $u_1 \cdot u_2$ mathematisch beschrieben werden. Die Volterra-Kerne in obiger Gleichung entsprechen genau denen im Wiener-Modell.

Eine weitere Möglichkeit eines nichtlinearen Eingriffs ist, dass die Eingangsgröße u_2 an eine zeitinvariante Übertragungsfunktion $F_2(s)$ angelegt wird, deren Ausgang bei Punkt 3 eingreift. Die Beschreibung eines solchen Systems ändert sich nicht, wenn der Ausgang der Übertragungsfunktion in Punkt 1 eingreift, d.h. eine Multiplikation vor oder nach der statischen Nichtlinearität führt auf die gleiche Systembeschreibung. In Abb. 4.12 sind diese Zusammenhänge noch einmal veranschaulicht.

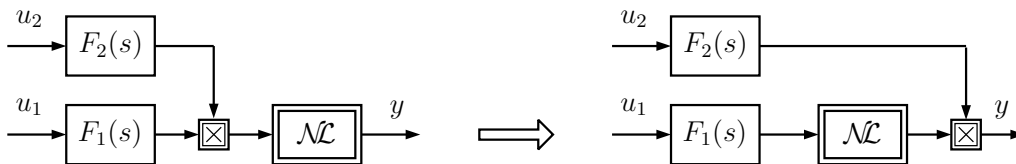


Abb. 4.12: Nichtlineare Verkopplung eines Wiener-Modells mit einer Übertragungsfunktion im Eingriffspunkt 1 und 3

Diese Strukturen entsprechen jedoch einer vereinfachten Form von einer Multiplikation zweier Wiener-Modelle, wie sie in Abb. 4.13 dargestellt ist.

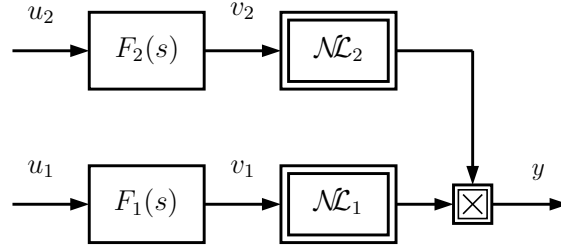


Abb. 4.13: *MISO-System aus zwei ausgangsseitig multiplikativ verkoppelten Wiener-Modellen*

Die Systembeschreibung für die Multiplikation zweier Wiener-Modelle ergibt sich wie folgt:

$$\begin{aligned}
y[k] = & a_{10}a_{20} + a_{11}a_{20} \sum_{i_1=1}^m h_1[i_1] u_1[k - i_1] + a_{10}a_{21} \sum_{i_2=1}^m h_2[i_2] u_2[k - i_2] + \dots + \\
& + a_{10}a_{2q} \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m h_2[i_{21}] \dots h_2[i_{2q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] + \dots + \quad (4.24) \\
& + a_{11}a_{2q} \sum_{i_{11}=1}^m \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m h_1[i_{11}] h_2[i_{21}] \dots h_2[i_{2q}] u_1[k - i_{11}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] + \dots + \\
& + a_{1q}a_{2q} \sum_{i_{11}=1}^m \dots \sum_{i_q=i_{q-1}}^m \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m h_1[i_{11}] \dots h_1[i_{1q}] h_2[i_{21}] \dots h_2[i_{2q}] \cdot \\
& \cdot u_1[k - i_{11}] \dots u_1[k - i_{1q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}]
\end{aligned}$$

Aufgrund der Komplexität der obigen Systembeschreibung wird an dieser Stelle ebenfalls darauf verzichtet die Volterra-Funktionalpotenzreihe für r Eingangssignale zu formulieren. Jedoch soll die obige Gleichung, die die nichtlineare Verkopplung zweier Wiener-Modelle beschreibt, in eine Volterra-Funktionalpotenzreihe überführt werden:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i_1=1}^m g_1[i_1] u_1[k - i_1] + \sum_{i_2=1}^m g_2[i_2] u_2[k - i_2] + \dots + \\
& + \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m g_2[i_{21}, \dots, i_{2q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}] + \dots + \quad (4.25) \\
& + \sum_{i_{11}=1}^m \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m g_1[i_{11}, i_{21}, \dots, i_{2q}] u_1[k - i_{11}] u_2[k - i_{22}] \dots u_2[k - i_{2q}] + \dots + \\
& + \sum_{i_{11}=1}^m \dots \sum_{i_{1q}=i_{1(q-1)}}^m \sum_{i_{21}=1}^m \dots \sum_{i_{2q}=i_{2(q-1)}}^m g_1[i_{11}, \dots, i_{1q}, i_{21}, \dots, i_{2q}] \cdot \\
& \cdot u_1[k - i_{11}] \dots u_1[k - i_{1q}] u_2[k - i_{21}] \dots u_2[k - i_{2q}]
\end{aligned}$$

Die Systembeschreibung nach Gl. (4.25) beinhaltet mehrere Volterra-Kerne gleicher Ordnung, die sich durch ihre Indizes unterscheiden, d.h. dass sowohl symmetrische als auch unsymmetrische Volterra-Kerne in dieser Systembeschreibung enthalten sind. In Abb. 4.14 sind alle Strukturen für $r = 2$ Eingangssignale dargestellt, die durch einen nichtlinearen Eingriff in ein Wiener-Modell mit Gl. (4.25) beschrieben werden können.

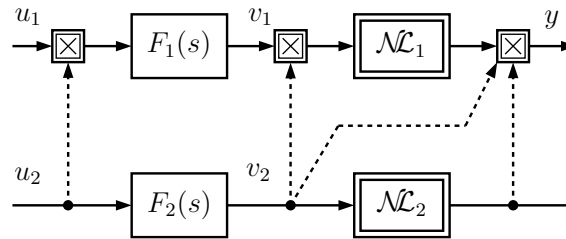


Abb. 4.14: Zwei nichtlinear verkoppelte Wiener-Modelle nach Gl. (4.25)

In Abb. 4.14 muss wiederum beachtet werden, dass in der Regel jeweils nur eine Verkopplung (gestrichelte Linie) und keine Mehrfachverkopplungen durch Gl. (4.25) darstellbar sind.

4.3 Identifikation von Mehrgrößensystemen

In den Kapiteln 4.1 und 4.2 wurde gezeigt, wie aus den Systembeschreibungen von linear und nichtlinear verkoppelten Hammerstein- und Wiener-Modellen die jeweilige Volterra-Funktionalpotenzreihe abgeleitet werden kann. Dabei wurde klar, dass bereits bei zwei Eingangssignalen eine hohe Komplexität der Systembeschreibungen erreicht wird, die eine Verallgemeinerung dieser Gleichungen auf beliebig viele Eingangssignale wenig sinnvoll erscheinen lassen. Es konnte jedoch gezeigt werden, dass neben den bereits bekannten Volterra-Kernen aus der Beschreibung von SISO-Systemen und den in der Literatur bisher noch nicht bekannten unsymmetrischen Volterra-Kernen aus der Beschreibung von MISO-Systemen, keine weiteren Volterra-Kerne existieren. Bei den folgenden Betrachtungen zur Identifikation soll die Anzahl der Eingangssignale aus Gründen der Anschaulichkeit auf $r = 2$ beschränkt bleiben. Sollte für Anwendungsfälle in der Praxis eine höhere Anzahl an Eingangssignalen erforderlich sein, muss diese spezielle Systembeschreibung für den Einzelfall bestimmt werden.

Ausgangspunkt für die Identifikation von MISO-Systemen ist eine auf Basis der Systembeschreibungen aus den Kapiteln 4.1 und 4.2 abgeleitete Volterra-Funktionalpotenzreihe. Die Identifikation von MISO-Systemen erfolgt ebenfalls in einer Ausgangsfehlerstruktur, wie sie in Abb. 3.2 dargestellt ist. Als Eingang für den Identifikationsalgorithmus dient der Eingangsvektor \underline{u} .

Die abgeleiteten Volterra-Funktionalpotenzreihen müssen zunächst in eine vektoriel-

le Schreibweise überführt werden, in der jedes Element $g[i, \dots]$ eines Volterra-Kerns einem Parameter θ_i entspricht. Wie schon bei der Identifikation von blockorientierten SISO-Systemen würde eine Identifikation ohne die Einführung orthonormaler Basisfunktionen zu einem enorm hohen Rechenaufwand führen. Aus diesem Grund ist die Einführung orthonormaler Basisfunktionen bei der Identifikation von MISO-Systemen zwingend erforderlich.

Bei einer Identifikation ist das Anregesignal von entscheidender Bedeutung. Ein geeignetes Anregesignal für die Identifikation von SISO-Systemen war das amplitudenmodulierte Pseudo-Rausch-Binärsignal. Dieses Anregesignal eignet sich auch für die Identifikation von MISO-Systemen, wobei im Falle von MISO-Systemen darauf geachtet werden muss, dass für jeden Eingang ein gegenüber den anderen Eingängen unabhängiges Signal verwendet wird. Das bedeutet, dass im Falle mehrerer Eingänge der gesamte Bereich im Raum \mathbb{R}^r idealerweise abgedeckt werden muss.

4.3.1 Parameterreduktion durch Einführung orthonormaler Basisfunktionen bei Mehrgrößensystemen

In Kapitel 3.3.4.3 wurde bereits gezeigt, dass bei der Identifikation von SISO-Systemen die Anzahl an unbekanntem Parametern wesentlich verringert werden kann. Bei MISO-Systemen kommt der Parameterreduktion mittels Basisfunktionen eine noch gewichtigere Rolle zu. Beispielhaft wird in diesem Kapitel die Parameterreduktion von Volterra-Kernen bis zur zweiten Ordnung beschrieben. Die Erweiterung auf eine Parameterreduktion von Volterra-Kernen beliebiger Ordnung erfolgt im jeweiligen Anschluss. Die bereits aus den SISO-Systemen bekannten Volterra-Kerne werden ebenso behandelt, wie der neu hinzugekommene unsymmetrische Volterra-Kern.

Bei den in den Kapiteln 4.1 und 4.2 betrachteten Systemen, wie auch bei den SISO-Systemen in Kapitel 3.3.4.2, treten lineare Volterra-Kerne auf. Als Beispiel ist ein beliebiger, linearer Volterra-Kern in Abb. 4.15 links dargestellt.

Ein linearer Volterra-Kern hängt nur von einem Index ab und stellt die Impulsantwort eines linearen Systems dar. Die Verrechnung mit den orthonormalen Basisfunktionen ist sehr einfach und kann mit dem reduzierten Gewichtsfolgevektor \underline{g}_r und der Basisfunktionenmatrix $\tilde{\mathbf{R}}$ wie folgt formuliert werden:

$$y_{lin}[k] = \sum_{i=1}^m g[i] u[k-i] \quad \xrightarrow{\text{OBFs}} \quad y_{lin}[k] = \sum_{i=1}^m \left(\sum_{j=1}^{m_r} g_r[j] \tilde{r}_j[i] \right) u[k-i] \quad (4.26)$$

Durch die Einführung orthonormaler Basisfunktionen ist es möglich, die Anzahl der Elemente eines linearen Volterra-Kerns von m auf m_r zu reduzieren.

Die Elemente eines reduzierten Gewichtsfolgevektors \underline{g}_r entsprechen nicht mehr ex-

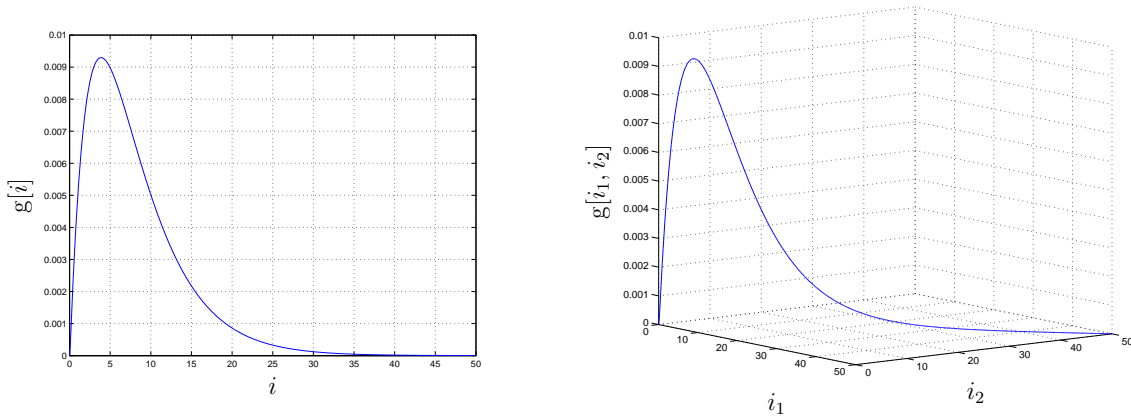


Abb. 4.15: *Linearer Volterra-Kern (links) und Volterra-Kern zweiter Ordnung (rechts) — nur auf der Hauptdiagonale besetzt*

akten Punkten einer Impulsantwort, wie es bei einem nicht reduzierten Gewichtsfolgevektor \underline{g} der Fall ist. Sie sind ein Maß dafür, mit wie viel Gewicht die einzelnen orthonormalen Basisfunktionen in die Systembeschreibung eingehen. Wird die Beschreibung des reduzierten linearen Volterra-Kerns in vektorieller Form durchgeführt, ergibt sich:

$$\begin{aligned}
 y_{lin}[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\
 \underline{\mathcal{A}}_{dyn}^T[k] &= \underline{u}^T \cdot \tilde{\mathbf{R}}^T \\
 \underline{u}^T[k] &= [u[k-1], \dots, u[k-m]] \\
 \underline{\theta}^T &= [g_r[1], \dots, g_r[m_r]]
 \end{aligned} \tag{4.27}$$

Die Parameter $\underline{\theta} \in \mathbb{R}^{m_r \times 1}$ stellen die unbekanntenen Gewichte der Basisfunktionen dar.

Eine weitere Kategorie auftretender Volterra-Kerne sind Volterra-Kerne höherer Ordnung, die ausschließlich auf der Hauptdiagonale besetzt sind. Solche Volterra-Kerne treten bei Hammerstein-Strukturen im SISO-Fall als auch bei unterschiedlich verschalteten MISO-Systemen auf. In Abb. 4.15 rechts ist ein Volterra-Kern zweiter Ordnung, der ausschließlich auf der Hauptdiagonale besetzt ist, dargestellt. Die mathematische Beschreibung eines solchen Volterra-Kerns ergibt sich nach der Einführung orthonormaler Basisfunktionen zu:

$$y_{diag}[k] = \sum_{i=1}^m g[i, i] u^2[k-i] \xrightarrow{\text{OBFs}} y_{diag}[k] = \sum_{i=1}^m \left(\sum_{j=1}^{m_r} g_r[j, j] \tilde{r}_j[i] \right) u^2[k-i] \tag{4.28}$$

Durch die Parameterreduktion kann die Anzahl der unbekanntenen Gewichte, wie bei den linearen Volterra-Kernen, von m auf m_r Elemente verringert werden. Die vektorielle Beschreibung dieser Art von Volterra-Kernen lautet für eine beliebige Ordnung

q wie folgt:

$$\begin{aligned}
 y_{diag}[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\
 \underline{\mathcal{A}}_{dyn}^T[k] &= \underline{u}^{qT} \cdot \tilde{\mathbf{R}}^T \\
 \underline{u}^{qT}[k] &= [u^q[k-1], \dots, u^q[k-m]] \\
 \underline{\theta}^T &= [g_r[1, \dots, 1], \dots, g_r[m_r, \dots, m_r]]
 \end{aligned} \tag{4.29}$$

Die Parametervektor $\underline{\theta} \in \mathbb{R}^{m_r \times 1}$ beinhaltet wiederum die unbekanntenen Gewichte der Basisfunktionen.

Ein weiterer Volterra-Kern, der im Falle eines SISO-Systems bereits beim Wiener-Modell aufgetreten ist und auch in den MISO-Systembeschreibungen aus den Kapiteln 4.1 und 4.2 vorhanden ist, ist der symmetrische Volterra-Kern höherer Ordnung. In Abb. 4.16 links ist beispielhaft ein symmetrischer Volterra-Kern zweiter Ordnung dargestellt.

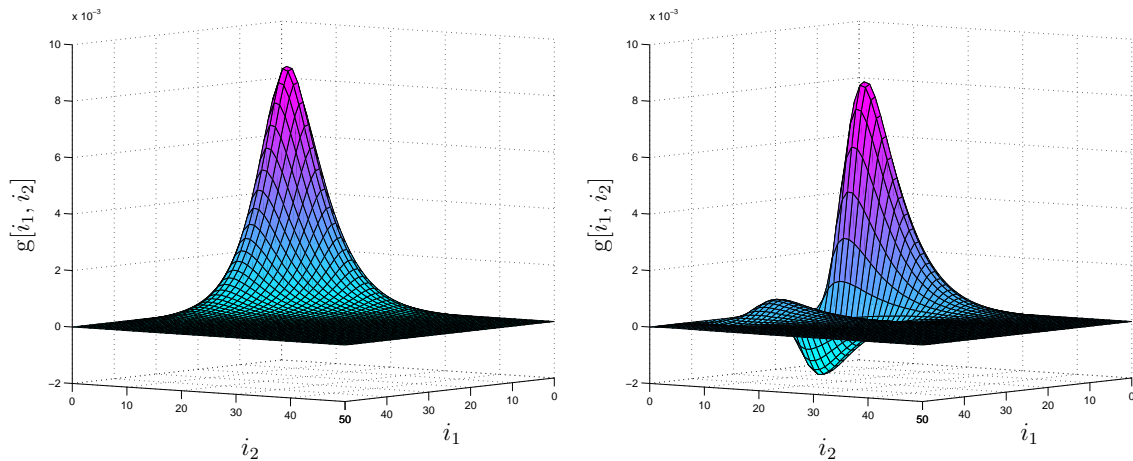


Abb. 4.16: *Symmetrischer Volterra-Kern zweiter Ordnung (links) und unsymmetrischer Volterra-Kern zweiter Ordnung (rechts)*

Die Symmetrie ergibt sich für diesen Volterra-Kern bezüglich der beiden Achsen i_1 und i_2 , d.h. bezüglich der Hauptdiagonale des Volterra-Kerns. Für einen symmetrischen Volterra-Kern zweiter Ordnung erfolgt die Parameterreduktion durch:

$$\begin{aligned}
 y_{sym}[k] &= \sum_{i_1=1}^m \sum_{i_2=i_1}^m g[i_1, i_2] u[k-i_1] u[k-i_2] \\
 \xrightarrow{\text{OBFs}} y_{sym}[k] &= \sum_{i_1=1}^m \sum_{i_2=i_1}^m \left(\sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} g_r[j_1, j_2] \tilde{r}_{j_1}[i_1] \tilde{r}_{j_2}[i_2] \right) u[k-i_1] u[k-i_2]
 \end{aligned} \tag{4.30}$$

Ein symmetrischer Volterra-Kern beliebiger Ordnung besitzt vor der Parameterreduktion durch orthonormale Basisfunktionen $\binom{m+q-1}{q}$ Elemente, wobei die Elemente

bezüglich der Hauptdiagonale symmetrisch sind und daher nicht alle m^q Elemente bestimmt werden müssen. Nach der Parameterreduktion bleiben noch $\binom{m_r+q-1}{q}$ Elemente übrig. Die Beschreibung eines reduzierten, symmetrischen Volterra-Kerns beliebiger Ordnung ergibt sich durch:

$$y_{sym}[k] = \sum_{i_1=1}^m \cdots \sum_{i_q=i_{q-1}}^m \left(\sum_{j_1=1}^{m_r} \cdots \sum_{j_q=j_{q-1}}^{m_r} g_r[j_1, \dots, j_q] \tilde{r}_{j_1}[i_1] \cdots \tilde{r}_{j_q}[i_q] \right) u[k-i_1] \cdots u[k-i_q] \quad (4.31)$$

In Kapitel 3.3.4.3 wurde der $*$ -Operator eingeführt und in Anhang A definiert. Durch den $*$ -Operator ist es möglich symmetrische Volterra-Kerne beliebiger Ordnung kompakt in Vektorschreibweise darzustellen. Die Parameter $\underline{\theta} \in \mathbb{R}^{\binom{m_r+q-1}{q} \times 1}$ stellen die unbekanntenen Gewichte der Basisfunktionen dar.

$$\begin{aligned} y_{sym}[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\ \underline{\mathcal{A}}_{dyn}^T[k] &= \left[\left(\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*q} \right] \\ \underline{u}^T[k] &= \left[u[k-1], \dots, u[k-m] \right] \\ \underline{\theta}^T &= \left[g_r[1, \dots, 1], \dots, g_r[m_r, \dots, m_r] \right] \end{aligned} \quad (4.32)$$

Die letzte Art von Volterra-Kernen sind unsymmetrische Volterra-Kerne höherer Ordnung. Die unsymmetrischen Volterra-Kerne höherer Ordnung treten ausschließlich bei MISO-Systemen auf [59] und sind bisher in der Literatur noch nicht erwähnt worden. Unsymmetrische Volterra-Kerne entstehen immer dann, wenn zwei zeitinvariante Übertragungsfunktionen mit unterschiedlichen Impulsantworten ausgangsseitig miteinander multipliziert werden. In Abb. 4.16 rechts ist beispielhaft ein unsymmetrischer Volterra-Kern zweiter Ordnung dargestellt. Unsymmetrische Volterra-Kerne besitzen genauso wie die symmetrischen Volterra-Kerne unterschiedliche Indizes. In Abb. 4.16 ist zu erkennen, dass sich die Impulsantworten in Richtung von i_1 und von i_2 deutlich unterscheiden. Die Impulsantwort in Richtung von i_1 schwingt nicht über, während die Impulsantwort in Richtung von i_2 ein deutliches Überschwingen zeigt. Die Parameterreduktion eines unsymmetrischen Volterra-Kerns zweiter Ordnung lässt sich wie folgt beschreiben:

$$\begin{aligned} y_{usym}[k] &= \sum_{i_1=1}^m \sum_{i_2=1}^m g[i_1, i_2] u_1[k-i_1] u_2[k-i_2] \\ \xrightarrow{\text{OBFs}} y_{usym}[k] &= \sum_{i_1=1}^m \sum_{i_2=1}^m \left(\sum_{j_1=1}^{m_r} \sum_{j_2=1}^{m_r} g_r[j_1, j_2] \tilde{r}_{j_1}[i_1] \tilde{r}_{j_2}[i_2] \right) u_1[k-i_1] u_2[k-i_2] \end{aligned} \quad (4.33)$$

Ein unsymmetrischer Volterra-Kern höherer Ordnung besitzt vor der Parameterreduktion m^r Elemente, wobei durch die Einführung orthonormaler Basisfunktionen die Anzahl der zu berechnenden Elemente auf m_r^r gesenkt werden kann. Die Parameterreduktion für einen unsymmetrischen Volterra-Kern höherer Ordnung ist

gegeben durch:

$$y_{usym}[k] = \sum_{i_1=1}^m \cdots \sum_{i_r=1}^m \left(\sum_{j_1=1}^{m_r} \cdots \sum_{j_r=1}^{m_r} g_r[j_1, \dots, j_r] \tilde{r}_{j_1}[i_1] \cdots \tilde{r}_{j_r}[i_r] \right) u_1[k-i_1] \cdots u_r[k-i_r] \quad (4.34)$$

Um die Beschreibung von unsymmetrischen Volterra-Kernen in einer kompakten Vektorschreibweise darstellen zu können, wird analog zum $*$ -Operator bei symmetrischen Volterra-Kernen der \otimes -Operator eingeführt. Der \otimes -Operator ist ebenfalls in Anhang A definiert. Die Beschreibung eines reduzierten, unsymmetrischen Volterra-Kerns beliebiger Ordnung ergibt sich mit dem \otimes -Operator zu:

$$\begin{aligned} y_{usym}[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\ \underline{\mathcal{A}}_{dyn}^T[k] &= \left[\left(\underline{u}_1^T[k] \cdot \tilde{\mathbf{R}}^T \right) \otimes \cdots \otimes \left(\underline{u}_r^T[k] \cdot \tilde{\mathbf{R}}^T \right) \right] \\ \underline{u}_i^T[k] &= \left[u_i[k-1], \dots, u_i[k-m] \right] \quad \text{mit } i = 1 \dots r \\ \underline{\theta}^T &= \left[g_r[1, \dots, 1], \dots, g_r[m_r, \dots, m_r] \right] \end{aligned} \quad (4.35)$$

Die Parametervektor $\underline{\theta} \in \mathbb{R}^{m_r \times \dots \times m_r \times 1}$ beinhaltet wiederum die unbekanntenen Gewichte der Basisfunktionen.

In Kapitel 4.2 wurde gezeigt, dass bei MISO-Systemen auch Volterra-Kerne höherer Ordnung entstehen können, die teilweise symmetrisch und teilweise unsymmetrisch sind⁷. Solche Kerne können durch eine kombinierte Anwendung des $*$ -Operators und des \otimes -Operators beschrieben werden.

Nachdem in diesem Kapitel die Parameterreduktion der unterschiedlichen Volterra-Kerne beliebiger Ordnung erläutert wurde, soll die Parameterreduktion auf die in den Kapiteln 4.1 und 4.2 abgeleiteten Volterra-Funktionalpotenzreihen angewendet werden. Ziel ist es, eine vektorielle Systembeschreibung zu finden, die letztendlich die Grundlage für die Identifikation des MISO-Systems darstellt. Aus Gründen der Anschaulichkeit soll dies an zwei Beispielen erfolgen, die in Abb. 4.17 dargestellt sind.

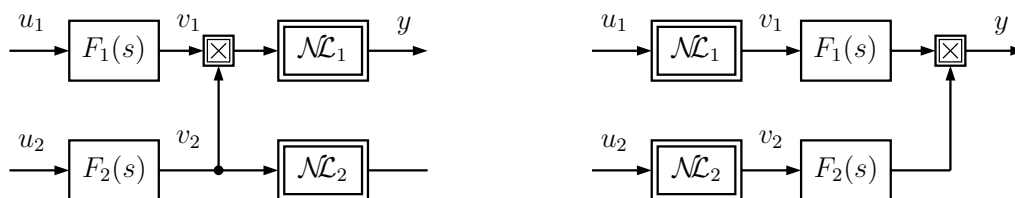


Abb. 4.17: Zwei nichtlinear verkoppelte Wiener-Modelle (links) und zwei nichtlinear verkoppelte Hammerstein-Modelle (rechts) als Beispiele zur Anwendung der Parameterreduktion

In Abb. 4.17 links sind zwei nichtlinear verkoppelte Wiener-Modelle und rechts zwei

⁷Die Ordnung dieser Kerne muss mindestens gleich drei sein.

nichtlinear verkoppelte Hammerstein-Modelle dargestellt. Der Grad der Nichtlinearitäten soll $q = 2$ sein. Für alle MISO-Systeme, die an dieser Stelle nicht gesondert betrachtet werden, gilt dieselbe Vorgehensweise.

Die Volterra-Funktionalpotenzreihe für das zu identifizierende System aus nichtlinear verkoppelten Wiener-Modellen ergibt sich nach Kapitel 4.2 zu:

$$\begin{aligned}
 y[k] = & g_0 + \sum_{i_1=1}^m \sum_{i_2=1}^m g[i_1, i_2] u_1[k - i_1] u_2[k - i_2] + \\
 & + \sum_{i_{11}=1}^m \sum_{i_{12}=i_{11}}^m \sum_{i_{21}=1}^m \sum_{i_{22}=i_{21}}^m g[i_{11}, i_{12}, i_{21}, i_{22}] u_1[k - i_{11}] u_1[k - i_{12}] u_2[k - i_{21}] u_2[k - i_{22}]
 \end{aligned} \tag{4.36}$$

In dieser Volterra-Funktionalpotenzreihe sind in der ersten Zeile der Beharrungswert und ein unsymmetrischer Volterra-Kern zweiter Ordnung aufgeführt. Dem Beharrungswert wird ein zu bestimmender Parameter θ_1 zugeordnet, dem unsymmetrischen Volterra-Kern zweiter Ordnung m_r^2 Parameter. In der zweiten Zeile befindet sich ein Volterra-Kern vierter Ordnung, der zwei symmetrische Anteile enthält. Bezüglich den Richtungen i_{11} und i_{12} bzw. i_{21} und i_{22} ist er symmetrisch, bezüglich den Richtungen i_{1j} und i_{2j} mit $j = 1, 2$ ist er unsymmetrisch. Für das zu identifizierende System lässt sich nach Anwendung der Parameterreduktion die folgende vektorielle Formulierung finden:

$$\begin{aligned}
 y[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\
 \underline{\mathcal{A}}_{dyn}^T[k] &= \left[1, \left(\underline{u}_1^T[k] \cdot \tilde{\mathbf{R}}^T \right) \otimes \left(\underline{u}_2^T[k] \cdot \tilde{\mathbf{R}}^T \right), \left(\underline{u}_1^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*2} \otimes \left(\underline{u}_2^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*2} \right] \\
 \underline{u}_i^T[k] &= \left[u_i[k - 1], \dots, u_i[k - m] \right] \quad \text{mit } i = 1, 2 \\
 \underline{\theta}^T &= \left[g_0, g_r[1, 1], \dots, g_r[m_r, m_r], g_r[1, 1, 1, 1], \dots, g_r[m_r, m_r, m_r, m_r] \right]
 \end{aligned} \tag{4.37}$$

Ohne die Einführung orthonormaler Basisfunktionen müssten $1 + m^2 + \binom{m+q-1}{q}^2$ Parameter identifiziert werden. Durch deren Einführung kann die Anzahl der zu bestimmenden Parameter auf $1 + m_r^2 + \binom{m_r+q-1}{q}^2$ reduziert werden. Es ist zu erkennen, dass die Systembeschreibung linear in den Parameter ist und dass der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}[k]$ ausschließlich Vergangenheitswerte der Eingangssignale u_1 und u_2 enthält. Der Vektor $\underline{\theta}$ stellt den optimalen Parametervektor dar, der durch Identifikation bestimmt werden soll.

Für das System aus nichtlinear verkoppelten Hammerstein-Modellen nach Abb. 4.17 rechts lässt sich die Volterra-Funktionalpotenzreihe wie folgt formulieren:

$$\begin{aligned}
y[k] = & g_0 + \sum_{i_1=1}^m g_1[i_1] u_1[k - i_1] + \sum_{i_2=1}^m g_2[i_2] u_2[k - i_2] + \\
& + \sum_{i_1=1}^m g_1[i_1, i_1] u_1^2[k - i_1] + \sum_{i_2=1}^m g_2[i_2, i_2] u_2^2[k - i_2] + \\
& + \sum_{i_1=1}^m \sum_{i_2=1}^m g_3[i_1, i_2] u_1[k - i_1] u_2[k - i_2] + \sum_{i_1=1}^m \sum_{i_2=1}^m g_4[i_1, i_2] u_1^2[k - i_1] u_2[k - i_2] + \\
& + \sum_{i_1=1}^m \sum_{i_2=1}^m g_5[i_1, i_2] u_1[k - i_1] u_2^2[k - i_2] + \sum_{i_1=1}^m \sum_{i_2=1}^m g_6[i_1, i_2] u_1^2[k - i_1] u_2^2[k - i_2]
\end{aligned}$$

Auf den ersten Blick ist nicht zu erkennen, dass für dieses System eine sehr einfache Vektordarstellung möglich ist. In der Systembeschreibung befinden sich zwei lineare Volterra–Kerne, zwei Volterra–Kerne zweiter Ordnung, die ausschließlich auf der Hauptdiagonale besetzt sind, und vier unsymmetrische Volterra–Kerne zweiter Ordnung. Für dieses System lässt sich nach Anwendung der Parameterreduktion die folgende vektorielle Formulierung finden:

$$\begin{aligned}
y[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\
\underline{\mathcal{A}}_{dyn}[k] &= [\underline{u}_{Ham,1}^T \otimes \underline{u}_{Ham,2}^T] \\
\underline{u}_{Ham,i}^T &= \left[1, \underline{u}_i^T \cdot \tilde{\mathbf{R}}^T, \underline{u}_i^{2T} \cdot \tilde{\mathbf{R}}^T \right] \quad \text{mit } i = 1, 2 \\
\underline{u}_i^T &= \left[u_i[k-1], \dots, u_i[k-m] \right] \\
\underline{\theta}^T &= \left[g_0, g_{r1}[1], \dots, g_{r2}[m_r], \dots, g_{r6}[m_r, m_r] \right]
\end{aligned} \tag{4.38}$$

In diesem Beispiel kann durch die Einführung orthonormaler Basisfunktionen die Anzahl der zu bestimmenden Parameter von $(2 \cdot m + 1)^2$ auf $(2 \cdot m_r + 1)^2$ reduziert werden. Außerdem ist in diesem Beispiel zu erkennen, dass die Systembeschreibung linear in den Parameter ist und dass der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}[k]$ ausschließlich Vergangenheitswerte der Eingangssignale u_1 und u_2 enthält. Der Vektor $\underline{\theta}$ stellt wiederum den optimalen Parametervektor dar, der durch Identifikation bestimmt werden soll.

Abschließend sei an dieser Stelle noch einmal erwähnt, dass die beispielhaft dargelegte Vorgehensweise generell zur Identifikation nichtlinearer dynamischer MISO–Systeme angewendet werden kann. Aus der Beschreibung des zu identifizierenden Systems wird die Volterra–Funktionalpotenzreihe abgeleitet. Diese ist die Grundlage für die Einführung orthonormaler Basisfunktionen zum Zwecke der Parameterreduktion. Durch Anwendung des $*$ –Operators und des \otimes –Operators kann das System in eine kompakte Vektorschreibweise überführt werden, die die Grundlage der Identifikation darstellt. Alle MISO–Systeme lassen sich in der Form

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k]$$

darstellen. Die Systembeschreibungen sind linear in den Parametern und der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}$ enthält ausschließlich Vergangenheitswerte der Eingangssignale, so dass das System in einer Ausgangsfehleranordnung identifiziert werden kann. Das bedeutet, dass alle Vorteile von NOBF-Modellen mit einem Eingang und einem Ausgang auf NOBF-Modelle mit mehreren Eingängen und einem Ausgang übertragen werden können. Selbstverständlich nimmt die Anzahl der unbekannt Parameter im Falle von MISO-Systemen verglichen mit SISO-Systemen zu. Das Prinzip der Parameterreduktion mittels orthonormaler Basisfunktionen kann jedoch auch im MISO-Falle angewendet werden, so dass die Parameteranzahl in einem akzeptablen Bereich bleibt.

Angesichts der Tatsache, dass das entwickelte Identifikationsverfahren für MISO-Systeme in Kapitel 6 in einen Beobachter integriert werden soll, ist die Stabilität des Identifikationsverfahrens von entscheidender Bedeutung. Aus diesem Grund wird im folgenden Kapitel ein Stabilitätsbeweis nach Lyapunov geführt.

4.3.2 Stabilitätsbeweis nach Lyapunov

In diesem Abschnitt wird bewiesen, dass das vorgestellte Verfahren zur Identifikation nichtlinearer dynamischer MISO-Systeme auf Grundlage der Volterra-Funktionalpotenzreihe globales Konvergenzverhalten aufweist [25]. Wird ein Stabilitätsbeweis nach Lyapunov [63] geführt, gilt es zu zeigen, dass die zeitliche Ableitung der Lyapunov-Funktion $V(t)$ kleiner oder gleich Null ist, d.h. $\dot{V}(t) \leq 0$. Zusätzlich muss der Ausgangsfehler $e(t)$ zwischen dem zu identifizierenden System und dem Modell zu Null werden, d.h. $\lim_{t \rightarrow \infty} e(t) = 0$. Die daraus resultierende zeitliche Ableitung des Parameterfehlers $\underline{\Phi}(t) = \hat{\underline{\theta}}(t) - \underline{\theta}$ muss ebenfalls zu Null werden, d.h. $\lim_{t \rightarrow \infty} \dot{\underline{\Phi}}(t) = 0$. Das in den vorangegangenen Kapiteln abgeleitete Identifikationsverfahren basiert durchgehend auf zeitdiskreten Berechnungen. Aus diesem Grund müssen die Stabilitätsbedingungen auch in den zeitdiskreten Bereich überführt werden, d.h. es gilt zu beweisen, dass:

$$\begin{aligned} \Delta V[k] &\leq 0 \\ \lim_{k \rightarrow \infty} e_{\text{post}}[k] &= 0 \\ \lim_{k \rightarrow \infty} \Delta \underline{\Phi}[k] &= 0 \end{aligned} \tag{4.39}$$

Die betrachteten nichtlinearen dynamischen MISO-Systeme können generell in der Form

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k]$$

dargestellt werden⁸. Charakteristisch für diese Systembeschreibung ist, dass sie linear in den Parametern ist und der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}$ ausschließlich

⁸Der folgende Stabilitätsbeweis ist generell auf alle Systeme übertragbar, die linear in den Parametern sind.

Vergangenheitswerte der Eingangssignale enthält. In [38] werden diese Eigenschaften als Fehlermodell 1 bezeichnet. Der Vektor $\underline{\theta}$ enthält die optimalen Parameter, die durch Identifikation ermittelt werden sollen. Der Modellausgang $\hat{y}[k]$ kann analog zur Systembeschreibung wie folgt formuliert werden⁹:

$$\hat{y}[k] = \hat{\underline{\theta}}^T [k] \cdot \underline{\mathcal{A}}_{dyn} [k]$$

Der Vektor $\hat{\underline{\theta}}[k]$ enthält die identifizierten Parameter zum Zeitpunkt k . Aus den vorangegangenen Gleichungen kann die Abweichung zwischen dem Systemausgang $y[k]$ und dem Modellausgang $\hat{y}[k]$ für jeden Zeitpunkt k berechnet werden durch:

$$e_{\text{post}}[k] = \hat{y}[k] - y[k] = \hat{\underline{\theta}}^T [k] \cdot \underline{\mathcal{A}}_{dyn} [k] - \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn} [k] = \underline{\Phi}^T [k] \cdot \underline{\mathcal{A}}_{dyn} [k] \quad (4.40)$$

In Gl. (4.40) beschreibt der Ausgangsfehler $e_{\text{post}}[k]$ die Abweichung zwischen dem Systemausgang und dem Modellausgang zum momentanen Zeitpunkt k . Der Vektor $\underline{\Phi}[k] = \hat{\underline{\theta}}[k] - \underline{\theta}$ wird als Parameterfehlervektor bezeichnet und enthält für jeden Zeitpunkt k die Abweichungen zwischen den optimalen Parametern und den zu identifizierten Parametern.

Der folgende Stabilitätsbeweis nach Lyapunov soll anhand des rekursiven Least-Squares-Algorithmus durchgeführt werden, der bereits in Kapitel 2.7.1 vorgestellt wurde. Bei Anwendung des rekursiven Least-Squares-Algorithmus besteht das Problem, dass die identifizierten Parameterwerte $\hat{\underline{\theta}}$ erst zum darauf folgenden Zeitschritt verwendet werden können. Im Gegensatz zum A-posteriori-Ausgangsfehler $e_{\text{post}}[k]$ wird beim rekursiven Least-Squares-Algorithmus der A-priori-Ausgangsfehler $e[k]$ verwendet, d.h.:

$$e[k] = \hat{\underline{\theta}}^T [k-1] \cdot \underline{\mathcal{A}}_{dyn} [k] - \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn} [k] = \underline{\Phi}^T [k-1] \cdot \underline{\mathcal{A}}_{dyn} [k] \quad (4.41)$$

Zur Durchführung des Stabilitätsbeweises nach Lyapunov werden noch weitere Gleichungen benötigt, die aus der Herleitung des rekursiven Least-Squares-Algorithmus hervorgehen. Die folgenden Voraussetzungen gelten nach [25] für die Kovarianzmatrix $\mathbf{P}[k]$ und deren Inverse $\mathbf{P}^{-1}[k]$.

1. Die Matrizen $\mathbf{P}[k]$ und $\mathbf{P}^{-1}[k]$ sind für jeden Zeitpunkt k positiv definit, d.h. es gelten nach [51] die mathematischen Zusammenhänge:

$$\begin{aligned} \det \mathbf{P}[k] > 0 &\implies \mathbf{P}[k] > 0 && \forall k \\ \det \mathbf{P}^{-1}[k] > 0 &\implies \mathbf{P}^{-1}[k] > 0 && \forall k \end{aligned} \quad (4.42)$$

2. Die Inverse $\mathbf{P}^{-1}[k]$ der Matrix $\mathbf{P}[k]$ existiert für alle Zeitpunkte k und ist symmetrisch, d.h. es gelten nach [65] die mathematischen Zusammenhänge zwischen den Matrizen:

$$\mathbf{P}[k] = \mathbf{P}^T [k] \quad \forall k \quad \mathbf{P}[k] \mathbf{P}^{-1}[k] = \mathbf{E} \quad \forall k \quad (4.43)$$

⁹Der inhärente Approximationsfehler, der sich aus der begrenzten Parameteranzahl ergibt, wird an dieser Stelle vernachlässigt.

3. Für die Aktualisierung der invertierten Kovarianzmatrix $\mathbf{P}^{-1}[k]$, sowie durch Anwendung des Matrizeninversionslemmas [51] für die Kovarianzmatrix $\mathbf{P}[k]$, gilt nach [25] beim rekursiven Least-Squares-Algorithmus:

$$\begin{aligned} \mathbf{P}^{-1}[k] &= \mathbf{P}^{-1}[k-1] + \underline{\mathcal{A}}_{dyn}[k] \underline{\mathcal{A}}_{dyn}^T[k] \\ \mathbf{P}[k] \underline{\mathcal{A}}_{dyn}[k] &= \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} \end{aligned} \quad (4.44)$$

Für die Aktualisierung der Parameterwerte $\hat{\underline{\theta}}[k]$ gilt die Rekursionsgleichung aus Kapitel 2.7.1:

$$\hat{\underline{\theta}}[k] = \hat{\underline{\theta}}[k-1] - \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] e[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} \quad (4.45)$$

Wird Gl. (4.45) auf beiden Seiten mit $-\underline{\theta}$ erweitert, ergibt sich die Gleichung für die Aktualisierung des Parameterfehlers $\underline{\Phi}[k]$. Durch Einsetzen des Matrixinversionslemmas aus Gl. (4.44) kann die Gleichung weiter umgeformt werden zu:

$$\begin{aligned} \hat{\underline{\theta}}[k] - \underline{\theta} &= \hat{\underline{\theta}}[k-1] - \underline{\theta} - \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] e[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} \\ \implies \underline{\Phi}[k] &= \underline{\Phi}[k-1] - \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] e[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} = \\ &= \underline{\Phi}[k-1] - \mathbf{P}[k] \underline{\mathcal{A}}_{dyn}[k] e[k] \end{aligned} \quad (4.46)$$

Zwischen dem A-posteriori-Ausgangsfehler $e_{\text{post}}[k]$ nach Gl. (4.40) und dem A-priori-Ausgangsfehler $e[k]$ nach Gl. (4.41), gilt folgender Zusammenhang:

$$\begin{aligned} e_{\text{post}}[k] &= e[k] + \left(\hat{\underline{\theta}}^T[k] - \underline{\theta}^T \right) \underline{\mathcal{A}}_{dyn}[k] - \left(\hat{\underline{\theta}}^T[k-1] - \underline{\theta}^T \right) \underline{\mathcal{A}}_{dyn}[k] = \\ &= e[k] + \left(\hat{\underline{\theta}}^T[k] - \hat{\underline{\theta}}^T[k-1] \right) \underline{\mathcal{A}}_{dyn}[k] \end{aligned} \quad (4.47)$$

Durch Einsetzen von Gl. (4.45) zur Aktualisierung der Parameterwerte $\hat{\underline{\theta}}[k]$ in Gl. (4.47) folgt:

$$\begin{aligned} e_{\text{post}}[k] &= e[k] - \underline{\mathcal{A}}_{dyn}^T[k] \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] e[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} = \\ &= e[k] \frac{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] - \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} = \\ &= e[k] \frac{1}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} \end{aligned} \quad (4.48)$$

Wird Gl. (4.48) in Gl. (4.46) eingesetzt und anschließend Gl. (4.44) angewendet, so ergibt dies eine Vorschrift für die Aktualisierung des Parameterfehlers $\underline{\Phi}[k]$ in

Abhängigkeit vom A-posteriori-Fehler $e_{\text{post}}[k]$ der Form:

$$\begin{aligned}\underline{\Phi}[k] &= \underline{\Phi}[k-1] - \mathbf{P}[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e_{\text{post}}[k] (1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]) = \\ &= \underline{\Phi}[k-1] - \frac{\mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k] e_{\text{post}}[k]}{1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]} (1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]) = \\ &= \underline{\Phi}[k-1] - \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k] e_{\text{post}}[k]\end{aligned}\quad (4.49)$$

Als Lyapunov-Funktion für den Beweis globalen Konvergenzverhaltens im deterministischen Fall¹⁰ wird die folgende Funktion eingeführt:

$$V[k] = \underline{\Phi}^T[k] \mathbf{P}^{-1}[k] \underline{\Phi}[k] \quad (4.50)$$

Mit den Voraussetzungen für die Matrizen $\mathbf{P}[k]$ und $\mathbf{P}^{-1}[k]$ aus den Gleichungen (4.42) bis (4.44) und durch Einsetzen von Gl. (4.46) folgt:

$$\begin{aligned}V[k] &= (\underline{\Phi}[k-1] - \mathbf{P}[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e[k])^T \mathbf{P}^{-1}[k] (\underline{\Phi}[k-1] - \mathbf{P}[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e[k]) = \\ &= (\underline{\Phi}^T[k-1] - \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}^T[k] e[k]) (\mathbf{P}^{-1}[k] \underline{\Phi}[k-1] - \mathbf{P}^{-1}[k] \mathbf{P}[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e[k]) = \\ &= \underline{\Phi}^T[k-1] \mathbf{P}^{-1}[k] \underline{\Phi}[k-1] - 2 \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] e[k] + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}^T[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e^2[k]\end{aligned}\quad (4.51)$$

Mit Gl. (4.44) zur Aktualisierung der invertierten Kovarianzmatrix $\mathbf{P}^{-1}[k]$ lässt sich die Lyapunov-Funktion weiter umformen zu:

$$\begin{aligned}V[k] &= \underline{\Phi}^T[k-1] (\mathbf{P}^{-1}[k-1] + \underline{\mathcal{A}}_{\text{dyn}}[k] \underline{\mathcal{A}}_{\text{dyn}}^T[k]) \underline{\Phi}[k-1] - \\ &\quad - 2 \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] e[k] + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}^T[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e^2[k] = \\ &= \underline{\Phi}^T[k-1] \mathbf{P}^{-1}[k-1] \underline{\Phi}[k-1] + \underline{\Phi}^T[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k] \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] - \\ &\quad - 2 \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] e[k] + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}^T[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e^2[k] = \\ &= V[k-1] + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] (\underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1] - 2e[k]) + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}^T[k] \underline{\mathcal{A}}_{\text{dyn}}[k] e^2[k]\end{aligned}\quad (4.52)$$

Durch das Ersetzen des Ausgangsfehlers $e[k] = \underline{\mathcal{A}}_{\text{dyn}}^T[k] \underline{\Phi}[k-1]$ nach Gl. (4.41) und das Anwenden von Gl. (4.44) kann die Lyapunov-Funktion weiter vereinfacht werden zu:

$$\begin{aligned}V[k] &= V[k-1] + e[k] (e[k] - 2e[k]) + e^2[k] - \frac{e^2[k]}{1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]} \\ &= V[k-1] + e^2[k] - 2e^2[k] + e^2[k] - \frac{e^2[k]}{1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]} \\ &= V[k-1] - \frac{e^2[k]}{1 + \underline{\mathcal{A}}_{\text{dyn}}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{\text{dyn}}[k]}\end{aligned}$$

¹⁰Deterministischer Fall bedeutet ohne Einwirkung von Störungen.

Die Lösung dieser mathematischen Untersuchungen lässt sich wie folgt zusammenfassen:

$$\implies \Delta V[k] = V[k] - V[k-1] = -\frac{e^2[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} \quad (4.53)$$

Es ist zu erkennen, dass die zeitliche Differenz $\Delta V[k]$ stets negativ ist. Der Ausgangsfehler steht zur zweiten Potenz im Zähler des Bruchs, d.h. der Zähler ist immer positiv. In Gl. (4.42) wurde vorausgesetzt, dass die Kovarianzmatrix $\mathbf{P}[k]$ zu jedem Zeitpunkt positiv definit ist. Wird der Vektor $\underline{\mathcal{A}}_{dyn}[k]$ mit der Kovarianzmatrix verrechnet, so ergibt auch diese Berechnung ein stets positives Ergebnis, d.h. $\underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k] > 0$. Der Nenner des obigen Bruchs ist also auch stets positiv. Das Vorzeichen des dargestellten Bruchs ist negativ und somit auch die zeitliche Differenz der Lyapunov-Funktion für jeden Zeitpunkt k .

Wird die Differenz der Lyapunov-Funktion zum Zeitschritt k und $k=0$ gebildet, folgt:

$$V[k] - V[0] = \sum_{i=0}^k -\frac{e^2[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]}$$

Da die einzelnen Summanden alle negativ sind und die Stabilität nach Lyapunov voraussetzt, dass $V[0]$ begrenzt sein muss, gilt für den Zeitschritt k :

$$V[k] - V[0] \leq 0 \quad \implies V[k] \leq V[0] \forall k$$

Für k gegen ∞ bedeutet dies, dass auch die Summe über die einzelnen Summanden begrenzt sein muss, d.h. für die einzelnen Summanden und dem daraus resultierenden Ausgangsfehler muss gelten:

$$\lim_{k \rightarrow \infty} -\frac{e^2[k]}{1 + \underline{\mathcal{A}}_{dyn}^T[k] \mathbf{P}[k-1] \underline{\mathcal{A}}_{dyn}[k]} = 0 \quad \implies \lim_{k \rightarrow \infty} e[k] = 0 \quad (4.54)$$

Nach Gl. 4.48 folgt daraus unmittelbar für den A-posteriori-Fehler $e_{\text{post}}[k]$:

$$\implies \lim_{k \rightarrow \infty} e_{\text{post}}[k] = 0 \quad (4.55)$$

Mit Gl. (4.46) kann in gleicher Art und Weise gezeigt werden, dass auch die zeitliche Differenz des Parameterfehlers zu Null wird, da dieser in direktem Zusammenhang mit dem Ausgangsfehler steht:

$$\lim_{k \rightarrow \infty} \Delta \underline{\Phi}[k] = \lim_{k \rightarrow \infty} -\mathbf{P}[k] \underline{\mathcal{A}}_{dyn}[k] e[k] = 0 \quad (4.56)$$

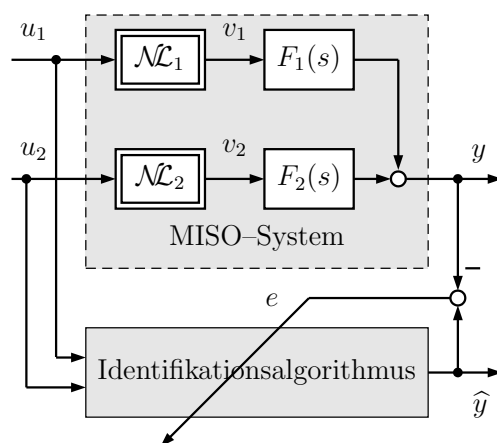
Mit Gl. (4.56) ist die Führung des Stabilitätsbeweises nach Lyapunov für die Identifikation nichtlinearer dynamischer MISO-Systeme abgeschlossen. Der rekursive Least-Squares-Algorithmus besitzt in Verbindung mit Fehlermodell 1 globales Konvergenzverhalten, d.h. die identifizierten Parameterwerte $\hat{\underline{\theta}}$, die Änderung des Parameterfehlers $\Delta \underline{\Phi}[k]$ und der A-priori-Fehler $e[k]$ sowie der A-posteriori-Fehler $e_{\text{post}}[k]$ sind für jeden Zeitschritt k begrenzt.

4.3.3 Identifikationsbeispiele

Die Theorie zur Identifikation nichtlinearer dynamischer MISO-Systeme auf Grundlage der Volterra-Funktionalpotenzreihe wird in diesem Kapitel an verschiedenen Identifikationsbeispielen veranschaulicht.

4.3.3.1 Beispiel 1 — Linear verkoppelte Hammerstein-Modelle

Als erstes Beispiel wird die Identifikation von zwei linear verkoppelten Hammerstein-Modellen betrachtet. Zunächst erfolgt die Identifikation ohne Parameterreduktion mittels orthonormaler Basisfunktionen und anschließend mit Parameterreduktion. Ein Vergleich soll die Vorteile der Einführung von Basisfunktionen verdeutlichen. Das betrachtete System einschließlich Identifikationsstruktur ist in Abb. 4.18 dargestellt.



Das System ist wie folgt gegeben:

$$\mathcal{NL}_1(u_1) = -0.5 \cdot u_1 + 1.8 \cdot u_1^2$$

$$\mathcal{NL}_2(u_2) = -1.0 \cdot u_2 + 1.2 \cdot u_2^2$$

$$F_1(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1}$$

$$F_2(s) = \frac{1}{32 \cdot s^2 + 8 \cdot s + 1}$$

Die Zeitkonstanten von $F_1(s)$ und $F_2(s)$ sind auf einen Simulationszeitschritt von $1s$ normiert.

Abb. 4.18: Identifikationsbeispiel 1

Die beiden Nichtlinearitäten sind zweiten Grades und enthalten keinen Gleichanteil. Die Übertragungsfunktion $F_1(s)$ besitzt zwei reelle Pole, während die Übertragungsfunktion $F_2(s)$ ein konjugiert komplexes Polpaar besitzt. Das betrachtete System soll zunächst ohne Parameterreduktion mittels orthonormaler Basisfunktionen identifiziert werden. Aus Voruntersuchungen wurden für die Identifikation die Einstellwerte nach Tabelle 4.1 ermittelt.

Die Simulationsdauer beträgt insgesamt 600 Zeitschritte. Nach 500 Zeitschritten wird die Parameteradaption eingestellt und das identifizierte Modell wird als reines Parallelmodell betrieben. Das System wird durch zwei voneinander unabhängige Zufallssignale im Bereich $[-1; 1]$ angeregt. Als Adaptionsverfahren kommt der rekursive Least-Squares-Algorithmus zum Einsatz. In Abb. 4.19 ist der Verlauf der Identifikation und das Konvergenzverhalten der Parameter dargestellt.

Einstellwerte	Erläuterung
$q = 2$	Grad der Nichtlinearität
$m = 50$	Antwortlänge
$r = 2$	Anzahl an Eingangssignalen
$p = r \cdot m \cdot q + 1 = 201$	Parameteranzahl
$h = 1s$	Abtastzeit

Tabelle 4.1: Einstellwerte für Identifikationsbeispiel 1

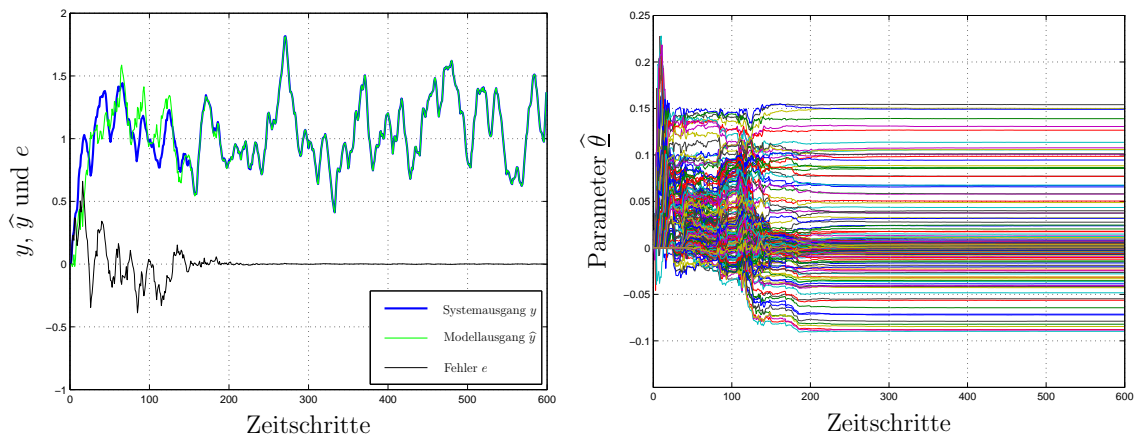


Abb. 4.19: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) ohne Einführung orthonormaler Basisfunktionen

Es ist zu erkennen, dass der Fehler e gegen Null geht und die Parameter $\hat{\theta}_i$ konvergieren. Die Anzahl der Zeitschritte entspricht ungefähr der Anzahl unbekannter Parameter. Auch nach 500 Zeitschritten wird der Fehler e im Parallellauf nicht größer. In Abb. 4.20 sind die vorgegebenen und identifizierten statischen Nichtlinearitäten \mathcal{NL}_1 und \mathcal{NL}_2 dargestellt.

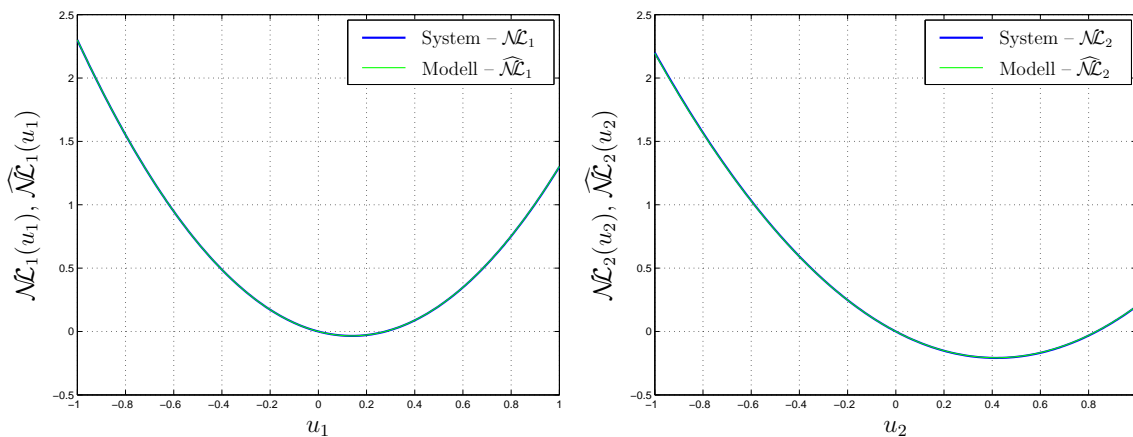


Abb. 4.20: Vorgegebene und identifizierte Nichtlinearitäten \mathcal{NL}_1 (links) und \mathcal{NL}_2 (rechts) ohne Einführung orthonormaler Basisfunktionen

Die identifizierten Nichtlinearitäten¹¹ kommen exakt auf den vorgegebenen Nichtlinearitäten zu liegen. In den Abbildungen 4.21 und 4.22 sind die Volterra-Kerne erster und zweiter Ordnung veranschaulicht.

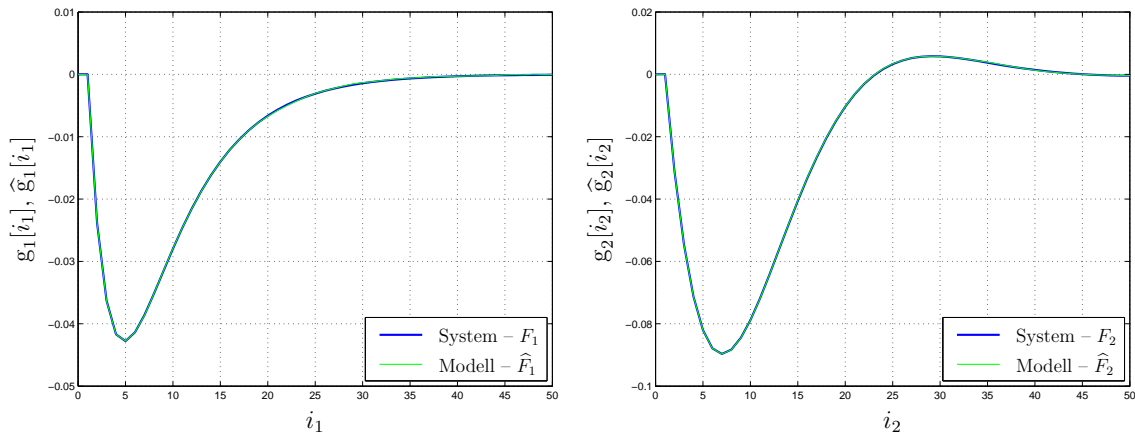


Abb. 4.21: Lineare Volterra-Kerne 1 (links) und 2 (rechts) ohne Einführung orthonormaler Basisfunktionen

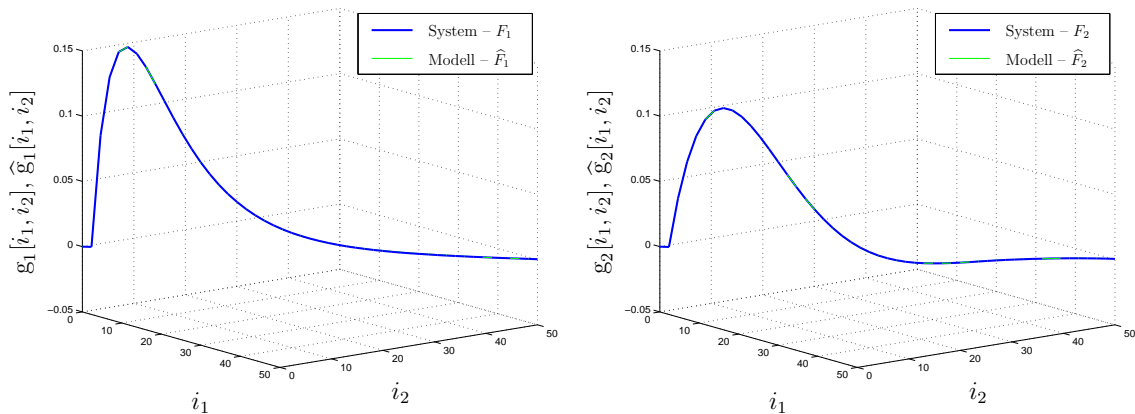


Abb. 4.22: Quadratische Volterra-Kerne 1 (links) und 2 (rechts) ohne Einführung orthonormaler Basisfunktionen

Die identifizierten Volterra-Kerne entsprechen exakt der Vorgabe. Die Kerne haben analog zu den Impulsantworten von $F_1(s)$ und $F_2(s)$ unterschiedliche Form. Die Volterra-Kerne zweiter Ordnung sind ausschließlich auf der Hauptdiagonale besetzt, was für linear verkoppelte Hammerstein-Modelle charakteristisch ist.

Die Anzahl der unbekannt Parameter ist ohne die Einführung orthonormaler Basisfunktion sehr hoch. Im Folgenden soll eine Parameterreduktion mit $m_r = 6$ und $m_r = 20$ Basisfunktionen erfolgen. Es werden die Einstellwerte nach Tabelle 4.2 verwendet.

¹¹Die identifizierten Nichtlinearitäten wurden aus dem Identifikationsergebnis zurückgerechnet. Die Rückrechnung des Identifikationsergebnisses wird in Kapitel 4.4 genauer behandelt.

Einstellwerte	Erläuterung
$q = 2$	Grad der Nichtlinearität
$m = 50$	Antwortlänge
$r = 2$	Anzahl an Eingangssignalen
$m_r = 6$ bzw. 20	Basisfunktionenanzahl
$\zeta = 11.65$	Formfaktor
$p_6 = r \cdot m_r \cdot q + 1 = 25$	Parameteranzahl bei $m_r = 6$
$p_{20} = r \cdot m_r \cdot q + 1 = 81$	Parameteranzahl bei $m_r = 20$
$h = 1s$	Abtastzeit

Tabelle 4.2: Einstellwerte für Identifikationsbeispiel 1 bei Einführung orthonormaler Basisfunktionen

Zunächst soll der Fall mit $m_r = 6$ Basisfunktionen betrachtet werden, wie es in den Einstellregeln nach Tabelle 3.2 empfohlen wird. Bezüglich der Anregung und des Parallellaufs gelten gleiche Bedingungen wie vorher. In Abb. 4.23 sind wiederum der Identifikationsverlauf und die Konvergenz der Parameter bei Einführung von 6 orthonormalen Basisfunktionen dargestellt.

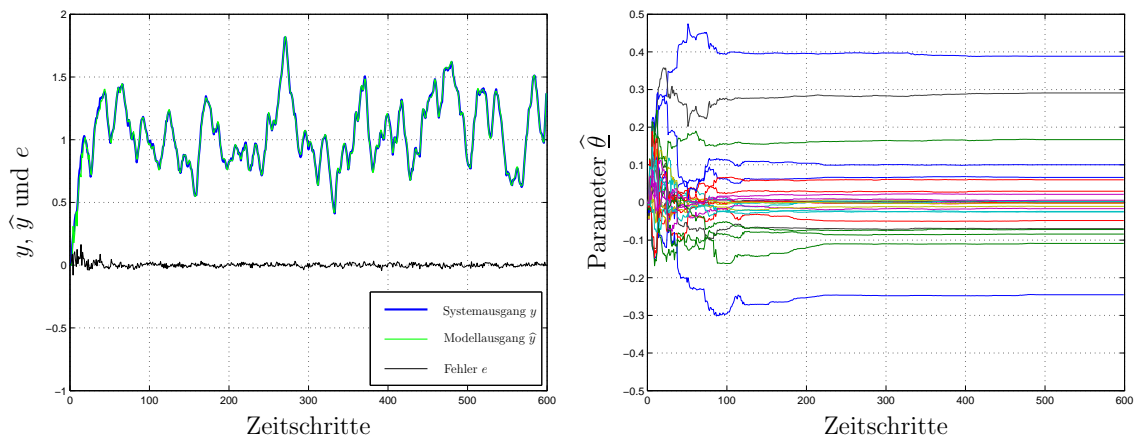


Abb. 4.23: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) bei Einführung von 6 Basisfunktionen

Es ist zu erkennen, dass ein gewisser Restfehler bleibt, der aus der Parameterreduktion mittels Basisfunktionen resultiert. Dieser Fehler wird im Parallellauf ab 500 Zeitschritten nicht größer. Die Parameteranzahl wurde mit $p = 25$ drastisch reduziert. In Abb. 4.24 sind die vorgegebenen und identifizierten statischen Nichtlinearitäten \mathcal{NL}_1 und \mathcal{NL}_2 dargestellt.

Trotz des Restfehlers bei der Identifikation werden die statischen Nichtlinearitäten sehr gut identifiziert. In den Abbildungen 4.25 und 4.26 sind die rekonstruierten¹² Volterra-Kerne erster und zweiter Ordnung veranschaulicht.

¹²Die Rekonstruktion der Volterra-Kerne wird in Kapitel 4.4 genauer erläutert.

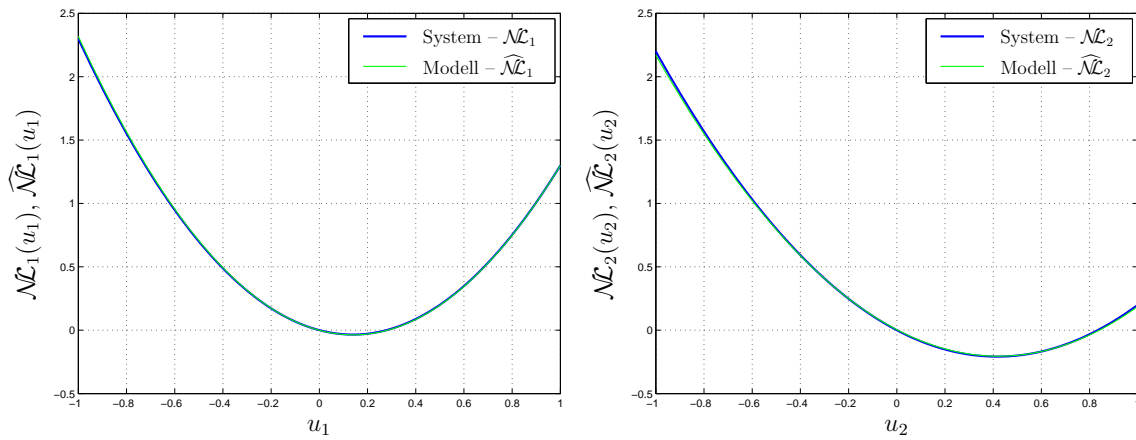


Abb. 4.24: Vorgegebene und identifizierte Nichtlinearitäten $\mathcal{N}_{\mathcal{L}_1}$ (links) und $\mathcal{N}_{\mathcal{L}_2}$ (rechts) bei Einführung von 6 Basisfunktionen

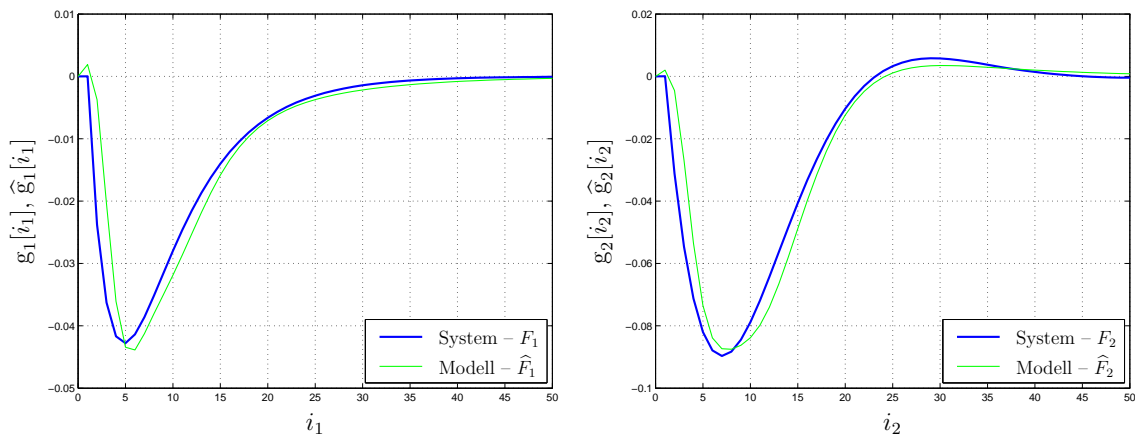


Abb. 4.25: Lineare Volterra-Kerne 1 (links) und 2 (rechts) bei Einführung von 6 Basisfunktionen

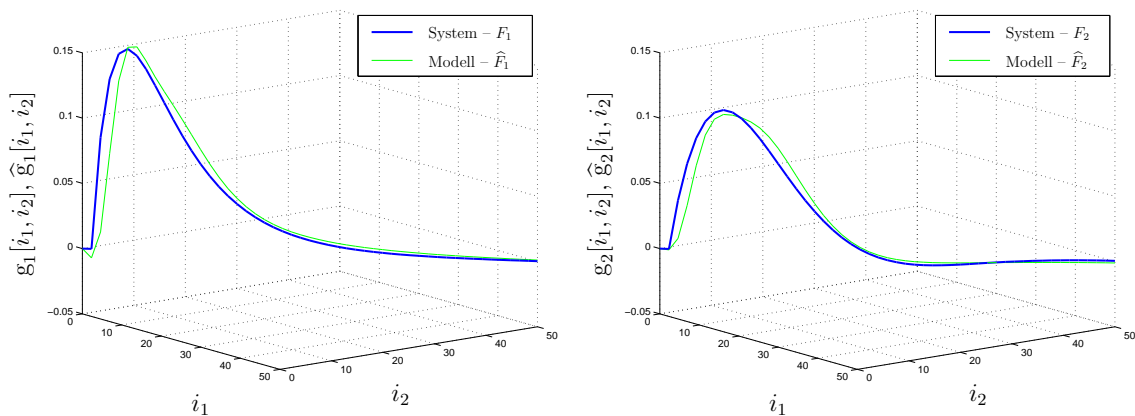


Abb. 4.26: Quadratische Volterra-Kerne 1 (links) und 2 (rechts) bei Einführung von 6 Basisfunktionen

Es ist zu erkennen, dass die Volterra–Kerne durch die geringe Anzahl an Basisfunktionen nicht exakt nachgebildet werden können, woraus auch der Restfehler bei der Identifikation resultiert. Die Anzahl an Basisfunktionen ist jedoch ein Einstellparameter mit dem ein Kompromiss zwischen Genauigkeit und Parameteranzahl getroffen werden muss. Aus diesem Grund wird das System noch einmal mit $m_r = 20$ orthonormalen Basisfunktionen identifiziert. In Abb. 4.27 sind wiederum der Identifikationsverlauf und die Konvergenz der Parameter dargestellt.

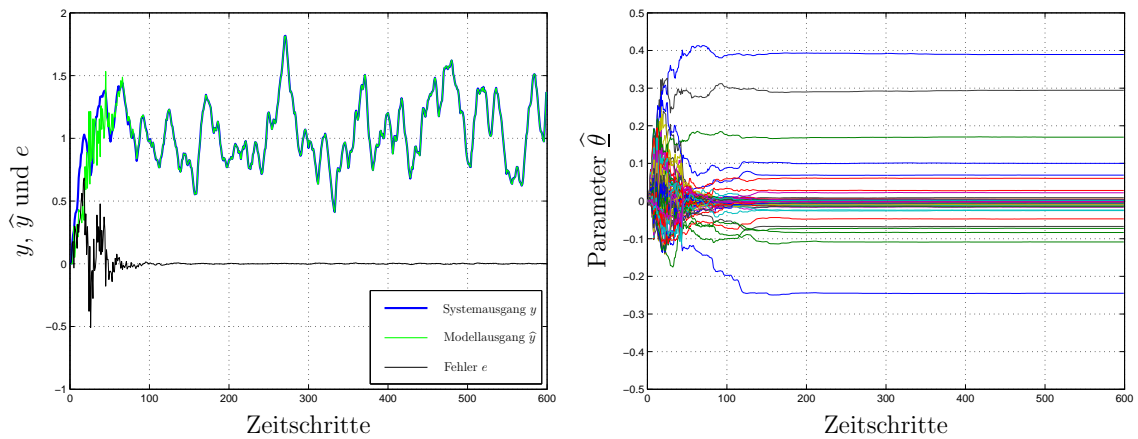


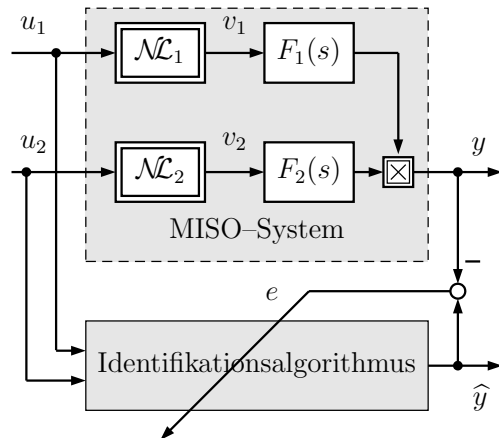
Abb. 4.27: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) bei Einführung von 20 Basisfunktionen

Der Fehler e geht mit 20 Basisfunktionen deutlich gegen Null. Es ist kein Unterschied zu Abb. 4.19 links zu erkennen, wo die Identifikation ohne Parameterreduktion erfolgte. Die Anzahl der Parameter ist mit $p = 81$ um mehr als die Hälfte reduziert worden. Auf die Darstellung der statischen Nichtlinearitäten und der Volterra–Kerne wird an dieser Stelle verzichtet.

4.3.3.2 Beispiel 2 — Nichtlinear verkoppelte Hammerstein–Modelle

Als zweites Beispiel wird die Identifikation von zwei nichtlinear verkoppelten Hammerstein–Modellen betrachtet. Um die Identifikation eines unsymmetrischen Volterra–Kerns genauer zu veranschaulichen, wird zunächst der Spezialfall betrachtet, dass das System aus zwei ausgangsseitig multiplikativ verkoppelten Übertragungsfunktionen besteht. Anschließend erfolgt die Identifikation des allgemeineren Falls von nichtlinear verkoppelten Hammerstein–Modellen.

Das betrachtete System einschließlich Identifikationsstruktur ist in Abb. 4.28 dargestellt.



Zunächst soll für das System gelten:

$$\mathcal{NL}_1(u_1) = u_1$$

$$\mathcal{NL}_2(u_2) = u_2$$

$$F_1(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1}$$

$$F_2(s) = \frac{1}{32 \cdot s^2 + 8 \cdot s + 1}$$

Die Zeitkonstanten von $F_1(s)$ und $F_2(s)$ sind wiederum auf einen Simulationszeitschritt von 1s normiert.

Abb. 4.28: Identifikationsbeispiel 2

Das System besteht aus zwei ausgangseitig nichtlinear verkoppelten Übertragungsfunktionen. Die Übertragungsfunktion $F_1(s)$ besitzt zwei reelle Pole, während die Übertragungsfunktion $F_2(s)$ ein konjugiert komplexes Polpaar besitzt. Das betrachtete System besitzt als einzigen Kern einen unsymmetrischen Volterra-Kern zweiter Ordnung. Das System kann bei Einführung orthonormaler Basisfunktionen wie folgt beschrieben werden:

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k]$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left(\underline{u}_1^T[k] \cdot \tilde{\mathbf{R}}^T \right) \circledast \left(\underline{u}_2^T[k] \cdot \tilde{\mathbf{R}}^T \right)$$

$$\underline{u}_i^T[k] = \left[u_i[k-1], u_i[k-2], \dots, u_i[k-m] \right] \quad \text{mit } i = 1, 2$$

$$\underline{\theta}^T = \left[g_r[1, 1], \dots, g_r[m_r, m_r] \right]$$

Für die Identifikation werden die Einstellwerte nach Tabelle 4.3 verwendet.

Einstellwerte	Erläuterung
$m = 60$	Antwortlänge
$r = 2$	Anzahl an Eingangssignalen
$m_r = 10$ bzw. 20	Basisfunktionenanzahl
$\zeta = 9.8$	Formfaktor
$p_{10} = m_r^r = 100$	Parameteranzahl bei $m_r = 10$
$p_{20} = m_r^r = 400$	Parameteranzahl bei $m_r = 20$
$h = 1s$	Abtastzeit

Tabelle 4.3: Einstellwerte für Identifikationsbeispiel 2 — Nichtlinear verkoppelte Übertragungsfunktionen

Ohne die Einführung orthonormaler Basisfunktionen müssten $p = m^r = 3600$ Parameterwerte bestimmt werden. Durch die Einführung der Basisfunktionen kann

die Anzahl der zu identifizierenden Parameter auf 100 bzw. 400 reduziert werden. Die Simulationsdauer beträgt insgesamt 1200 Zeitschritte. Nach 1000 Zeitschritten wird die Parameteradaption eingestellt und das identifizierte Modell wird als reines Parallelmodell betrieben. Das System wird durch zwei voneinander unabhängige Zufallssignale im Bereich $[-1; 1]$ angeregt. Als Adaptionsverfahren kommt wiederum der rekursive Least-Squares-Algorithmus zum Einsatz. In Abb. 4.29 bzw. Abb. 4.30 ist der Verlauf der Identifikation und das Konvergenzverhalten der Parameter für 10 bzw. 20 orthonormale Basisfunktionen dargestellt.

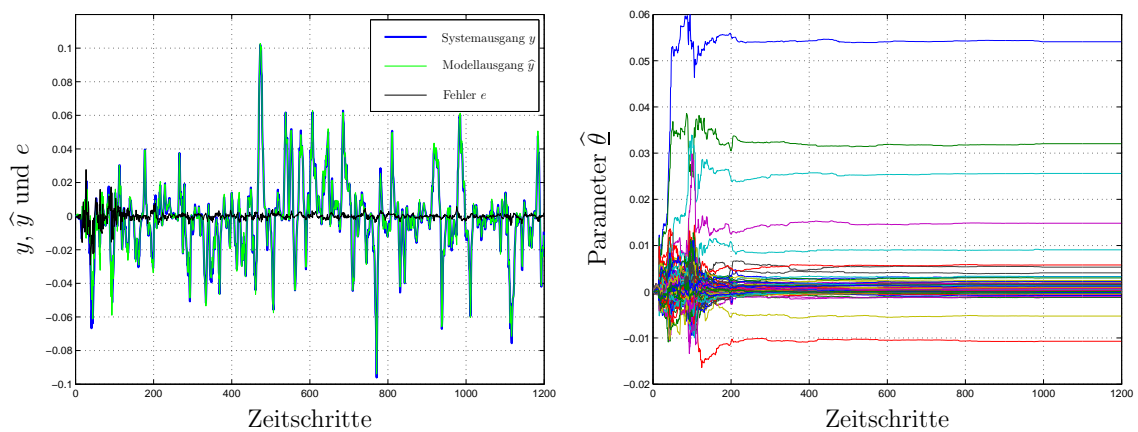


Abb. 4.29: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) bei Einführung von 10 Basisfunktionen

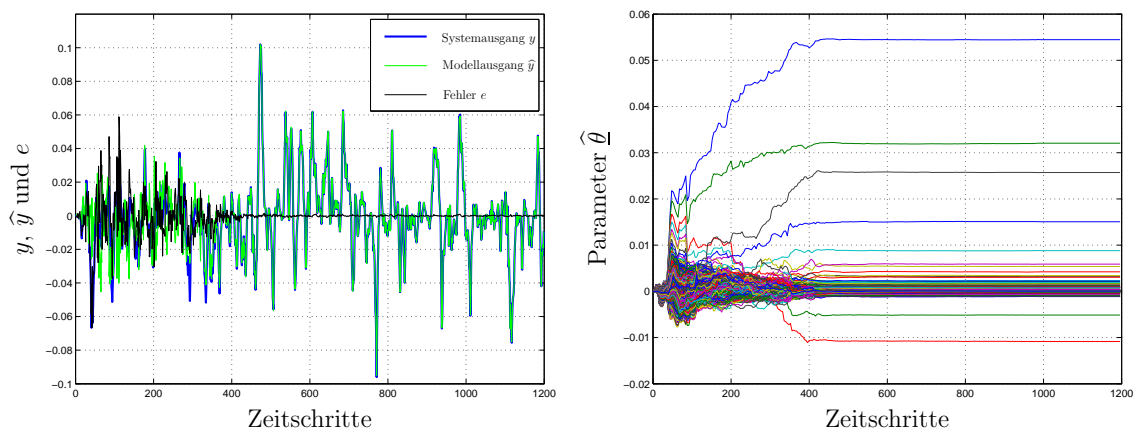


Abb. 4.30: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) bei Einführung von 20 Basisfunktionen

In Abb. 4.29 links ist zu erkennen, dass die starke Parameterreduktion zu einem deutlichen Restfehler führt. Aufgrund dieses Restfehlers konvergieren die Parameter nicht exakt (vgl. Abb. 4.29 rechts). Eine Vergrößerung des Fehlers im Parallellauf ist jedoch nicht festzustellen. Ein deutlich besseres Ergebnis hinsichtlich Identifikationsverlauf und Konvergenz der Parameter ist mit 20 Basisfunktionen zu erreichen

(vgl. Abb. 4.30). Allerdings ist die Anzahl der Parameter viermal so hoch, wie bei 10 Basisfunktionen, aber dennoch deutlich geringer als ohne Einführung orthonormaler Basisfunktionen. In Abb. 4.31 sind die identifizierten unsymmetrischen Volterra-Kerne zweiter Ordnung für 10 (links) und 20 (rechts) Basisfunktionen dargestellt.

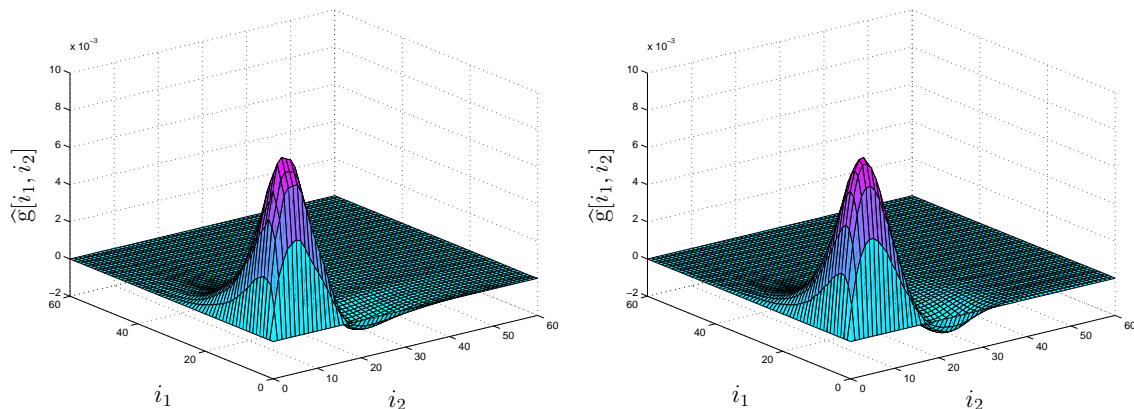


Abb. 4.31: *Unsymmetrischer Volterra-Kern mit 10 (links) und 20 (rechts) Basisfunktionen*

Es ist deutlich zu erkennen, dass es sich in Abb. 4.31 um einen unsymmetrischen Volterra-Kern zweiter Ordnung handelt. In den Richtungen von i_1 und i_2 sind die unterschiedlichen Impulsantworten der Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ zu erkennen. Auf den ersten Blick unterscheiden sich die beiden Volterra-Kerne in Abb. 4.31 nur unwesentlich. Jedoch führen bereits minimale Abweichungen im Kernverlauf zu einer deutlichen Erhöhung des Ausgangsfehlers e , wie in den Abbildungen 4.29 links und 4.30 links zu erkennen ist.

Bei den bisherigen Untersuchungen wurde davon ausgegangen, dass die Einstellwerte zur Einführung orthonormaler Basisfunktionen nur für das Gesamtsystem ermittelt werden können. Das bedeutet, dass bei unsymmetrischen Volterra-Kernen ein Kompromiss bei der Antwortlänge m , bei der Basisfunktionenanzahl m_r und vor allem beim Formfaktor ζ getroffen werden muss. Im Folgenden soll davon ausgegangen werden, dass für die Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ jeweils getrennt die Einstellwerte für die orthonormalen Basisfunktionen ermittelt werden können¹³, so dass für die i_1 - und die i_2 -Richtung unterschiedliche Basisfunktionen verwendet werden können, die jeweils optimal an die Systemdynamik angepasst sind. Das System

¹³Dies kann z.B. dadurch erfolgen, dass ein Eingangssignal konstant gehalten wird und bezüglich des zweiten Eingangssignals die Sprungantwort des Systems ermittelt wird. Es ist jedoch zu beachten, dass dies bei realen Systemen selten möglich ist.

kann mit unterschiedlichen Basisfunktionenmatrizen wie folgt beschrieben werden:

$$\begin{aligned}
 y[k] &= \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \\
 \underline{\mathcal{A}}_{dyn}^T[k] &= \left(\underline{u}_1^T[k] \cdot \tilde{\mathbf{R}}_1^T \right) \otimes \left(\underline{u}_2^T[k] \cdot \tilde{\mathbf{R}}_2^T \right) \\
 \underline{u}_1^T[k] &= \left[u_1[k-1], u_1[k-2], \dots, u_1[k-m_1] \right] \\
 \underline{u}_2^T[k] &= \left[u_2[k-1], u_2[k-2], \dots, u_2[k-m_2] \right] \\
 \underline{\theta}^T &= \left[g_r[1,1], \dots, g_r[m_{r1}, m_{r2}] \right]
 \end{aligned}$$

Für die Identifikation werden die Einstellwerte nach Tabelle 4.4 verwendet.

Einstellwerte	Erläuterung
$m_1 = m_2 = 60$	Antwortlänge
$r = 2$	Anzahl an Eingangssignalen
$m_{r1} = 15$ und $m_{r2} = 26$	Basisfunktionenanzahl
$\zeta_1 = 10.85$ und $\zeta_2 = 13.1$	Formfaktor
$p = m_{r1} \cdot m_{r2} = 390$	Parameteranzahl
$h = 1s$	Abtastzeit

Tabelle 4.4: Einstellwerte für Identifikationsbeispiel 2 mit unterschiedlichen orthonormalen Basisfunktionen für $F_1(s)$ und $F_2(s)$

Abbildung 4.32 zeigt den Verlauf der Identifikation und das Konvergenzverhalten der Parameter für diesen Fall.

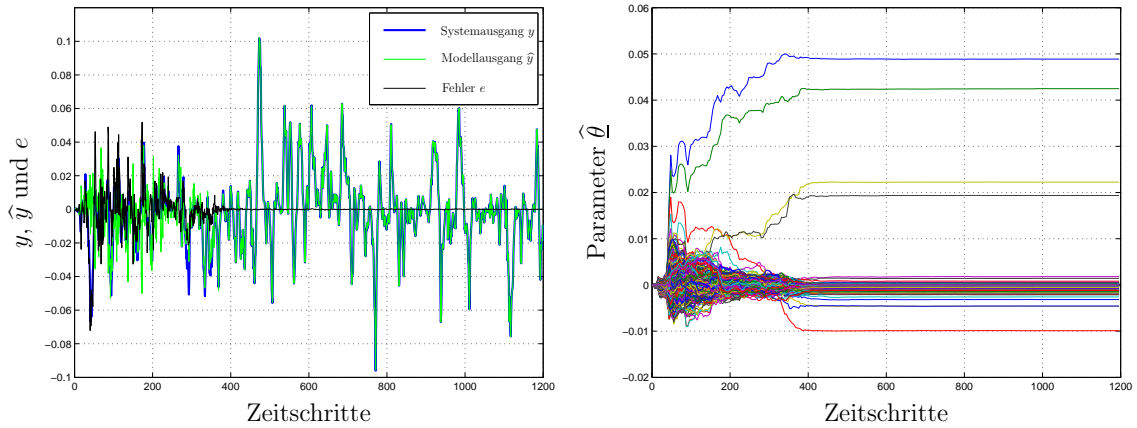


Abb. 4.32: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts) mit unterschiedlichen Basisfunktionen in i_1 - und i_2 -Richtung

Es ist zu erkennen, dass der Fehlerverlauf e und das Konvergenzverhalten der Parameter im Vergleich zu Abb. 4.30 leicht verbessert werden konnte, obwohl sich die

Parameteranzahl etwas reduziert hat. Somit ist festzuhalten, dass die Optimierung der Einstellwerte für die orthonormalen Basisfunktionen in i_1 - und i_2 -Richtung prinzipiell Vorteile bringt. Voraussetzung dafür ist jedoch, dass die unterschiedlichen Systemdynamiken in Vorversuchen ermittelt werden können.

Im Folgenden wird die Identifikation des allgemeineren Falls von nichtlinear verkoppelten Hammerstein-Modellen betrachtet. Das System in Abb. 4.28 enthält die statischen Nichtlinearitäten:

$$\begin{aligned}\mathcal{N}_1(u_1) &= 0.1 + 0.7 \cdot u_1 - 0.9 \cdot u_1^2 \\ \mathcal{N}_2(u_2) &= 0.1 - 0.5 \cdot u_2 + 1.0 \cdot u_2^2\end{aligned}$$

Die Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ bleiben unverändert. Dieses System kann mit Gl. (4.38) beschrieben werden und enthält zwei lineare Volterra-Kerne, zwei Volterra-Kerne zweiter Ordnung, die ausschließlich auf der Hauptdiagonale besetzt sind, und vier unsymmetrische Volterra-Kerne zweiter Ordnung. Für die Identifikation werden die Einstellwerte nach Tabelle 4.5 verwendet.

Einstellwerte	Erläuterung
$q = 2$	Grad der Nichtlinearität
$m = 60$	Antwortlänge
$r = 2$	Anzahl an Eingangssignalen
$m_r = 20$	Basisfunktionenanzahl
$\zeta = 9.8$	Formfaktor
$p = (2 \cdot m_r + 1)^r = 1681$	Parameteranzahl
$h = 1s$	Abtastzeit

Tabelle 4.5: *Einstellwerte für Identifikationsbeispiel 2 — Zwei ausgangsseitig nichtlinear verkoppelte Hammerstein-Modelle*

Die Parameteranzahl kann durch die Einführung orthonormaler Basisfunktionen von ursprünglich 14641 auf 1681 reduziert werden. Die Anzahl der orthonormalen Basisfunktionen wird mit $m_r = 20$ bewusst hoch gewählt, um ein gutes Identifikationsergebnis zu erzielen. Die Simulationsdauer beträgt insgesamt 5000 Zeitschritte. Nach 4500 Zeitschritten wird die Parameteradaptation eingestellt und das identifizierte Modell wird als reines Parallelmodell betrieben. In Abb. 4.33 ist der Identifikationsverlauf und die Konvergenz der Parameter veranschaulicht. Es ist zu erkennen, dass der Fehler e gegen Null geht und auch im Parallellauf nicht ansteigt. Die Konvergenz der Parameter (vgl. Abb. 4.33 rechts) verlängert sich entsprechend der Parameteranzahl, die trotz Parameterreduktion noch hoch ist. Auf die Darstellung der statischen Nichtlinearitäten und der einzelnen Volterra-Kerne wird an dieser Stelle verzichtet.

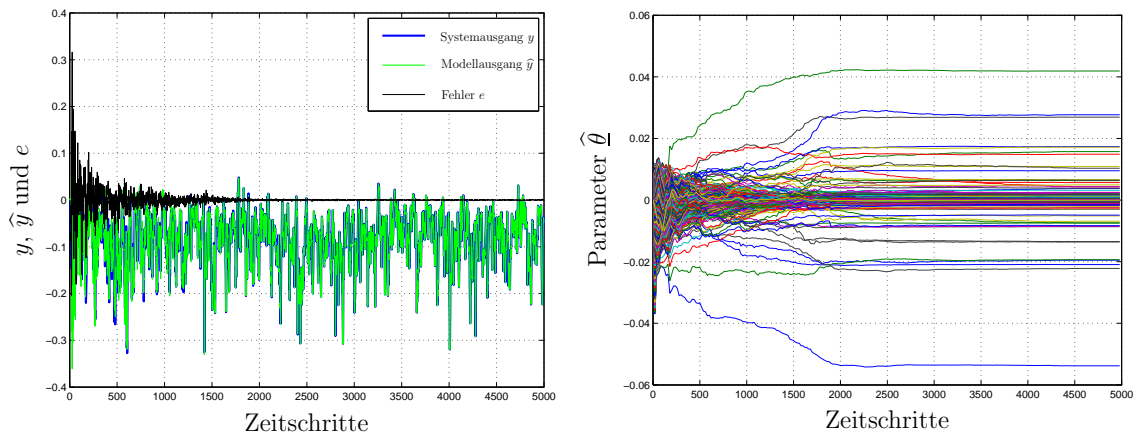


Abb. 4.33: Identifikationsverlauf (links) und Konvergenz der Parameter (rechts)

4.4 Rekonstruktion der blockorientierten Modellstruktur

Bei der Identifikation von nichtlinearen dynamischen Systemen auf Basis der Volterra-Funktionalpotenzreihe kann das Identifikationsergebnis nicht direkt in ein blockorientiertes Modell übertragen werden¹⁴. Das Identifikationsergebnis beschreibt zunächst nur das Ein-/Ausgangsverhalten des Systems. Ist jedoch bei der Identifikation eine Systemstruktur angenommen worden, kann diese aus dem Identifikationsergebnis wieder rekonstruiert werden. In Abb. 4.34 ist das Prinzip der Rekonstruktion am Beispiel eines Hammerstein-Modells dargestellt.

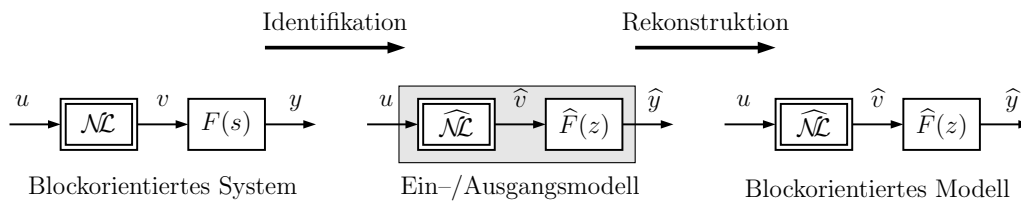


Abb. 4.34: Prinzip der Rekonstruktion einer blockorientierten Modellstruktur

Das in Abb. 4.34 verdeutlichte Prinzip gilt in analoger Weise auch für MISO-Systeme und wurde bereits in Kapitel 4.3.3 zur Veranschaulichung der Identifikationsergebnisse angewendet. Die Rekonstruktion der Modellstruktur kann aus unterschiedlichen Gründen sinnvoll sein. Der Entwurf von Regelstrategien kann durch die rekonstruierte Modellstruktur deutlich erleichtert werden. Das Hammerstein-Modell kann zum Beispiel, durch Vorschalten eines inversen Übertragungsblockes der rekonstruierten statischen Nichtlinearität $\widehat{\mathcal{NL}}$, linearisiert werden. Außerdem hat ein rekon-

¹⁴Diese Aussage gilt insbesondere bei der Einführung orthonormaler Basisfunktionen zur Parameterreduktion.

struiertes Modell Vorteile hinsichtlich der Rechenzeit. Im Folgenden wird zunächst die Rekonstruktion von SISO–Modellstrukturen betrachtet. Diese bilden die Grundlage zur Rekonstruktion der MISO–Strukturen aus den vorangegangenen Kapiteln.

4.4.1 Rekonstruktion von SISO–Modellstrukturen

In [30, 44, 60] wurde bereits die Rekonstruktion von SISO–Modellstrukturen untersucht. Die Ergebnisse sollen an dieser Stelle kurz zusammengefasst werden.

4.4.1.1 Rekonstruktion der Hammerstein–Modellstruktur

Das Identifikationsergebnis beschreibt zunächst nur das Ein–/Ausgangsverhalten des Systems. Das Ein–/Ausgangsverhalten ist durch die geschätzten Parameterwerte $\hat{\underline{\theta}}$ festgelegt. Diese sind der Ausgangspunkt für die Rekonstruktion der Modellstruktur. Für ein Hammerstein–Modell ergibt sich eine Parameteranzahl abhängig vom Grad q der statischen Nichtlinearität und der Anzahl orthonormalen Basisfunktionen m_r zu $p = q \cdot m_r + 1$. Der identifizierte Parametervektor sieht allgemein wie folgt aus:

$$\hat{\underline{\theta}}^T = \left[\hat{g}_0, \hat{\theta}[1], \dots, \hat{\theta}[m_r], \hat{\theta}[1, 1], \dots, \hat{\theta}[m_r, m_r], \dots, \hat{\theta}[m_r, \dots, m_r] \right] \quad (4.57)$$

Dieser Vektor besitzt entsprechend Gl. (3.58) eine gewisse Struktur, d.h. die Elemente dieses Vektors lassen sich in Gruppen unterschiedlicher Zugehörigkeit einteilen. Der erste Parameter beschreibt den Gleichanteil g_0 des Systems, die weiteren m_r Parameter sind die geschätzten Gewichte $\hat{\underline{\theta}}_1$ der orthonormalen Basisfunktionen, die den linearen Kern charakterisieren, bis zu den letzten m_r Elementen des Parametervektors, die zusammen mit den orthonormalen Basisfunktionen den Kern q -ten Grades approximieren. Mit dieser Erkenntnis kann der Parametervektor in folgender Struktur angeschrieben werden:

$$\hat{\underline{\theta}}^T = \left[\hat{g}_0, \hat{\underline{\theta}}_1^T, \dots, \hat{\underline{\theta}}_q^T \right] \quad \hat{\underline{\theta}}_i \in \mathbb{R}^{m_r \times 1} \quad \text{mit} \quad i = 1 \dots q \quad (4.58)$$

Die Gewichtsfolgen der einzelnen Volterra–Kerne können durch die gewichtete Überlagerung der orthonormalen Basisfunktionen rekonstruiert werden. Für die einzelnen Volterra–Kerne ergibt sich:

$$\hat{g}[i] = \sum_{j=1}^{m_r} \hat{\theta}[j] \tilde{r}_j[i] \quad \hat{g}[i, i] = \sum_{j=1}^{m_r} \hat{\theta}[j, j] \tilde{r}_j[i] \quad \dots \quad \hat{g}[i, \dots, i] = \sum_{j=1}^{m_r} \hat{\theta}[j, \dots, j] \tilde{r}_j[i] \quad (4.59)$$

Aus den rekonstruierten Volterra–Kernen können durch einen Vergleich mit der allgemeinen Beschreibung eines Hammerstein–Modells nach Gl. (3.50), die Koeffizienten der statischen Nichtlinearität und die Impulsantwort des linearen dynamischen Systems berechnet werden. Aufgrund der Tatsache, dass ein beliebiger

Verstärkungsfaktor K nicht eindeutig der statischen Nichtlinearität \mathcal{NL} oder der Übertragungsfunktion $F(s)$ zugeordnet werden kann, wird die Annahme getroffen, dass der Verstärkungsfaktor von $F(s)$ gleich Eins ist, d.h. $\sum_{i=1}^m h[i] = 1$. Für diesen Fall können die Koeffizienten der statischen Nichtlinearität berechnet werden zu:

$$\hat{a}_0 = \hat{g}_0 \quad \hat{a}_1 = \sum_{i=1}^m \hat{g}[i] \quad \hat{a}_2 = \sum_{i=1}^m \hat{g}[i, i] \quad \dots \quad \hat{a}_q = \sum_{i=1}^m \hat{g}[i, \dots, i] \quad (4.60)$$

Die statische Nichtlinearität ist durch die Polynomkoeffizienten $\hat{a}_0 \dots \hat{a}_q$ rekonstruiert.

Zur Bestimmung der Impulsantwort der Übertragungsfunktion kann ein beliebiger Volterra-Kern durch seinen zugehörigen Polynomkoeffizienten geteilt werden.

$$\hat{h}_k[i] = \frac{1}{\hat{a}_k} \cdot \underbrace{\hat{g}[i, \dots, i]}_k \quad \text{mit } k = 1 \dots q \quad (4.61)$$

Die berechneten Impulsantworten $\hat{h}_k[i]$ sollten alle exakt gleich sein, da nur eine Dynamik im Hammerstein-Prozess vorhanden ist. Ist der Systemausgang jedoch während der Identifikation gestört, können die einzelnen Impulsantworten $\hat{h}_k[i]$ voneinander abweichen. In diesem Fall empfiehlt es sich eine Mittelung der einzelnen Impulsantworten mit einer anschließenden Normierung auf die angenommene Verstärkung durchzuführen.

An dieser Stelle ist das Ziel der Rekonstruktion einer blockorientierten Modellstruktur erreicht, jedoch kann es wünschenswert sein, den linearen Block in Form einer Übertragungsfunktion darzustellen. Eine analytische Vorschrift für die Umrechnung der Impulsantwort in eine rational darstellbare Übertragungsfunktion existiert nicht [31]. Aus diesem Grund wird die rational darstellbare Übertragungsfunktion durch Lösung eines überbestimmten linearen Gleichungssystems z.B. mit Hilfe des rekursiven Least-Squares-Algorithmus aus der bekannten Impulsantwort bestimmt. Die mathematische Formulierung kann mit dem Systemgrad n der rationalen Übertragungsfunktion, angeschrieben werden durch:

$$\hat{F}(z) = \sum_{i=1}^m \hat{h}[i] z^{-i} \quad \xrightarrow{\text{RLS}} \quad \hat{F}(z) = \frac{\hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + \dots + \hat{b}_n z^{-n}}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \dots + \hat{a}_n z^{-n}}$$

Als Lösung des überbestimmten Gleichungssystems ergeben sich die Koeffizienten der rationalen zeitdiskreten Übertragungsfunktion.

4.4.1.2 Rekonstruktion der Wiener-Modellstruktur

Ausgangspunkt für die Rekonstruktion der Wiener-Modellstruktur ist wiederum der geschätzte Parametervektor $\hat{\underline{\theta}}$. Für das Wiener-Modell ergibt sich eine Parameteranzahl abhängig vom Grad q der Nichtlinearität \mathcal{NL} und der Anzahl orthonormalen

Basisfunktionen m_r zu $p = \binom{m_r+q}{q}$. Der identifizierte Parametervektor sieht allgemein wie folgt aus:

$$\underline{\hat{\theta}}^T = \left[\hat{g}_0, \hat{\theta}[1], \dots, \hat{\theta}[m_r], \hat{\theta}[1, 1], \hat{\theta}[1, 2], \dots, \hat{\theta}[m_r, m_r], \dots, \hat{\theta}[m_r, \dots, m_r] \right] \quad (4.62)$$

Dieser Vektor besitzt nach Gl. (3.61) ebenfalls eine gewisse Struktur. Die Vektoren $\hat{\theta}_i$ in Gl. (4.63) beinhalten die Gewichte der orthonormalen Basisfunktionen zur Rekonstruktion des Volterra-Kerns i -ten Grades:

$$\underline{\hat{\theta}}^T = \left[\hat{g}_0, \hat{\theta}_1^T, \dots, \hat{\theta}_q^T \right] \quad \text{mit} \quad \hat{\theta}_i \in \mathbb{R}^{[\binom{m_r+i}{i} - \binom{m_r+i-1}{i-1}]} \times 1 \quad i = 1 \dots q \quad (4.63)$$

Die Gewichtsfolgen der einzelnen Volterra-Kerne können durch die gewichtete Überlagerung der orthonormalen Basisfunktionen rekonstruiert werden. Für die einzelnen Volterra-Kerne ergibt sich beim Wiener-Modell:

$$\begin{aligned} \hat{g}[i_1] &= \sum_{j_1=1}^{m_r} \hat{\theta}[j_1] \tilde{r}_{j_1}[i_1] \\ \hat{g}[i_1, i_2] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \hat{\theta}[j_1, j_2] \tilde{r}_{j_2}[i_2] \tilde{r}_{j_1}[i_1] \\ &\vdots \\ \hat{g}[i_1, \dots, i_q] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \dots \sum_{j_q=j_{q-1}}^{m_r} \hat{\theta}[j_1, \dots, j_q] \tilde{r}_{j_q}[i_q] \dots \tilde{r}_{j_1}[i_1] \end{aligned} \quad (4.64)$$

Aus den rekonstruierten Volterra-Kernen können durch einen Vergleich mit der allgemeinen Beschreibung eines Wiener-Modells nach Gl. (3.53), die Koeffizienten der statischen Nichtlinearität und die Impulsantwort des linearen dynamischen Systems berechnet werden. Der Verstärkungsfaktor der Übertragungsfunktion wird wieder zu Eins angenommen. Die Koeffizienten der statischen Nichtlinearität berechnen sich wie folgt:

$$\begin{aligned} \hat{a}_0 &= \hat{g}_0 & \hat{a}_1 &= \sum_{i_1=1}^m \hat{g}[i_1] \\ \hat{a}_2 &= \sum_{i_1=1}^m \sum_{i_2=i_1}^m \hat{g}[i_1, i_2] & \dots & \hat{a}_q = \sum_{i_1=1}^m \dots \sum_{i_q=i_{q-1}}^m \hat{g}[i_1, \dots, i_q] \end{aligned} \quad (4.65)$$

Zur Bestimmung der Impulsantwort der Übertragungsfunktion kann prinzipiell ein beliebiger Volterra-Kern verwendet werden. Jedoch gestaltet sich die Berechnung bei Volterra-Kernen höherer Ordnung schwieriger als beim Hammerstein-Modell. Am einfachsten erfolgt die Berechnung der Impulsantwort aus dem linearen Kern:

$$\hat{h}[i] = \frac{1}{\hat{a}_1} \cdot \hat{g}[i_1] \quad (4.66)$$

Mit den Gleichungen (4.65) und (4.66) ist das Ziel der Rekonstruktion einer blockorientierten Modellstruktur erreicht. Der lineare Übertragungsblock kann noch bei Bedarf, wie bei der Rekonstruktion der Hammerstein-Modellstruktur beschrieben, in eine rationale Übertragungsfunktion umgerechnet werden.

4.4.2 Rekonstruktion von MISO-Modellstrukturen

Im Kapitel 4.4.1 wurde die Rekonstruktion von SISO-Systemen behandelt. Die Elemente des identifizierten Parametervektors $\hat{\underline{\theta}}$ wurden zunächst den einzelnen Kernen des Hammerstein- bzw. Wiener-Modells zugeordnet. Anschließend wurden die verschiedenen Volterra-Kerne durch Verrechnung der identifizierten Gewichte mit den orthonormalen Basisfunktionen rekonstruiert. Durch einen Koeffizientenvergleich mit den allgemeinen Beschreibungen von Hammerstein- bzw. Wiener-Modell konnte auf die Polynomkoeffizienten der statischen Nichtlinearität und auf die Impulsantwort des linearen dynamischen Systems zurückgerechnet werden.

Diese prinzipielle Vorgehensweise ist auch bei der Rekonstruktion von MISO-Systemen anwendbar. Dabei gestaltet sich die Rekonstruktion von linear verkoppelten Modellstrukturen relativ einfach, da die Erkenntnisse zur Rekonstruktion von SISO-Systemen vollständig genutzt werden können. Die Rekonstruktion von nichtlinear verkoppelten Modellstrukturen gestaltet sich hingegen deutlich komplizierter.

4.4.2.1 Rekonstruktion von linear verkoppelten Modellstrukturen

Eine allgemeingültige formale Beschreibung zur Rekonstruktion von linear verkoppelten MISO-Modellstrukturen zu finden ist annähernd unmöglich. Aus diesem Grund wird lediglich auf die prinzipielle Vorgehensweise eingegangen und die wesentlichen Unterschiede und Gemeinsamkeiten im Vergleich zur Rekonstruktion von SISO-Modellstrukturen erläutert.

Zunächst werden linear verkoppelte Hammerstein-Modelle betrachtet. In Gl. (4.10) ist zu erkennen, dass in der Beschreibung linear verkoppelter Hammerstein-Modelle ausschließlich lineare Volterra-Kerne und Volterra-Kerne höherer Ordnung, die nur auf der Hauptdiagonale besetzt sind, vorkommen. Die Parameterreduktion mittels orthonormaler Basisfunktionen wurde für solche Kerne in Kapitel 4.3.1 beschrieben. Der Parametervektor $\hat{\underline{\theta}}$, der wiederum Ausgangspunkt der Rekonstruktion ist, weist ebenso wie beim Hammerstein-Modell eine Struktur auf. Das erste Element im Parametervektor beschreibt den Gleichanteil \hat{g}_0 . Die weiteren Elemente des Parametervektors können zu Vektoren zusammengefasst werden, die jeweils die Gewichte der orthonormalen Basisfunktionen zur Beschreibung eines bestimmten Volterra-Kerns darstellen. Die Rekonstruktion der einzelnen Volterra-Kerne erfolgt durch die Verrechnung mit den orthonormalen Basisfunktionen analog zu Gl. (4.59). Bevor auf die einzelnen Polynomkoeffizienten zurückgerechnet werden kann, muss eine Annahme über die Verstärkungsfaktoren der einzelnen Übertragungsfunktionen $F_k(s)$ im

System getroffen werden¹⁵. Durch einen Koeffizientenvergleich mit der allgemeinen Systembeschreibung können aus den rekonstruierten Volterra-Kernen die Polynomkoeffizienten der statischen Nichtlinearitäten \mathcal{NL}_k sowie die Impulsantworten $h_k[i]$ bestimmt werden. Dabei ist zu beachten, dass der identifizierte Gleichanteil \hat{g}_0 nicht mehr eindeutig auf die Gleichanteile der einzelnen statischen Nichtlinearitäten aufgeteilt werden kann.

In Gl. (4.22) ist zu erkennen, dass in der Beschreibung linear verkoppelter Wiener-Modelle neben dem bereits vom Wiener-Modell bekannten linearen Volterra-Kern und den symmetrischen Volterra-Kernen auch unsymmetrische Volterra-Kerne auftreten können. Die Parameterreduktion mittels orthonormaler Basisfunktionen wurde für unsymmetrische Volterra-Kerne ebenfalls in Kapitel 4.3.1 beschrieben. Der erste Schritt ist wiederum die Rekonstruktion der Volterra-Kerne, anhand der im Parametervektor $\hat{\theta}$ enthaltenen Gewichte und den orthonormalen Basisfunktionen. Das erste Element im Parametervektor beschreibt den Systemoffset \hat{g}_0 . Die Rekonstruktion der linearen bzw. der symmetrischen Volterra-Kerne erfolgt durch die Verrechnung mit den orthonormalen Basisfunktionen analog zu Gl. (4.64). Ein beliebiger unsymmetrischer Volterra-Kern der Ordnung r kann wie folgt rekonstruiert werden:

$$\hat{g}[i_1, \dots, i_r] = \sum_{j_1=1}^{m_r} \sum_{j_2=1}^{m_r} \cdots \sum_{j_r=1}^{m_r} \hat{\theta}[j_1, \dots, j_r] \tilde{r}_{j_r}[i_r] \cdots \tilde{r}_{j_1}[i_1] \quad (4.67)$$

An dieser Stelle sei erwähnt, dass die Rekonstruktion der unsymmetrischen Volterra-Kerne bei linear verkoppelten Wiener-Modellen zur Berechnung der Polynomkoeffizienten und der Impulsantworten nicht unbedingt erforderlich ist, da sowohl die Polynomkoeffizienten als auch die Impulsantworten allein aus den linearen und symmetrischen Volterra-Kernen berechnet werden können. Ist eine Annahme über die Verstärkungsfaktoren der einzelnen Übertragungsfunktionen $F_k(s)$ im System getroffen, können sowohl die Polynomkoeffizienten der statischen Nichtlinearitäten \mathcal{NL}_k als auch die Impulsantworten $h_k[i]$ durch einen Koeffizientenvergleich mit der allgemeinen Systembeschreibung bestimmt werden. Zu beachten ist, dass der identifizierte Gleichanteil \hat{g}_0 nicht in allen Fällen eindeutig auf die Gleichanteile der einzelnen statischen Nichtlinearitäten aufgeteilt werden kann.

4.4.2.2 Rekonstruktion von nichtlinear verkoppelten Modellstrukturen

Auch im Falle von nichtlinear verkoppelten MISO-Modellstrukturen wird ausschließlich auf die prinzipielle Vorgehensweise eingegangen und die wesentlichen Unterschiede und Gemeinsamkeiten im Vergleich zur Rekonstruktion von SISO-Modellstrukturen bzw. linear verkoppelten MISO-Modellstrukturen eingegangen.

In Gl. (4.18) ist zu erkennen, dass in der Beschreibung nichtlinear verkoppelter Hammerstein-Modelle lineare Volterra-Kerne, Volterra-Kerne höherer Ordnung,

¹⁵Dies kann z.B. sein $\sum_{i=1}^m h_k[i] = 1$.

die nur auf der Hauptdiagonale besetzt sind, und unsymmetrische Volterra-Kerne höherer Ordnung vorkommen. Der Parametervektor $\hat{\underline{\theta}}$, der wiederum Ausgangspunkt der Rekonstruktion ist, weist ebenso wie in allen vorangegangenen Fällen eine Struktur auf. Das erste Element im Parametervektor beschreibt den Systemoffset \hat{g}_0 . Die weiteren Elemente des Parametervektors können wie vorher zu Vektoren zusammengefasst werden, die jeweils die Gewichte der orthonormalen Basisfunktionen zur Beschreibung eines bestimmten Volterra-Kerns darstellen. Die Rekonstruktion der einzelnen Volterra-Kerne erfolgt durch die Verrechnung mit den orthonormalen Basisfunktionen analog zu den Gleichungen (4.59) und (4.67). Zur Berechnung der Polynomkoeffizienten muss eine Annahme über die Verstärkungsfaktoren der einzelnen Übertragungsfunktionen $F_k(s)$ im System getroffen werden. Im Gegensatz zu den SISO-Modellstrukturen und auch den linear verkoppelten MISO-Modellstrukturen ergibt sich bei nichtlinear verkoppelten Hammerstein-Modellen an dieser Stelle ein überbestimmtes nichtlineares Gleichungssystem für die Polynomkoeffizienten, das nicht mehr analytisch gelöst werden kann. Die einzige Möglichkeit zur Bestimmung der Polynomkoeffizienten ist, mittels eines nichtlinearen Ausgleichsverfahrens ein wenn möglich globales Minimum dieses Gleichungssystems zu finden¹⁶. Mit der Kenntnis der Polynomkoeffizienten können die Impulsantworten $h_k[i]$ der Übertragungsfunktionen z.B. aus den linearen Volterra-Kernen leicht bestimmt werden. In Gl. (4.25) ist zu erkennen, dass in der Beschreibung nichtlinear verkoppelter Wiener-Modelle lineare Volterra-Kerne, symmetrische und unsymmetrische Volterra-Kerne höherer Ordnung auftreten. Die Rekonstruktion der einzelnen Volterra-Kerne durch die Verrechnung des Parametervektors $\hat{\underline{\theta}}$ mit den entsprechenden orthonormalen Basisfunktionen erfolgt analog zu den Gleichungen (4.64) und (4.67). Die Polynomkoeffizienten sind wiederum nur durch die Lösung eines überbestimmten nichtlinearen Gleichungssystems mittels eines nichtlinearen Ausgleichsverfahrens zu bestimmen. Mit der Kenntnis der Polynomkoeffizienten können auch die Impulsantworten $h_k[i]$ leicht bestimmt werden. Die Rekonstruktion von MISO-Modellstrukturen fand bereits in Kapitel 4.3.3 Anwendung. Weitere Beispiele zur Rekonstruktion von linear und nichtlinear verkoppelten MISO-Modellstrukturen sind in [60] zu finden.

4.5 Kurzzusammenfassung

In diesem Kapitel wurde die Identifikation nichtlinearer dynamischer Mehrgrößensysteme auf Basis der Volterra-Funktionalpotenzreihe umfassend untersucht. Ausgehend von dem Hammerstein- und dem Wiener-Modell wurden anhand der Untersuchung von möglichen linearen und nichtlinearen Eingriffspunkten weiterer Eingangsgrößen allgemeine Systembeschreibungen für linear und nichtlinear verkop-

¹⁶Bei der Rekonstruktion wurde im Rahmen dieser Arbeit der Levenberg-Marquardt-Algorithmus verwendet.

pelte Hammerstein- und Wiener-Modelle abgeleitet. Es musste jedoch festgestellt werden, dass die Komplexität solcher Systembeschreibungen sehr schnell ein Ausmaß annimmt, das die Formulierung einer allgemeingültigen Systembeschreibung für eine beliebige Anzahl an Eingangsgrößen unmöglich macht. Dennoch konnten durch die Untersuchung von Systemen mit zwei Eingangsgrößen umfangreiche Erkenntnisse über die Beschreibung von Mehrgrößensystemen mittels der Volterra-Funktionalpotenzreihe gewonnen werden [21, 22, 59]. Besonders ist die Entdeckung des unsymmetrischen Volterra-Kerns höherer Ordnung herauszuheben, der bisher in der Literatur nicht zu finden war.

Aufbauend auf den Systembeschreibungen für MISO-Systeme wurde die Identifikation von Mehrgrößensystemen behandelt. Ausgangspunkt waren die abgeleiteten Volterra-Funktionalpotenzreihen, deren Kerne aus dem Grund der Parameterreduktion durch die Überlagerung orthonormaler Basisfunktionen beschrieben werden mussten. Durch die Einführung des \otimes -Operators konnte auch für unsymmetrische Volterra-Kerne eine kompakte Schreibweise gefunden werden. Anhand eines Stabilitätsbeweises nach Lyapunov wurde die garantierte Stabilität der Identifikation sowie die Konvergenz der Parameter bei Verwendung des rekursiven Least-Squares-Algorithmus dargelegt. Die abgeleitete Theorie wurde mit zwei Simulationsbeispielen veranschaulicht.

Abschließend wurde die Rekonstruktion der Identifikationsergebnisse auf eine blockorientierte Modellstruktur genauer betrachtet. Grundlage war die Rekonstruktion der bekannten SISO-Modellstrukturen. Es zeigte sich, dass sich im Falle von linear verkoppelten MISO-Modellstrukturen keine gravierenden Unterschiede ergaben, wohingegen bei der Rekonstruktion nichtlinear verkoppelter MISO-Modellstrukturen ein überbestimmtes nichtlineares Gleichungssystem zur Bestimmung der Polynomkoeffizienten zu lösen ist.

5 Erweiterung des Verfahrens mittels statischer Funktionsapproximatoren

In den Kapiteln 3 und 4 wurde angenommen, dass die statischen Nichtlinearitäten von blockorientierten Systemen als Polynome q -ten Grades approximiert werden können. Dies stellt eine Einschränkung der Allgemeinheit dar. Zwar können prinzipiell alle stetigen nichtlinearen Funktionen durch Polynome approximiert werden, diese besitzen jedoch speziell bei hohem Grad q in der Praxis unerwünschte Eigenschaften. Aus diesem Grund soll im Folgenden eine Erweiterung und Verallgemeinerung des Identifikationsverfahrens aus Kapitel 4 erfolgen.

5.1 Motivation und Prinzip des erweiterten Verfahrens

Alle bisherigen Identifikationsansätze unterliegen der Einschränkung, dass die statischen Nichtlinearitäten durch Polynome q -ten Grades approximiert werden können [27, 30]. Muss zur Approximation der Grad einer Nichtlinearität sehr hoch gewählt werden, führt dies in der Praxis oft zu schlechten Ergebnissen, da das Polynom zu Schwingungen neigt und bei der Extrapolation gegen $\pm\infty$ strebt. Die Lösung dieses Problems ist ein Identifikationsansatz, der zur Approximation der statischen Nichtlinearitäten einen allgemeinen statischen Funktionsapproximator aus Kapitel 2 verwendet¹. Um die Linearität der Parameter in der Volterra-Funktionalpotenzreihe zu erhalten, muss der Funktionsapproximator selbst linear in den Parametern sein. Als statischer Funktionsapproximator hat sich das GRNN bewährt. Die Struktur des GRNN ist in Abb. 5.1 noch einmal dargestellt². Der Ausgang des GRNN (vgl. Kapitel 2.4) berechnet sich zu:

$$y_{\mathcal{N}} = \underline{\theta}_{\mathcal{N}}^T \cdot \underline{\mathcal{A}}(\underline{u})$$

Der Vektor $\underline{\theta}_{\mathcal{N}}$ beinhaltet die q Stützwerte. Die Aktivierung des i -ten Stützwertes kann entsprechend Gl.(2.6) bestimmt werden.

¹In [28] wird ein RBF-Netz vorgeschlagen. Die Nachteile des RBF-Netzes wurden bereits in Kapitel 2.3 diskutiert. Außerdem werden in [28] im Unterschied zu diesem Kapitel Übertragungsfunktionen durch Differenzgleichungen beschrieben.

²Die Nomenklatur wurde auf dieses Kapitel angepasst.

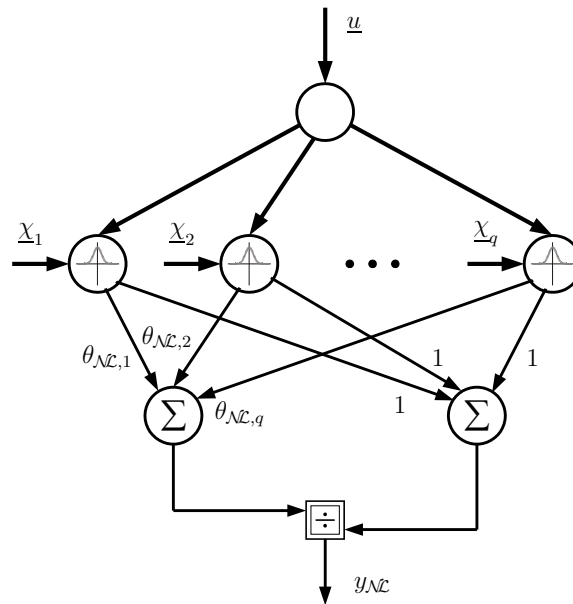


Abb. 5.1: Struktur des GRNN

Das GRNN wird im Folgenden als universeller Funktionsapproximator verwendet, um die statischen Nichtlinearitäten in den betrachteten blockorientierten Systemen zu beschreiben.

5.2 Erweiterung des Identifikationsverfahrens am Beispiel des Hammerstein-Modells

5.2.1 Beschreibung des Hammerstein-Modells mit Funktionsapproximatoransatz

Zunächst wird ein Hammerstein-Modell betrachtet. Das Hammerstein-Modell ist zur Veranschaulichung noch einmal in Abb. 5.2 dargestellt.

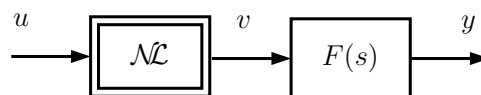


Abb. 5.2: Hammerstein-Modell

Für die statische Nichtlinearität \mathcal{NL} gilt:

$$v(u) = \underline{\theta}_{\mathcal{NL}}^T \cdot \underline{\mathcal{A}}(u) = \theta_{\mathcal{NL},1} \cdot \mathcal{A}_1(u) + \dots + \theta_{\mathcal{NL},q} \cdot \mathcal{A}_q(u) \quad (5.1)$$

Die Nichtlinearität wird durch ein GRNN mit q Stützwerten approximiert. Zu jedem Stützwert gehört eine Aktivierung $\mathcal{A}_1(u) \dots \mathcal{A}_q(u)$. Die Aktivierungen sind vom Eingangssignal u der statischen Nichtlinearität abhängig. Für den eindimensionalen Fall vereinfacht sich die Berechnung der Aktivierungen zu:

$$\mathcal{A}_i(u) = \frac{\exp\left[\frac{(u-\chi_i)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{(u-\chi_j)^2}{-2\sigma^2}\right]} \quad (5.2)$$

Mit der Standardabweichung σ wird der Grad der Überlappung der einzelnen Aktivierungsfunktionen eingestellt³.

Die Übertragungsfunktion wird wie bisher durch die Faltungssumme beschrieben, wobei als konstante obere Grenze die Antwortlänge m eingeführt wird⁴:

$$y[k] = \sum_{i=0}^k h[i]v[k-i] \quad \Longrightarrow \quad y[k] = \sum_{i=1}^m h[i]v[k-i] \quad (5.3)$$

Durch das Einsetzen von Gl. (5.1) in Gl. (5.3) ergibt sich die Systembeschreibung eines Hammerstein-Modells bei Approximation der Nichtlinearität durch ein GRNN:

$$y[k] = \theta_{\mathcal{NL},1} \sum_{i=1}^m h[i]\mathcal{A}_1(u[k-i]) + \dots + \theta_{\mathcal{NL},q} \sum_{i=1}^m h[i]\mathcal{A}_q(u[k-i]) \quad (5.4)$$

Ein Vergleich von Gl. (5.4) mit der Volterra-Funktionalpotenzreihe nach Gl. (3.45) zeigt, dass obige Gleichung nicht mehr exakt in die Volterra-Funktionalpotenzreihe überführt werden kann. Anders als bei der Approximation der statischen Nichtlinearität mit einem Polynom geht in Gl. (5.4) die Eingangsgröße u als Argument der Aktivierungsfunktionen in den Systemausgang ein. Wichtig ist jedoch, dass der Systemausgang nach wie vor linear in den Parametern ist. Werden die Aktivierungsfunktionen $\mathcal{A}_1(u) \dots \mathcal{A}_q(u)$ als Eingangssignale interpretiert, so kann durch Koeffizientenvergleich eine Analogie zur Volterra-Funktionalpotenzreihe hergestellt werden. Entsprechend den Volterra-Kernen ergibt sich:

$$g_1[i] = \theta_{\mathcal{NL},1} \cdot h[i] \quad g_2[i] = \theta_{\mathcal{NL},2} \cdot h[i] \quad \dots \quad g_q[i] = \theta_{\mathcal{NL},q} \cdot h[i] \quad (5.5)$$

Bei den Gewichtsfolgen in Gl. (5.5) soll ebenfalls von Volterra-Kernen gesprochen werden. Es wurde bereits berücksichtigt, dass im Falle des Hammerstein-Modells nur die Diagonalelemente der Volterra-Kerne besetzt sind. Ein Parameter g_0 , der den Systemoffset beschreibt, ist in Gl. (5.4) nicht nötig, da ein Systemoffset vom GRNN automatisch gelernt wird. So ergibt sich die Volterra-Reihe⁵ für Gl. (5.4) zu:

$$y[k] = \sum_{i=1}^m g_1[i] \mathcal{A}_1(u[k-i]) + \dots + \sum_{i=1}^m g_q[i] \mathcal{A}_q(u[k-i]) \quad (5.6)$$

³Für die Standardabweichung wird in der Regel ein auf den Stützstellenabstand normierter Wert angegeben: $\sigma_{norm} = \frac{\sigma}{\Delta x}$

⁴Es wird angenommen, dass das lineare System nicht sprunghaft ist.

⁵Im erweiterten Identifikationsansatz wird der Begriff Volterra-Reihe verwendet, um eine Abgrenzung zu Volterra-Funktionalpotenzreihen herzustellen.

Die Anzahl der unbekannt Parameter in Gl. (5.6) beträgt $p = q \cdot m$. Die Anzahl der Stützpunkte entspricht formal dem Grad der Nichtlinearität. Der Parameter g_0 für den Systemoffset kann eingespart werden. Auch bei dem erweiterten Identifikationsansatz ist eine Parameterreduktion erforderlich. Analog zum Polynomansatz werden orthonormale Basisfunktionen zur Parameterreduktion eingesetzt. Die Systembeschreibung ergibt sich nach der Einführung orthonormaler Basisfunktionen entsprechend Gl. (3.58) zu:

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (5.7)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[\underline{\mathcal{A}}_1^T[k] \cdot \tilde{\mathbf{R}}^T, \underline{\mathcal{A}}_2^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \tilde{\mathbf{R}}^T \right] \\ \underline{\mathcal{A}}_i^T[k] &= \left[\mathcal{A}_i(u[k-1]), \mathcal{A}_i(u[k-2]), \dots, \mathcal{A}_i(u[k-m]) \right] \\ \underline{\theta}^T &= \left[g_{r,1}[1], \dots, g_{r,1}[m_r], g_{r,2}[1], \dots, g_{r,2}[m_r], \dots, g_{r,q}[m_r] \right] \end{aligned}$$

Für die Approximation der Gewichtsfolgen durch orthonormale Basisfunktionen muss für jeden Stützpunkt jeweils ein Vektor $\underline{\mathcal{A}}_i[k]$ gebildet werden, der die Aktivierungsfunktion $\mathcal{A}_i(u)$ zu den Eingangssignalen $u[k-1] \dots u[k-m]$ enthält. Die nach den Stützstellen sortierten Aktivierungsvektoren $\underline{\mathcal{A}}_1 \dots \underline{\mathcal{A}}_q$ müssen mit der Basisfunktionenmatrix $\tilde{\mathbf{R}}$ multipliziert werden. Der Parametervektor $\underline{\theta}$ enthält die reduzierten Gewichtsfolgen $g_{r,i}$ und kann nach der Zugehörigkeit der einzelnen Parameter zu den entsprechenden Stützstellen des GRNN wie folgt dargestellt werden:

$$\underline{\theta}^T = [\underline{\theta}_1^T, \underline{\theta}_2^T, \dots, \underline{\theta}_q^T] \quad (5.8)$$

Die Dimensionen der verwendeten Vektoren und Matrizen in den Gleichungen (5.7) und (5.8) sind abhängig von der Antwortlänge m , der Anzahl verwendeter Basisfunktionen m_r und der Stützstellenanzahl q des GRNN. Es gilt:

$$\begin{aligned} \tilde{\mathbf{R}} &\in \mathbb{R}^{m_r \times m} \\ \underline{\mathcal{A}}_1^T, \underline{\mathcal{A}}_2^T, \dots, \underline{\mathcal{A}}_q^T &\in \mathbb{R}^{1 \times m} \\ \underline{\theta}_1, \underline{\theta}_2, \dots, \underline{\theta}_q &\in \mathbb{R}^{m_r \times 1} \\ \underline{\theta} &\in \mathbb{R}^{(q \cdot m_r) \times 1} \end{aligned} \quad (5.9)$$

Die Anzahl der Parameter konnte durch die Einführung orthonormaler Basisfunktionen von $p = q \cdot m$ auf $p = q \cdot m_r$ verringert werden.

5.2.2 Vergleich zwischen Polynom- und Funktionsapproximatoransatz

Die Verbesserung, die mit dem in Kapitel 5.2.1 beschriebenen Identifikationsverfahren erzielt werden kann, soll anhand eines Simulationsbeispiels verdeutlicht werden [18]. Gegeben ist das folgende Hammerstein-Modell:

$$\mathcal{NL}(u) = \arctan(10 \cdot u) \quad F(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1}$$

Die Zeitkonstanten von $F(s)$ sind auf einen Simulationszeitschritt von $1s$ normiert. Die statische Nichtlinearität wurde als Arcustangensfunktion gewählt, da diese Funktion in der Praxis häufig vorkommt⁶ und zudem nur durch ein Polynom hohen Grades beschrieben werden kann.

Im Folgenden werden der Identifikationsansatz mit dem Polynom und der Identifikationsansatz mit dem GRNN zur Approximation der statischen Nichtlinearität miteinander verglichen. Für die Identifikation werden die Einstellwerte nach Tabelle 5.1 verwendet.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl bzw. Grad der Nichtlinearität
$\sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 50$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.65$	Formfaktor
$p_{GRNN} = m_r \cdot q = 210$	Parameteranzahl bei GRNN
$p_{POLY} = m_r \cdot q + 1 = 211$	Parameteranzahl bei Polynom
$h = 1s$	Abtastzeit

Tabelle 5.1: *Einstellwerte für Vergleich von Funktionsapproximator- und Polynomansatz*

Die Simulationsdauer beträgt insgesamt 3000 Zeitschritte. Nach 2500 Zeitschritten wird die Parameteradaption eingestellt und die identifizierten Modelle werden als reine Parallelmodelle betrieben. Die Systeme werden durch ein Zufallssignal im Bereich $[-1; 1]$ angeregt. Als Adaptionverfahren kommt der rekursive Least-Squares-Algorithmus zum Einsatz.

In Abb. 5.3 ist ein Vergleich der Fehlerverläufe beider Ansätze dargestellt. Es ist zu erkennen, dass der Fehler e beim Funktionsapproximatoransatz deutlich kleiner wird als beim Polynomansatz. Die Anzahl der Parameter ist bis auf den Parameter für den Systemoffset beim Polynomansatz gleich. Im Parallellauf bleibt der Fehler beim Funktionsapproximatoransatz klein, während er beim Polynomansatz etwas größer wird. Dies kann durch das Konvergenzverhalten der Parameter, das in Abb. 5.4 veranschaulicht ist, begründet werden. Die Parameter des Funktionsapproximatoransatzes können am Ende der Identifikation als konvergiert angesehen werden. Dies gilt nicht für die Parameter des Polynomansatzes, wo viele Parameter noch nicht annähernd konvergiert haben. Dies wird auch bei der Betrachtung der aus den Parametern rekonstruierten statischen Nichtlinearitäten deutlich, die in Abb. 5.5 veranschaulicht sind. Durch das lokale Approximationsverhalten des GRNN kann jede beliebige statische Nichtlinearität nachgebildet werden. Im Gegensatz dazu wirkt sich ein Polynomkoeffizient im gesamten Eingangsbereich aus. Das führt zu langen Konvergenzzeiten, da die Polynomkoeffizienten höherer Ordnung erst kon-

⁶Sättigungscharakteristiken und Reibkennlinien werden häufig als Arcustangensfunktionen modelliert.

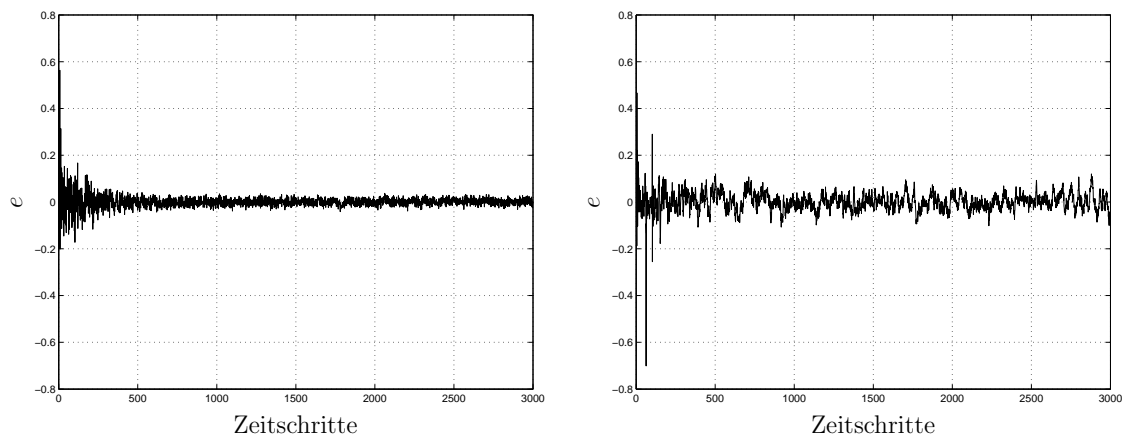


Abb. 5.3: Vergleich der Fehlerverläufe des Funktionsapproximatoransatzes (links) und des Polynomansatzes (rechts)

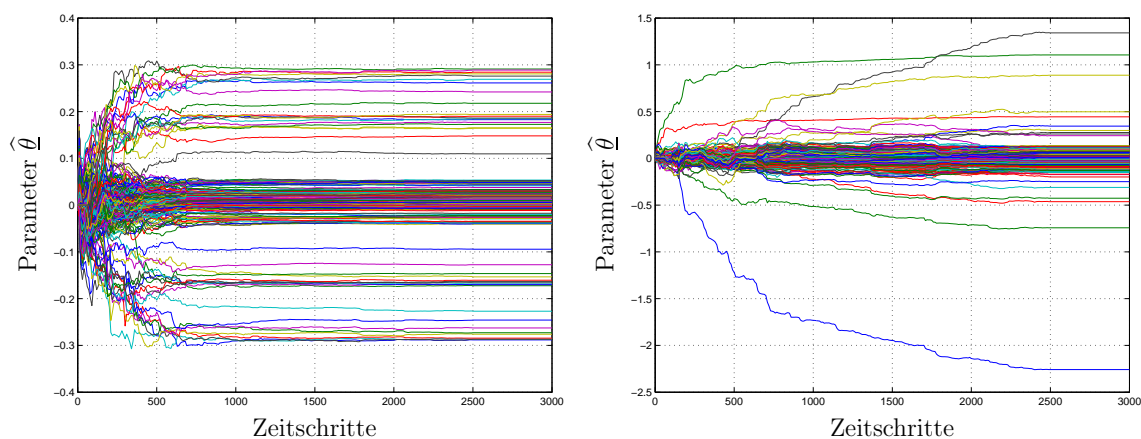


Abb. 5.4: Vergleich der Parameterkonvergenz des Funktionsapproximatoransatzes (links) und des Polynomansatzes (rechts)

vergieren können, wenn die Polynomkoeffizienten niedrigerer Ordnung konvergiert haben. Außerdem ist zu erkennen, dass der Polynomansatz zu Schwingungen im Verlauf der identifizierten Nichtlinearität führt. Einen weiteren gravierenden Nachteil des Polynomansatzes zeigt Abb. 5.6, wo das Extrapolationsverhalten beider Ansätze dargestellt ist. Während das GRNN konstant auf den Wert des letzten Stützwerthes extrapoliert, strebt das Polynom bereits bei geringem Überschreiten der Grenze zur Extrapolation gegen $\pm\infty$. Dieses Verhalten kann in der Praxis unangenehme Folgen haben.

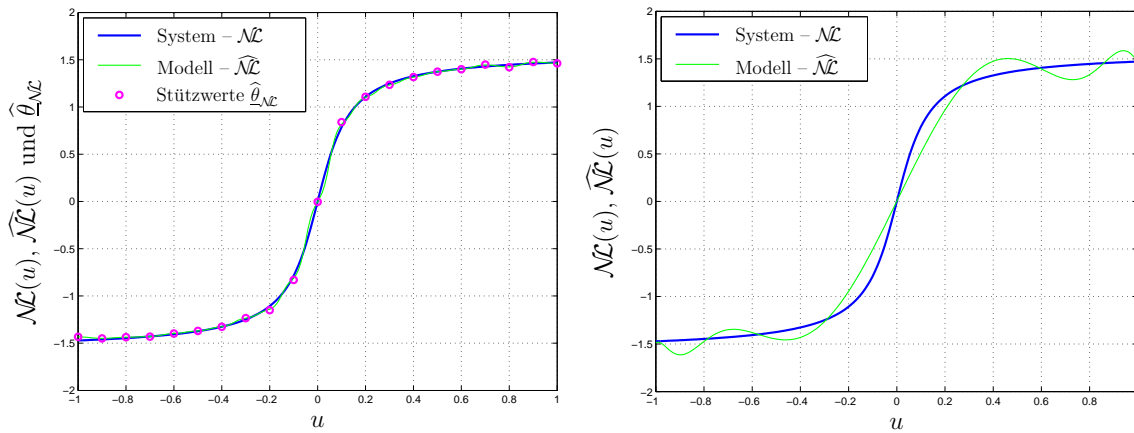


Abb. 5.5: Vergleich der identifizierten Nichtlinearitäten des Funktionsapproximatoransatzes (links) und des Polynomansatzes (rechts)

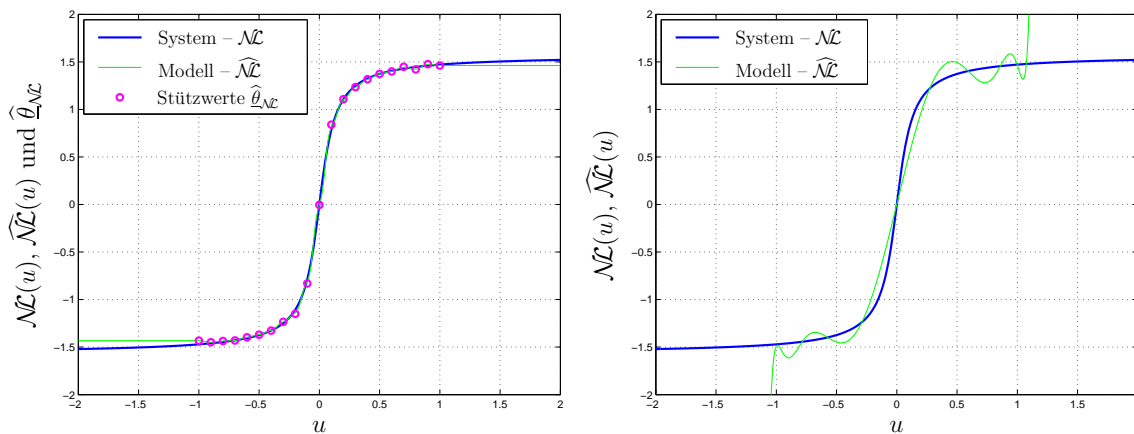


Abb. 5.6: Vergleich des Extrapolationsverhaltens der identifizierten Nichtlinearitäten des Funktionsapproximatoransatzes (links) und des Polynomansatzes (rechts)

Die Rekonstruktion der blockorientierten Modellstruktur, wie sie bereits zur Darstellung der statischen Nichtlinearitäten in Abb. 5.5, genutzt wurde, funktioniert entsprechend der Vorgehensweise in Kapitel 4.4.1.1. Der identifizierte Parametervektor hat folgende Struktur:

$$\underline{\hat{\theta}}^T = \left[\underline{\hat{\theta}}_1^T, \underline{\hat{\theta}}_2^T, \dots, \underline{\hat{\theta}}_q^T \right] \quad \underline{\hat{\theta}}_i \in \mathbb{R}^{m_r \times 1} \quad i = 1 \dots q \quad (5.10)$$

Zunächst werden aus dem Vektor $\underline{\hat{\theta}}_i$ und den in der Matrix $\tilde{\mathbf{R}}$ enthaltenen orthogonalen Basisfunktionen die einzelnen Gewichtsfolgen für jeden Stützwert zurückgerechnet:

$$\underline{\hat{g}}_1 = \tilde{\mathbf{R}}^T \underline{\hat{\theta}}_1 \quad \dots \quad \underline{\hat{g}}_q = \tilde{\mathbf{R}}^T \underline{\hat{\theta}}_q \quad (5.11)$$

Wird von der Annahme ausgegangen, dass der Verstärkungsfaktor des linearen Systems gleich Eins ist, d.h. $\sum_{i=1}^m h[i] = 1$, berechnen sich die Stützwerte des GRNN zu:

$$\hat{\theta}_{\mathcal{NL},1} = \sum_{i=1}^m \hat{g}_1[i] \quad \dots \quad \hat{\theta}_{\mathcal{NL},q} = \sum_{i=1}^m \hat{g}_q[i] \quad (5.12)$$

Die identifizierte Nichtlinearität kann rekonstruiert werden, indem das GRNN mit den nach Gl. (5.12) berechneten Stützwerten über dem Eingangsraum ausgewertet wird. Die Impulsantwort kann aus einer beliebigen Gewichtsfolge \hat{g}_j nach Gl. (5.5) bestimmt werden zu:

$$\hat{h}[i] = \frac{1}{\hat{\theta}_{\mathcal{NL},j}} \cdot \hat{g}_j[i] \quad j = 1 \dots q \quad (5.13)$$

Mit den Gleichungen (5.12) und (5.13) ist die blockorientierte Struktur des Modells vollständig rekonstruiert.

5.3 Erweiterung des Identifikationsverfahrens am Beispiel des Wiener-Modells

5.3.1 Beschreibung des Wiener-Modells mit Funktionsapproximatoransatz

Im Folgenden wird die Erweiterung des Identifikationsverfahrens mittels eines statischen Funktionsapproximators am Beispiel eines Wiener-Modells betrachtet. Zur Veranschaulichung ist das Wiener-Modell noch einmal in Abb. 5.7 dargestellt.

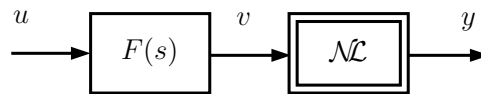


Abb. 5.7: Wiener-Modell

Das Wiener-Modell ist ein blockorientiertes Modell, das aus einer Übertragungsfunktion $F(s)$ gefolgt von einer statischen Nichtlinearität \mathcal{NL} besteht. Die Übertragungsfunktion wird wiederum durch die Faltungssumme und die statische Nichtlinearität aus den bereits aufgezeigten Gründen durch ein GRNN beschrieben:

$$v[k] = \sum_{i=1}^m h[i] u[k-i] \quad (5.14)$$

$$y[k] = \underline{\theta}_{\mathcal{NL}}^T \cdot \underline{\mathcal{A}}(v[k]) = \theta_{\mathcal{NL},1} \cdot \mathcal{A}_1(v[k]) + \dots + \theta_{\mathcal{NL},q} \cdot \mathcal{A}_q(v[k]) \quad (5.15)$$

Die Aktivierungsfunktionen des GRNN sind im Falle des Wiener-Modells vom Ausgangssignal v der Übertragungsfunktion abhängig. Sie können für den eindimensionalen Fall entsprechend Gl. (5.2) berechnet werden. Wird Gl. (5.14) in (5.15) eingesetzt, ergibt sich die Systembeschreibung für ein Wiener-Modell wie folgt:

$$y[k] = \theta_{\mathcal{N},1} \cdot \mathcal{A}_1 \left(\sum_{i=1}^m h[i] u[k-i] \right) + \dots + \theta_{\mathcal{N},q} \cdot \mathcal{A}_q \left(\sum_{i=1}^m h[i] u[k-i] \right) \quad (5.16)$$

Im Gegensatz zu Gl. (5.4) kann bei Gl. (5.16) keine Analogie zur Volterra-Funktionalpotenzreihe festgestellt werden. In Gl. (5.4) konnten die unbekannt Stützwerte des GRNN sowie die unbekannt Elemente der Impulsantwort des linearen Systems als Gewichtsfolgen interpretiert werden. Diese Gewichtsfolgen waren die unbekannt Parameter des Hammerstein-Modells, die linear in den Modellausgang eingingen. Bei der Beschreibung des Wiener-Modells nach Gl. (5.16) können die unbekannt Stützwerte des GRNN und die unbekannt Elemente der Impulsantwort nicht mehr in Form von Gewichtsfolgen identifiziert werden, da die unbekannt Elemente der Impulsantwort in die Berechnung der Aktivierungsfunktionen des GRNN und somit nichtlinear in das Ausgangssignal eingehen. Diese Tatsache soll noch einmal verdeutlicht werden, indem Gl. (5.2) für die Berechnung der Aktivierungsfunktionen des GRNN in Gl. (5.16) eingesetzt wird:

$$\begin{aligned} y[k] &= \theta_{\mathcal{N},1} \cdot \mathcal{A}_1 \left(\sum_{i=1}^m h[i] u[k-i] \right) + \dots + \theta_{\mathcal{N},q} \cdot \mathcal{A}_q \left(\sum_{i=1}^m h[i] u[k-i] \right) = \\ &= \theta_{\mathcal{N},1} \cdot \frac{\exp \left[\frac{(h[1] u[k-1] + \dots + h[m] u[k-m] - \chi_1)^2}{-2\sigma^2} \right]}{\sum_{j=1}^q \exp \left[\frac{(h[1] u[k-1] + \dots + h[m] u[k-m] - \chi_j)^2}{-2\sigma^2} \right]} + \dots + \\ &+ \theta_{\mathcal{N},q} \cdot \frac{\exp \left[\frac{(h[1] u[k-1] + \dots + h[m] u[k-m] - \chi_q)^2}{-2\sigma^2} \right]}{\sum_{j=1}^q \exp \left[\frac{(h[1] u[k-1] + \dots + h[m] u[k-m] - \chi_j)^2}{-2\sigma^2} \right]} \end{aligned} \quad (5.17)$$

Die unbekannt Stützwerte $\theta_{\mathcal{N},i}$ gehen zwar linear in das Ausgangssignal y ein, jedoch ist in Gl. (5.17) klar zu erkennen, dass die unbekannt Elemente der Impulsantwort $h[i]$ das Ausgangssignal y nichtlinear beeinflussen. Das bedeutet, dass die vorgestellte Theorie zur Volterra-Reihe auf dieses Identifikationsproblem nicht mehr angewendet werden kann. Dennoch ist eine Identifikation des Wiener-Modells mit einem Funktionsapproximatoransatz nach Gl. (5.17) möglich und soll im Folgenden vorgestellt werden [69].

5.3.1.1 Berechnung der Gradienten für das Lerngesetz

Aufgrund der Tatsache, dass die unbekannt Parameter nichtlinear in das Ausgangssignal eingehen, muss ein nichtlineares Adaptionverfahren aus Kapitel 2.7 eingesetzt werden. Als einfachstes nichtlineares Adaptionverfahren wird das Gradientenabstiegsverfahren nach Gl. (2.25) verwendet⁷. Die zeitdiskreten Adaptiongleichungen für die Parameter der statischen Nichtlinearität $\hat{\theta}_{\mathcal{NL},i}$ und der Übertragungsfunktion $\hat{\theta}_{lin,i}$ ergeben sich wie folgt:

$$\begin{aligned}\hat{\theta}_{\mathcal{NL},i}[k] &= \hat{\theta}_{\mathcal{NL},i}[k-1] - \eta_{\mathcal{NL}} \cdot e[k] \cdot \frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{\mathcal{NL},i}} & i = 1 \dots q \\ \hat{\theta}_{lin,i}[k] &= \hat{\theta}_{lin,i}[k-1] - \eta_{lin} \cdot e[k] \cdot \frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin,i}} & i = 1 \dots m\end{aligned}\tag{5.18}$$

Insgesamt sind $p = q + m$ unbekannte Parameter zu adaptieren. Dies erscheint auf den ersten Blick sehr wenig, allerdings ist zu bedenken, dass diese mit einem Gradientenabstiegsverfahren adaptiert werden müssen, das im Vergleich zu einem linearen Adaptionverfahren sehr langsam konvergiert.

Die Hauptaufgabe bei der Adaption der Parameter besteht in der Berechnung der Gradienten $\partial \hat{y}[k] / \partial \hat{\theta}_{\mathcal{NL},i}$ und $\partial \hat{y}[k] / \partial \hat{\theta}_{lin,i}$. Zunächst werden die partiellen Ableitungen des Ausgangs $\hat{y}[k]$ nach den Parametern der statischen Nichtlinearität $\hat{\theta}_{\mathcal{NL},i}$ bestimmt. Zu diesem Zweck wird die Systembeschreibung nach Gl. (5.17) in eine Modellbeschreibung umgewandelt.

$$\begin{aligned}\hat{y}[k] &= \hat{\theta}_{\mathcal{NL},1} \cdot \mathcal{A}_1(\hat{v}[k]) + \dots + \hat{\theta}_{\mathcal{NL},q} \cdot \mathcal{A}_q(\hat{v}[k]) = \\ &= \hat{\theta}_{\mathcal{NL},1} \cdot \frac{\exp\left[\frac{\left(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_1\right)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{\left(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_j\right)^2}{-2\sigma^2}\right]} + \dots + \\ &+ \hat{\theta}_{\mathcal{NL},q} \cdot \frac{\exp\left[\frac{\left(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_q\right)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{\left(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_j\right)^2}{-2\sigma^2}\right]}\end{aligned}\tag{5.19}$$

⁷Das Gradientenabstiegsverfahren wird auch für die unbekannt Stützwerte des GRNN eingesetzt, obwohl diese Parameter linear in das Ausgangssignal eingehen.

Für die partiellen Ableitungen nach den Parametern der statischen Nichtlinearität ergibt sich somit:

$$\frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{\mathcal{NL},i}} = \frac{\exp \left[\frac{(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_i)^2}{-2\sigma^2} \right]}{\sum_{j=1}^q \exp \left[\frac{(\hat{\theta}_{lin,1} \cdot u[k-1] + \dots + \hat{\theta}_{lin,m} \cdot u[k-m] - \chi_j)^2}{-2\sigma^2} \right]} = \mathcal{A}_i(\hat{v}[k]) \quad (5.20)$$

Die partiellen Ableitungen nach den Parametern der statischen Nichtlinearität sind in diesem Fall einfach die Aktivierungsfunktionen des GRNN mit dem Eingangssignal $\hat{v}[k]$. Die Berechnung der partiellen Ableitungen des Ausgangssignals $\hat{y}[k]$ nach den Parametern $\hat{\theta}_{lin,i}$ der Übertragungsfunktion gestaltet sich erheblich schwieriger. Durch Anwendung der Quotientenregel ergibt sich:

$$\begin{aligned} \frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin,i}} &= \sum_{j=1}^q \hat{\theta}_{\mathcal{NL},j} \cdot \left\{ - \frac{\exp \left[\frac{(\hat{v}[k] - \chi_j)^2}{-2\sigma^2} \right]}{\sum_{p=1}^q \exp \left[\frac{(\hat{v}[k] - \chi_p)^2}{-2\sigma^2} \right]} \cdot \frac{(\hat{v}[k] - \chi_j) \cdot u[k-i]}{\sigma^2} + \right. \\ &+ \left. \frac{\exp \left[\frac{(\hat{v}[k] - \chi_j)^2}{-2\sigma^2} \right]}{\sum_{p=1}^q \exp \left[\frac{(\hat{v}[k] - \chi_p)^2}{-2\sigma^2} \right]} \cdot \sum_{l=1}^q \frac{(\hat{v}[k] - \chi_l) \cdot u[k-i]}{\sigma^2} \cdot \frac{\exp \left[\frac{(\hat{v}[k] - \chi_l)^2}{-2\sigma^2} \right]}{\sum_{p=1}^q \exp \left[\frac{(\hat{v}[k] - \chi_p)^2}{-2\sigma^2} \right]} \right\} \end{aligned} \quad (5.21)$$

Werden in Gl. (5.21) die Aktivierungsfunktionen des GRNN eingesetzt, kann wie folgt vereinfacht werden:

$$\frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin,i}} = \sum_{j=1}^q \hat{\theta}_{\mathcal{NL},j} \cdot \left\{ -\mathcal{A}_j \cdot \frac{(\hat{v}[k] - \chi_j) \cdot u[k-i]}{\sigma^2} + \mathcal{A}_j \cdot \sum_{l=1}^q \frac{(\hat{v}[k] - \chi_l) \cdot u[k-i]}{\sigma^2} \cdot \mathcal{A}_l \right\} \quad (5.22)$$

Durch weitere Umformungen ergibt sich letztendlich für die partiellen Ableitungen nach den Parametern der Übertragungsfunktion folgender Ausdruck:

$$\frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin,i}} = \frac{u[k-i]}{\sigma^2} \cdot \sum_{j=1}^q \hat{\theta}_{\mathcal{NL},j} \cdot \mathcal{A}_j \cdot \left\{ \sum_{l=1}^q [(\hat{v}[k] - \chi_l) \cdot \mathcal{A}_l] - (\hat{v}[k] - \chi_j) \right\} \quad (5.23)$$

Mit den Gleichungen (5.18) und (5.20) sowie (5.23) sind die Lerngesetze für die Parameter der statischen Nichtlinearität und die Parameter der Übertragungsfunktion vollständig gegeben. Trotz der auf den ersten Blick geringen Parameteranzahl von $p = q + m$ kommt es aufgrund des Gradientenabstiegsverfahrens zu extrem langen Konvergenzzeiten, so dass eine Parameterreduktion durch die Einführung orthonormaler Basisfunktionen erforderlich ist.

5.3.1.2 Berechnung der Gradienten bei Einführung von Basisfunktionen

Die Einführung orthonormaler Basisfunktionen ändert nichts an der Tatsache, dass die unbekannt Parameter nichtlinear in das Ausgangssignal eingehen. Die Adaption der unbekannt Parameter erfolgt wieder nach dem Gradientenabstiegsverfahren. Die zeitdiskreten Adaptionsgleichungen für die Parameter der statischen Nichtlinearität $\widehat{\theta}_{\mathcal{NL},i}$ und der Übertragungsfunktion $\widehat{\theta}_{lin,i}$ ergeben sich wie folgt:

$$\begin{aligned}\widehat{\theta}_{\mathcal{NL},i}[k] &= \widehat{\theta}_{\mathcal{NL},i}[k-1] - \eta_{\mathcal{NL}} \cdot e[k] \cdot \frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{\mathcal{NL},i}} & i = 1 \dots q \\ \widehat{\theta}_{lin,i}[k] &= \widehat{\theta}_{lin,i}[k-1] - \eta_{lin} \cdot e[k] \cdot \frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{lin,i}} & i = 1 \dots m_r\end{aligned}\quad (5.24)$$

Durch die Einführung orthonormaler Basisfunktionen müssen anstatt m nur noch m_r lineare Parameter bestimmt werden. Die Gesamtparameteranzahl reduziert sich somit auf $p = q + m_r$. Die Hauptaufgabe besteht wiederum in der Berechnung der Gradienten. Für die partiellen Ableitungen nach den Parametern der statischen Nichtlinearität ergibt sich bei Einführung orthonormaler Basisfunktionen:

$$\frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{\mathcal{NL},i}} = \frac{\exp \left[\frac{\left(\widehat{\theta}_{lin,1} \cdot \underline{u}^T \cdot \tilde{r}_1 + \dots + \widehat{\theta}_{lin,m_r} \cdot \underline{u}^T \cdot \tilde{r}_{m_r} - \chi_i \right)^2}{-2\sigma^2} \right]}{\sum_{j=1}^q \exp \left[\frac{\left(\widehat{\theta}_{lin,1} \cdot \underline{u}^T \cdot \tilde{r}_1 + \dots + \widehat{\theta}_{lin,m_r} \cdot \underline{u}^T \cdot \tilde{r}_{m_r} - \chi_j \right)^2}{-2\sigma^2} \right]} = \mathcal{A}_i(\widehat{v}[k]) \quad (5.25)$$

Die partiellen Ableitungen für die Parameter der statischen Nichtlinearität ändern sich bis auf die Berechnung von $\widehat{v}[k]$ nicht. Die Berechnung von $\widehat{v}[k]$ erfolgt in diesem Fall mit Hilfe der gewichteten Überlagerung von Basisfunktionen.

Die partiellen Ableitungen nach den Parametern der Übertragungsfunktion berechnen sich ausgehend von Gl. (5.23) bei Einführung orthonormaler Basisfunktionen wie folgt:

$$\frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{lin,i}} = \frac{\underline{u}^T \cdot \tilde{r}_i}{\sigma^2} \cdot \sum_{j=1}^q \widehat{\theta}_{\mathcal{NL},j} \cdot \mathcal{A}_j \cdot \left\{ \sum_{l=1}^q \left[(\widehat{v}[k] - \chi_l) \cdot \mathcal{A}_l \right] - (\widehat{v}[k] - \chi_j) \right\} \quad (5.26)$$

$\widehat{v}[k]$ wird entsprechend Gl. (5.25) berechnet. Wird Gl. (5.26) mit Gl. (5.23) verglichen, ist lediglich ein Unterschied bei der Berechnung des Terms vor der ersten Summe zu erkennen. Der entscheidende Punkt ist jedoch, dass die Parameteranzahl durch die Einführung der Basisfunktionen von m auf m_r reduziert werden konnte, was letztendlich zu deutlich kürzeren Konvergenzzeiten der Parameter führt.

5.3.2 Beispiel — Wiener-Modell mit Funktionsapproximatoransatz

Nachdem in Kapitel 5.3.1 die mathematischen Grundlagen zur Identifikation des Wiener-Modells mit einem Funktionsapproximatoransatz gelegt wurden, sollen die theoretischen Ausführungen in diesem Kapitel an einem Simulationsbeispiel veranschaulicht werden. Gegeben ist das folgende Wiener-Modell:

$$F(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1} \quad \mathcal{NL}(v) = \arctan(10 \cdot v)$$

Bevor zu den Identifikationsergebnissen übergegangen wird, soll an dieser Stelle noch auf ein Problem hingewiesen werden, welches sich bei der Identifikation des Wiener-Modells ergibt. Aufgrund der Tatsache, dass der statische Nichtlinearität die Übertragungsfunktion vorgeschaltet ist, ist es sehr schwierig, die statische Nichtlinearität und infolge dessen auch das GRNN zur Approximation dieser statischen Nichtlinearität im Randbereich des vorgegebenen Eingangsbereiches anzuregen. Zur Lösung dieses Problems wurde für das GRNN eine Anhebungslogik entwickelt, die aus dem momentanen Identifikationsergebnis des GRNN im angeregten Eingangsbereich auf den Verlauf der zu approximierenden Nichtlinearität im nicht angeregten Eingangsbereich schließt. Die genaue Funktion dieser Anhebungslogik kann in [1] nachgelesen werden. An dieser Stelle soll nur kurz auf die prinzipielle Funktionsweise eingegangen werden. Zunächst wird für jede Aktivierungsfunktion \mathcal{A}_i des GRNN festgestellt, ob sie eine gewisse Mindestaktivierung überschritten hat, d.h. $\mathcal{A}_i > \mathcal{A}_{min}$. In einem zusätzlichen Vektor, der für jede Aktivierungsfunktion die logischen Werte Null oder Eins enthalten kann, wird das Überschreiten der Mindestaktivierung durch eine Eins gekennzeichnet und für den Rest der Adaption vermerkt. Anschaulich bedeutet die Einführung der Mindestaktivierung, dass bei der Adaption der Parameter detektiert wird, ob das Eingangssignal des GRNN im Bereich des Zentrums der Aktivierungsfunktion \mathcal{A}_i war und der zugehörige Stützwert somit als aktiviert und gelernt betrachtet werden kann. Durch die Kenntnis der aktivierten und nicht aktivierten Stützwerte kann für die nicht aktivierten Stützwerte auch ein plausiblerer Wert als der beliebig vorgegebene Anfangswert gefunden werden. Im Falle, dass die nicht aktivierten Stützwerte im Randbereich des Eingangsbereiches liegen, können die nicht aktivierten Stützwerte z.B. auf den Wert des nächstgelegenen aktivierten Stützwertes gesetzt werden. In [1] wurden noch aufwändigere Anhebungsverfahren entwickelt, die aus der Kenntnis von zwei bzw. drei nächstgelegenen Stützwerten, die nicht aktivierten Stützwerte linear bzw. unter Berücksichtigung des Steigungs- und Krümmungsverhaltens anheben. Das Anhebungsverfahren, das sowohl das Steigungs- als auch das Krümmungsverhalten der im angeregten Eingangsbereich approximierten Nichtlinearität berücksichtigt, kommt im Folgenden zum Einsatz [69].

Für die Identifikation werden die Einstellwerte nach Tab. 5.2 verwendet.

Einstellwerte	Erläuterung
$q = 41$	Stützstellenanzahl
$\sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 2 \cdot 50 = 100$	Antwortlänge
$m_r = 20$	Basisfunktionenanzahl
$\zeta = 2 \cdot 10.65 = 21.3$	Formfaktor
$p = q + m_r = 61$	Parameteranzahl
$\eta_{lin} = 8.0 \cdot 10^{-5}$	Lernschrittweite für $\hat{\theta}_{lin,i}$
$\eta_{NL} = 5.0 \cdot 10^{-3}$	Lernschrittweite für $\hat{\theta}_{NL,i}$
$h = 0.5s$	Abtastzeit

Tabelle 5.2: Einstellwerte für die Identifikation des Wiener-Modells

Aus Gründen der Genauigkeit des Identifikationsergebnisses wird die Abtastzeit auf $0.5s$ reduziert. Die Antwortlänge m und der Formfaktor ζ der Basisfunktionen verdoppeln sich dadurch im Vergleich zu den vorangegangenen Identifikationsbeispielen. Die Simulationsdauer beträgt insgesamt $7 \cdot 10^6$ Zeitschritte. Nach $6.4 \cdot 10^6$ Zeitschritten wird die Parameteradaption eingestellt und das identifizierte Modell wird als Parallelmodell betrieben. Das System wird durch ein Zufallssignal im Bereich $[-1; 1]$ angeregt. In Abb. 5.8 links ist der Fehlerverlauf dargestellt, welcher auf der rechten Seite noch einmal zu Lernbeginn veranschaulicht ist.

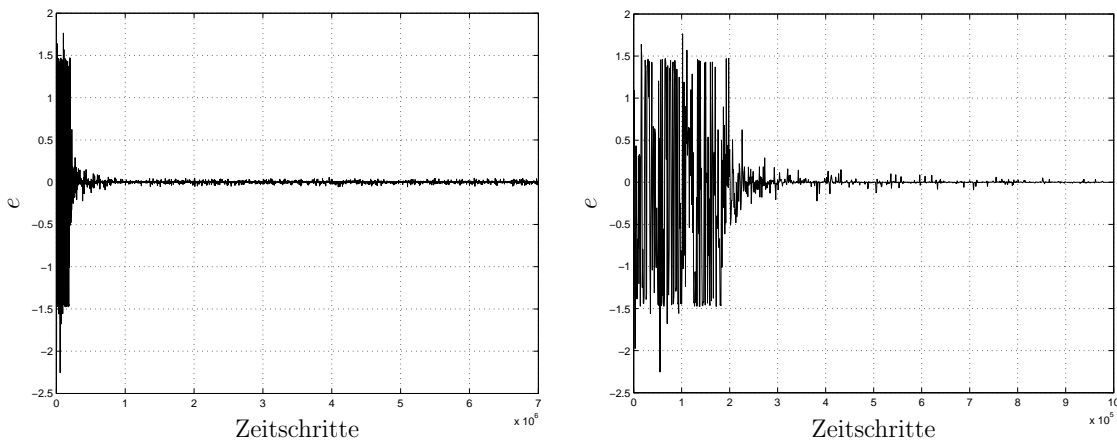


Abb. 5.8: Fehlerverlauf des Wiener-Modells mit Funktionsapproximatoransatz (links) und Vergrößerung zu Lernbeginn (rechts)

Es ist deutlich zu erkennen, dass der Fehler e klein wird und auch im Parallelbetrieb nicht merklich zunimmt. Die Anhebungslogik, die bei ca. $2 \cdot 10^5$ Zeitschritten einsetzt, wirkt sich sehr positiv auf die Adaptionsgeschwindigkeit der Parameter und somit auf den Fehlerverlauf aus, wie in Abb. 5.8 rechts zu sehen ist. In Abb. 5.9 ist das Konvergenzverhalten der Parameter der statischen Nichtlinearität (links) und der Übertragungsfunktion (rechts) dargestellt. Auch bei der Parameterkonvergenz

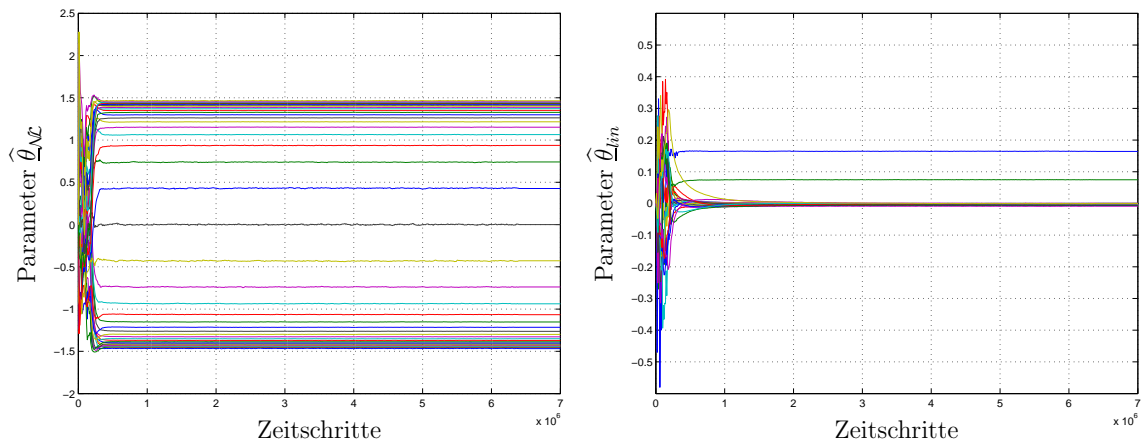


Abb. 5.9: Konvergenz der Parameter der statischen Nichtlinearität (links) und der Übertragungsfunktion (rechts)

ist der positive Einfluss der Anhebungslogik zu erkennen. Die identifizierte statische Nichtlinearität und die identifizierte Impulsantwort, die mit der Basisfunktionenmatrix rekonstruiert werden kann, sind in Abb. 5.10 gezeigt.

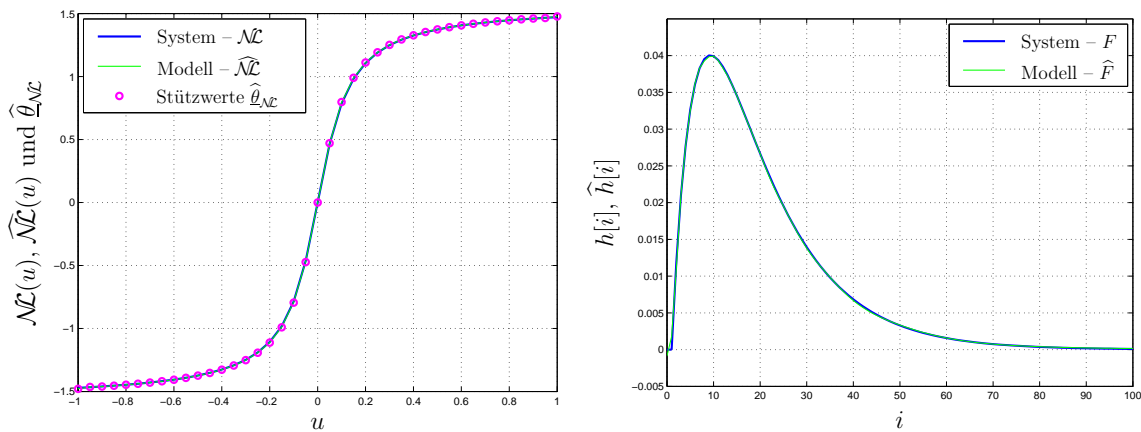


Abb. 5.10: Identifizierte Nichtlinearität (links) und Impulsantwort (rechts)

Die Identifikationsergebnisse verdeutlichen, dass auch das Wiener-Modell mit dem Funktionsapproximatoransatz gelernt werden kann. Allerdings zeigte sich, dass sich die Systembeschreibung nicht mehr auf die Volterra-Funktionalpotenzreihe zurückführen lässt. Dies ist darin begründet, dass die unbekannt Parameter nichtlinear in das Ausgangssignal eingehen.

5.4 Erweiterung des Funktionsapproximatoransatzes auf Mehrgrößensysteme

In den Kapiteln 5.2 und 5.3 wurden die Vorteile, die sich durch die Approximation der statischen Nichtlinearität durch ein GRNN anstatt eines Polynoms ergeben, ausführlich dargelegt. Die Simulationsbeispiele haben gezeigt, dass sowohl das Hammerstein- als auch das Wiener-Modell mit hoher Genauigkeit identifiziert werden kann. Die Adaptionsverfahren sind jedoch aufgrund der Tatsache, dass beim Wiener-Modell die Parameter nichtlinear in das Ausgangssignal eingehen unterschiedlich.

Im Folgenden soll beispielhaft gezeigt werden, dass die vorgestellten Verfahren zur Identifikation eines Hammerstein- bzw. Wiener-Modells auch auf Mehrgrößensysteme, wie sie in Kapitel 4 für den Polynomansatz ausführlich behandelt wurden, angewendet werden können.

5.4.1 Identifikation von nichtlinear verkoppelten Hammerstein-Modellen

Das im Folgenden betrachtete System aus zwei nichtlinear verkoppelten Hammerstein-Modellen ist in Abb. 5.11 dargestellt [19].

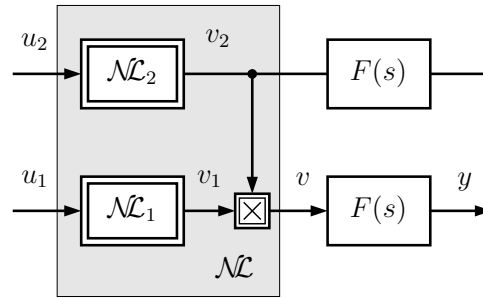


Abb. 5.11: Hammerstein-Modell mit zweidimensionaler Nichtlinearität

Die statische Nichtlinearität $\mathcal{NL}(u_1, u_2) = \mathcal{NL}_1(u_1) \cdot \mathcal{NL}_2(u_2)$ kann in diesem Fall durch ein zweidimensionales GRNN approximiert werden. Für die statische Nichtlinearität gilt:

$$\mathcal{NL}(\underline{u}) = \underline{\theta}_{\mathcal{NL}}^T \cdot \underline{\mathcal{A}}(\underline{u}) = \theta_{\mathcal{NL},1} \cdot \mathcal{A}_1(\underline{u}) + \dots + \theta_{\mathcal{NL},q} \cdot \mathcal{A}_q(\underline{u}) \quad (5.27)$$

Die Nichtlinearität wird durch ein GRNN mit q_{1D} Stützwerten je Eingangsdimension beschrieben. Die gesamte Anzahl an Stützwerten ist somit $q = q_{1D}^2$. Die Aktivierungsfunktionen $\mathcal{A}_1(\underline{u}) \dots \mathcal{A}_q(\underline{u})$ berechnen sich für den zweidimensionalen Fall wie

folgt:

$$\mathcal{A}_i(\underline{u}) = \frac{\exp\left[-\frac{1}{2}\left(\frac{(u_1 - \chi_{1,i})^2}{\sigma_1^2} + \frac{(u_2 - \chi_{2,i})^2}{\sigma_2^2}\right)\right]}{\sum_{j=1}^q \exp\left[-\frac{1}{2}\left(\frac{(u_1 - \chi_{1,j})^2}{\sigma_1^2} + \frac{(u_2 - \chi_{2,j})^2}{\sigma_2^2}\right)\right]} \quad (5.28)$$

Wird Gl. (5.27) in die Faltungssumme nach Gl. (5.3) eingesetzt, ergibt sich als Systembeschreibung:

$$y[k] = \theta_{\mathcal{NL},1} \sum_{i=1}^m h[i] \mathcal{A}_1(\underline{u}[k-i]) + \dots + \theta_{\mathcal{NL},q} \sum_{i=1}^m h[i] \mathcal{A}_q(\underline{u}[k-i]) \quad (5.29)$$

Im Vergleich zu der Systembeschreibung des Hammerstein-Modells nach Gl. (5.4) sind in diesem Fall die Aktivierungsfunktionen von einem Eingangsvektor \underline{u} abhängig, der die Komponenten u_1 und u_2 enthält. Die Gewichtsfolgen ergeben sich entsprechend Gl. (5.5). Somit kann die Volterra-Reihe für diesen Fall wie folgt formuliert werden:

$$y[k] = \sum_{i=1}^m g_1[i] \mathcal{A}_1(\underline{u}[k-i]) + \dots + \sum_{i=1}^m g_q[i] \mathcal{A}_q(\underline{u}[k-i]) \quad (5.30)$$

Die Anzahl der unbekannt Parameter beträgt in Gl. (5.30) wiederum $p = q \cdot m$, wobei sich q aus dem Produkt der Stützwerteanzahlen je Dimension ergibt. Nach der Einführung orthonormaler Basisfunktionen zum Zwecke der Parameterreduktion folgt:

$$y[k] = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (5.31)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[\underline{\mathcal{A}}_1^T[k] \cdot \tilde{\mathbf{R}}^T, \underline{\mathcal{A}}_2^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \tilde{\mathbf{R}}^T \right] \\ \underline{\mathcal{A}}_i^T[k] &= \left[\mathcal{A}_i(\underline{u}[k-1]), \mathcal{A}_i(\underline{u}[k-2]), \dots, \mathcal{A}_i(\underline{u}[k-m]) \right] \\ \underline{\theta}^T &= \left[g_{r,1}[1], \dots, g_{r,1}[m_r], g_{r,2}[1], \dots, g_{r,2}[m_r], \dots, g_{r,q}[m_r] \right] \end{aligned}$$

Die Vektoren $\underline{u}[k-i]$, die in die Berechnung der Aktivierungsfunktionen eingehen, enthalten die Eingangssignale $u_1[k-i]$ und $u_2[k-i]$. Die Struktur des Parametervektors $\underline{\theta}$ und die Dimensionen der einzelnen Vektoren und Matrizen sind analog zu den Gleichungen (5.8) und (5.9).

Die Erweiterung des Identifikationsansatzes auf MISO-Systeme wird anhand eines Simulationsbeispiels überprüft. Betrachtet wird das Hammerstein-Modell nach Abb. 5.11. Für die zweidimensionale statische Nichtlinearität und für die Übertragungsfunktion gilt:

$$\mathcal{NL}(\underline{u}) = \arctan(5 \cdot u_1) \cdot \arctan(5 \cdot u_2) \quad F(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1}$$

Die statische Nichtlinearität wurde bewusst so gewählt, dass sie unter keinen Umständen durch ein Polynom zufriedenstellend approximiert werden kann.

Für die Identifikation werden die Einstellwerte nach Tabelle 5.3 verwendet.

Einstellwerte	Erläuterung
$q = 11^2 = 121$	Stützstellenanzahl
$\sigma_{norm1} = \sigma_{norm2} = \sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 50$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.65$	Formfaktor
$p = m_r \cdot q = 1210$	Parameteranzahl
$h = 1s$	Abtastzeit

Tabelle 5.3: *Einstellwerte für die Identifikation eines Hammerstein-Modells mit zweidimensionaler statischer Nichtlinearität*

Die Simulationsdauer beträgt insgesamt $1 \cdot 10^5$ Zeitschritte. Nach $9.5 \cdot 10^4$ Zeitschritten wird die Parameteradaption eingestellt und das identifizierte Modell wird als reines Parallelmodell betrieben. Das System wird durch ein Zufallssignal im Bereich $[-1; 1]$ angeregt. Als Adaptionsverfahren kommt wiederum der rekursive Least-Squares-Algorithmus zum Einsatz. In Abb. 5.12 ist der Fehlerverlauf (links) und die Konvergenz der Parameter (rechts) dargestellt.

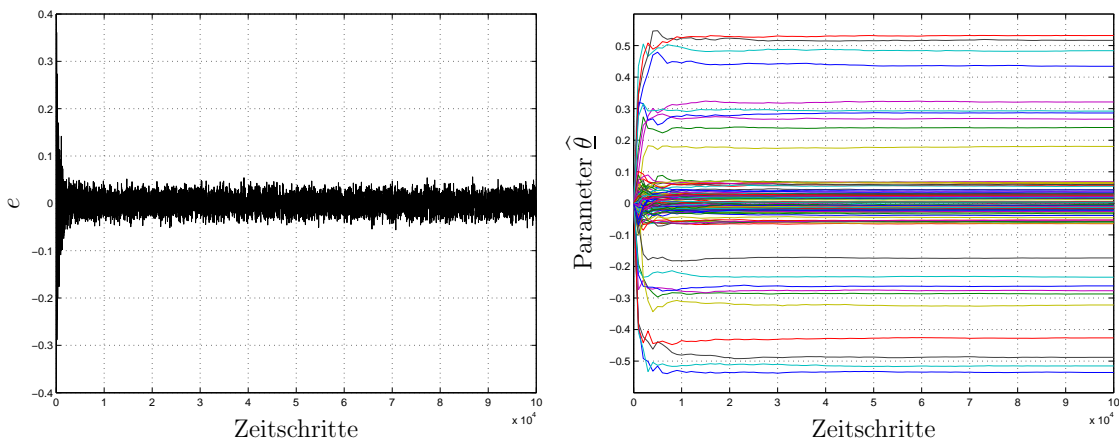


Abb. 5.12: *Fehlerverlauf des Hammerstein-Modells mit zweidimensionaler Nichtlinearität (links) und Konvergenz von ausgewählten Parametern (rechts)*

Es ist zu erkennen, dass der Fehler e klein wird und die Parameter konvergieren. Auch im Parallelbetrieb steigt der Fehler nicht merklich an. Allerdings bleibt aufgrund der relativ geringen Anzahl an Basisfunktionen und der begrenzten Stützstellenanzahl ein Restfehler. In Abb. 5.13 sind die vorgegebene und die identifizierte statische Nichtlinearität einander gegenübergestellt. Die Rückrechnung der blockorientierten Struktur erfolgte analog zu Kapitel 5.2.

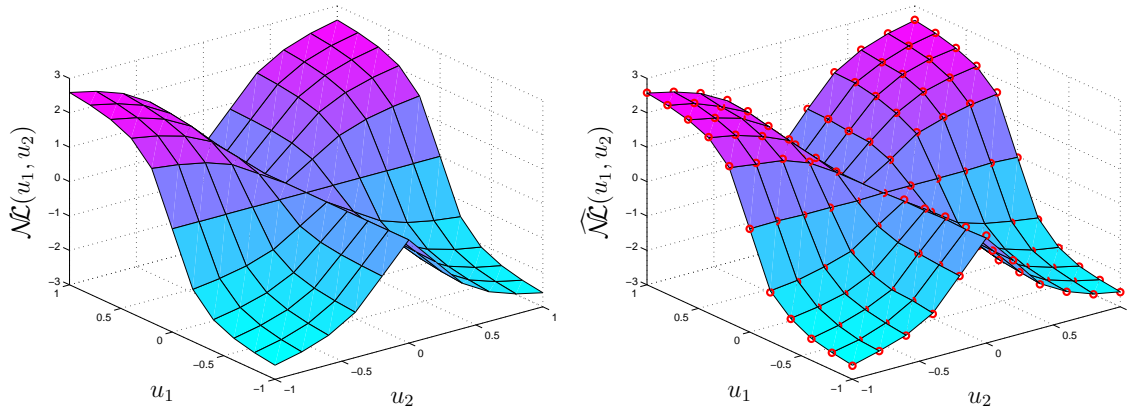


Abb. 5.13: Vergleich zwischen vorgegebener statischer Nichtlinearität (links) und identifizierter statischer Nichtlinearität (rechts)

Die statische Nichtlinearität wird durch das GRNN sehr gut identifiziert. Ein Polynom wäre in diesem Fall nicht annähernd in der Lage, die statische Nichtlinearität zufriedenstellend zu approximieren. In Abb. 5.14 links ist der Approximationsfehler der statischen Nichtlinearität und rechts die identifizierte Impulsantwort veranschaulicht.

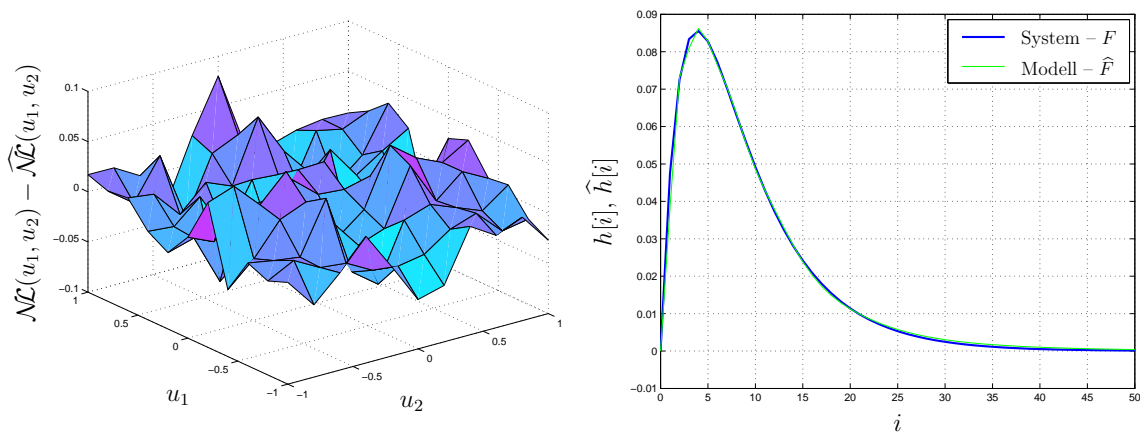


Abb. 5.14: Identifikationsfehler der statischen Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

Das Identifikationsbeispiel zeigt, dass der Funktionsapproximatoransatz auch im Falle von MISO–Hammerstein–Modellen anwendbar ist und sehr gute Ergebnisse erzielt werden können. Allerdings muss die Stützwerteanzahl und die Anzahl an orthonormierten Basisfunktionen sorgsam gewählt werden, da ansonsten die Gesamtparameteranzahl zu groß wird.

5.4.2 Identifikation von nichtlinear verkoppelten Wiener-Modellen

In diesem Kapitel wird das Identifikationsverfahren mit GRNN zur Approximation der statischen Nichtlinearität auf Wiener-Modelle mit mehr als einer Eingangsgröße erweitert [69]. Beispielhaft werden zwei nichtlinear verkoppelte Wiener-Modelle, wie sie in Abb. 5.15 dargestellt sind, betrachtet.

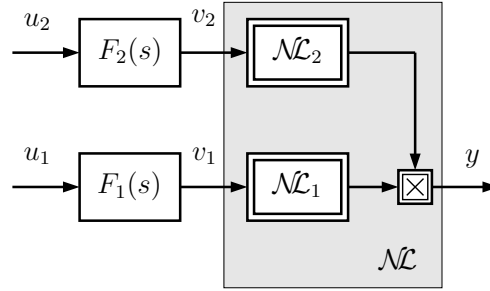


Abb. 5.15: Wiener-Modelle mit zweidimensionaler Nichtlinearität

Die Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ werden durch Faltungssummen und die statischen Nichtlinearitäten \mathcal{N}_1 und \mathcal{N}_2 durch ein zweidimensionales GRNN beschrieben. Es gilt:

$$v_1[k] = \sum_{i=1}^m h_1[i] u_1[k-i] \quad v_2[k] = \sum_{i=1}^m h_2[i] u_2[k-i] \quad (5.32)$$

$$y[k] = \underline{\theta}_{\mathcal{N}}^T \cdot \underline{\mathcal{A}}(\underline{v}[k]) = \theta_{\mathcal{N},1} \cdot \mathcal{A}_1(\underline{v}[k]) + \dots + \theta_{\mathcal{N},q} \cdot \mathcal{A}_q(\underline{v}[k]) \quad (5.33)$$

Die Aktivierungsfunktionen $\mathcal{A}_1(\underline{v}[k]) \dots \mathcal{A}_q(\underline{v}[k])$ des GRNN sind in diesem Fall von den Ausgangssignalen $v_1[k]$ und $v_2[k]$ der Übertragungsfunktionen abhängig und berechnen sich zu:

$$\mathcal{A}_i(\underline{v}[k]) = \frac{\exp \left[-\frac{1}{2} \left(\frac{(v_1[k] - \chi_{1,i})^2}{\sigma_1^2} + \frac{(v_2[k] - \chi_{2,i})^2}{\sigma_2^2} \right) \right]}{\sum_{j=1}^q \exp \left[-\frac{1}{2} \left(\frac{(v_1[k] - \chi_{1,j})^2}{\sigma_1^2} + \frac{(v_2[k] - \chi_{2,j})^2}{\sigma_2^2} \right) \right]} \quad (5.34)$$

Die Stützwerteanzahl der zweidimensionalen statischen Nichtlinearität ergibt sich wie in Kapitel 5.4.1 aus dem Produkt der Stützwerteanzahlen der einzelnen Dimensionen, d.h. bei gleicher Anzahl je Dimension zu $q = q_{1D}^2$.

Aus den Gleichungen (5.32) bis (5.34) ist zu erkennen, dass auch im mehrdimensionalen Fall die unbekannt Elemente der Impulsantworten nichtlinear in das Ausgangssignal y eingehen. Aus diesem Grund wird analog zu Kapitel 5.3 auf das Gradientenabstiegsverfahren zurückgegriffen. Die Aufgabe besteht wiederum in der Berechnung der partiellen Ableitungen für die Parameter der Übertragungsfunktionen und der statischen Nichtlinearität. Bei Einführung orthonormaler Basisfunktionen

ergeben sich die zeitdiskreten Adaptionsgleichungen für die Parameter der statischen Nichtlinearität $\widehat{\theta}_{\mathcal{NL},i}$ und der Übertragungsfunktionen $\widehat{\theta}_{lin1,i}$ sowie $\widehat{\theta}_{lin2,i}$ wie folgt:

$$\begin{aligned}\widehat{\theta}_{\mathcal{NL},i}[k] &= \widehat{\theta}_{\mathcal{NL},i}[k-1] - \eta_{\mathcal{NL}} \cdot e[k] \cdot \frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{\mathcal{NL},i}} & i = 1 \dots q \\ \widehat{\theta}_{lin1,i}[k] &= \widehat{\theta}_{lin1,i}[k-1] - \eta_{lin1} \cdot e[k] \cdot \frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{lin1,i}} & i = 1 \dots m_r \\ \widehat{\theta}_{lin2,i}[k] &= \widehat{\theta}_{lin2,i}[k-1] - \eta_{lin2} \cdot e[k] \cdot \frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{lin2,i}} & i = 1 \dots m_r\end{aligned}\quad (5.35)$$

Insgesamt sind $p = q + 2 \cdot m_r$ unbekannte Parameter zu adaptieren. Die Berechnung der partiellen Ableitungen wird mit der folgenden Modellbeschreibung durchgeführt:

$$\begin{aligned}\widehat{y}[k] &= \widehat{\theta}_{\mathcal{NL},1} \cdot \mathcal{A}_1(\widehat{v}[k]) + \dots + \widehat{\theta}_{\mathcal{NL},q} \cdot \mathcal{A}_q(\widehat{v}[k]) = \\ &= \widehat{\theta}_{\mathcal{NL},1} \cdot \frac{\exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,1})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,1})^2}{\sigma_2^2} \right)\right]}{\sum_{j=1}^q \exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,j})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,j})^2}{\sigma_2^2} \right)\right]} + \dots + \\ &+ \widehat{\theta}_{\mathcal{NL},q} \cdot \frac{\exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,q})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,q})^2}{\sigma_2^2} \right)\right]}{\sum_{j=1}^q \exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,j})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,j})^2}{\sigma_2^2} \right)\right]}\end{aligned}\quad (5.36)$$

Mit den Eingangssignalen $\widehat{v}_1[k]$ und $\widehat{v}_2[k]$ der Aktivierungsfunktionen:

$$\widehat{v}_1[k] = \widehat{\theta}_{lin1,1} \cdot \underline{u}_1^T \cdot \tilde{\mathbf{r}}_1 + \dots + \widehat{\theta}_{lin1,m_r} \cdot \underline{u}_1^T \cdot \tilde{\mathbf{r}}_{m_r} \quad (5.37)$$

$$\widehat{v}_2[k] = \widehat{\theta}_{lin2,1} \cdot \underline{u}_2^T \cdot \tilde{\mathbf{r}}_1 + \dots + \widehat{\theta}_{lin2,m_r} \cdot \underline{u}_2^T \cdot \tilde{\mathbf{r}}_{m_r} \quad (5.38)$$

Die Vektoren \underline{u}_1 und \underline{u}_2 enthalten m Vergangenheitswerte von den Eingangssignalen u_1 und u_2 .

Die partiellen Ableitungen nach den Parametern der statischen Nichtlinearität ergeben sich somit zu:

$$\frac{\partial \widehat{y}[k]}{\partial \widehat{\theta}_{\mathcal{NL},i}} = \frac{\exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,i})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,i})^2}{\sigma_2^2} \right)\right]}{\sum_{j=1}^q \exp\left[-\frac{1}{2} \left(\frac{(\widehat{v}_1[k] - \chi_{1,j})^2}{\sigma_1^2} + \frac{(\widehat{v}_2[k] - \chi_{2,j})^2}{\sigma_2^2} \right)\right]} = \mathcal{A}_i(\widehat{v}[k]) \quad (5.39)$$

Die partiellen Ableitungen des Ausgangssignals $\widehat{y}[k]$ nach den Parametern $\widehat{\theta}_{lin1,i}$ sowie $\widehat{\theta}_{lin2,i}$ der Übertragungsfunktionen berechnen sich bei analoger Vorgehensweise

wie in den Kapitel 5.3.1.1 und 5.3.1.2 zu:

$$\frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin1,i}} = \frac{\underline{u}_1^T \cdot \tilde{r}_i}{\sigma_1^2} \cdot \sum_{j=1}^q \hat{\theta}_{\mathcal{NL},j} \cdot \mathcal{A}_j \cdot \left\{ \sum_{l=1}^q [(\hat{v}_1[k] - \chi_{1,l}) \cdot \mathcal{A}_l] - (\hat{v}_1[k] - \chi_{1,j}) \right\} \quad (5.40)$$

$$\frac{\partial \hat{y}[k]}{\partial \hat{\theta}_{lin2,i}} = \frac{\underline{u}_2^T \cdot \tilde{r}_i}{\sigma_2^2} \cdot \sum_{j=1}^q \hat{\theta}_{\mathcal{NL},j} \cdot \mathcal{A}_j \cdot \left\{ \sum_{l=1}^q [(\hat{v}_2[k] - \chi_{2,l}) \cdot \mathcal{A}_l] - (\hat{v}_2[k] - \chi_{2,j}) \right\} \quad (5.41)$$

Mit den Gleichungen (5.35) bis (5.41) sind die Lerngesetze für die Parameter der statischen Nichtlinearität und die Parameter der Übertragungsfunktionen vollständig gegeben.

Die Erweiterung des vorgestellten Identifikationsansatzes auf MISO-Systeme wird anhand eines Simulationsbeispiels überprüft. Betrachtet wird das MISO-System nach Abb. 5.15. Für die zweidimensionale statische Nichtlinearität und für die Übertragungsfunktionen gilt:

$$F_1(s) = \frac{1}{21 \cdot s^2 + 10 \cdot s + 1} \quad F_2(s) = \frac{1}{32 \cdot s^2 + 8 \cdot s + 1}$$

$$\mathcal{NL}(\underline{v}) = \arctan(5 \cdot v_1) \cdot \arctan(5 \cdot v_2)^2$$

Die Übertragungsfunktion $F_1(s)$ besitzt zwei reelle Pole und die Übertragungsfunktion $F_2(s)$ ein konjugiert komplexes Polpaar. Für die Identifikation werden die Einstellwerte nach Tabelle 5.4 verwendet.

Einstellwerte	Erläuterung
$q = 11^2 = 121$	Stützstellenanzahl
$\sigma_{norm1} = \sigma_{norm2} = \sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 50$	Antwortlänge
$m_r = 8$	Basisfunktionenanzahl
$\zeta = 11.85$	Formfaktor
$p = 2 \cdot m_r + q = 137$	Parameteranzahl
$\eta_{lin1} = \eta_{lin2} = 1.0 \cdot 10^{-5}$	Lernschrittweite für $\hat{\theta}_{lin1,i}$ und $\hat{\theta}_{lin2,i}$
$\eta_{\mathcal{NL}} = 5.0 \cdot 10^{-4}$	Lernschrittweite für $\hat{\theta}_{\mathcal{NL},i}$
$h = 1s$	Abtastzeit

Tabelle 5.4: Einstellwerte für die Identifikation von zwei nichtlinear verkoppelten Wiener-Modellen mit zweidimensionalem GRNN zur Approximation der statischen Nichtlinearitäten

Die Simulationsdauer beträgt insgesamt $5 \cdot 10^7$ Zeitschritte. Nach $4.8 \cdot 10^7$ Zeitschritten wird die Parameteradaption eingestellt und das identifizierte Modell wird als Parallelmodell betrieben. Das System wird durch Zufallssignale im Bereich $[-1; 1]$ angeregt. Entsprechend Kapitel 5.3.2 kommt eine Anhebungslogik für die nicht aktivierten Stützwerte des GRNN zum Einsatz. In Abb. 5.16 ist der Fehlerverlauf

(links) und die Konvergenz der Parameter der statischen Nichtlinearität (rechts) dargestellt.

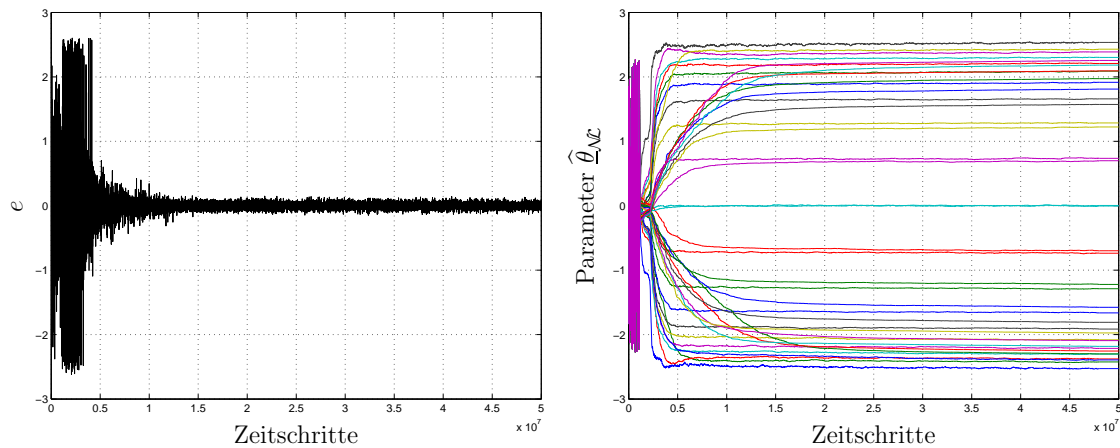


Abb. 5.16: Fehlerverlauf der nichtlinear verkoppelten Wiener-Modelle (links) und Konvergenz von ausgewählten Parametern der statischen Nichtlinearität (rechts)

In Abb. 5.17 ist die Konvergenz der Parameter der Übertragungsfunktionen $F_1(s)$ und $F_2(s)$ gezeigt.

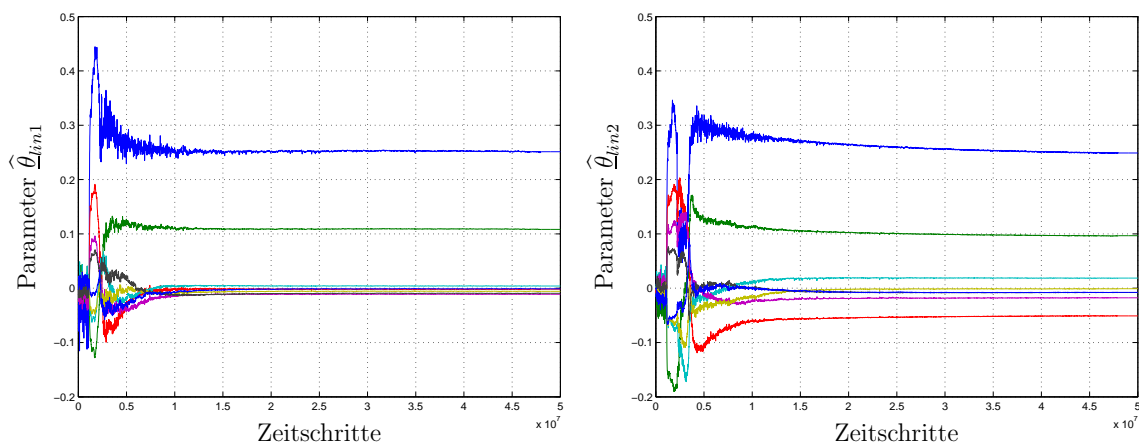


Abb. 5.17: Konvergenz der Parameter der Übertragungsfunktionen $F_1(s)$ (links) und $F_2(s)$ (rechts)

In Abb. 5.16 links ist zu erkennen, dass der Fehler e klein wird und auch im Parallelbetrieb nicht merklich ansteigt. Das Einsetzen der Anhebungslogik bei ca. $3 \cdot 10^6$ Zeitschritten ist deutlich im Fehlerverlauf zu erkennen. Aufgrund der sehr langen Konvergenzzeiten der Parameter, bedingt durch das Gradientenabstiegsverfahren,

und der begrenzten Parameteranzahl bleibt ein gewisser Restfehler. Dennoch ist das Identifikationsergebnis als sehr gut zu bewerten. Die Auswertung des Identifikationsergebnisses für die statische Nichtlinearität ist in Abb. 5.18 gezeigt.

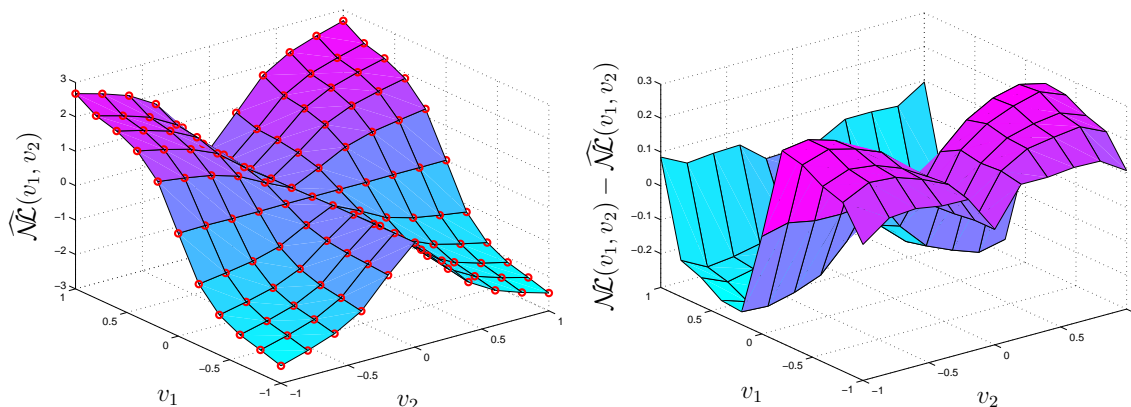


Abb. 5.18: Identifizierte statische Nichtlinearität (links) und Approximationsfehler (rechts)

Auch in diesem Falle wird die statische Nichtlinearität durch das GRNN sehr gut identifiziert. Mit dem Polynomansatz wäre ein Identifikationsergebnis dieser Qualität nicht möglich. In Abb. 5.19 sind die identifizierten Impulsantworten der Übertragungsfunktionen veranschaulicht.

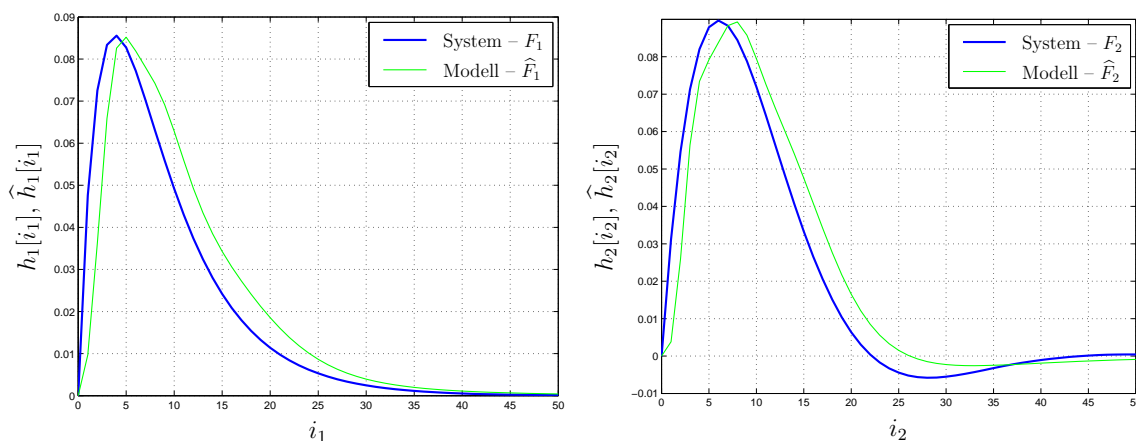


Abb. 5.19: Identifizierte Impulsantworten der Übertragungsfunktionen $F_1(s)$ (links) und $F_2(s)$ (rechts)

Die identifizierten Impulsantworten weichen aufgrund der geringen Basisfunktionsanzahl leicht von der Vorgabe ab. Das Beispiel verdeutlicht jedoch, dass das in Kapitel 5.3 vorgestellte Identifikationsverfahren auch auf Systeme mit mehreren Eingangsgrößen, die auf verkoppelten Wiener-Modellen basieren, anwendbar ist.

5.5 Kurzzusammenfassung

In diesem Kapitel wurde die Einschränkung aus den Kapiteln 3 und 4 aufgehoben, dass die statischen Nichtlinearitäten von blockorientierten Systemen durch Polynome q -ten Grades approximiert werden müssen. Zur Approximation der statischen Nichtlinearitäten wurde stattdessen das GRNN als universeller Funktionsapproximator verwendet.

Im Falle des Hammerstein-Modells zeigte sich, dass eine Analogie zur Volterra-Funktionalpotenzreihe hergestellt werden kann. Die Vorteile gegenüber dem Polynomansatz konnten am Beispiel eines Hammerstein-Modells, das eine Sättigungscharakteristik als statische Nichtlinearität beinhaltet, veranschaulicht werden.

Im Falle des Wiener-Modells zeigte sich, dass sich keine Analogie zur Volterra-Funktionalpotenzreihe mehr ergibt, da die unbekannt Parameter der Übertragungsfunktion nichtlinear in das Ausgangssignal eingehen. Dennoch ist es möglich mittels eines nichtlinearen Adaptionsverfahrens, ein Wiener-Modell mit einem GRNN zur Approximation der statischen Nichtlinearität zu identifizieren. Die Parameter konvergieren jedoch, bedingt durch das Gradientenabstiegsverfahren, vergleichsweise langsam.

Abschließend wurde beispielhaft für zwei nichtlinear verkoppelte Hammerstein-Modelle sowie für zwei nichtlinear verkoppelte Wiener-Modelle gezeigt, dass die für die jeweiligen SISO-Systeme abgeleiteten Identifikationsverfahren, problemlos zur Identifikation von Systemen mit mehreren Eingangsgrößen eingesetzt werden können. Somit wurde durch den Einsatz eines universellen Funktionsapproximators, die Praxistauglichkeit der abgeleiteten Identifikationsverfahren für Mehrgrößensysteme aus Kapitel 4 deutlich erhöht.

6 Beobachterentwurf für Systeme mit dynamischen Nichtlinearitäten

In den Kapiteln 4 und 5 wurde ausführlich die Identifikation von Mehrgrößensystemen auf Basis der Volterra-Funktionalpotenzreihe und deren Verallgemeinerung mittels eines Funktionsapproximatoransatzes behandelt. Dabei wurde davon ausgegangen, dass sowohl die Eingangssignale als auch das Ausgangssignal der betrachteten Systeme unmittelbar zur Verfügung stehen. In der Regel sind diese Systeme jedoch in ein komplexeres Gesamtsystem eingebettet, so dass die Eingangssignale nicht direkt vorgegeben werden können bzw. das Ausgangssignal nicht direkt gemessen werden kann.

Für den Fall einer unbekanntem statischen Nichtlinearität wurde für dieses Problem der Neuronale Beobachter (vgl. Kapitel 3.3.2) entwickelt, der es erlaubt statische Nichtlinearitäten innerhalb eines bekannten linearen dynamischen Systems zu identifizieren. Die Nachteile des Neuronalen Beobachters sind, dass das lineare dynamische System sowohl in seiner Struktur als auch in seinen Parametern exakt bekannt sein muss. Diese Voraussetzung schränkt die Anwendbarkeit des Neuronalen Beobachters bei komplexeren Systemen in der Praxis ein. Aus diesem Grund konzentrieren sich die nachfolgenden Forschungsarbeiten auf dem Gebiet des Neuronalen Beobachters darauf, die ursprünglich getroffenen Einschränkungen aufzuheben.

In [16, 15] werden neben den statischen Nichtlinearitäten auch die linearen Parameter des in seiner Struktur vollständig bekannten Systems identifiziert. Hierzu wird das Blockschaltbild eines Systems in ein strukturiertes rekurrentes Neuronales Netz (SRNN) überführt. Auf das SRNN wird in Kapitel 8 noch genauer eingegangen.

In der vorliegenden Arbeit soll der Neuronale Beobachter in der Richtung erweitert werden, dass auch sog. dynamische Nichtlinearitäten identifiziert werden können. Anschaulich bedeutet das, dass nicht mehr die gesamte Struktur des Systems exakt bekannt sein muss, sondern dass für ein Teilsystem eine grobe Strukturkenntnis ausreichend ist¹. In Abb. 6.1 sind die unterschiedlichen Erweiterungen des Neuronalen Beobachters grafisch veranschaulicht. Dynamische Nichtlinearitäten sind nichtlineare blockorientierte Systeme, wie sie in den Kapiteln 4 und 5 schon genauer betrachtet wurden. Eine genaue Definition des Begriffs der dynamischen Nichtlinearität erfolgt im nächsten Abschnitt.

¹Theoretisch ist es auch möglich, Teilsysteme ohne jegliche Vorkenntnisse über die Struktur zu identifizieren. Dies führt in der Praxis jedoch zu einer extrem hohen Parameteranzahl.

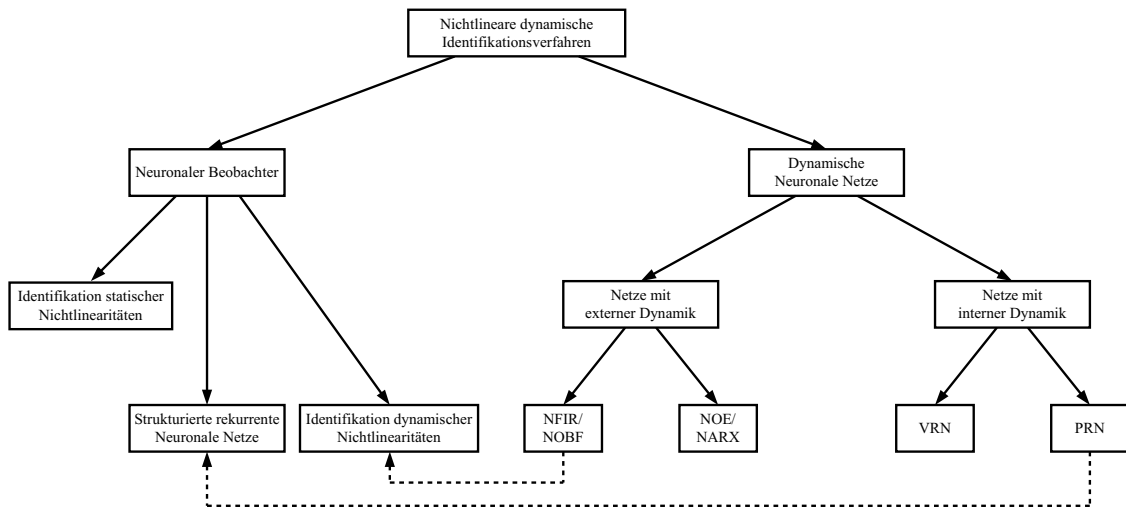


Abb. 6.1: Erweiterungen des Neuronalen Beobachters

Die grundlegende Idee der Erweiterung des Neuronalen Beobachters ist, dass ein komplexes nichtlineares dynamisches System in ein lineares dynamisches Teilsystem, das in seiner Struktur und seinen Parametern exakt bekannt ist, und in ein nichtlineares dynamisches Teilsystem, das nur grob in seiner Struktur und nicht in seinen Parametern bekannt ist, eingeteilt werden kann. Dieses unbekanntes nichtlineare dynamische Teilsystem stellt die dynamische Nichtlinearität dar. Das grobe Strukturwissen ist durch die blockorientierte Struktur der dynamischen Nichtlinearität gegeben (NFIR-/NOBF-Modelle).

In der Praxis ist es oftmals erheblich leichter eine Vorstellung über die Struktur eines Systems in Form von statischen Nichtlinearitäten, Übertragungsfunktionen und mathematischen Operationen wie z.B. Summationen und Multiplikationen zu erlangen, als eine exakte physikalische Modellbildung durchzuführen. Der Neuronale Beobachter zur Identifikation dynamischer Nichtlinearitäten ist somit immer dann vorteilhaft einzusetzen, wenn die exakte Modellierung eines unbekanntes nichtlinearen Teilsystems aufgrund von mangelnder Systemkenntnis nicht oder nur mit einem unverhältnismäßig hohem Aufwand möglich ist. Außerdem ist es möglich in der dynamischen Nichtlinearität auch strukturelle Nichtlinearitäten wie Multiplikationen im Blockschaltbild mit zu berücksichtigen. Der Neuronale Beobachter zur Identifikation statischer Nichtlinearitäten ist als Spezialfall im erweiterten Beobachterentwurf enthalten.

6.1 Systeme mit dynamischen Nichtlinearitäten

Bevor ganze Systeme mit dynamischen Nichtlinearitäten genauer betrachtet werden können, muss zunächst der Begriff der dynamischen Nichtlinearität genauer erläutert werden.

Eine dynamische Nichtlinearität $\mathcal{NL}_{dyn}(\underline{u}): \mathbb{R}^N \rightarrow \mathbb{R}$ ist ein nichtlineares System, welches sich in statische Nichtlinearitäten und in zeitinvariante Übertragungsfunktionen mit abklingender Impulsantwort separieren lässt. Hierbei können die einzelnen Blöcke linear und nichtlinear miteinander verkoppelt sein, jedoch darf es keine Ausgangsrückkopplung zwischen den einzelnen Blöcken geben.

Anschaulich werden unter der obigen Definition einer dynamischen Nichtlinearität blockorientierte Systeme, wie sie in Kapitel 4 und 5 behandelt wurden, verstanden. Als System mit separierbarer dynamischer Nichtlinearität wird eine Strecke bezeichnet, die sich nach folgender Zustandsdarstellung beschreiben lässt²:

$$\begin{aligned}\dot{\underline{x}} &= \underline{\mathbf{A}} \cdot \underline{x} + \underline{\mathbf{b}} \cdot u + \underline{\mathbf{k}} \cdot \mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \\ y &= \underline{\mathbf{c}}^T \cdot \underline{x} + d \cdot u\end{aligned}\tag{6.1}$$

Der Vektor $\underline{\mathbf{k}}$ bezeichnet einen konstanten Einkopplungsvektor. In Abb. 6.2 ist das Blockschaltbild eines Systems mit separierbarer dynamischer Nichtlinearität veranschaulicht.

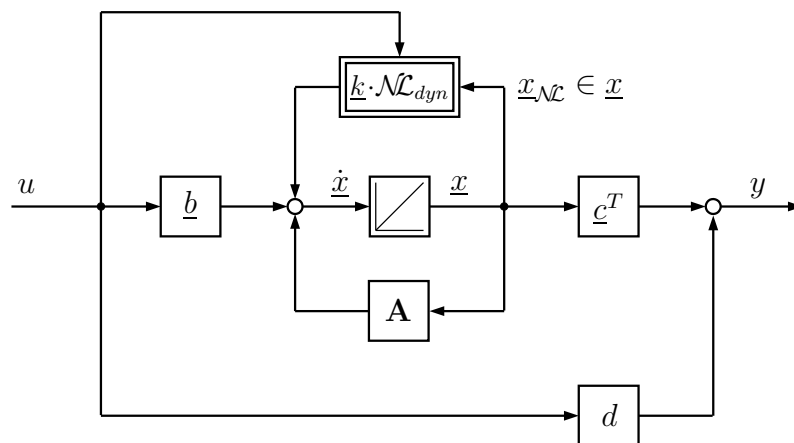


Abb. 6.2: Blockschaltbild eines Systems mit dynamischer Nichtlinearität

²Hier wird ausschließlich der Fall einer SISO-Strecke betrachtet.

In Gl. (6.1) bzw. Abb. 6.2 bezeichnet

- u den skalaren Systemeingang,
- $\underline{x} \in \mathbb{R}^{n \times 1}$ den Zustandsvektor mit n Zuständen,
- $\mathbf{A} \in \mathbb{R}^{n \times n}$ die Systemmatrix des linearen Streckenteils,
- $\underline{b} \in \mathbb{R}^{n \times 1}$ den Einkopplungsvektor des Systemeingangs,
- $\underline{x}_{\mathcal{N}} \in \underline{x}$ den Vektor der Zustandsgrößen von denen die dynamische Nichtlinearität abhängt,
- $\mathcal{N}_{dyn}(\underline{x}_{\mathcal{N}}, u)$ die dynamische Nichtlinearität,
- $\underline{k} \in \mathbb{R}^{n \times 1}$ den Einkopplungsvektor der dynamischen Nichtlinearität,
- $\underline{c} \in \mathbb{R}^{n \times 1}$ den Auskopplungsvektor,
- d den Durchgriff des Systemeingangs auf den Systemausgang und
- y den skalaren Systemausgang.

In Gl. (6.1) ist deutlich die Trennung zwischen dem bekannten linearen Teilsystem und der unbekannt dynamischen Nichtlinearität zu erkennen³. Neben den Systemzuständen \underline{x} , existieren weitere Systemzustände innerhalb der dynamischen Nichtlinearität, welche jedoch aufgrund der mangelnden Strukturkenntnis nicht explizit modelliert werden können. Anschaulich bedeutet das, dass die mangelnde Strukturkenntnis der dynamischen Nichtlinearität es nicht erlaubt, die Systemstruktur auf einer Ebene zu veranschaulichen, in der einzelne Integratoren bzw. Systemzustände betrachtet werden können. Wird als dynamische Nichtlinearität beispielsweise ein Hammerstein-Modell angenommen, das eine Übertragungsfunktion $F(s)$ enthält, so ist damit noch keine Aussage darüber getroffen, wie viele Systemzustände die dynamische Nichtlinearität enthält. Eine solche Aussage würde eine detaillierte Strukturkenntnis erfordern. Wäre diese Strukturkenntnis vorhanden und die Parameter der Übertragungsfunktion bekannt, könnte die Übertragungsfunktion der linearen Zustandsdarstellung zugeordnet werden und die Identifikation könnte sich auf die statische Nichtlinearität beschränken. Wäre hingegen nur die Struktur der Übertragungsfunktion bekannt, nicht aber ihre Parameter, könnte die Übertragungsfunktion zwar der linearen Zustandsdarstellung zugeordnet werden, jedoch wären die linearen Parameter nicht vollständig bekannt. Das bedeutet, dass der Neuronale Beobachter zur Identifikation statischer Nichtlinearitäten in diesem Fall nicht angewendet werden könnte. Für eine gemeinsame Identifikation der linearen Parameter und der statischen Nichtlinearitäten wurde in [16] das SRNN entwickelt, das somit auf dieses Problem angewendet werden könnte.

³Es wird deshalb auch von einer separierbaren Nichtlinearität gesprochen.

In Kapitel 8 wird gezeigt werden, dass es mit einer Kombination der Theorie zu den strukturierten rekurrenten Netzen und der Theorie zur Identifikation dynamischer Nichtlinearitäten möglich ist, Systeme mit unbekanntem linearen Parametern und statischen sowie dynamischen Nichtlinearitäten zu identifizieren. Zunächst soll jedoch der Beobachterentwurf für Systeme mit dynamischen Nichtlinearitäten genauer betrachtet werden.

6.2 Beobachterentwurf

Zur Anwendung der im Folgenden beschriebenen Beobachtertheorie werden zunächst alle notwendigen Voraussetzungen noch einmal zusammengefasst⁴:

- Das betrachtete System ist gemäß Gl. (6.1) bzw. Abb. 6.2 darstellbar.
- Das lineare dynamische Teilsystem ist durch die Parameter \mathbf{A} , \underline{b} , \underline{c}^T , d bekannt und zeitinvariant.
- Das lineare dynamische Teilsystem ist zustandsbeobachtbar⁵. Dies ist genau dann der Fall, wenn die Beobachtbarkeitsmatrix \mathbf{Q}_{obs} regulär ist, d.h. wenn gilt:

$$\det(\mathbf{Q}_{obs}) \neq 0 \quad \text{mit} \quad \mathbf{Q}_{obs} = [\underline{c}, \mathbf{A}^T \underline{c}, \dots, (\mathbf{A}^T)^{(n-1)} \underline{c}] \quad (6.2)$$

- Der Einkopplungsvektor \underline{k} der dynamischen Nichtlinearität ist bekannt und konstant.
- Die Auswirkungen der dynamischen Nichtlinearität müssen im Ausgangssignal y des Systems sichtbar sein. Die Sichtbarkeit einer Nichtlinearität ist entsprechend Gl. (3.25) genau dann erfüllt, wenn gilt:

$$H_S(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot \underline{k} \neq 0 \quad (6.3)$$

- Das Ausgangssignal der dynamischen Nichtlinearität wird als zusätzliches Eingangssignal des linearen Teilsystems betrachtet. Nur unter dieser Voraussetzung können weiterhin die bekannten linearen mathematischen Methoden angewandt werden.

Die nachfolgenden Ausführungen zum Beobachterentwurf erfolgen in enger Anlehnung an [32, 58]. Der Beobachterentwurf erfolgt dabei in einer kontinuierlichen Darstellungsform. Im Gegensatz dazu geht der eigentliche Identifikationsalgorithmus von

⁴Diese Voraussetzungen decken sich größtenteils mit denen des Neuronalen Beobachters zur Identifikation statischer Nichtlinearitäten [58].

⁵Dies muss im Falle eines nicht messbaren Eingangsraumes der dynamischen Nichtlinearität vorausgesetzt werden.

einer zeitdiskreten Darstellung aus. Dies hat zur Folge, dass durch die Verwendung eines zeitdiskreten Identifikationsalgorithmus in einer zeitkontinuierlichen Beobachterdarstellung, ein formaler Widerspruch entsteht. Dieser Widerspruch wird jedoch aus Gründen einer übersichtlicheren Gesamtdarstellung in Kauf genommen. Außerdem relativiert sich dieser Widerspruch im Hinblick auf die Anwendung der vorgestellten Theorie in modernen Simulationsprogrammen, wie z.B. MATLAB/Simulink. Die zeitkontinuierliche Darstellung soll in diesem Zusammenhang zum Ausdruck bringen, dass jeder beliebige zeitdiskrete Integrationsalgorithmus zum Einsatz kommen kann, wie z.B. die Euler–Vorwärts–Approximation⁶.

6.2.1 Beobachterentwurf bei messbarem Eingangsraum der dynamischen Nichtlinearität

Zunächst wird davon ausgegangen, dass der Eingangsraum der dynamischen Nichtlinearität $\mathcal{N}_{dyn}(\underline{x}_{\mathcal{N}}, u)$ messbar ist, d.h. dass alle Zustände $\underline{x}_{\mathcal{N}} \in \underline{x}$, von denen die dynamische Nichtlinearität abhängig ist, messbar sind. Für das System nach Gl. (6.1) wird ein Zustandsbeobachter nach Luenberger angesetzt, der um den Approximationsalgorithmus für die dynamische Nichtlinearität $\widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{N}}, u)$ erweitert ist⁷:

$$\begin{aligned}\dot{\hat{\underline{x}}} &= \mathbf{A} \cdot \hat{\underline{x}} + \underline{b} \cdot u + \underline{k} \cdot \widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{N}}, u) - \underline{l} \cdot e \\ \hat{y} &= \underline{c}^T \cdot \hat{\underline{x}} + d \cdot u\end{aligned}\tag{6.4}$$

Der Beobachterfehler ist dabei definiert zu:

$$e = \hat{y} - y\tag{6.5}$$

Für die Dimensionierung des Beobachtervektors \underline{l} gelten alle bekannten Einstellvorschriften für den linearen Beobachterentwurf [34] (z.B. Polvorgabe, LQ–Optimierung, Kalman–Filter–Theorie) zur Erzielung eines asymptotisch stabilen und ausreichend schnellen Beobachterverhaltens, so dass darauf an dieser Stelle nicht weiter eingegangen werden soll. In Abb. 6.3 ist die resultierende Struktur von System und Beobachter in Zustandsdarstellung verdeutlicht.

⁶Durch Verwendung der Euler–Vorwärts–Approximation könnte die zeitkontinuierliche Darstellungsform des Beobachters sehr einfach diskretisiert werden und somit wäre der formale Widerspruch gelöst.

⁷Bei diesem Ansatz wird die dynamische Nichtlinearität als Eingang des linearen Teilsystems interpretiert, so dass weiter mit den bekannten linearen Methoden gearbeitet werden kann [32].

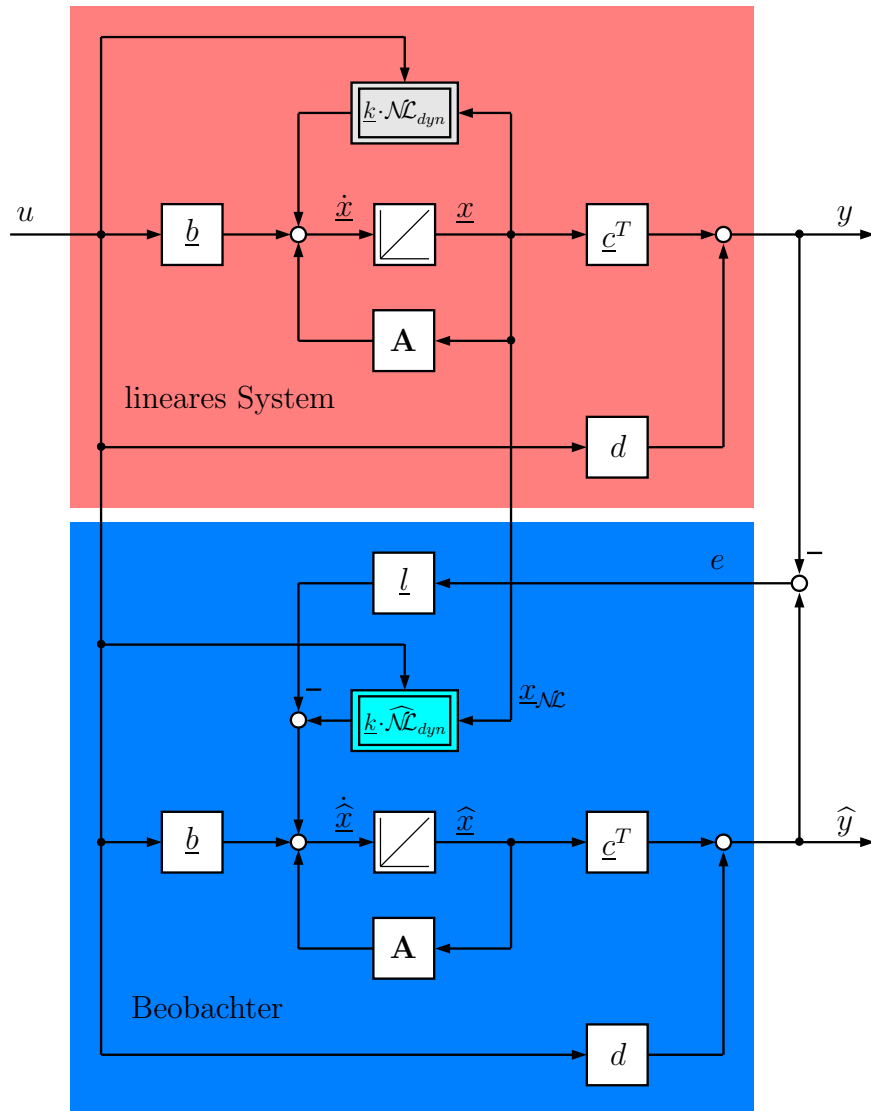


Abb. 6.3: *Neuronaler Zustandsbeobachter bei messbarem Eingangsraum der dynamischen Nichtlinearität*

Zur Ableitung eines stabilen Lerngesetzes ist die Bestimmung der sog. Fehlerübertragungsfunktion $H(s)$ nötig. Die Berechnung dieser Fehlerübertragungsfunktion soll im Folgenden kurz skizziert werden. Zunächst wird der Beobachterfehler in den Zuständen definiert zu:

$$\underline{e}_Z = \hat{\underline{x}} - \underline{x} \quad (6.6)$$

Für die Fehlerdifferentialgleichung des Zustandsfehlers gilt somit:

$$\dot{\underline{e}}_Z = (\mathbf{A} - \underline{l} \cdot \underline{c}^T) \cdot \underline{e}_Z + \underline{k} \cdot \left[\widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{N}}, u) - \mathcal{N}_{dyn}(\underline{x}_{\mathcal{N}}, u) \right] \quad (6.7)$$

Der Identifikationsalgorithmus für die dynamische Nichtlinearität wird im Lerngesetz jedoch mit dem Ausgangsfehler e adaptiert. Zur einfacheren Darstellung wird

auf die Laplace–Notation übergegangen. Die Signale $\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ und $\widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ werden dazu als normale Zeitsignale aufgefasst, deren Laplace–Transformation existiert. Die Fehlerdifferentialgleichung ergibt sich somit zu⁸:

$$s \cdot \underline{e}_Z(s) = (\mathbf{A} - \underline{l} \cdot \underline{c}^T) \cdot \underline{e}_Z(s) + \underline{k} \cdot \underbrace{\mathcal{L} \left\{ \left[\widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) - \mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \right] \right\}}_{e_{\mathcal{NL}}(s)} \quad (6.8)$$

Das Umformen von Gl. (6.8) führt zu:

$$\underline{e}_Z(s) = (s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)^{-1} \cdot \underline{k} \cdot e_{\mathcal{NL}}(s) \quad (6.9)$$

Der Ausgangsfehler e ergibt sich somit zu:

$$e(s) = \underline{c}^T \cdot \underline{e}_Z(s) = \underbrace{\underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)^{-1} \cdot \underline{k}}_{H(s)} \cdot e_{\mathcal{NL}}(s) \quad (6.10)$$

Die Fehlerübertragungsfunktion

$$H(s) = \frac{e(s)}{e_{\mathcal{NL}}(s)} = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)^{-1} \cdot \underline{k} \quad (6.11)$$

beschreibt das Übertragungsverhalten des Beobachters vom Angriffspunkt der dynamischen Nichtlinearität zum Systemausgang. Der Fehler $e_{\mathcal{NL}}$ zwischen der realen und identifizierten dynamischen Nichtlinearität wird durch $H(s)$ zum Systemausgang übertragen und kann dort als Beobachterfehler e gemessen werden.

Analog zu Kapitel 3.3.2 müssen abhängig von der Fehlerübertragungsfunktion $H(s)$ zwei Fälle zur Ableitung eines stabilen Lerngesetzes unterschieden werden. Kriterium für die Unterscheidung ist die SPR–Eigenschaft, d.h. die Frage, ob die Fehlerübertragungsfunktion eine streng positive reelle Funktion ist. Die Definition einer streng positiv reellen Funktion wurde bereits in Kapitel 3.3.2.1 gegeben [38].

Für die Ableitung eines stabilen Lerngesetzes ist eine zwingende Voraussetzung, dass der Identifikationsalgorithmus für die dynamische Nichtlinearität linear in den unbekannt Parametern ist⁹. Bei Vernachlässigung des unvermeidbaren Approximationsfehlers, der mit steigender Antwortlänge, Basisfunktionenanzahl und Stützstellen– bzw. Polynomkoeffizientenanzahl immer kleiner wird, kann die optimal adaptierte dynamische Nichtlinearität wie folgt dargestellt werden:

$$\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u) = \underline{\theta}^T \cdot \underline{A}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \quad (6.12)$$

Der konstante Parametervektor $\underline{\theta}$ beschreibt die optimal adaptierten Parameter der dynamischen Nichtlinearität und geht linear in das Ausgangssignal ein. Die Parameter sind unbekannt und müssen durch eine Identifikation bestimmt werden.

⁸Aufgrund der Stabilität der Zustandsmatrix \mathbf{A} klingt der Einfluss der Anfangswerte ab. Somit kann der Anfangsfehler $\underline{e}_Z(t=0)$ weggelassen werden.

⁹Dies ist bei allen Identifikationsalgorithmen aus Kapitel 4 und 5 mit Ausnahme des Wiener–Modells auf Basis des Funktionsapproximatoransatzes der Fall.

Der Ausgang des Approximationsalgorithmus berechnet sich zu:

$$\widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) = \widehat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \quad (6.13)$$

Werden die Gleichungen (6.12) und (6.13) in die Gleichungen (6.8) bis (6.10) eingesetzt, ergibt sich unter der Annahme, dass sich $\widehat{\underline{\theta}}$ nur langsam ändert:

$$e(s) = H(s) \cdot \mathcal{L} \left\{ \left(\widehat{\underline{\theta}}^T - \underline{\theta}^T \right) \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \right\} \quad (6.14)$$

Wird der Parameterfehlervektor $\underline{\Phi} = \widehat{\underline{\theta}} - \underline{\theta}$ eingesetzt, folgt die in der Literatur [38] bekannte Fehlergleichung:

$$e(s) = H(s) \cdot \mathcal{L} \left\{ \underline{\Phi}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \right\} \quad (6.15)$$

Erfüllt die Fehlerübertragungsfunktion $H(s)$ die SPR-Bedingung, kann zur Adaption der unbekannt Parameter das Fehlermodell 3 nach [38] verwendet werden¹⁰. Für die Adaptionsgleichung gilt im Zeitbereich¹¹:

$$\dot{\underline{\Phi}} = \dot{\widehat{\underline{\theta}}} = -\eta \cdot e \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \quad \eta > 0 \quad (6.16)$$

In Abb. 6.4 ist das Fehlermodell 3 noch einmal grafisch veranschaulicht¹².

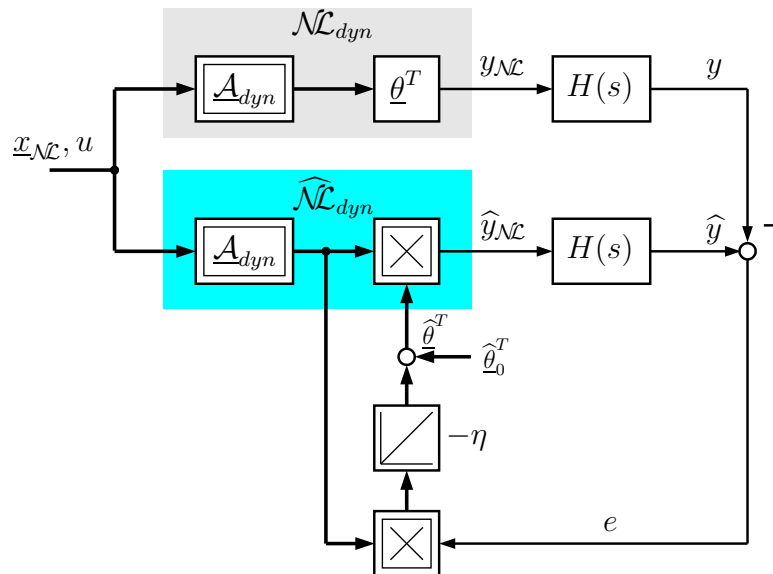


Abb. 6.4: Lerngesetz nach Fehlermodell 3 zur Adaption von $\widehat{\mathcal{N}}_{dyn}$

¹⁰Die Fehlermodelle 3 und 4 basieren auf dem Gradientenabstiegsverfahren. In Kapitel 7 erfolgt eine Erweiterung der Fehlermodelle auf den RLS-Algorithmus.

¹¹Es gilt $e(s) = \mathcal{L}\{e\}$.

¹²In den Abbildungen 6.4 und 6.5 werden aus Gründen einer besseren Übersicht die Abkürzungen $y_{\mathcal{NL}}(s) = \mathcal{L}\{\mathcal{N}_{dyn}(\underline{x}_{\mathcal{NL}}, u)\}$ und $\widehat{y}_{\mathcal{NL}}(s) = \mathcal{L}\{\widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{NL}}, u)\}$ eingeführt.

Anschaulich bedeutet das Fehlermodell 3, dass trotz des Einflusses der Fehlerübertragungsfunktion $H(s)$ das Gradientenabstiegsverfahren unverändert eingesetzt werden kann.

Im allgemeineren Fall, dass die Fehlerübertragungsfunktion $H(s)$ nicht die SPR-Bedingung erfüllt, muss das in [38] als Fehlermodell 4 bezeichnete Lerngesetz verwendet werden. Für dieses Lerngesetz gilt:

$$\begin{aligned} \dot{\underline{\Phi}} = \dot{\underline{\hat{\theta}}} &= -\eta \cdot e_e \cdot \mathcal{L}^{-1} \left\{ \underbrace{H(s) \cdot \mathcal{L} \{ \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \}}_{\text{verzögerte dynamische Aktivierung}} \right\} \\ &= -\eta \cdot e_e \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_{Vdyn}(s) \} \quad \eta > 0 \end{aligned} \quad (6.17)$$

Das Lerngesetz nach Fehlermodell 4 wird auch als Verfahren der verzögerten Aktivierung bezeichnet [32]. Zur Adaption der Parameter wird ein erweiterter Fehler e_e gebildet, der sich aus dem Beobachterfehler e und einem zusätzlichen Fehler e_z zusammensetzt. Für den erweiterten Fehler e_e ergibt sich somit¹³:

$$e_e(s) = e(s) + e_z(s) \quad (6.18)$$

mit¹⁴

$$e_z(s) = \underline{\hat{\theta}}^T \cdot H(s) \cdot \mathcal{L} \{ \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \} - H(s) \cdot \mathcal{L} \{ \underline{\hat{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \} \quad (6.19)$$

Das Fehlermodell 4 ist in Abb. 6.5 grafisch veranschaulicht.

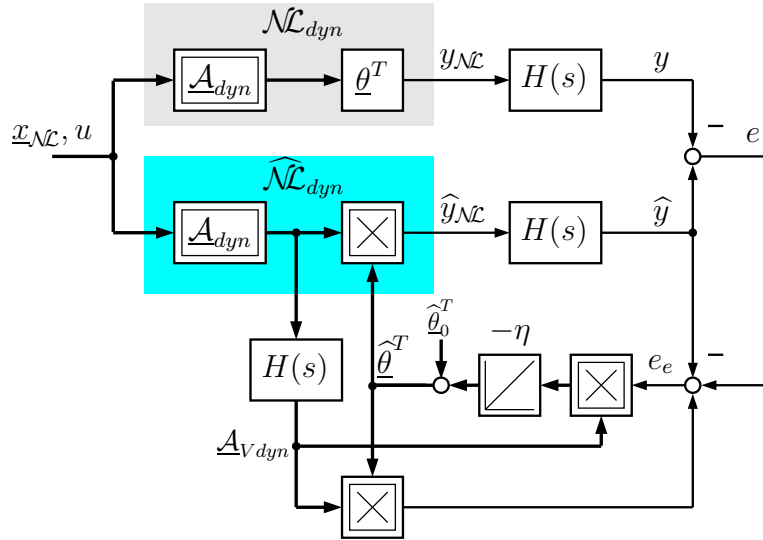


Abb. 6.5: Lerngesetz nach Fehlermodell 4 zur Adaption von $\widehat{\mathcal{N}}_{dyn}$

Die verzögerte dynamische Aktivierung bewirkt, dass der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ so lange verzögert wird, bis die Auswirkungen des Fehlers $e_{\mathcal{NL}}$

¹³Es gilt $e_e(s) = \mathcal{L} \{ e_e \}$.

¹⁴Es wird angenommen, dass sich der Parametervektor $\underline{\hat{\theta}}$ nur sehr langsam ändert.

im Beobachterfehler e sichtbar werden. Der erweiterte Fehler e_e wird gebildet um die dynamischen Auswirkungen der Adaption der unbekannt Parameter im Fehler zu kompensieren. Dies kann wie folgt gezeigt werden [38]. Wird in Gl. (6.19) zunächst der Vektor $\underline{\mathcal{A}}_{V_{dyn}}$ für die verzögerte dynamische Aktivierung eingeführt, ergibt sich¹⁵:

$$e_z(t) = \widehat{\underline{\theta}}^T(t) \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_{V_{dyn}}(s) \} - \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \widehat{\underline{\theta}}^T(t) \cdot \underline{\mathcal{A}}_{dyn}(t) \right\} \right\} \quad (6.20)$$

Wenn $\widehat{\underline{\theta}}^T(t) = \underline{\theta}$ zutrifft, wobei $\underline{\theta}$ der zeitinvariante optimale Parametervektor ist, dann folgt aus Gl. (6.20):

$$\delta(t) = \underline{\theta}^T \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_{V_{dyn}}(s) \} - \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}(t) \right\} \right\} \quad (6.21)$$

Das Signal $\delta(t)$ ist ein exponentiell abklingendes Signal, so dass gilt:

$$\lim_{t \rightarrow \infty} \delta(t) = 0 \quad (6.22)$$

Durch Einführen des Parameterfehlervektors $\underline{\Phi}(t) = \widehat{\underline{\theta}}(t) - \underline{\theta}$ und Umformen von Gl. (6.20) ergibt sich für den zusätzlichen Fehler e_z :

$$e_z(t) = \underline{\Phi}^T(t) \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_{V_{dyn}}(s) \} - \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\Phi}^T(t) \cdot \underline{\mathcal{A}}_{dyn}(t) \right\} \right\} + \delta(t) \quad (6.23)$$

Wird in die Definition des erweiterten Fehlers nach Gl. (6.18) der Beobachterfehler

$$e(t) = \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\Phi}^T(t) \cdot \underline{\mathcal{A}}_{dyn}(t) \right\} \right\} \quad (6.24)$$

sowie der zusätzliche Fehler e_z nach Gl. (6.23) eingesetzt, folgt:

$$e_e(t) = \underline{\Phi}^T(t) \cdot \mathcal{L}^{-1} \{ \underline{\mathcal{A}}_{V_{dyn}}(s) \} + \delta(t) \quad (6.25)$$

Für Gl. (6.25) und (6.17) kann analog zu [32, 38] asymptotische Stabilität [63] nach Lyapunov bewiesen werden. Der Beweis soll an dieser Stelle jedoch nicht geführt werden.

Anschaulich können die obigen Gleichungen folgendermaßen interpretiert werden. Wenn im Laufe der Parameteradaption $\lim_{t \rightarrow \infty} \widehat{\underline{\theta}}(t) = \underline{\theta}$ gilt¹⁶, geht der zusätzliche Fehler $e_z(t)$ gegen Null und damit strebt der erweiterte Fehler $e_e(t)$ gegen den Beobachterfehler e . Nach Gl. (6.25) strebt jedoch der erweiterte Fehler $e_e(t)$ gegen Null, da sowohl der Term $\underline{\Phi}(t)$ als auch der Term $\delta(t)$ gegen Null gehen. Folglich muss auch der Beobachterfehler $e(t)$ gegen Null streben. Die Verwendung der verzögerten dynamischen Aktivierung $\underline{\mathcal{A}}_{V_{dyn}}(s)$ sowie des erweiterten Fehlers $e_e(t)$ führen in dem Fall, dass die Fehlerübertragungsfunktion $H(s)$ nicht die SPR-Bedingung erfüllt, somit wieder zu einem stabilen Lerngesetz.

¹⁵Zur besseren Veranschaulichung wird im Folgenden die Zeitabhängigkeit der einzelnen Signale explizit dargestellt. Dabei wird angenommen, dass sich $\widehat{\underline{\theta}}(t)$ nur sehr langsam ändert.

¹⁶Es wird vorausgesetzt, dass der Systemeingang die Persistent-Excitation-Bedingung erfüllt [38].

6.2.2 Beobachterentwurf bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität

Im Gegensatz zu Kapitel 6.2.1 ist im Folgenden die vollständige Messbarkeit des Eingangsraumes der dynamischen Nichtlinearität $\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ nicht mehr gefordert. In diesem Fall müssen die beobachteten Signale $\hat{\underline{x}}_{\mathcal{NL}}$ als Eingangssignale für die Identifikation der dynamischen Nichtlinearität verwendet werden. Hierzu ist, wie am Anfang von Kapitel 6.2 bereits ausgeführt wurde, die vollständige Zustandsbeobachtbarkeit eine zwingende Voraussetzung. Wird ein Zustandsbeobachter analog zu Kapitel 6.2.1 entworfen, gilt:

$$\begin{aligned}\dot{\hat{\underline{x}}} &= \mathbf{A} \cdot \hat{\underline{x}} + \underline{b} \cdot u + \underline{k} \cdot \widehat{\mathcal{NL}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u) - \underline{l} \cdot e \\ \hat{\underline{y}} &= \underline{c}^T \cdot \hat{\underline{x}} + d \cdot u\end{aligned}\quad (6.26)$$

Der Identifikationsalgorithmus $\widehat{\mathcal{NL}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u)$ hat in Gl. (6.26) den Systemeingang u sowie die beobachteten Systemzustände $\hat{\underline{x}}_{\mathcal{NL}} \in \hat{\underline{x}}$ als Eingangssignale. In Abb. 6.6 ist die resultierende Struktur von System und Beobachter in Zustandsdarstellung verdeutlicht.

Die Dimensionierung des Beobachtervektors \underline{l} sowie die Berechnung der Fehlerübertragungsfunktion $H(s)$ erfolgt analog zum vorangegangenen Kapitel. An dieser Stelle muss der Frage nachgegangen werden, ob die bekannten Fehlermodelle 3 und 4 weiter anwendbar sind. Für die zu approximierende dynamische Nichtlinearität gilt in diesem Fall:

$$\widehat{\mathcal{NL}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u) \neq \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \quad (6.27)$$

Der Grund dafür ist, dass der Eingangsraum der dynamischen Nichtlinearität nicht mehr als Messgröße, sondern nur noch als beobachtete Größe vorliegt. Somit kann der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ nicht bestimmt werden. Während der Identifikation gilt folglich:

$$\underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \neq \underline{\mathcal{A}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u) \quad (6.28)$$

Um die Stabilität der bekannten Fehlermodelle 3 und 4 dennoch zu beweisen, werden formal sog. virtuelle Parameter eingeführt [32]. Für die virtuellen Parameter kann folgender Ansatz gemacht werden:

$$\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{NL}}, u) = \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) = \check{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u) \quad (6.29)$$

Die virtuellen Parameter $\check{\underline{\theta}}$ beschreiben die Bewegung der dynamischen Nichtlinearität im Beobachterzustandsraum $\hat{\underline{x}}_{\mathcal{NL}} \in \hat{\underline{x}}$. Sie sind zeitvariant und ändern sich, wenn sich $\underline{x}_{\mathcal{NL}}$ und $\hat{\underline{x}}_{\mathcal{NL}}$ und somit auch die Aktivierungen $\underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u)$ und $\underline{\mathcal{A}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u)$ im Verlauf der Identifikation angleichen. Die virtuellen Parameter streben folglich gegen die optimalen Parameter, wenn sich die Beobachterzustände den Zuständen der Regelstrecke angleichen, da dann

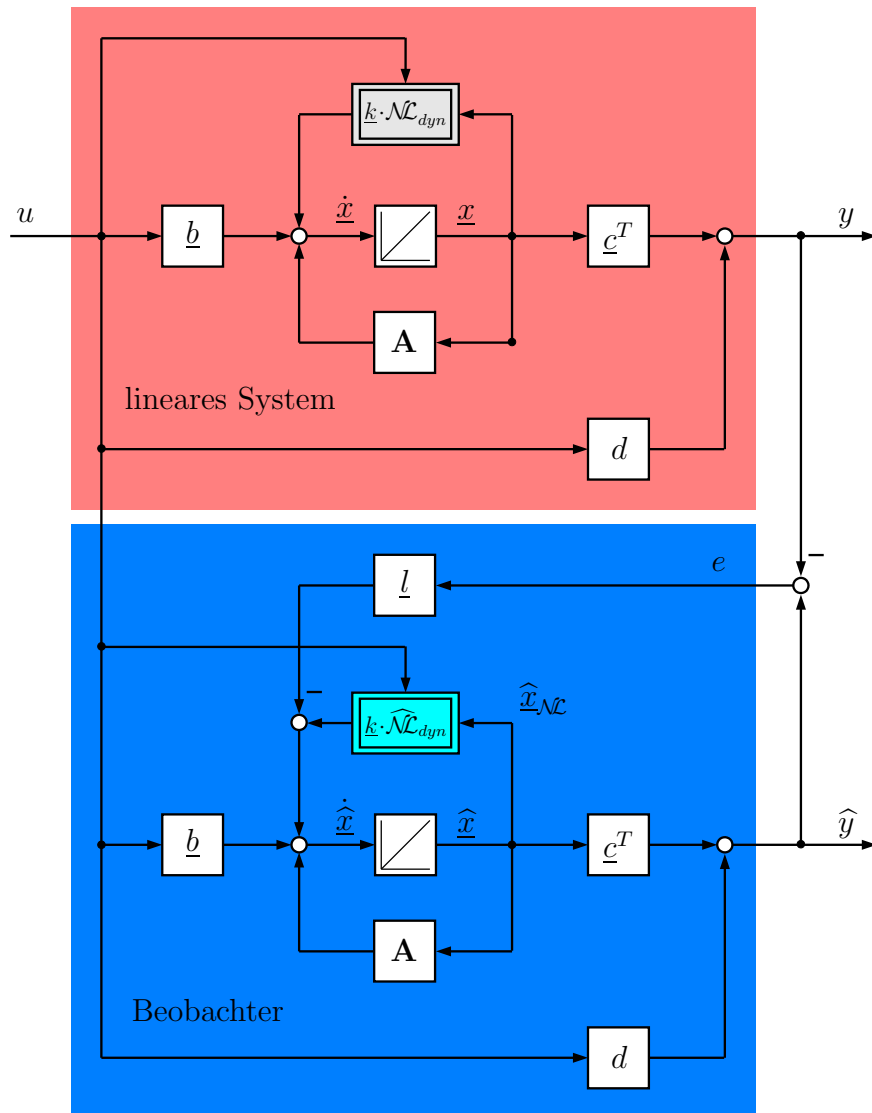


Abb. 6.6: *Neuronaler Zustandsbeobachter bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität*

$$\hat{\underline{x}} \rightarrow \underline{x} \Rightarrow \hat{\underline{A}}_{dyn}(\hat{\underline{x}}_{\mathcal{NL}}, u) \rightarrow \underline{A}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \quad (6.30)$$

$$\hat{\underline{x}} \rightarrow \underline{x} \Rightarrow \hat{\underline{\theta}} \rightarrow \underline{\theta} \quad (6.31)$$

gilt. Es muss nun gezeigt werden, wann dies der Fall ist¹⁷. Mit Gl. (6.29) folgt für den Beobachterfehler:

¹⁷Zur Beweisführung soll das Fehlermodell 4 betrachtet werden.

$$\begin{aligned}
e(s) &= H(s) \cdot \mathcal{L} \left\{ \widehat{\mathcal{N}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) - \mathcal{N}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \right\} \\
&= H(s) \cdot \mathcal{L} \left\{ \widehat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) - \underline{\theta}^T \cdot \underline{\mathcal{A}}_{dyn}(\underline{x}_{\mathcal{NL}}, u) \right\} \\
&= H(s) \cdot \mathcal{L} \left\{ \left(\widehat{\underline{\theta}}^T - \underline{\theta}^T \right) \cdot \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\}
\end{aligned} \tag{6.32}$$

Der Parameterfehlervektor wird definiert zu:

$$\underline{\Phi} = \widehat{\underline{\theta}} - \underline{\theta} \tag{6.33}$$

Im Falle eines nicht messbaren Eingangsraumes gilt folglich:

$$e(s) = H(s) \cdot \mathcal{L} \left\{ \underline{\Phi}^T \cdot \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} \tag{6.34}$$

Für diese Fehlergleichung kann wieder die aus [38] bekannte global stabile Adaptionsgleichung

$$\dot{\underline{\Phi}} = -\eta \cdot e_e \cdot \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} \right\} \tag{6.35}$$

mit dem zu verwendenden Fehler¹⁸

$$e_e(s) = e(s) + \widehat{\underline{\theta}}^T \cdot H(s) \cdot \mathcal{L} \left\{ \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} - H(s) \cdot \mathcal{L} \left\{ \widehat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} \tag{6.36}$$

angewandt werden. Infolge der Zeitvarianz der virtuellen Parameter kann an dieser Stelle aber nicht gefolgert werden, dass $\dot{\underline{\Phi}} = \dot{\widehat{\underline{\theta}}}$ ist. Im Folgenden wird daher das Adaptionsgesetz

$$\dot{\widehat{\underline{\theta}}} = -\eta_{virt} \cdot e_e \cdot \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} \right\} \tag{6.37}$$

verwendet, was für die Änderung des Parameterfehlervektors bedeutet:

$$\dot{\underline{\Phi}} = \dot{\widehat{\underline{\theta}}} - \dot{\underline{\theta}} = - \left[\eta_{virt} \cdot e_e \cdot \mathcal{L}^{-1} \left\{ H(s) \cdot \mathcal{L} \left\{ \underline{\mathcal{A}}_{dyn}(\widehat{\underline{x}}_{\mathcal{NL}}, u) \right\} \right\} + \dot{\underline{\theta}} \right] \tag{6.38}$$

Aus den Gleichungen (6.35) und (6.38) ist ersichtlich, dass ein stabiler Lernvorgang genau dann gewährleistet ist, wenn sich das Vorzeichen von $\dot{\underline{\Phi}}$ trotz $\dot{\underline{\theta}}$ nicht ändert, d.h. wenn der Term $\dot{\widehat{\underline{\theta}}}$ gegenüber $\dot{\underline{\theta}}$ überwiegt. Da die Lernschrittweite η_{virt} positiv sein muss, ergibt sich eine Bedingung der Art:

$$0 < \eta_{min} < \eta_{virt} < \eta_{max} \tag{6.39}$$

Die beschriebene Problematik entspricht anschaulich dargestellt, dem Nachlernen von zeitvarianten Parametern. Dies ist möglich, wenn der Identifikationsalgorithmus schneller lernt, als sich die Parameter der dynamischen Nichtlinearität ändern. Die Obergrenze der Lernschrittweite wird durch die zeitdiskrete Realisierung des Integrationsalgorithmus bestimmt, die Untergrenze hingegen durch $\dot{\underline{\theta}}$. Eine analytische Methode zur Bestimmung dieser Grenzen η_{min} und η_{max} ist jedoch bisher nicht bekannt.

Zusammenfassend kann somit festgestellt werden, dass das in Kapitel 6.2.1 vorgestellte Verfahren auch bei nicht messbaren Zuständen eingesetzt werden kann.

¹⁸Es gilt $e_e(s) = \mathcal{L} \{e_e\}$.

6.3 Identifikation von global integrierenden Systemen

In diesem Kapitel wird zum Zwecke der Veranschaulichung der vorgestellten Beobachtertheorie zur Identifikation dynamischer Nichtlinearitäten das Problem von global integrierenden Systemen noch einmal aufgegriffen. In Kapitel 3.2.2 wurde bereits darauf hingewiesen, dass lineare Modelle ohne Ausgangsrückkopplung nicht in der Lage sind, instabile bzw. grenzstabile Systeme zu identifizieren. Dies liegt daran, dass bei Modellen ohne Ausgangsrückkopplung die Identifikation auf der Impulsantwort beruht, die im Falle von instabilen bzw. global integrierenden Systemen nicht abklingt. Diese Problematik bleibt selbstverständlich bei den nichtlinearen Modellen ohne Ausgangsrückkopplung und den vorgestellten Erweiterungen aus Kapitel 4 und 5 erhalten.

In [30] wird für die Identifikation von global integrierenden Systemen eine Differentiation des Ausgangssignals vorgeschlagen, so dass wieder ein System mit abklingender Impulsantwort entsteht. In der Simulation ist diese Differentiation des unverrauschten Ausgangssignals leicht möglich, in der Praxis wird die Identifikation mittels des differenzierten Ausgangssignals aufgrund von Messrauschen schlechte Ergebnisse liefern.

Aus diesem Grund wird im Folgenden gezeigt, dass es mit Hilfe einer Beobachterstruktur möglich ist, global integrierende Systeme ohne Differentiation des Ausgangssignals zu identifizieren. Gleichzeitig wird gezeigt, dass es sich bei den Identifikationen um einfache Fälle der vorgestellten Beobachtertheorie handelt. Als konkrete Beispiele werden ein global integrierendes Hammerstein– sowie ein global integrierendes Wiener–Modell untersucht. Am Beispiel der Identifikation eines Wiener–Modells mittels Funktionsapproximatoransatz werden jedoch auch die Grenzen der vorgestellten Beobachtertheorie verdeutlicht, die sich daraus ergeben, dass die unbekannt Parameter in diesem Fall nichtlinear in den Modellausgang eingehen.

6.3.1 Identifikation eines global integrierenden Hammerstein–Modells

In Abb. 6.7 ist das global integrierende Hammerstein–Modell, das im Folgenden betrachtet wird, dargestellt. Entsprechend Abb. 6.3 kann das Gesamtsystem in eine unbekannt dynamische Nichtlinearität $\mathcal{NL}_{dyn}(u)$ und ein in seiner Struktur und in seinen Parametern bekanntes lineares System aufgespalten werden. Das Hammerstein–Modell stellt in diesem Fall die dynamische Nichtlinearität dar, während das bekannte lineare Teilsystem lediglich aus einem einzelnen Integrator besteht¹⁹. Das global integrierende Hammerstein–Modell kann entsprechend Gl. (6.1) wie folgt im Zustandsraum formuliert werden:

$$\dot{x} = \mathcal{NL}_{dyn}(u) \qquad y = x \qquad (6.40)$$

¹⁹Die Zeitkonstante des Integrators ist Eins und somit bekannt.

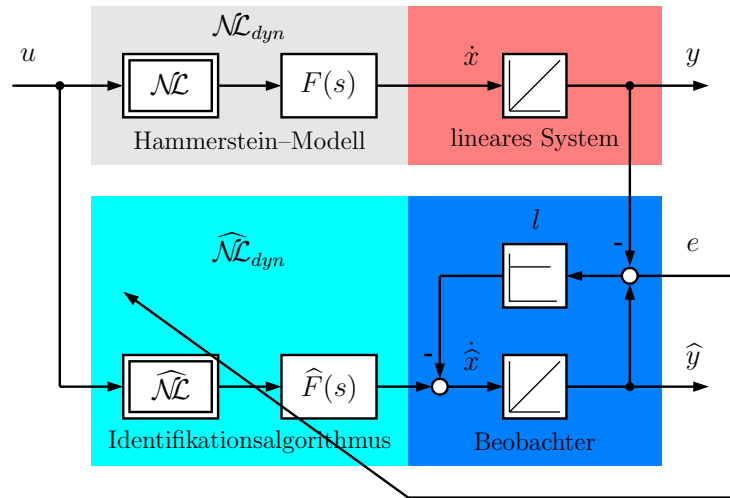


Abb. 6.7: Global integrierendes Hammerstein-Modell

Aus Abb. 6.7 wird anschaulich deutlich, wieso für ein global integrierendes System eine Beobachterstruktur zwingend erforderlich ist, wenn die Differentiation des Ausgangssignals vermieden werden soll. Während des Lernvorgangs entsprechen die geschätzten Parameter $\hat{\theta}$ der dynamischen Nichtlinearität nicht den optimalen Parametern $\underline{\theta}$, so dass der Identifikationsalgorithmus ein fehlerhaftes Signal liefert. Dieses fehlerhafte Signal führt dazu, dass selbst wenn die geschätzten Parameter im Verlauf der Identifikation mit den optimalen Parametern identisch werden, die Adaption der Parameter nicht aufhört, da aufgrund des anfänglich fehlerhaften Ausgangssignals, der Integrator des Modells einen unterschiedlichen Wert gespeichert hat, als der Integrator des Systems. Der Beobachterfehler e , mit dem die Adaption der Parameter erfolgt, wird somit niemals Null, so dass die fortwährende Adaption der Parameter letztendlich zu einem instabilen Identifikationsverlauf führt. Dasselbe Problem ergibt sich natürlich, wenn zu Anfang der Identifikation der Initialisierungswert des Modellintegrators nicht mit dem System übereinstimmt.

Abhilfe schafft die Identifikation in einer Beobachterstruktur. Mit dem Beobachterfehler e wird über den Rückführkoeffizienten l der Wert des Modellintegrators korrigiert. Somit können sowohl ein falscher Initialisierungswert als auch falsche Eingangssignale des Modellintegrators ausgeglichen werden. Für den Beobachter des bekannten linearen Teilsystems gilt²⁰:

$$\dot{\hat{x}} = \widehat{\mathcal{N}}_{dyn}(u) - l \cdot e \quad \hat{y} = \hat{x} \quad (6.41)$$

Für die Identifikation der dynamischen Nichtlinearität wird im Folgenden der Funktionsapproximatoransatz für das Hammerstein-Modell nach Kapitel 5.2.1 verwen-

²⁰Die für die Identifikation erforderliche Sichtbarkeit der Nichtlinearität kann leicht überprüft werden.

det. Für den Identifikationsalgorithmus gilt somit:

$$\widehat{\mathcal{N}}_{dyn}(u) = \widehat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad \text{mit} \quad \underline{\mathcal{A}}_{dyn}^T[k] = \left[\underline{\mathcal{A}}_1^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \tilde{\mathbf{R}}^T \right] \quad (6.42)$$

Der Identifikationsalgorithmus ist linear in den unbekanntenen Parametern $\widehat{\underline{\theta}}$, so dass abhängig von der Fehlerübertragungsfunktion $H(s)$, eines der in Kapitel 6.2 vorgestellten Fehlermodelle angewendet werden kann. Für das global integrierende Hammerstein-Modell ergibt sich die Fehlerübertragungsfunktion $H(s)$ nach Gl. (6.11) wie folgt:

$$H(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + l \cdot \underline{c}^T)^{-1} \cdot \underline{k} = \frac{1}{s + l} \quad (6.43)$$

Aus Gl. (6.43) geht hervor, dass es sich bei der Fehlerübertragungsfunktion um ein PT_1 -Glied handelt. Ein PT_1 -Glied erfüllt die SPR-Bedingung, so dass Fehlermodell 3 zur Parameteradaption angewendet werden kann²¹.

Anhand eines Simulationsbeispiels soll die Funktionsweise der Identifikation in Beobachterstruktur verdeutlicht werden. Es wird ein Hammerstein-Modell, wie es in Kapitel 5.2.2 gegeben ist, betrachtet. Die Identifikation erfolgt zunächst ohne Beobachterrückführung, wobei die Integratoren im Modell und im System gleich initialisiert sind. Anschließend erfolgt die Identifikation mit Beobachterrückführung. Für die Identifikation werden die Einstellwerte nach Tabelle 6.1 verwendet.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 100$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 21.3$	Formfaktor
$l = 0$ bzw. $l = 1$	Beobachterrückführung
$p = m_r \cdot q = 210$	Parameteranzahl
$h = 0.5s$	Abtastzeit

Tabelle 6.1: Einstellwerte für die Identifikation eines global integrierenden Hammerstein-Modells

In Abb. 6.8 sind die Ergebnisse bei einer Identifikation ohne Beobachterrückführung, d.h. bei $l = 0$, dargestellt.

²¹Es empfiehlt sich jedoch aufgrund kürzerer Konvergenzzeiten auch in diesem Fall Fehlermodell 4 anzuwenden.

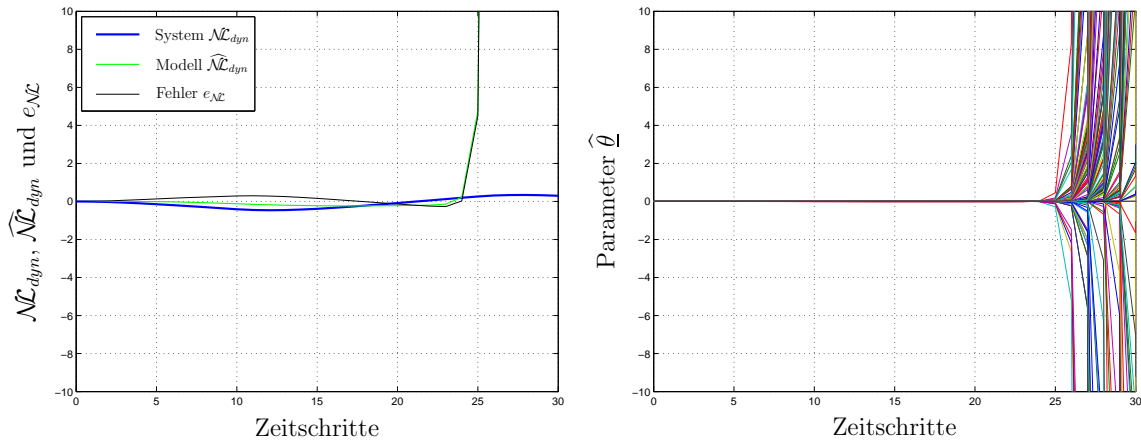


Abb. 6.8: *Identifikation ohne Beobachter — Identifikationsverlauf (links) und Verlauf der Parameter (rechts)*

In Abb. 6.8 links ist der Verlauf des Ausgangs der dynamischen Nichtlinearität $\mathcal{N}_{dyn}(u)$ sowie des Ausgangs des Identifikationsalgorithmus $\widehat{\mathcal{N}}_{dyn}(u)$ dargestellt. Zusätzlich ist der Fehler $e_{\mathcal{N}}$ zwischen der vorgegebenen und identifizierten dynamischen Nichtlinearität veranschaulicht. Dieser Fehler

$$e_{\mathcal{N}} = \widehat{\mathcal{N}}_{dyn} - \mathcal{N}_{dyn} \quad (6.44)$$

dient ausschließlich zur Visualisierung des Lernergebnisses. Die Adaption der Parameter erfolgt mit dem Fehler e . Wie in Abb. 6.8 aus dem Signal- und Parameterverlauf zu erkennen ist, wird die Identifikation bereits nach wenigen Zeitschritten instabil, was das Ergebnis der in diesem Kapitel beschriebenen Problematik ist.

Im Gegensatz dazu zeigt Abb. 6.9 den Fehler- und Parameterverlauf einer stabilen Identifikation in Beobachterstruktur.

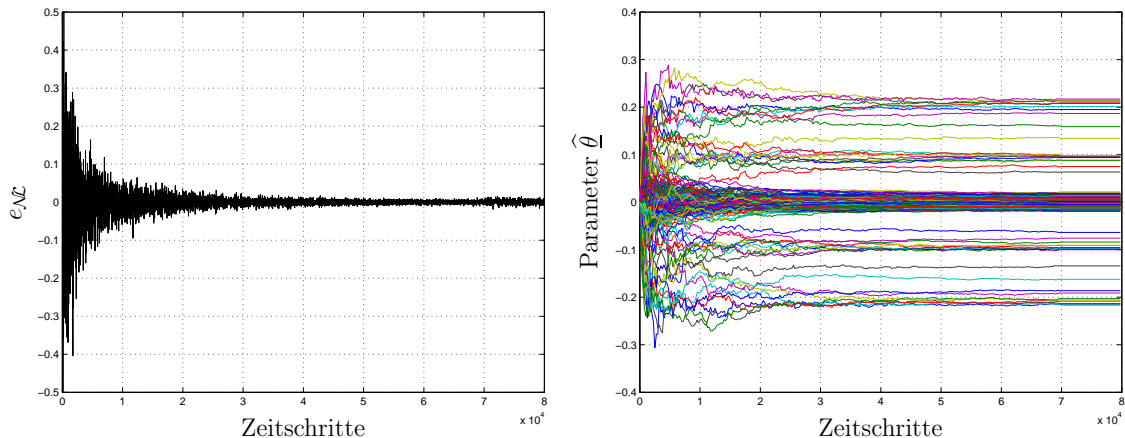


Abb. 6.9: *Identifikation mit Beobachter — Fehlerverlauf (links) und Konvergenz der Parameter (rechts)*

Die Simulationsdauer betrug insgesamt 80000 Zeitschritte. Nach 70000 Zeitschritten

wurde die Parameteradaption eingestellt und der Beobachterrückführkoeffizient auf Null gesetzt, so dass die identifizierte dynamische Nichtlinearität als reines Parallelmodell betrieben wurde. Das System wurde durch ein Zufallssignal im Bereich $[-1; 1]$ angeregt.

In Abb. 6.9 ist zu erkennen, dass der Fehler $e_{\mathcal{N}\mathcal{L}}$ sehr klein wird und die Parameter $\hat{\theta}$ relativ gut konvergieren. Ein kleiner Restfehler, der auf die begrenzte Anzahl an Stützwerten und Basisfunktionen zurück zu führen ist, bleibt jedoch, so dass die Parameter nicht auf einen exakten Wert konvergieren können. Dies führt auch dazu, dass der Fehler im Parallellauf leicht ansteigt. Dadurch wird jedoch nicht der Erfolg der Identifikation geschmälert, da ohne die vorgestellte Beobachterstruktur überhaupt keine Identifikation möglich gewesen wäre.

In Abb. 6.10 ist die identifizierte statische Nichtlinearität sowie die identifizierte Impulsantwort dargestellt.

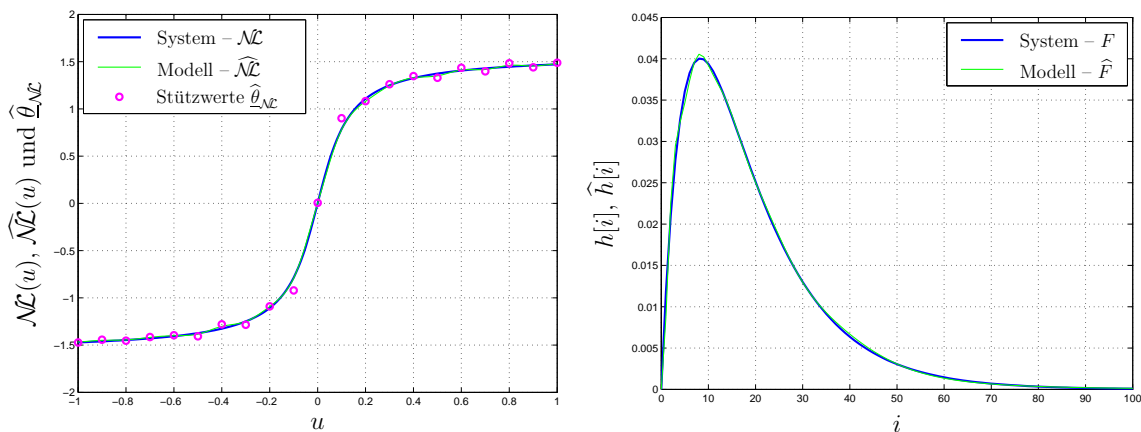


Abb. 6.10: Identifizierte statische Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

Auch die Lernergebnisse für die statische Nichtlinearität und die Impulsantwort verdeutlichen, dass sich die vorgestellte Beobachterstruktur hervorragend zur Identifikation von global integrierenden Systemen eignet. Wie sich jedoch im Folgenden Kapitel zeigen wird, ist für die Anwendung der vorgestellten Beobachtertheorie, die Linearität des Identifikationsalgorithmus in den unbekanntem Parametern zwingende Voraussetzung.

6.3.2 Identifikation eines global integrierenden Wiener-Modells

In diesem Kapitel wird die Identifikation eines global integrierenden Wiener-Modells genauer betrachtet. Hierbei müssen im Folgenden zwei Fälle unterschieden werden. Wie in Kapitel 6.2 bereits verdeutlicht wurde, ist für die Anwendung der vorgestellten Beobachtertheorie die Linearität des Identifikationsalgorithmus in den Parametern

tern zwingende Voraussetzung. Diese Linearität der Parameter ist beim Wiener-Modell lediglich im Falle des Polynomansatzes gegeben, wie ein Vergleich mit Kapitel 3.3.4.2 bzw. 3.3.4.3 zeigt. Auch die Erweiterungen hinsichtlich Mehrgrößensystemen auf Basis des Wiener-Modells sind linear in den Parametern, so dass die vorgestellte Beobachtertheorie uneingeschränkt anwendbar ist. Im Gegensatz dazu ist das Wiener-Modell im Falle des Funktionsapproximatoransatzes nichtlinear in den Parametern, wie die Gleichungen aus Kapitel 5.3 verdeutlichen. Dennoch ist die Identifikation eines global integrierenden Wiener-Modells auf Basis des Funktionsapproximatoransatzes möglich, wie im Folgenden gezeigt wird. Zunächst ist in Abb. 6.11 ein global integrierendes Wiener-Modell veranschaulicht.

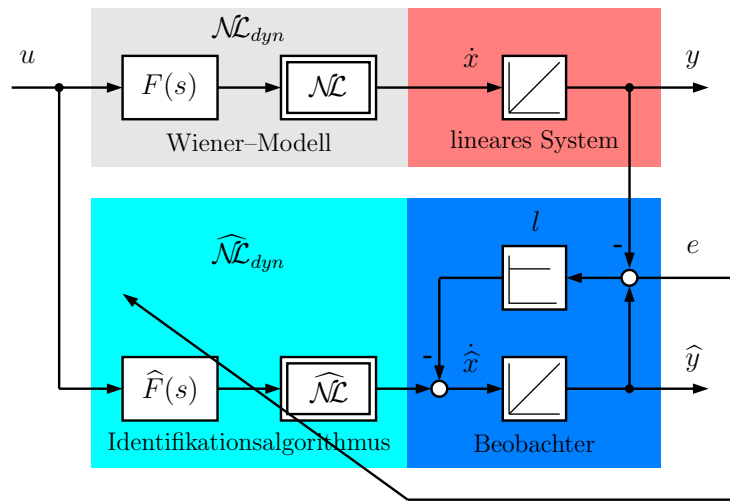


Abb. 6.11: Global integrierendes Wiener-Modell

Die formale Beschreibung eines global integrierenden Wiener-Modells und des zugehörigen Beobachters entspricht den Gleichungen (6.40) und (6.41). Bei der Berechnung des Ausgangssignals $\hat{\mathcal{N}}_{dyn}(u)$ des Identifikationsalgorithmus muss unterschieden werden, ob zur Identifikation des Wiener-Modells der Polynom- oder Funktionsapproximatoransatz verwendet wird. Bei Verwendung des Polynomansatzes kann das Ausgangssignal entsprechend Gl. (3.61) wie folgt berechnet werden:

$$\hat{\mathcal{N}}_{dyn}(u) = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad \text{mit} \quad \underline{\mathcal{A}}_{dyn}^T[k] = \left[1, \underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \left(\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*q} \right] \quad (6.45)$$

Wie aus Gl. (6.45) deutlich zu erkennen ist, ist der Identifikationsalgorithmus linear in den unbekannt Parametern $\hat{\underline{\theta}}$, so dass abhängig von der Fehlerübertragungsfunktion $H(s)$, uneingeschränkt Fehlermodell 3 bzw. 4 angewendet werden kann.

Wird zur Identifikation des Wiener-Modells der Funktionsapproximatoransatz verwendet, berechnet sich das Ausgangssignal des Identifikationsalgorithmus folgendermaßen:

$$\hat{\mathcal{N}}_{dyn}(u) = \hat{\underline{\theta}}_{\mathcal{N}}^T \cdot \underline{\mathcal{A}} \left(u, \hat{\underline{\theta}}_{lin} \right) = \hat{\theta}_{\mathcal{N},1} \cdot \mathcal{A}_1 \left(u, \hat{\underline{\theta}}_{lin} \right) + \dots + \hat{\theta}_{\mathcal{N},q} \cdot \mathcal{A}_q \left(u, \hat{\underline{\theta}}_{lin} \right) \quad (6.46)$$

mit

$$\mathcal{A}_i(u, \hat{\theta}_{lin}) = \frac{\exp\left[\frac{(\hat{\theta}_{lin,1} \cdot \underline{u}^T \cdot \tilde{r}_1 + \dots + \hat{\theta}_{lin,m_r} \cdot \underline{u}^T \cdot \tilde{r}_{m_r} - \chi_i)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{(\hat{\theta}_{lin,1} \cdot \underline{u}^T \cdot \tilde{r}_1 + \dots + \hat{\theta}_{lin,m_r} \cdot \underline{u}^T \cdot \tilde{r}_{m_r} - \chi_j)^2}{-2\sigma^2}\right]} \quad i = 1 \dots q \quad (6.47)$$

Wie aus den Gleichungen (6.46) und (6.47) zu erkennen ist, gehen die Parameter der Übertragungsfunktion in die Aktivierungsfunktionen des GRNN und somit nichtlinear in den Ausgang des Identifikationsalgorithmus ein. Diese Problematik trat bereits in Kapitel 5.3 auf. Für das Wiener-Modell bedeutet dies, dass bei Verwendung des Funktionsapproximatoransatzes die in Kapitel 6.2 vorgestellte Beobachtertheorie nicht anwendbar ist. Das heißt jedoch nicht, dass das global integrierende Wiener-Modell in diesem Fall nicht identifiziert werden kann. Das dargelegte Problem ist in ähnlicher Form bereits aus den strukturierten rekurrenten Netzen [16] bekannt. Im Hinblick auf die Kombination der vorgestellten Verfahren zur Identifikation dynamischer Nichtlinearitäten mit den strukturierten rekurrenten Netzen in Kapitel 8, soll an dieser Stelle die Theorie aus [16] aufgegriffen und zur Identifikation eines global integrierenden Wiener-Modells verwendet werden.

Zur Adaption der Parameter wird analog zu Kapitel 5.3.1.2 das Gradientenabstiegsverfahren verwendet. Die Adaptionsgleichungen lauten in zeitkontinuierlicher Schreibweise²²:

$$\begin{aligned} \dot{\hat{\theta}}_{\mathcal{NL},i} &= -\eta_{\mathcal{NL}} \cdot e \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}_{\mathcal{NL},i}} & i = 1 \dots q \\ \dot{\hat{\theta}}_{lin,i} &= -\eta_{lin} \cdot e \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}_{lin,i}} & i = 1 \dots m_r \end{aligned} \quad (6.48)$$

Zur Bestimmung der partiellen Ableitungen für die Parameter der statischen Nichtlinearität und der Übertragungsfunktion werden zunächst die Beobachtergleichungen (6.41) betrachtet. Es gilt:

$$\frac{\partial \hat{x}}{\partial \hat{\theta}_{\mathcal{NL},i}} = \frac{\partial \widehat{\mathcal{NL}}_{dyn}(u)}{\partial \hat{\theta}_{\mathcal{NL},i}} - l \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}_{\mathcal{NL},i}} + l \cdot \underbrace{\frac{\partial y}{\partial \hat{\theta}_{\mathcal{NL},i}}}_{=0} \quad (6.49)$$

Durch das Einsetzen von $\hat{x} = \hat{y}$ folgt:

$$\frac{\partial \hat{y}}{\partial \hat{\theta}_{\mathcal{NL},i}} + l \cdot \frac{\partial \hat{y}}{\partial \hat{\theta}_{\mathcal{NL},i}} = \frac{\partial \widehat{\mathcal{NL}}_{dyn}(u)}{\partial \hat{\theta}_{\mathcal{NL},i}} \quad (6.50)$$

²²Im Zusammenhang mit dem Beobachter wird die zeitkontinuierliche Schreibweise bevorzugt. Eine zeitdiskrete Darstellung ist jedoch z.B. durch Verwendung der Euler-Vorwärts-Integrationsregel leicht möglich.

Wird $\partial \widehat{\mathcal{N}}_{dyn}(u) / \partial \widehat{\theta}_{\mathcal{NL},i}$ als Eingangs- und $\partial \widehat{y} / \partial \widehat{\theta}_{\mathcal{NL},i}$ als Ausgangssignal einer zeitkontinuierlichen Übertragungsfunktion betrachtet, so ergibt sich durch die Transformation von Gl. (6.50) in den Laplace-Bereich:

$$\mathcal{L} \left\{ \frac{\partial \widehat{y}}{\partial \widehat{\theta}_{\mathcal{NL},i}} \right\} = \frac{1}{s+l} \cdot \mathcal{L} \left\{ \frac{\partial \widehat{\mathcal{N}}_{dyn}(u)}{\partial \widehat{\theta}_{\mathcal{NL},i}} \right\} \quad (6.51)$$

Analog zu Gl. (6.51) ergibt sich für die partiellen Ableitungen nach den linearen Parametern:

$$\mathcal{L} \left\{ \frac{\partial \widehat{y}}{\partial \widehat{\theta}_{lin,i}} \right\} = \frac{1}{s+l} \cdot \mathcal{L} \left\{ \frac{\partial \widehat{\mathcal{N}}_{dyn}(u)}{\partial \widehat{\theta}_{lin,i}} \right\} \quad (6.52)$$

Die partiellen Ableitungen der dynamischen Nichtlinearität $\widehat{\mathcal{N}}_{dyn}$ nach den unbekannt Parametern wurden bereits in den Gleichungen (5.25) und (5.26) berechnet. Durch das Einsetzen dieser Gleichungen folgt:

$$\mathcal{L} \left\{ \frac{\partial \widehat{y}}{\partial \widehat{\theta}_{\mathcal{NL},i}} \right\} = \frac{1}{s+l} \cdot \mathcal{L} \left\{ \mathcal{A}_i \left(u, \widehat{\theta}_{lin} \right) \right\} \quad (6.53)$$

$$\mathcal{L} \left\{ \frac{\partial \widehat{y}}{\partial \widehat{\theta}_{lin,i}} \right\} = \frac{1}{s+l} \cdot \mathcal{L} \left\{ \frac{u^T \cdot \tilde{r}_i}{\sigma^2} \cdot \sum_{j=1}^q \widehat{\theta}_{\mathcal{NL},j} \cdot \mathcal{A}_j \cdot \left\{ \sum_{l=1}^q [(\widehat{v} - \chi_l) \cdot \mathcal{A}_l] - (\widehat{v} - \chi_j) \right\} \right\} \quad (6.54)$$

Das geschätzte Eingangssignal \widehat{v} der statischen Nichtlinearität berechnet sich aus der Faltung des Eingangssignals u mit der gewichteten Überlagerung von orthonormalen Basisfunktionen. Die Berechnungsvorschrift kann aus Gl. (6.47) entnommen werden. Mit den Gleichungen (6.48) und (6.53) bzw. (6.54) sind die Lerngesetze für die unbekannt Parameter vollständig gegeben. Anschaulich bedeuten die Gleichungen, dass die Ausgangssignale der partiellen Ableitungen der dynamischen Nichtlinearität nach den unbekannt Parametern durch ein dynamisches System verzögert werden. Dieses dynamische System stellt die Übertragungsfunktion des Beobachters für das bekannte lineare Teilsystem dar.

Die Identifikation eines global integrierenden Wiener-Modells soll im Folgenden an einem Simulationsbeispiel veranschaulicht werden. Als Beispiel dient das Wiener-Modell aus Kapitel 5.3.2. Für die Identifikation werden die Einstellwerte nach Tabelle 6.2 verwendet. Die Simulationsdauer beträgt insgesamt $7 \cdot 10^6$ Zeitschritte. Nach $6.4 \cdot 10^6$ Zeitschritten wird die Parameteradaption eingestellt und der Beobachterrückführkoeffizient auf Null gesetzt, so dass die identifizierte dynamische Nichtlinearität als reines Parallelmodell betrieben wird. Das System wird durch ein Zufallssignal im Bereich $[-1; 1]$ angeregt. In Abb. 6.12 ist der Fehlerverlauf dargestellt.

Einstellwerte	Erläuterung
$q = 41$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 100$	Antwortlänge
$m_r = 20$	Basisfunktionenanzahl
$\zeta = 21.3$	Formfaktor
$l = 1$	Beobachterrückführung
$p = m_r + q = 61$	Parameteranzahl
$\eta_{lin} = 8.0 \cdot 10^{-5}$	Lernschrittweite für $\hat{\theta}_{lin,i}$
$\eta_{NL} = 5.0 \cdot 10^{-3}$	Lernschrittweite für $\hat{\theta}_{NL,i}$
$h = 0.5s$	Abtastzeit

Tabelle 6.2: Einstellwerte für die Identifikation eines global integrierenden Wiener-Modells

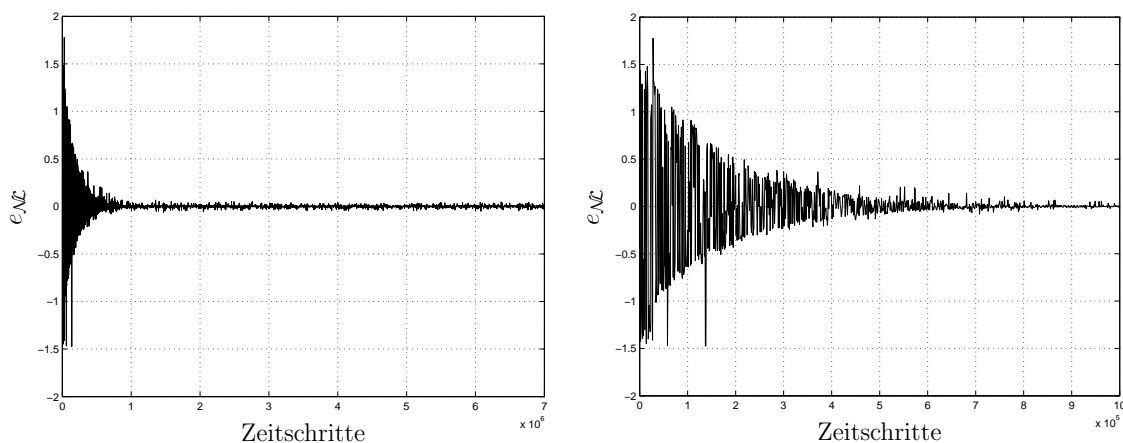


Abb. 6.12: Identifikation mit Beobachter — Fehlerverlauf (links) und Vergrößerung bei Lernbeginn (rechts)

Es ist zu erkennen, dass der Fehler e_{NL} sehr klein wird, jedoch bleibt ein kleiner Restfehler, der auf die begrenzte Anzahl an Stützpunkten und Basisfunktionen zurück zu führen ist. Ein Anstieg des Fehlers im Parallellauf ist in Abb. 6.12 links nicht zu erkennen, was die Qualität des Ergebnisses unterstreicht. In Abb. 6.13 ist das Konvergenzverhalten der Parameter der statischen Nichtlinearität (links) und der Übertragungsfunktion (rechts) veranschaulicht. Sowohl die Parameter der statischen Nichtlinearität als auch die der Übertragungsfunktion zeigen ein sehr gutes Konvergenzverhalten. In Abb. 6.14 ist die identifizierte statische Nichtlinearität sowie die identifizierte Impulsantwort dargestellt. Die Lernergebnisse für die statische Nichtlinearität und die Impulsantwort verdeutlichen, dass mit Hilfe einer Beobachterstruktur die Identifikation von global integrierenden Wiener-Modellen auch für den Fall möglich ist, dass die unbekannt Parameter nichtlinear in den Ausgang des Identifikationsalgorithmus eingehen. Hierzu müssen jedoch die Adaptionsgleichun-

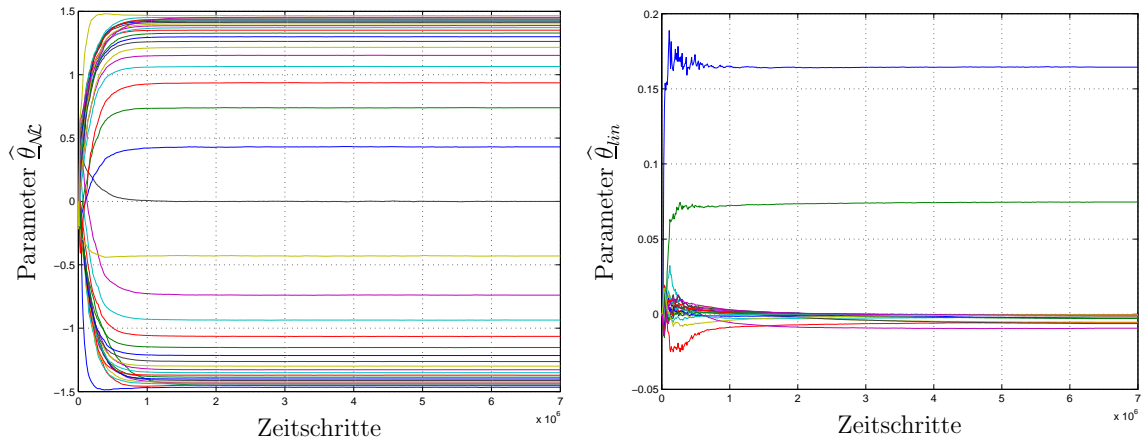


Abb. 6.13: Identifikation mit Beobachter — Konvergenz der Parameter

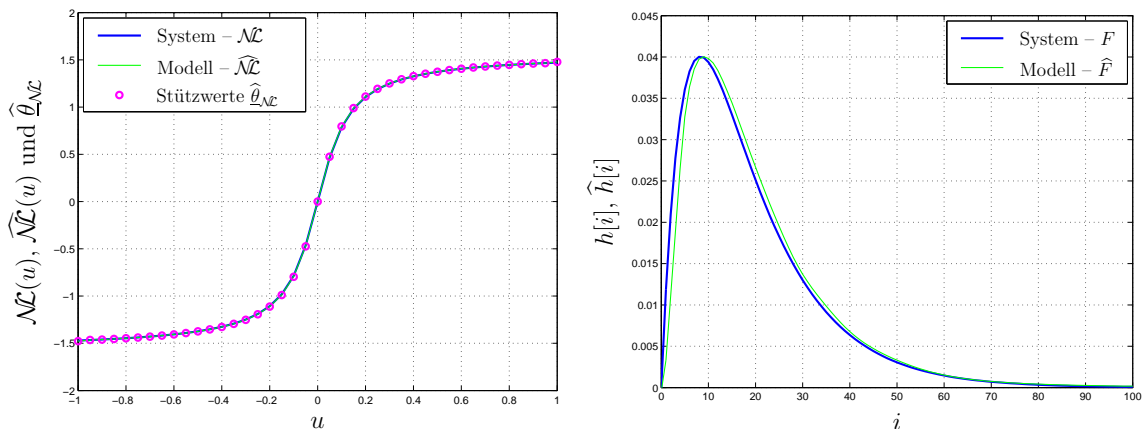


Abb. 6.14: Identifizierte statische Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

gen entsprechend angepasst werden. Trotz der sehr guten Identifikationsergebnisse, ist ein deutlicher Nachteil gegenüber dem Fall zu erkennen, dass die unbekannt Parameter linear in den Ausgang des Identifikationsalgorithmus eingehen und somit eines der vorgestellten Fehlermodelle verwendet werden kann. Ein Vergleich der Konvergenzzeiten beim global integrierenden Hammerstein- und Wiener-Modell zeigt, dass die Parameterkonvergenz beim global integrierenden Hammerstein-Modell und somit bei Verwendung der vorgestellten Fehlermodelle ca. um den Faktor 10 schneller erfolgt. Die Aufhebung der Einschränkung, dass der Identifikationsalgorithmus linear in den unbekannt Parametern sein muss, wird somit durch längere Konvergenzzeiten erkauft.

6.4 Beobachterentwurf und Identifikation — Simulationsbeispiel

Nachdem in Kapitel 6.3 global integrierende Systeme als einfachste Form einer Identifikation von dynamischen Nichtlinearitäten mit Hilfe der vorgestellten Beobachtertheorie untersucht wurden, wird in diesem Kapitel ein praxisbezogeneres Simulationsbeispiel betrachtet [19]. Das betrachtete System mit dynamischer Nichtlinearität und der zugehörige Beobachter einschließlich Identifikationsalgorithmus ist in Abb. 6.15 veranschaulicht.

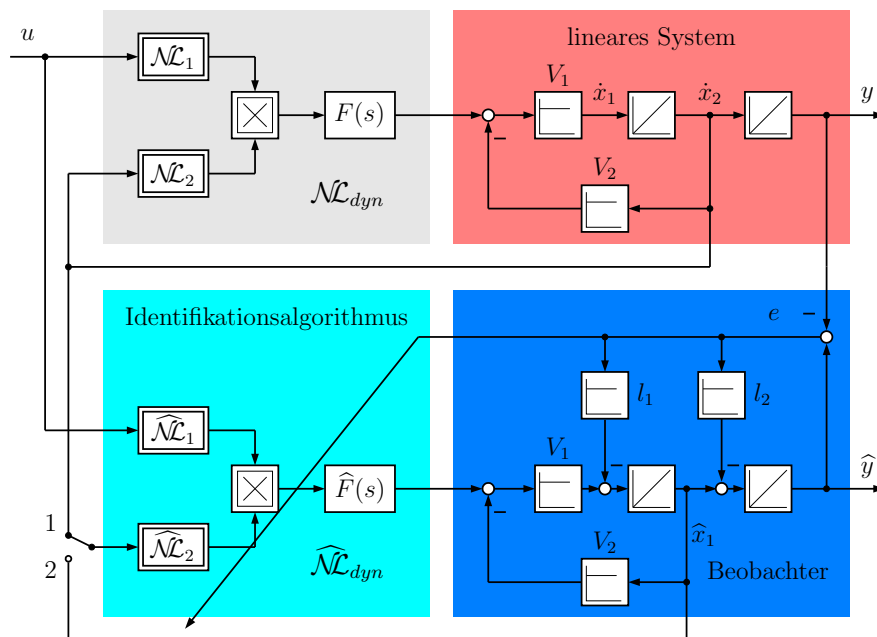


Abb. 6.15: Identifikationsbeispiel

Das System kann entsprechend Gl. (6.1) im Zustandsraum wie folgt dargestellt werden:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{\underline{x}}} = \underbrace{\begin{bmatrix} -V_1 \cdot V_2 & 0 \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\underline{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\underline{b}} \cdot u + \underbrace{\begin{bmatrix} V_1 \\ 0 \end{bmatrix}}_{\underline{k}} \cdot \mathcal{N}_{dyn}(x_1, u) \quad (6.55)$$

$$y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\underline{c}^T} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Die Parameter V_1 und V_2 des linearen dynamischen Teilsystems sind bekannt und zeitinvariant. Folglich gilt dies auch für die Systemmatrix \mathbf{A} und den Einkopplungsvektor \underline{k} der dynamischen Nichtlinearität. Der Einkopplungsvektor \underline{b} des Systemein-

gangs ist Null, da der Systemeingang ausschließlich über die dynamische Nichtlinearität auf den ersten Zustand wirkt. Der Auskopplungsvektor \underline{c} ist durch den Zusammenhang, dass der zweite Systemzustand gleichzeitig der Ausgang des Systems ist, festgelegt. Wie sich in Kapitel 7 noch zeigen wird, ist diese Struktur des bekannten linearen Teilsystems typisch für mechanische Anordnungen. Der Eingang des linearen Teilsystems stellt in der Regel ein Moment dar. Dieses Moment beschleunigt eine träge Masse, die durch den Parameter V_1 charakterisiert ist²³. Die Zustände x_1 und x_2 beschreiben die Winkelgeschwindigkeit und die Position der mechanischen Anordnung. Der Parameter V_2 kann z.B. als winkelgeschwindigkeitsabhängiges Reibmoment betrachtet werden, das dem antreibenden Moment und somit der Bewegung entgegen wirkt²⁴. Als dynamische Nichtlinearität wird ein Hammerstein-Modell mit zweidimensionaler statischer Nichtlinearität betrachtet. Die Eingangsgrößen sind der Systemeingang u und der Zustand $x_{\mathcal{NL}} = x_1$. Die dynamische Nichtlinearität beschreibt in der Regel das nichtlineare dynamische Übertragungsverhalten eines Stellgliedes, das als Ausgangssignal ein Moment liefert. Das Eingangssignal u stellt folglich den Momentensollwert, als Ausgangssignal einer Regelung, dar. Zusätzliche Abhängigkeiten der dynamischen Nichtlinearität, wie z.B. vom Zustand x_1 sind möglich. An dieser Stelle soll noch einmal die Problematik des strukturellen Vorwissens veranschaulicht werden. Wie an dem Beispiel deutlich wird, ist die Struktur des mechanischen Teilsystems in der Regel exakt bekannt, da die Zusammenhänge zwischen Moment, Winkelgeschwindigkeit und Position bestens bekannt sind. Auch die linearen Parameter, wie z.B. das Trägheitsmoment, können relativ leicht berechnet werden²⁵. Im Gegensatz dazu gibt es für das Übertragungsverhalten von Stellgliedern in der Regel nur eine grobe Strukturvorstellung. Es ist beispielsweise bekannt, dass Stellglieder eine Stellgrößenbeschränkung besitzen, die in Form einer statischen Nichtlinearität angenommen werden kann und außerdem selbst ein dynamisches Übertragungsverhalten aufweisen, das sich aus der Momentenregelung und der Dynamik des Stellgliedes ergibt. Eine exakte physikalische Modellierung dieses Übertragungsverhaltens wäre mit erheblichem Aufwand verbunden, so dass für das dynamische Verhalten des Stellgliedes eine einfache Übertragungsfunktion angenommen wird ohne die exakte Struktur zu wissen.

Die Identifikation der dynamischen Nichtlinearität in der Beobachterstruktur setzt die Sichtbarkeit der dynamischen Nichtlinearität und die Beobachtbarkeit des linearen Teilsystems²⁶ voraus. Für die Übertragungsfunktion $H_S(s)$ zur Überprüfung der Sichtbarkeit gilt entsprechend Gl. (6.3):

$$H_S(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot \underline{k} = \frac{V_1}{s^2 + s \cdot V_1 \cdot V_2} \neq 0 \quad (6.56)$$

²³Der Parameter V_1 stellt das reziproke Trägheitsmoment dar.

²⁴Die Modellierung einer Reibung als linearer Parameter ist selbstverständlich in Realität nicht ausreichend. An dieser Stelle soll jedoch nur ein qualitativer Bezug zur Praxis hergestellt werden.

²⁵Ist dies nicht der Fall müssen diese Parameter ebenfalls in die Identifikation einbezogen werden. Hierzu sei auf die strukturierten rekurrenten Netze in [16] bzw. Kapitel 8 verwiesen.

²⁶Zustandsbeobachtbarkeit muss nur vorausgesetzt werden, wenn der Eingangsraum der dynamischen Nichtlinearität nicht messbar ist.

Somit ist die dynamische Nichtlinearität am Systemausgang sichtbar und kann mit dem Beobachterfehler e gelernt werden.

Für die Beobachtbarkeit des linearen Teilsystems gilt:

$$\det \mathbf{Q}_{obs} = \det [\underline{c}, \mathbf{A}^T \underline{c}] = \det \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = -1 \neq 0 \quad (6.57)$$

Damit sind alle Voraussetzungen für den Beobachterentwurf erfüllt. Der Beobachter kann im Falle eines nicht messbaren Eingangsraumes der dynamischen Nichtlinearität entsprechend Gl. (6.26) wie folgt beschrieben werden:

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} -V_1 \cdot V_2 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} V_1 \\ 0 \end{bmatrix} \cdot \widehat{\mathcal{N}}_{dyn}(\hat{x}_1, u) - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \cdot e \quad (6.58)$$

$$\hat{y} = [0 \quad 1] \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}$$

Die Gleichungen (6.58) gelten für den Fall, dass der Eingangsraum der dynamischen Nichtlinearität nicht messbar ist (vgl. Abb. 6.15, Fall 2). Ist im Gegensatz dazu der Eingangsraum messbar (vgl. Abb. 6.15, Fall 1), so ist die dynamische Nichtlinearität von x_1 und u abhängig, d.h. $\widehat{\mathcal{N}}_{dyn}(x_1, u)$. Der Beobachterentwurf für das lineare Teilsystem kann mit bekannten Verfahren erfolgen, so z.B. durch Polvorgabe oder nach der Kalman–Filter–Theorie.

Für die Identifikation der dynamischen Nichtlinearität wird der Algorithmus aus Kapitel 5.4.1 verwendet. Somit gilt für den Identifikationsalgorithmus:

$$\widehat{\mathcal{N}}_{dyn}(\hat{x}_1, u) = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (6.59)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[\underline{\mathcal{A}}_1^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \tilde{\mathbf{R}}^T \right] \\ \underline{\mathcal{A}}_i^T[k] &= \left[\mathcal{A}_i(\hat{x}_1[k-1], u[k-1]), \dots, \mathcal{A}_i(\hat{x}_1[k-m], u[k-m]) \right] \end{aligned}$$

Zur Adaption der unbekannt Parameter $\hat{\underline{\theta}}$ muss die Fehlerübertragungsfunktion $H(s)$ nach Gl. (6.11) bestimmt werden. Für $H(s)$ ergibt sich:

$$H(s) = \frac{e(s)}{e_{\mathcal{N}}(s)} = \frac{V_1}{s^2 + s \cdot (V_1 \cdot V_2 + l_2) + (l_1 + V_1 \cdot V_2 \cdot l_2)} \quad (6.60)$$

Die Fehlerübertragungsfunktion erfüllt in diesem Fall nicht die SPR–Bedingung, so dass Fehlermodell 4 angewendet werden muss.

Im Folgenden wird angenommen, dass für die zweidimensionale statische Nichtlinearität und die Übertragungsfunktion gilt:

$$\begin{aligned} \mathcal{N}(x_1, u) &= \mathcal{N}_1(u) \cdot \mathcal{N}_2(x_1) = [0.5 \cdot \arctan(5 \cdot u)] \cdot [\arctan(5 \cdot x_1)^2 + 1] \\ F(s) &= \frac{1}{21 \cdot s^2 + 10 \cdot s + 1} \end{aligned}$$

Für die Parameter des bekannten linearen Teilsystems gilt:

$$V_1 = 0.02 \qquad V_2 = 3.85$$

Für die Identifikation werden die Einstellwerte nach Tabelle 6.3 verwendet.

Einstellwerte	Erläuterung
$q = 11^2 = 121$	Stützstellenanzahl
$\sigma_{norm1} = \sigma_{norm2} = \sigma_{norm} = 0.6$	normierte Standardabweichung des GRNN
$m = 50$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.65$	Formfaktor
$\underline{l}^T = [0.52, 1.02]$	Beobachterrückführungen
$p = m_r \cdot q = 1210$	Parameteranzahl
$\eta = 5$	Lernschrittweite
$h = 1s$	Abtastzeit

Tabelle 6.3: Einstellwerte für die Identifikation

Die Simulationsdauer beträgt insgesamt $1 \cdot 10^6$ Zeitschritte. Nach $9 \cdot 10^5$ Zeitschritten wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird durch ein Zufallssignal im Bereich $u \in [-1; 1]$ angeregt, so dass sich $x_1 \in [-0.4; 0.4]$ ergibt. Zunächst erfolgt die Identifikation unter der Annahme, dass x_1 messbar ist. In Abb. 6.16 ist der Fehlerverlauf (links) und die Konvergenz ausgewählter Parameter (rechts) dargestellt.

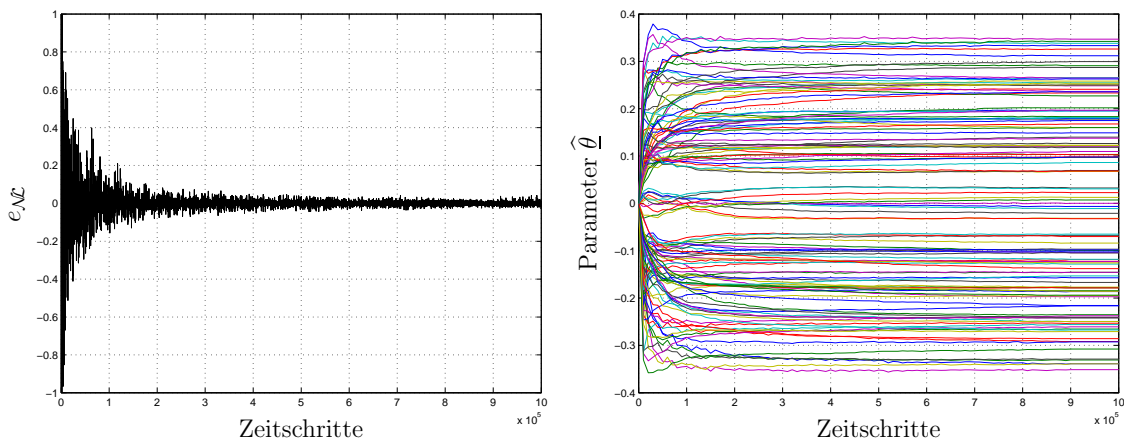


Abb. 6.16: Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz ausgewählter Parameter (rechts)

Es ist zu erkennen, dass der Fehler e_{NL} klein wird und die Parameter konvergieren. Auch im Parallelbetrieb steigt der Fehler nicht merklich an. Allerdings bleibt aufgrund der relativ geringen Anzahl an Basisfunktionen und der begrenzten Stützwertanzahl ein Restfehler. In Abb. 6.17 sind die vorgegebene und die identifizierte

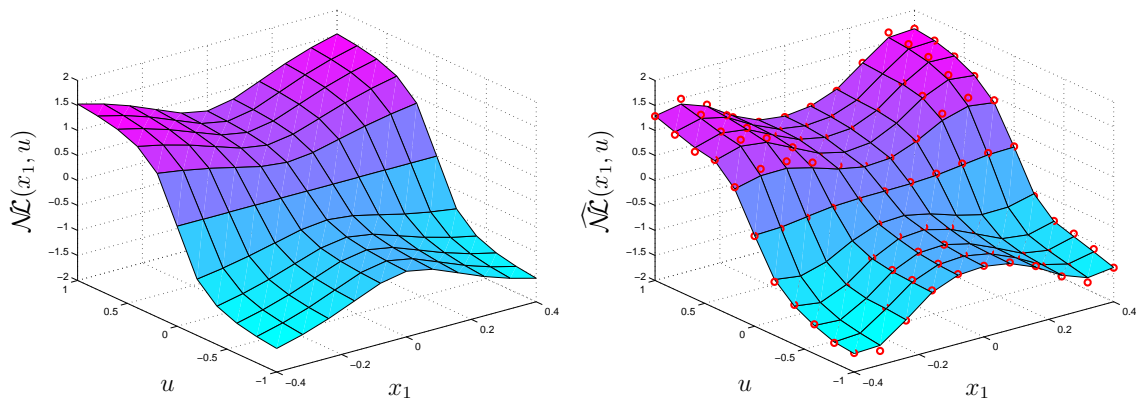


Abb. 6.17: Vergleich zwischen vorgegebener (links) und identifizierter (rechts) statischer Nichtlinearität

statische Nichtlinearität einander gegenüber gestellt. Die statische Nichtlinearität wird durch das GRNN relativ gut identifiziert. Dadurch, dass das Eingangssignal x_1 nicht direkt vorgegeben werden kann, wird die Nichtlinearität bezüglich x_1 in den Randbereichen nicht genug angeregt. Es wurde versucht, die dadurch bedingte langsamere Konvergenz der Parameter durch eine etwas erhöhte Standardabweichung des GRNN auszugleichen. Dies führt insgesamt zu einem verschliffeneren Verlauf der identifizierten statischen Nichtlinearität. In Abb. 6.18 ist der Approximationsfehler der statischen Nichtlinearität und eine identifizierte Impulsantwort veranschaulicht.

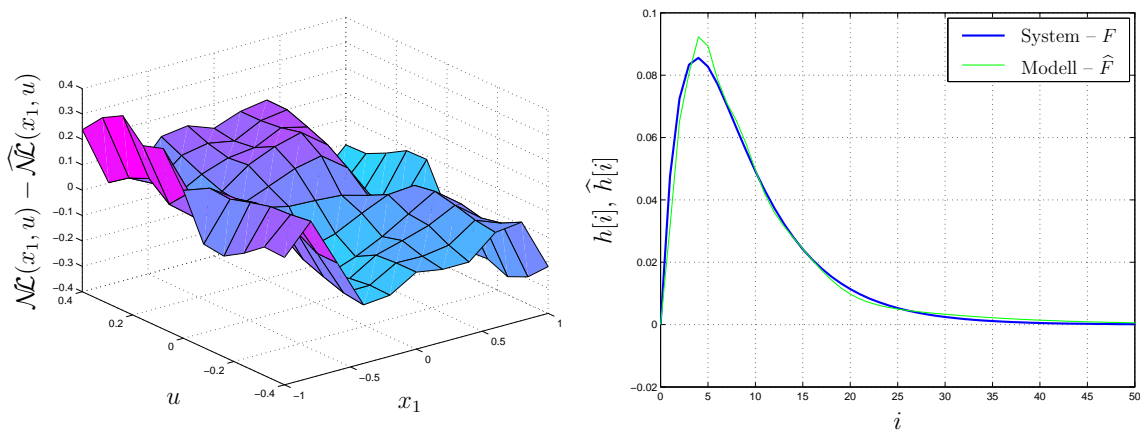


Abb. 6.18: Identifikationsfehler der statischen Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

Ein Approximationsfehler in der statischen Nichtlinearität ist vor allem in den Randbereichen von x_1 zu erkennen, was aus der Anregung in diesen Bereichen resultiert. Dieser Fehler führt zusammen mit der begrenzten Basisfunktionenanzahl und dem relativ großen Abtastzeitschritt auch zu Fehlern in der Impulsantwort. Dennoch ist

das Identifikationsergebnis im Hinblick auf den sehr kleinen Fehler im Parallelbetrieb des Modells (vgl. Abb. 6.16 links) als sehr gut zu bewerten.

Im Folgenden wird angenommen, dass der Eingangsraum der dynamischen Nichtlinearität nicht vollständig gemessen werden kann, d.h. dass die Eingangsgröße x_1 nicht messbar ist und zur Identifikation die beobachtete Zustandsgröße \hat{x}_1 verwendet werden muss. Für die Identifikation gelten die Einstellungen und Rahmenbedingungen, wie für den Fall des vollständig messbaren Eingangsraumes. In Abb. 6.19 ist der Fehlerverlauf (links) und die Konvergenz ausgewählter Parameter (rechts) dargestellt.

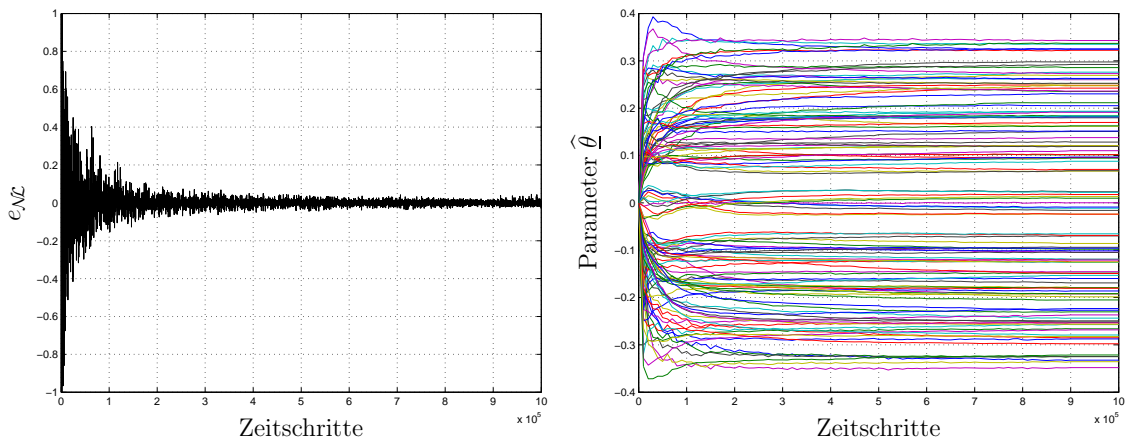


Abb. 6.19: Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz ausgewählter Parameter (rechts)

Es ist zu erkennen, dass der Fehler $e_{\mathcal{N}}$ ebenfalls klein wird und die Parameter konvergieren. In Abb. 6.20 sind die vorgegebene und die identifizierte statische Nichtlinearität einander gegenüber gestellt.

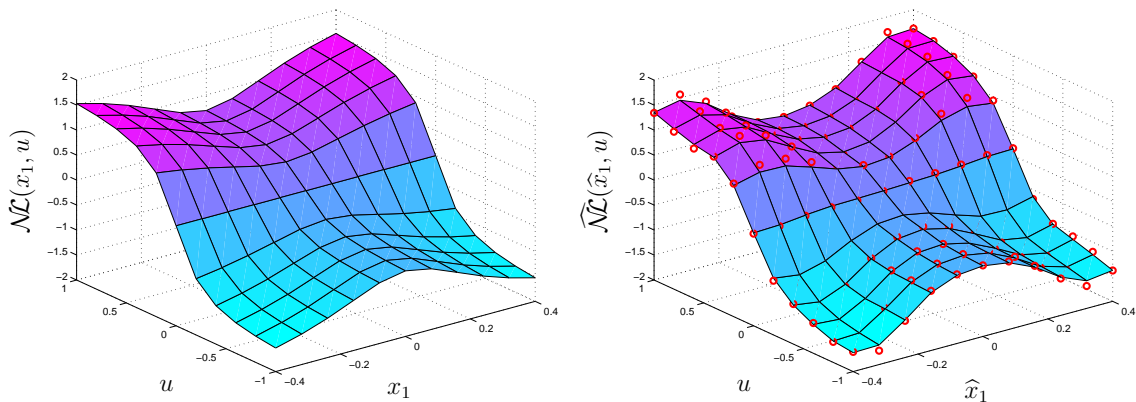


Abb. 6.20: Vergleich zwischen vorgegebener (links) und identifizierter (rechts) statischer Nichtlinearität

Das Ergebnis für die statische Nichtlinearität ist kaum zu unterscheiden von dem Ergebnis bei vollständig messbarem Eingangsraum der dynamischen Nichtlinearität. In Abb. 6.21 links ist der Approximationsfehler der statischen Nichtlinearität und rechts die identifizierte Impulsantwort veranschaulicht.

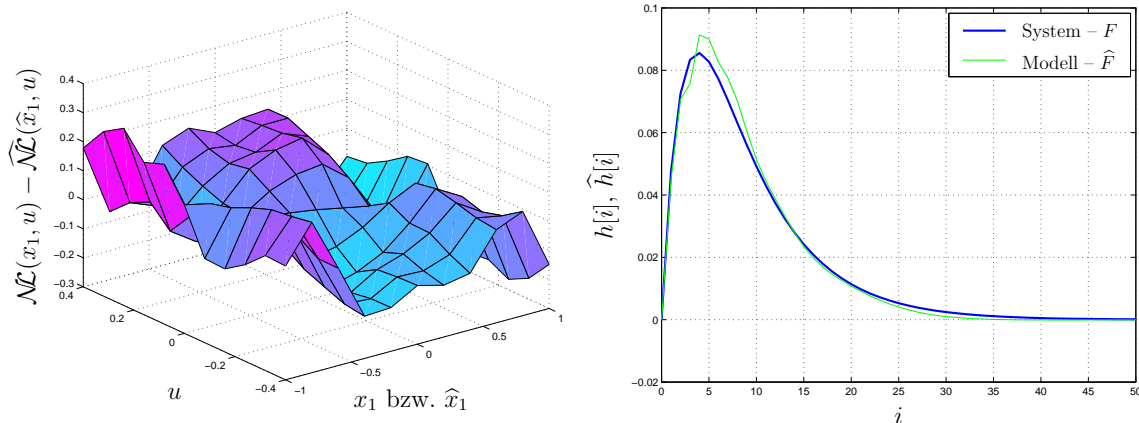


Abb. 6.21: Identifikationsfehler der statischen Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

Auch der Approximationsfehler und die identifizierte Impulsantwort zeigen nur marginale Unterschiede, so dass an dieser Stelle festgehalten werden kann, dass auch bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität eine Identifikation mit sehr guten Ergebnissen erzielt werden kann.

6.5 Kurzzusammenfassung

In diesem Kapitel wurde gezeigt, dass es mit Hilfe eines Beobachterentwurfs möglich ist, dynamische Nichtlinearitäten innerhalb eines komplexeren Gesamtsystems zu identifizieren. Hierzu wurde zunächst der Begriff der dynamischen Nichtlinearität definiert und die betrachtete Klasse von nichtlinearen dynamischen Systemen im Zustandsraum beschrieben. Es wurde gezeigt, dass es durch einen Beobachterentwurf für das bekannte lineare Teilsystem möglich ist, eine dynamische Nichtlinearität, die in ihrer Struktur und in ihren Parametern weitestgehend unbekannt ist, zu identifizieren. Aufgrund der Tatsache, dass eine Fehlerbildung nur am Systemausgang möglich ist, musste die sog. Fehlerübertragungsfunktion berechnet werden, mit der es schließlich möglich war ein stabiles Lerngesetz abzuleiten. Hierbei musste zwischen dem Fehlermodell 3 und 4 unterschieden werden, abhängig davon, ob die Fehlerübertragungsfunktion streng positiv reell ist. Die Gültigkeit der abgeleiteten Fehlerübertragungsfunktion sowie der Fehlermodelle konnte auch für den Fall bewiesen werden, dass der Eingangsraum der dynamischen Nichtlinearität nicht messbar ist und somit beobachtete Eingangsgrößen verwendet werden müssen.

Zur Veranschaulichung der Theorie wurden zunächst global integrierende Systeme betrachtet. Am Beispiel eines global integrierenden Hammerstein- und Wiener-Modells wurde gezeigt, dass eine zwingende Voraussetzung für die Anwendung der vorgestellten Theorie die Linearität der unbekannt Parameter im Ausgangssignal des Identifikationsalgorithmus ist. Im Hinblick auf die Verbindung der vorgestellten Theorie zur Identifikation dynamischer Nichtlinearitäten mit den strukturierten rekurrenten Netzen in Kapitel 8 wurde jedoch auch der Fall untersucht, dass die unbekannt Parameter nichtlinear in den Ausgang des Identifikationsalgorithmus eingehen.

Abschließend wurde an einem komplexeren Simulationsbeispiel die Leistungsfähigkeit sowie die Praxisrelevanz des vorgestellten Identifikationsverfahrens verdeutlicht.

7 Anwendung des Verfahrens zur Identifikation mechatronischer Systeme

In diesem Kapitel werden zwei Anwendungsbeispiele des vorgestellten Verfahrens zur Identifikation von dynamischen Nichtlinearitäten genauer betrachtet. Das erste Beispiel befasst sich mit der Identifikation der nichtlinearen Umrichter­dynamik eines mechatronischen Antriebssystems. Das zweite Beispiel behandelt die Identifikation der biomechanischen Eigenschaften des Muskel–Sehne–Komplexes am menschlichen Oberarm. Die Unterschiedlichkeit der beiden Beispiele zeigt schon an dieser Stelle das weite Anwendungsgebiet des vorgestellten Verfahrens.

7.1 Identifikation eines mechatronischen Antriebssystems

Als erstes Anwendungsbeispiel soll ein nichtlineares mechatronisches Antriebssystem identifiziert werden [20]. Das betrachtete Antriebssystem ist in Abb. 7.1 dargestellt.

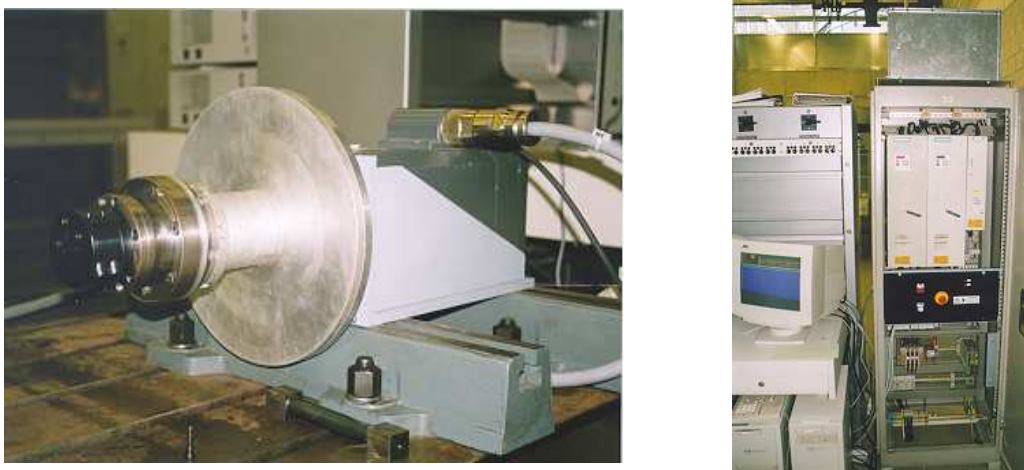


Abb. 7.1: Komponenten des Antriebssystems

Das gesamte System besteht aus einer permanent­erregten Synchronmaschine mit ei-

nem integrierten Encoder zur Rotorlageerfassung und einem zugehörigen Umrichter. Die wichtigsten technischen Daten der Anlage sind im Anhang B zusammengefasst¹. Aus Voruntersuchungen [2] ergab sich das in Abb. 7.2 dargestellte Blockschaltbild des Systems.

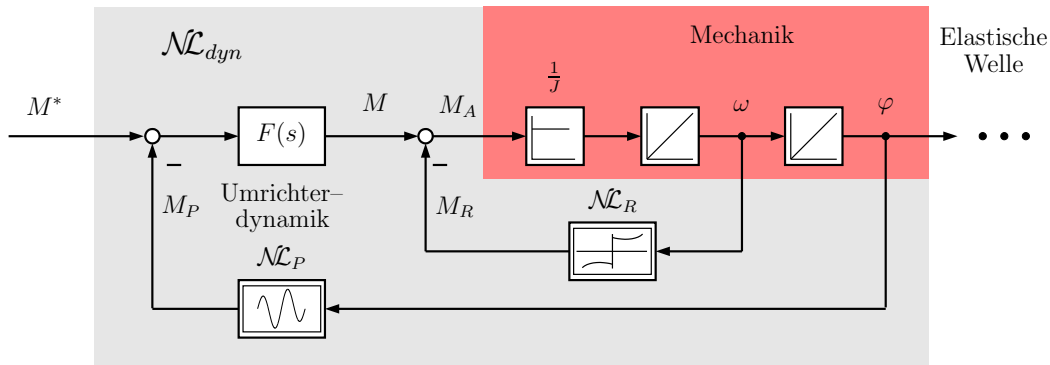


Abb. 7.2: Blockschaltbild des betrachteten Antriebssystems

Die elektrische Maschine ist normalerweise über eine Welle mit einer Last gekoppelt. Diese Last ist in der Regel drehzahl- bzw. positionsgeregelt. Je nach Beschaffenheit der Welle, kann es bei Momentenänderungen im System zu ausgeprägten Torsionsschwingungen zwischen dem Antrieb und der Last kommen, die schließlich zu einer instabilen Regelung führen können [53]². Wird im Gegensatz dazu für die Regelung die antriebsseitige Drehzahl bzw. Position verwendet, kann es dynamisch zu gravierenden Fehlern zwischen den Zustandsgrößen der Antriebs- und der Lastseite kommen. Um eine exakte und stabile Regelung der Last realisieren zu können, muss somit das Verhalten der elastischen Welle sowie der Nichtlinearitäten im System berücksichtigt werden. Eine geeignete Methode zur Bestimmung des Verhaltens eines nichtlinearen Zweimassensystems ist eine Identifikation mit einem strukturierten rekurrenten Netz [16]³. Hierzu muss jedoch das Eingangsmoment M , das in der Regel nicht messbar ist, bekannt sein. Dieses Moment M ist aufgrund der nichtlinearen Umrichter-dynamik nicht identisch mit dem vorgegebenen Sollmoment M^* . Zur Berechnung des Momentes M muss ein Modell bestimmt werden, das die nichtlineare Umrichter-dynamik hinreichend genau beschreibt. Dies stellt die Motivation für dieses Anwendungsbeispiel dar.

In Abb. 7.2 ist zu erkennen, dass das bekannte lineare Teilsystem aus einer Integrator-kette besteht. Die Zustandsgrößen sind die Winkelgeschwindigkeit und die Position der Maschine. Das Massenträgheitsmoment J der Maschine kann sehr genau berechnet bzw. aus Herstellerangaben bestimmt werden. Somit ist das lineare Teilsystem in seiner Struktur und seinen Parametern bekannt. Die Struktur der

¹In diesem Anwendungsbeispiel wird lediglich Maschine I mit Umrichter I betrachtet.

²Diese Problematik ist in der Literatur auch unter dem Stichwort *Zweimassensystem* bzw. *Mehrmassensystem* zu finden.

³Die Federsteifigkeit einer elastischen Welle kann näherungsweise durch eine Messung bestimmt werden. Die Dämpfung dagegen kann nur äußerst schwierig durch Messungen quantifiziert werden.

dynamischen Nichtlinearität beruht auf der Annahme, dass der Momentenaufbau mit Hilfe einer Übertragungsfunktion modelliert werden kann⁴. Zusätzlich wird eine maschinenlageabhängige statische Nichtlinearität $\mathcal{NL}_P(\varphi)$ mit einer periodischen Charakteristik berücksichtigt. Der Grund für die Annahme einer Abhängigkeit des Maschinenmomentes von der Rotorlage ist, dass die im Umrichter implementierte feldorientierte Regelung zur Approximation des nicht messbaren Luftspaltflusses ein vereinfachtes Maschinenmodell verwendet. Wenn jedoch der geschätzte und der reale Fluss, im Speziellen die Flusslagen, nicht übereinstimmen, dann kommt es aufgrund des falsch berechneten Flusses zu einem Fehlverhalten des feldorientierten Drehmomentregelkreises. Dieses Fehlverhalten drückt sich in einer maschinenlageabhängigen Drehmomentschwankung aus⁵. Als weitere statische Nichtlinearität ist die winkelschwindigkeitsabhängige Lagerreibung $\mathcal{NL}_R(\omega)$ der Maschine zu berücksichtigen. Die Lagerreibung wirkt der Bewegung entgegen und ist bei $\omega = 0 \frac{rad}{s}$ aufgrund der Haftreibung unstetig. Wird das System entsprechend Gl. (6.1) dargestellt, ergibt sich:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \mathcal{NL}_{dyn}(\underline{x}, u) \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{mit} \quad \underline{x}^T = [\omega, \varphi] \quad \text{und} \quad u = M^* \end{aligned} \tag{7.1}$$

Im Folgenden soll das betrachtete Antriebssystem identifiziert werden. Hierzu wird zunächst der Identifikationsalgorithmus sowie der Beobachter in einer Simulationsumgebung getestet. Anschließend erfolgt die Validierung an der realen Anlage.

7.1.1 Identifikation in der Simulationsumgebung

Für den Beobachterentwurf wird bei diesem Anwendungsbeispiel von einem nicht messbaren Eingangsraum der dynamischen Nichtlinearität ausgegangen. Diese Annahme wird so getroffen, da die Winkelgeschwindigkeit ω , die sich durch Differentiation der gemessenen Rotorlage φ berechnet, in Realität stark verrauscht ist und somit als Eingangsgröße für die dynamische Nichtlinearität ungeeignet ist. Dies ist insbesondere deswegen so, da die Winkelgeschwindigkeit als Eingangssignal für die Approximation der unstetigen Lagerreibung dient und sich verrauschte Eingangssignale aufgrund der Unstetigkeit bei einem Drehrichtungswechsel äußerst negativ auf das Identifikationsergebnis auswirken. Die Sichtbarkeit der dynamischen Nichtlinearität sowie die Beobachtbarkeit des linearen Teilsystems kann für das Anwendungs-

⁴In der Übertragungsfunktion sind die dynamischen Effekte der feldorientierten Drehmomentregelung und des Momentenaufbaus in der Maschine selbst zusammengefasst.

⁵Diese Abhängigkeit konnte bei einfachen Vorversuchen an der Anlage festgestellt werden.

beispiel leicht gezeigt werden. Somit gilt für den Beobachter entsprechend Gl. (6.26):

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \widehat{\mathcal{N}}_{dyn}(\hat{\underline{x}}, u) - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \cdot e \quad (7.2)$$

$$\hat{y} = [0 \ 1] \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \quad \text{mit} \quad \hat{\underline{x}}^T = [\hat{\omega}, \hat{\varphi}] \quad \text{und} \quad u = M^*$$

Der Identifikationsalgorithmus für die dynamische Nichtlinearität kann aus folgenden Überlegungen abgeleitet werden. Die periodische statische Nichtlinearität bildet zusammen mit der Übertragungsfunktion ein Hammerstein-Modell mit der Rotorlage φ als Eingangsgröße. In dieses Hammerstein-Modell greift eine zweite Eingangsgröße M^* linear ein⁶. Zusätzlich greift am Ausgang des Hammerstein-Modells eine dritte Eingangsgröße ein, die bei einer isolierten Betrachtung der dynamischen Nichtlinearität einen Durchgriff darstellt. In Kapitel 4 wurden die Fälle mit einem Durchgriff prinzipiell ausgeschlossen. Dies stellt jedoch kein Problem dar, da der Identifikationsalgorithmus leicht um einen Durchgriff einer weiteren Eingangsgröße erweitert werden kann. Der Durchgriff berechnet sich in diesem Fall allerdings als Ausgangssignal einer weiteren statischen Nichtlinearität, die ebenfalls identifiziert werden muss. Somit ergibt sich für den Identifikationsalgorithmus folgender Ansatz:

$$\widehat{\mathcal{N}}_{dyn}(\hat{\underline{x}}, u) = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (7.3)$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left[\underbrace{\mathcal{A}_1[k], \dots, \mathcal{A}_q[k]}_{\text{Eingang } \hat{x}_1}, \underbrace{\mathcal{A}_{A,1}^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \mathcal{A}_{B,K}^T[k] \cdot \tilde{\mathbf{R}}^T}_{\text{Eingang } \hat{x}_2}, \underbrace{u^T[k] \cdot \tilde{\mathbf{R}}^T}_{\text{Eingang } u} \right]$$

Der dynamische Aktivierungsvektor $\underline{\mathcal{A}}_{dyn}$ besteht aus drei Teilen. Der erste Teil beschreibt die Approximation der statischen Nichtlinearität $\widehat{\mathcal{N}}_R(\hat{x}_1)$, deren Ausgang den Durchgriff darstellt. Aufgrund der Tatsache, dass es sich bezüglich der Eingangsgröße \hat{x}_1 um ein rein statisches Approximationsproblem handelt, wird ein einfaches GRNN angesetzt. Für dessen Aktivierungsfunktionen gilt:

$$\mathcal{A}_i[k] = \frac{\exp\left[\frac{(\hat{x}_1 - \chi_i)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{(\hat{x}_1 - \chi_j)^2}{-2\sigma^2}\right]} \quad \text{mit} \quad i = 1 \dots q \quad (7.4)$$

Aufgrund der Unstetigkeit der Lagerreibung bei $\omega = 0 \frac{rad}{s}$ müssen für die Identifikation zwei GRNNs eingesetzt werden, die abhängig von \hat{x}_1 trainiert werden⁷.

Der zweite Teil des dynamischen Aktivierungsvektors beschreibt ein Hammerstein-Modell mit einer periodisch statischen Nichtlinearität $\widehat{\mathcal{N}}_P(\hat{x}_2)$. Zur Approximation dieser periodischen Nichtlinearität wird das HANN aus Kapitel 2.5 eingesetzt⁸. Die

⁶Dies entspricht dem Fall aus Abb. 4.2 in Kapitel 4.1.1.

⁷In Gl. (7.3) wurde dies aus Gründen der Übersichtlichkeit nicht dargestellt.

⁸Der Gleichanteil im HANN wird dabei vernachlässigt, da die periodische Nichtlinearität keinen Offset besitzt.

m zurückliegenden Werte der einzelnen Aktivierungsfunktionen des HANN werden zu Vektoren zusammengefasst und zum Zwecke der Parameterreduktion mit den orthonormalen Basisfunktionen verrechnet. Für die Aktivierungsvektoren des HANN gilt:

$$\underline{\mathcal{A}}_{A,i}^T[k] = \left[\mathcal{A}_{A,i}(\hat{x}_2[k-1]), \dots, \mathcal{A}_{A,i}(\hat{x}_2[k-m]) \right] \quad \text{mit} \quad i = 1 \dots K \quad (7.5)$$

$$\underline{\mathcal{A}}_{B,i}^T[k] = \left[\mathcal{A}_{B,i}(\hat{x}_2[k-1]), \dots, \mathcal{A}_{B,i}(\hat{x}_2[k-m]) \right]$$

Der dritte Teil des dynamischen Aktivierungsvektors beschreibt eine Übertragungsfunktion, die das Verhalten bezüglich der Eingangsgröße u charakterisiert. Die m Vergangenheitswerte von u werden hierzu in einem Vektor zusammengefasst und mit der Basisfunktionenmatrix multipliziert. Für den Vektor \underline{u} gilt:

$$\underline{u}^T[k] = \left[u[k-1], \dots, u[k-m] \right] \quad (7.6)$$

Das gesamte System sowie der zugehörige Beobachter einschließlich Identifikationsalgorithmus sind noch einmal in Abb. 7.3 veranschaulicht.

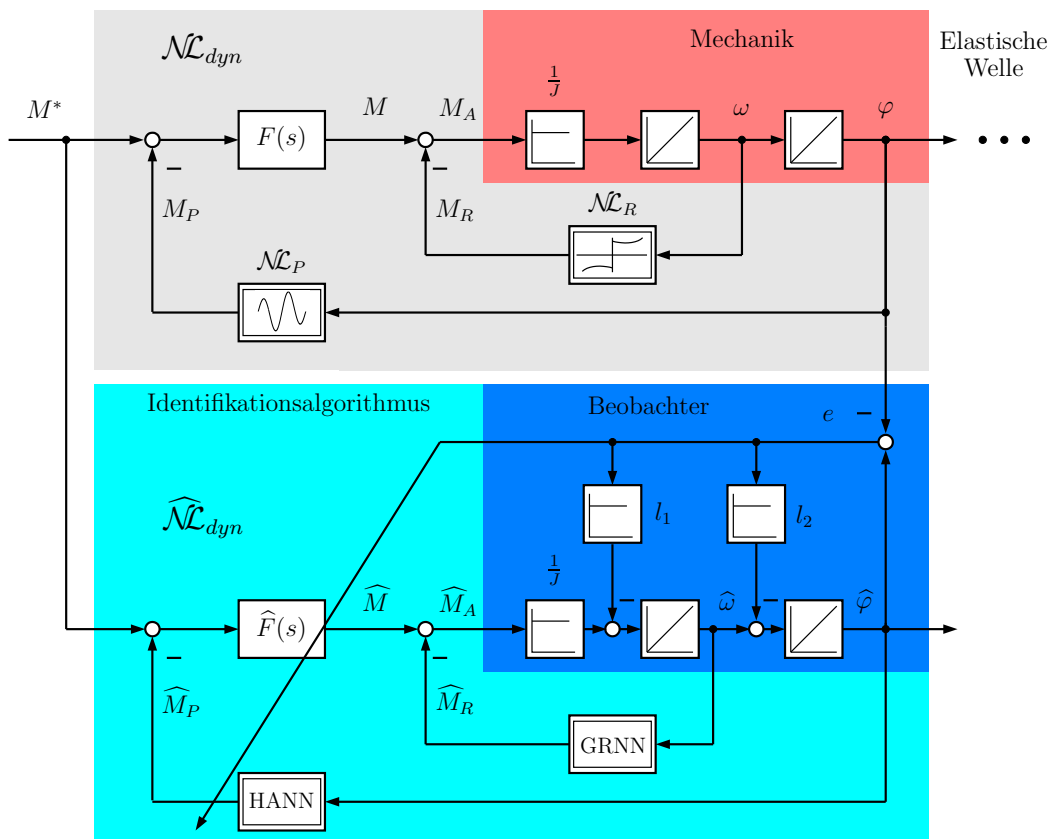


Abb. 7.3: Antriebssystem mit Beobachter und Identifikationsalgorithmus

Zur Adaption der unbekannt Parameter muss die Fehlerübertragungsfunktion $H(s)$ nach Gl. (6.11) bestimmt werden. Es ergibt sich:

$$H(s) = \frac{\frac{1}{J}}{s^2 + s \cdot l_2 + l_1} \quad (7.7)$$

Die Fehlerübertragungsfunktion erfüllt in diesem Fall nicht die SPR-Bedingung, so dass Fehlermodell 4 angewendet werden muss.

Für die Identifikation in der Simulationsumgebung wird eine Reibkennlinie nach Armstrong-Hélouvry [3] angenommen:

$$\mathcal{N}_{\mathcal{L}R}(x_1) = M_c + (M_h + M_c) \cdot \exp \left[- \left(\frac{x_1}{N_s} \right)^2 \right] + M_n \cdot x_1$$

mit

$$M_h = 0.3 \text{ Nm} \quad M_c = 0.75 \text{ Nm} \quad M_n = 0.001 \frac{\text{Nm s}}{\text{rad}} \quad N_s = 5 \frac{\text{rad}}{\text{s}}$$

M_h bezeichnet das Haftreibungsmoment, M_c den Coulombschen Reibanteil, M_n den viskosen Reibanteil und N_s die Losbrechdrehzahl. Für die periodische Nichtlinearität soll gelten:

$$\mathcal{N}_{\mathcal{L}P}(x_2) = M_{p1} \cdot \cos(2 \cdot x_2) - M_{p2} \cdot \sin(4 \cdot x_2)$$

mit

$$M_{p1} = 0.1 \text{ Nm} \quad M_{p2} = 0.3 \text{ Nm}$$

Für die Übertragungsfunktion wird ein PT_1 -Verhalten angenommen:

$$F(s) = \frac{1}{s \cdot T_{lin} + 1} \quad \text{mit} \quad T_{lin} = 0.003 \text{ s}$$

Für den Parameter des bekannten linearen Teilsystems gilt:

$$J = 0.166 \text{ kg m}^2$$

Als Einstellungen für die Identifikation werden die Werte nach Tabelle 7.1 verwendet.

Die Simulationsdauer beträgt insgesamt 10000 Sekunden. Nach 9500 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird mit einem Drehzahlregler, der als Solldrehzahl ein Zufallssignal erhält, im Bereich von $x_1 \in \left[-20 \frac{\text{rad}}{\text{s}} ; 20 \frac{\text{rad}}{\text{s}} \right]$ angeregt, so dass sich ein Moment von $u \in [-17 \text{ Nm} ; 17 \text{ Nm}]$ ergibt. Die Rotorlage wird bei dieser Anregung im Bereich $x_2 \in [-\pi \text{ rad} ; \pi \text{ rad}]$ durchfahren.

Einstellwerte	Erläuterung
$q = 21 \cdot 2 = 42$	Stützstellenanzahl der GRNNs
$\sigma_{norm} = 0.6$	normierte Standardabweichung der GRNNs
$K = 4$	Ordnung des HANN
$m = 25$	Antwortlänge
$m_r = 6$	Basisfunktionenanzahl
$\zeta = 4.95$	Formfaktor
$\underline{l}^T = [1020, 45.2]$	Beobachterrückführungen
$p = q + 2 \cdot K \cdot m_r + m_r = 96$	Parameteranzahl
$\eta_{GRNN} = 10^5$	Lernschrittweite der GRNNs
$\eta = 10^4$	Lernschrittweite der restlichen Parameter
$h = 0.001s$	Abtastzeit

Tabelle 7.1: Einstellwerte für die Identifikation

In Abb. 7.4 ist der Fehlerverlauf (links) und die Konvergenz der Parameter⁹ (rechts) dargestellt.

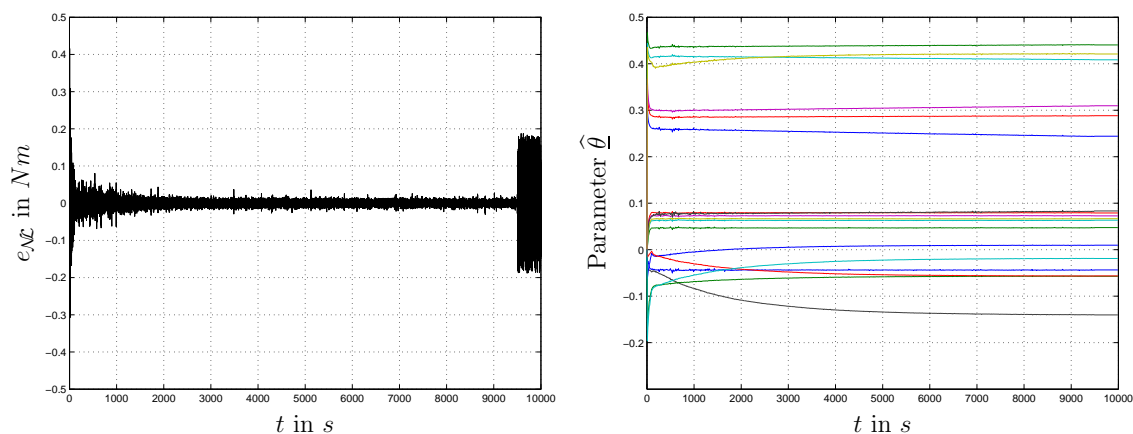


Abb. 7.4: Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz der Parameter (rechts)

Es ist zu erkennen, dass der Fehler $e_{\mathcal{N}}$ klein wird und die Parameter konvergieren. Allerdings steigt der Fehler im Parallelbetrieb wieder merklich an¹⁰. Dies ist darauf zurück zu führen, dass einige Parameter sehr langsam konvergieren. Auch die geringe Basisfunktionenanzahl trägt zu diesem Fehler bei. Dennoch ist das Ergebnis als gut zu bewerten, wenn beispielsweise die statischen Nichtlinearitäten betrachtet werden. In Abb. 7.5 links ist die identifizierte Reibkennlinie der vorgegebenen gegenübergestellt sowie rechts die Konvergenz der zugehörigen Parameter dargestellt. Die statische Reibkennlinie wird durch das GRNN sehr gut identifiziert. Lediglich um die Unstetigkeitsstelle gibt es geringfügige Abweichungen, die durch eine ungenügen-

⁹Die Parameter $\hat{\underline{\theta}}_R$ der Reibkennlinie werden gesondert dargestellt.

¹⁰Der Fehler ist im Parallellauf kleiner als zu Beginn der Identifikation mit aktiven Beobachter.

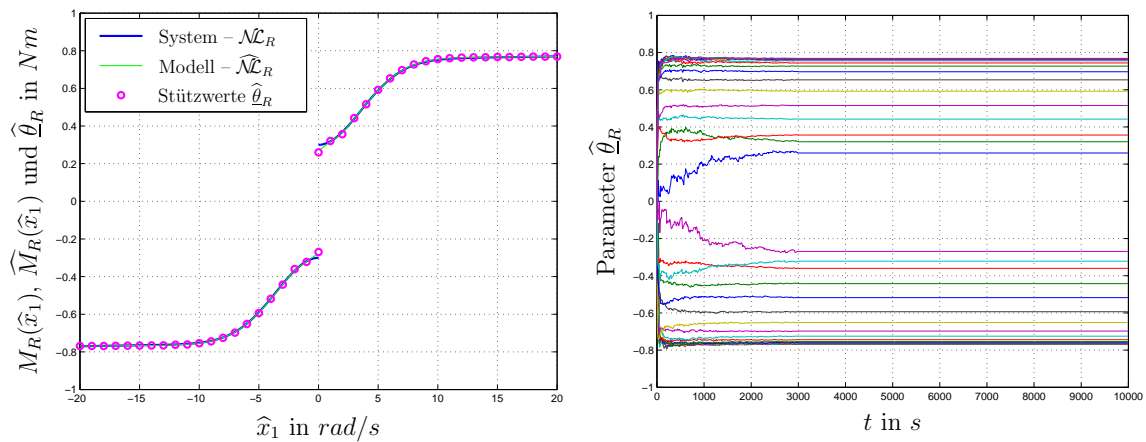


Abb. 7.5: Vergleich zwischen vorgegebener und identifizierter Reibkennlinie (links) sowie Konvergenz der Parameter (rechts)

de Anregung in diesem Bereich verursacht werden. Die zugehörigen Parameter der Reibkennlinie zeigen ein sehr gutes Konvergenzverhalten. Das Ergebnis der periodischen Nichtlinearität sowie der Impulsantwort ist in Abb. 7.6 veranschaulicht.

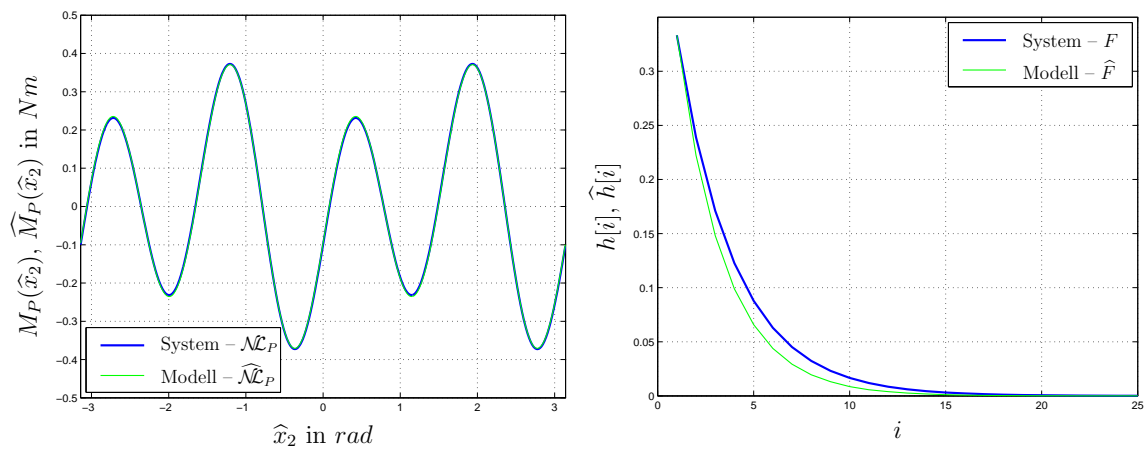


Abb. 7.6: Identifikationsergebnis der periodischen Nichtlinearität (links) und der identifizierten Impulsantwort (rechts)

Die Auswertung des HANN zeigt ein sehr gutes Identifikationsergebnis. Die Impulsantwort weicht jedoch etwas von der Vorgabe ab. Hierauf ist auch der Fehler im Parallelbetrieb zurück zu führen. Die Gründe wurden bereits genannt.

7.1.2 Validierung am realen System

In diesem Kapitel soll die Identifikation am realen System validiert werden. Als Einstellungen für die Identifikation werden die Werte nach Tabelle 7.2 verwendet.

Einstellwerte	Erläuterung
$q = 21 \cdot 2 = 42$	Stützstellenanzahl der GRNNs
$\sigma_{norm} = 1.6$	normierte Standardabweichung der GRNNs
$K = 4$	Ordnung des HANN
$m = 25$	Antwortlänge
$m_r = 6$	Basisfunktionenanzahl
$\zeta = 3.75$	Formfaktor
$\underline{l}^T = [204, 20.2]$	Beobachterrückführungen
$p = q + 2 \cdot K \cdot m_r + m_r = 96$	Parameteranzahl
$\eta_{GRNN} = 2000$	Lernschrittweite der GRNNs
$\eta = 400$	Lernschrittweite der restlichen Parameter
$h = 0.001s$	Abtastzeit

Tabelle 7.2: Einstellwerte für die Identifikation am realen System

Die Simulationsdauer beträgt insgesamt 1800 Sekunden. Nach 1700 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird wiederum mit einem Drehzahlregler, der als Solldrehzahl ein Zufallssignal erhält, im Bereich von $x_1 \in [-20 \frac{rad}{s}; 20 \frac{rad}{s}]$ angeregt, so dass sich ein Moment von $u \in [-17 Nm; 17 Nm]$ ergibt. Die Rotorlage wird bei dieser Anregung im Bereich von $x_2 \in [-\pi rad; \pi rad]$ durchfahren. In Abb. 7.7 ist der Fehler in der Winkelgeschwindigkeit (links) und die Konvergenz der Parameter¹¹ (rechts) dargestellt.

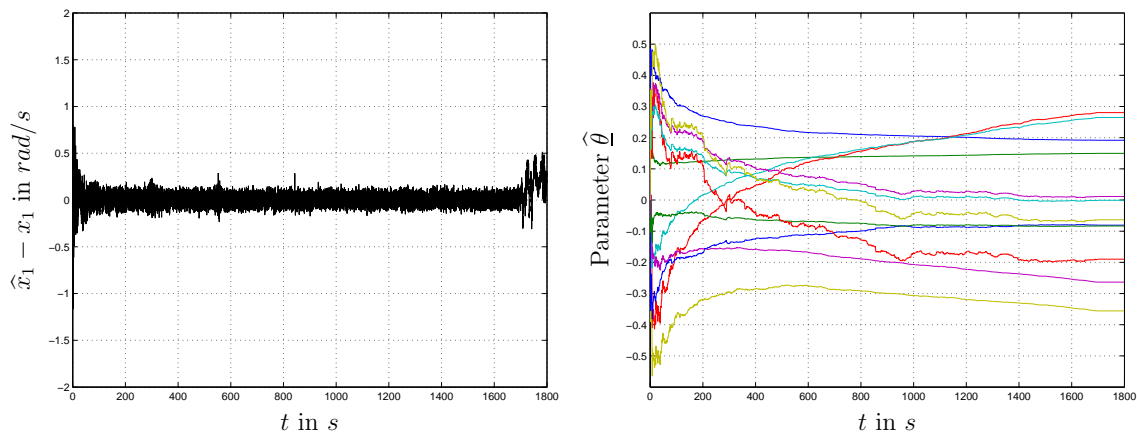


Abb. 7.7: Winkelgeschwindigkeitsfehler (links) und Konvergenz der Parameter (rechts)

Es ist zu erkennen, dass der Winkelgeschwindigkeitsfehler $\hat{x}_1 - x_1$ klein wird und die Parameter konvergieren. Dadurch, dass die Parameter relativ langsam konvergieren, steigt der Fehler im Parallelbetrieb wieder merklich an. Ein längerer Lernvorgang

¹¹Die Parameter $\hat{\theta}_R$ der Reibkennlinie werden wieder gesondert dargestellt.

würde das Ergebnis noch verbessern. Würde lediglich die Parameteradaption eingestellt werden und die Korrektur der Zustände über die Beobachterrückführungen weiterhin erfolgen, wäre kein Anstieg im Fehler zu erkennen. In Abb. 7.8 links ist die identifizierte Reibkennlinie sowie rechts die Konvergenz der zugehörigen Parameter dargestellt.

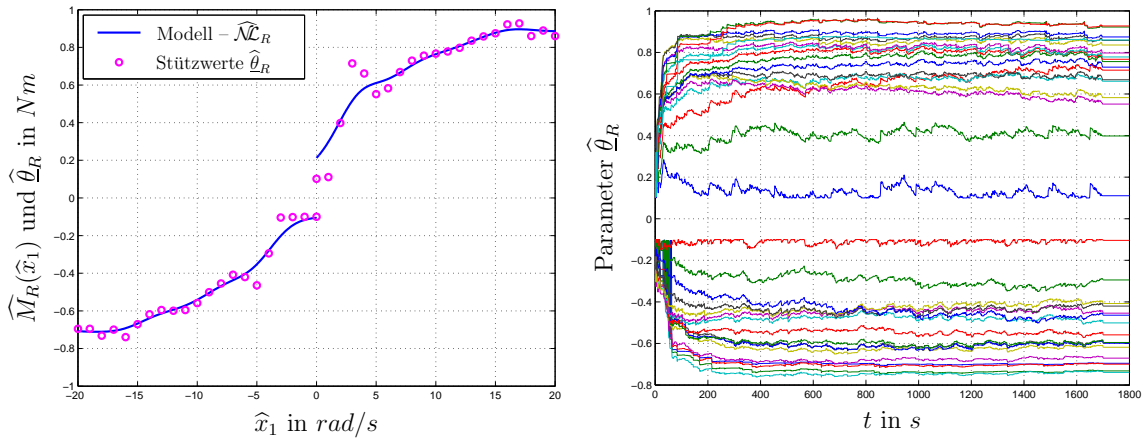


Abb. 7.8: Identifizierte Reibkennlinie (links) sowie Konvergenz der Parameter (rechts)

Die statische Reibkennlinie wird durch das GRNN gut identifiziert und hat einen plausiblen Verlauf. Die zugehörigen Parameter der Reibkennlinie zeigen ein gutes Konvergenzverhalten. Das Ergebnis der periodischen Nichtlinearität sowie der Impulsantwort ist in Abb. 7.9 veranschaulicht.

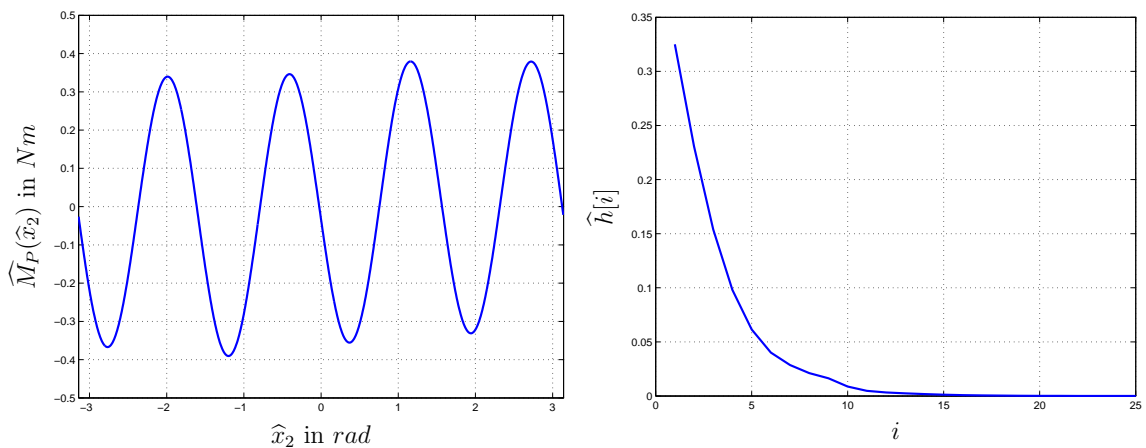


Abb. 7.9: Identifikationsergebnis der periodischen Nichtlinearität (links) und der identifizierten Impulsantwort (rechts)

Die Auswertung des HANN zeigt deutlich den Einfluss des periodischen Momentes. Die Impulsantwort ist aufgrund der langsamen Konvergenz der Parameter noch nicht vollständig gelernt. Eine Abhilfe für die langsame Parameterkonvergenz wird im nächsten Kapitel vorgestellt.

7.2 Identifikation der biomechanischen Eigenschaften des Muskel–Sehne–Komplexes am menschlichen Oberarm

In diesem Kapitel wird die Identifikation der biomechanischen Eigenschaften des Muskel–Sehne–Komplexes am menschlichen Oberarm untersucht. Die folgenden Ergebnisse entstanden im Rahmen einer Zusammenarbeit mit der Forschungsgruppe *Sensomotorische Integration* am Klinikum Rechts der Isar der TU München [10].

7.2.1 Motivation und Aufgabenstellung

Die repetitive periphere Magnetstimulation (RPMS) [7, 13] ist ein neues medizinisches Therapieverfahren, das zur Rehabilitation von gelähmten Patienten eingesetzt wird, die durch eine Schädigung des Gehirns, z.B. durch einen Schlaganfall, motorische Körperfunktionen, wie Arm- und Handbewegungen verloren haben. Derartige Schädigungen treten in Deutschland bei ca. einer halben Million Menschen auf. Die Betroffenen sind meist durch die entstandenen Funktionsverluste stark beeinträchtigt. Das therapeutische Ziel ist das Wiedererlernen der verloren gegangenen Bewegungen, vor allem das Reichen und das Präzisionsgreifen, welche im täglichen Leben die wichtigsten manipulatorischen Tätigkeiten sind. Das Konzept dieses Verfahrens beruht auf Untersuchungsergebnissen, die gezeigt haben, dass die sensomotorische Hirnrinde eine hohe Reorganisationstendenz besitzt, wenn durch Stimulation der sensorische Zustrom von Informationen aus den gelähmten Extremitäten reaktiviert wird. Für die RPMS ist ein Magnetstimulator entwickelt worden, der mittels Spulen ein pulsierendes Magnetfeld erzeugt. Durch dieses Feld werden elektrische Impulse generiert, die in die Nerven eindringen, über das periphere Nervensystem zum Muskel gelangen und seine Kontraktion auslösen. Durch die induzierte Bewegung wird ein sensomotorischer Zustrom zum zentralen Nervensystem erzeugt. Da die sensomotorische Hirnrinde eine sehr hohe Reorganisationstendenz besitzt, können durch mehrfache Wiederholung von Stimulationsvorgängen nicht aktive, ungeschädigte Gehirnbereiche die Funktionen der geschädigten Bereiche übernehmen. Dadurch ist es dem Patienten wieder möglich, Bewegungen mit den vorher gelähmten Körpereinheiten durchzuführen.

Der neue Ansatz der Stimulation des Muskels mit einem pulsierenden Magnetfeld ist insofern ein großer Fortschritt, da das der Magnetstimulation zu Grunde liegende elektrische Feld ausschließlich das sensomotorische System aktiviert und andere Nervenfasern, wie Temperatursysteme und Schmerzsysteme unbeeinflusst lässt. Im Gegensatz dazu werden bei einer elektrischen Stimulation alle Arten von Nervenfasern aktiviert. Somit wird bei diesem Verfahren nicht nur das sensomotorische System aktiviert, sondern auch die Schmerz- und Temperatursysteme. Dies hat zur Folge, dass die beiden Systeme über Reflexwege den stimulierten Muskel zusätzlich aktivieren und die Behandlung nicht immer komplett schmerzfrei verläuft. Der Vor-

teil der RPMS ist somit, dass die eigentliche Muskelstimulation nicht durch Reflexe, verursacht durch andere Systeme, beeinflusst wird. Zusätzlich erfolgt diese Art der Behandlung komplett schmerzfrei.

Die eigentliche Motivation für die im Folgenden vorgestellten Arbeiten ist die Identifikation der biomechanischen Eigenschaften des Muskel–Sehne–Komplexes während der Stimulationsphase zu ermöglichen und einen patientenspezifischen Entwurf eines Reglers und damit eine noch bessere Abstimmung der Behandlung auf den Patienten zu erlangen. Aus den Messungen des Winkels zwischen dem Unter– und Oberarm sowie der Kenntnis des vorgegebenen Stimulationsmusters soll es möglich sein die biomechanischen Parameter des Muskels zu ermitteln.

7.2.2 Modellbildung des biomechanischen Systems

Zunächst muss eine mathematische Modellbildung des Muskel–Sehne–Komplexes sowie des menschlichen Ellbogengelenkes durchgeführt werden. Im Folgenden werden nur die wesentlichen Ergebnisse zusammengefasst. Detaillierte Ausführungen zu der Modellierung können in nachgelesen werden [7, 10].

Das Gesamtmodell des Muskel–Sehne–Komplexes am menschlichen Oberarm bei einer Stimulation des Bizepsmuskels ist in Abb. 7.10 dargestellt.

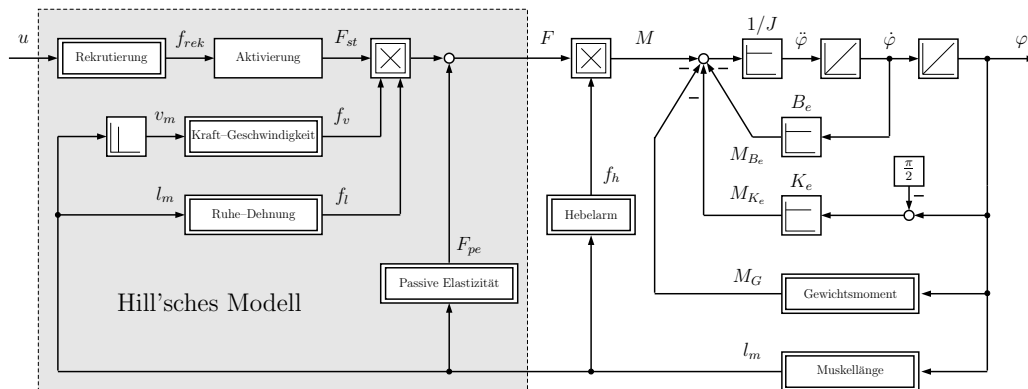


Abb. 7.10: Streckenmodell des Muskel–Sehne–Komplexes am menschlichen Oberarm

Der Muskel kann als zeitvariantes nichtlineares Stellglied interpretiert werden. Für die mathematische Modellierung der biomechanischen Eigenschaften des Muskels wird hier das sog. Hill'sche Modell [7] verwendet. Im Hill'schen Modell wird durch die Reaktion des Muskels auf die Stimulation u eine Kraft auf den Unterarm erzeugt. Die einzelnen Elemente des Hill'schen Modells haben folgende Bedeutung:

- Rekrutierung und Aktivierungsdynamik

Der durch die magnetische Stimulation angeregte motorische Anteil des Muskels wird durch die Rekrutierung und die Aktivierungsdynamik beschrieben. Die Rekrutierung beschreibt die Anzahl der motorischen Einheiten, die durch den Stimulus u in den Kontraktionsvorgang einbezogen werden. Die Rekrutierungskurve ist eine statische Nichtlinearität, die sich im Bereich der Stimulation als Funktion von der Pulsintensität annähern lässt:

$$f_{rek}(u) = k_1 \cdot \left[(u - u_{thr}) \cdot \arctan[\kappa_{thr} \cdot (u - u_{thr})] - (u - u_{sat}) \cdot \arctan[\kappa_{sat} \cdot (u - u_{sat})] \right] + k_2 \quad (7.8)$$

Die Eingangsgröße u ist das Signal, welches durch das pulsierende Magnetfeld erzeugt wird. Die Form der Rekrutierungskurve ist durch die Parameter u_{thr} und u_{sat} gegeben. Bei der unteren Grenze u_{thr} werden die ersten und bei der maximalen Pulsintensität u_{sat} werden alle motorischen Einheiten aktiviert. κ_{thr} und κ_{sat} sind die Kurvenkrümmungen im Schwell- und Sättigungsbereich, k_1 und k_2 ermöglichen eine Skalierung der Rekrutierungskurve. Die Rekrutierungskurve ist eine patientenspezifische statische Nichtlinearität, die aufgrund der Muskelermüdung zeitvariant ist. Für die Simulation wird eine zeitinvariante Rekrutierung nach Gl. (7.8) angenommen.

Je nach Anzahl der angeregten motorischen Einheiten erzeugt der Muskel eine Kraftantwort, die im Hill'schen Modell als Aktivierungsdynamik bezeichnet und für die Simulation durch ein PT_3 -Glied angenähert wird.

$$F(s) = \frac{k_{akt}}{(s + \alpha)^3} \quad (7.9)$$

k_{akt} ist ein Verstärkungsfaktor und α die reziproke Zeitkonstante. Wird die Rekrutierungskurve und die Aktivierungsdynamik als Einheit aufgefasst, so beschreibt dieses Element den mathematischen Zusammenhang zwischen der Stimulation u und der dadurch im Muskel erzeugten Kraft F_{st} . Die Rekrutierungskurve und die Aktivierungsdynamik bilden ein Hammerstein-Modell, das mit dem Funktionsapproximatoransatz aus Kapitel 5.2 identifiziert werden kann.

- Das lageabhängige Kraftbildungsvermögen des Muskels wird als Ruhe-Dehnungskurve f_l bezeichnet. Sie kann je nach Literatur durch ein Polynom dritten Grades oder eine Kombination einer Exponential- und einer Sinusfunktion beschrieben werden. Für die Simulationen wird folgender Ansatz ausgewählt:

$$f_l(l_m) = \exp \left[-\frac{\left(\frac{l_m}{l_{biceps}} - b \right)^2}{a} \right] \quad (7.10)$$

Die Parameter a und b sind muskelspezifische Konstanten. Als Eingangsgröße geht die Länge des Muskels l_m ein, welche sich in Abhängigkeit von der Lage des Unterarmes ändert. Mit dem Wert l_{biceps} , der die Muskellänge bei rechtem Winkel zwischen Unter- und Oberarm beschreibt, wird eine Normierung der Muskellänge l_m durchgeführt.

- Die Kraft–Geschwindigkeitskurve $f_v(v_m)$ beschreibt die Abnahme der entwickelbaren Kraft des Muskels mit ansteigender Geschwindigkeit v_m . Auch für sie werden in der Literatur verschiedene Ansätze angegeben. Für die Simulationen wird die folgende Gleichung gewählt:

$$f_v(v_m) = 0.54 \cdot \arctan(5.69 \cdot v_m + 0.51) + 0.745 \quad (7.11)$$

Als Eingangsgröße geht in diese Gleichung die Geschwindigkeit v_m , d.h. die zeitliche Änderung der Muskellänge l_m ein.

- Die passiven Eigenschaften des Muskels werden durch die passive Elastizität beschrieben, die mittels einer Exponentialfunktion dargestellt wird:

$$F_{pe}(l_m) = c_{pe} \cdot \exp \left[d_{pe} \cdot \left(\frac{l_m}{l_{biceps}} - 1 \right) \right] \quad (7.12)$$

Als Variable geht die Muskellänge l_m ein, die wiederum mit l_{biceps} normiert wird. Die Parameter c_{pe} und d_{pe} sind muskelspezifische Konstanten.

Die statischen Nichtlinearitäten und die Übertragungsfunktion des Hill'schen Modells beschreiben das patientenspezifische Muskelverhalten und sollen im Folgenden identifiziert werden. Ausgangsgröße ist die vom Muskel erzeugte Kraft F , die die Eingangsgröße für das mechanische Teilsystem darstellt.

Das mechanische Teilsystem beschreibt die Anatomie der Armes. Durch das Aufstellen der Momentengleichung im Ellbogengelenk wird die Ausgangsgröße des Systems, d.h. der Winkel φ im Ellbogengelenk bestimmt. Mit Hilfe des Winkels φ muss die Muskellänge l_m , die wiederum Eingangsgröße des Hill'schen Modells ist, sowie der Hebelarm f_h und das Moment M_G aufgrund der Gewichtskraft des Armes berechnet werden. In Abb. 7.11 ist die Analogie des im Folgenden verwendeten Modells zum anatomischen Aufbau des Armes dargestellt. Als Bezugssystem dient das orthonormierte Koordinatensystem (O, x^o, y^o) , dessen Ursprung im Punkt O liegt. Dem Punkt O werden die Koordinaten $(0,0)$ zugewiesen. Im Folgenden sollen die wichtigsten Punkte und Größen in Abb. 7.11 kurz erläutert werden. Der Punkt O stellt die Verbindung zwischen dem Oberarmknochen und dem Schultergelenk dar. Der Punkt S ist der Ansatz des Bizepsmuskels am Schultergelenk¹². Der Abstand zwischen dem Kontaktpunkt des Oberarmknochens mit dem Schultergelenk und dem Muskelansatz am Schultergelenk wird mit l_s bezeichnet. Da diese beiden Kontaktpunkte nicht in der gleichen Höhe miteinander verbunden sind, wird der Winkel ξ eingeführt. Der

¹²Der Trizepsmuskel wird in der Modellierung nicht berücksichtigt.

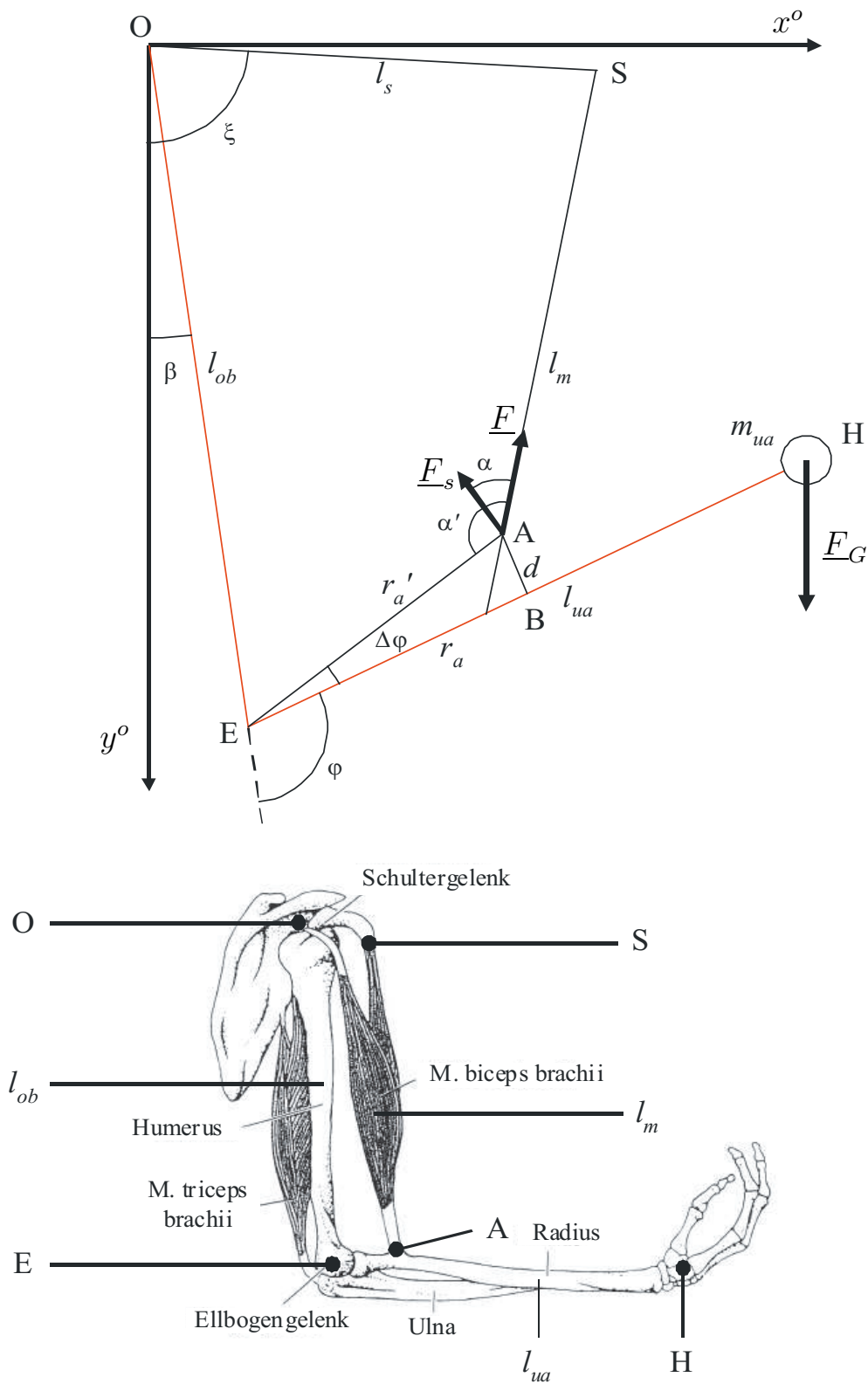


Abb. 7.11: Modell des Armes

dritte wichtige Punkt im Modell ist E, welcher das Ellbogengelenk darstellt. Der Ellbogen wird als ein gedämpftes Feder–Masse–System modelliert. Die Dämpfung wird durch den Wert B_e berücksichtigt und als Ellbogendämpfung bezeichnet. Die Ellbogensteifigkeit K_e wird als Federsteifigkeit einer Drehfeder angenommen. Das Feder–Masse–System ist in Abb.7.11 nicht dargestellt. Die Länge l_{ob} zwischen dem Punkt O und E ist die Länge des Oberarmes. Die Distanz vom Ellbogen E zum Massenmittelpunkt H der Hand, in welchem die gesamte Masse des Unterarmes konzentriert angenommen wird, ist die Länge des Unterarmes und wird im Modell mit l_{ua} bezeichnet. Auf den Massenmittelpunkt wirkt die Gewichtskraft F_G . Der Winkel zwischen dem Oberarm– und dem Unterarmknochen ist der Drehwinkel φ . Der Drehwinkel φ ist die Ausgangsgröße des Systems. Ein weiterer wichtiger Punkt ist A. An diesem Punkt setzt der Bizepsmuskel am Unterarm an und somit auch die durch den Muskel erzeugte Kraft F , die entlang des Muskels wirkt. Der Ansatz des Muskels erfolgt nicht in der Mittelachse des Unterarmknochens. Die Dicke des Unterarmknochens wird durch d berücksichtigt. Mit der senkrecht auf dem Hebelarm r'_a ansetzenden Kraft F_s kann ein Drehmoment erzeugt und somit der Unterarm angehoben werden. Die Länge des Hebelarmes r'_a entspricht der Distanz zwischen dem Muskelansatz am Unterarm A und dem Ellbogengelenk E. Der Abstand zwischen dem Punkt S und dem Punkt A wird als relative Muskellänge l_m bezeichnet, da sie in Abhängigkeit vom Drehwinkel φ variiert. Der Winkel $\Delta\varphi$ entsteht durch die Berücksichtigung der Dicke des Unterarmknochens. Er spannt sich zwischen der Mittelachse des Unterarmknochens und der Oberseite dieses Knochens auf. Der Winkel β muss eingeführt werden, da der Patient bei der Stimulation den Arm nur auf den Ellbogen auflegen darf und somit zwangsweise ein Winkel entsteht.

Für die Modellierung des menschlichen Ellbogengelenkes muss die Momentengleichung im Ellbogen aufgestellt werden. Für die Momentengleichung gilt:

$$F \cdot f_h(l_m) - J \cdot \ddot{\varphi} - B_e \cdot \dot{\varphi} - K_e \cdot (\varphi - \frac{\pi}{2}) - M_G(\varphi) = 0 \quad (7.13)$$

Die Muskelkraft F erzeugt über den Hebelarm $f_h(l_m)$ ein Moment M um das Ellbogengelenk, das den Unterarm anhebt und somit den Winkel φ , den der Unter– und Oberarm einschließen, verkleinert. Der Hebelarm f_h ergibt sich in Abhängigkeit der Muskellänge l_m wie folgt:

$$f_h(l_m) = \cos(\alpha(l_m)) \cdot r'_a \quad (7.14)$$

Der Faktor $\cos(\alpha(l_m))$ berechnet den Anteil der Kraft F , der senkrecht auf den eigentlichen Hebel r'_a wirkt. Der Winkel α ist wiederum von der Muskellänge l_m abhängig und berechnet sich zu:

$$\alpha(l_m) = \alpha'(l_m) - \frac{\pi}{2} \quad (7.15)$$

mit

$$\alpha'(l_m) = \arccos \frac{r_a'^2 + l_m^2 - [l_{ob} \cdot \sin(\beta) - l_s \cdot \sin(\xi)]^2 - [l_{ob} \cdot \cos(\beta) - l_s \cdot \cos(\xi)]^2}{2 \cdot l_m \cdot r_a'} \quad (7.16)$$

Die Muskellänge l_m ergibt sich in Abhängigkeit des Winkels φ zu:

$$l_m^2 = \left[r_a' \cdot \sin(\varphi + \Delta\varphi) \cdot \cos(\beta) + r_a' \cdot \cos(\varphi + \Delta\varphi) \cdot \sin(\beta) + l_{ob} \cdot \sin(\beta) - l_s \cdot \sin(\xi) \right]^2 + \left[r_a' \cdot \cos(\varphi + \Delta\varphi) \cdot \cos(\beta) - r_a' \cdot \sin(\varphi + \Delta\varphi) \cdot \sin(\beta) + l_{ob} \cdot \cos(\beta) - l_s \cdot \cos(\xi) \right]^2 \quad (7.17)$$

Für r_a' und $\Delta\varphi$ gilt:

$$r_a' = \sqrt{r_a^2 + d^2} \quad (7.18)$$

$$\Delta\varphi = \arctan \left(\frac{d}{r_a} \right) \quad (7.19)$$

Die Massenträgheit des Unterarmes sowie die Ellbogendämpfung B_e und die Ellbogensteifigkeit K_e wirken bewegungshemmend. Das Massenträgheitsmoment ist bestimmt durch:

$$J = m_{ua} \cdot l_{ua}^2 \quad (7.20)$$

Die Masse m_{ua} bezeichnet die im Punkt H konzentriert angenommene Masse des Unterarmes. Die Größe l_{ua} die Länge des Unterarmes. Die Ellbogensteifigkeit K_e trägt im Drehgelenkwinkelbereich $\varphi = [0 \text{ rad}; \frac{\pi}{2} \text{ rad}]$ zur Bewegung des Armes bei, im Bereich $\varphi = [\frac{\pi}{2} \text{ rad}; \frac{5\pi}{6} \text{ rad}]$ ¹³ hingegen wirkt sie drehhemmend. Um dieses Verhalten zu berücksichtigen, wird vom Winkel φ der Wert $\frac{\pi}{2} \text{ rad}$ abgezogen. Die Ellbogendämpfung B_e erzeugt in Abhängigkeit von der Winkelgeschwindigkeit $\dot{\varphi}$ ebenfalls ein Widerstandsmoment. Der letzte Term, der zum Gesamtwiderstandsmoment beiträgt, ist das Moment M_G , das aufgrund der Gewichtskraft erzeugt wird und sich wie folgt berechnet:

$$M_G = l_{ua} \cdot m_{ua} \cdot g \cdot \sin(\varphi + \beta) \quad (7.21)$$

Die Größe g bezeichnet die Erdbeschleunigung. Somit sind alle Zusammenhänge aus Abb. 7.10 erläutert. Aus Gründen der Plausibilität wird der Winkel φ auf einen Bereich von $[0 \text{ rad}; \frac{5\pi}{6} \text{ rad}]$ begrenzt. Für die nachfolgenden Simulationen werden die Modellparameter aus den Tabellen 7.3 und 7.4 verwendet. Als Anregesignal wird eine gewichtete Impulsfolge mit einer Frequenz von $f = 20 \text{ Hz}$ benutzt, welche die reale Magnetstimulation simuliert.

¹³ $\varphi_{\max} = \frac{5\pi}{6} \text{ rad}$ ist der obere mechanische Anschlag, ab dem keine Bewegung des Armes in positiver Drehrichtung mehr möglich ist.

Einstellwerte	Erläuterung
$K_e = 0.4 \text{ Nm/rad}$	Ellbogensteifigkeit
$B_e = 0.2 \text{ Nm s/rad}$	Ellbogendämpfung
$J = 0.037 \text{ kg m}^2$	Massenträgheitsmoment des Unterarmes
$m_{ua} = 0.3 \text{ kg}$	Gewicht des Unterarmes
$l_{ob} = 0.23 \text{ m}$	Länge des Oberarmes
$l_{ua} = 0.35 \text{ m}$	Länge des Unterarmes
$r_a = 0.04 \text{ m}$	Abstand zwischen Ellbogengelenk und Muskelansatz
$d = 0.01 \text{ m}$	Dicke des Unterarmknochens
$l_s = 0.05 \text{ m}$	Abstand zwischen Oberarm und oberem Muskelansatz
$\xi = 1.4923 \text{ rad}$	Winkel am oberen Muskelansatz
$\beta = 0.3491 \text{ rad}$	Einspannwinkel des Oberarmes

Tabelle 7.3: *Simulationsparameter für die Ellbogenmodellierung*

Einstellwerte	Erläuterung
$u_{thr} = 0.6$	minimale Pulsintensität
$u_{sat} = 0.9$	maximale Pulsintensität
$\kappa_{thr} = 5$	Kurvenkrümmung im Schwellbereich
$\kappa_{sat} = 25$	Kurvenkrümmung im Sättigungsbereich
$k_1 = 1.0580$	Skalierungsgröße
$k_2 = 0.6268$	Skalierungsgröße
$k_{akt} = 10^9 \text{ N}$	Verstärkungsfaktor
$\alpha = 28 \text{ 1/s}$	reziproke Zeitkonstante
$a = 0.4$	muskelspezifische Konstante
$b = 1$	muskelspezifische Konstante
$l_{biceps} = 0.1993 \text{ m}$	Länge des Muskels bei $\varphi = \frac{\pi}{2} \text{ rad}$
$c_{pe} = 0.0014$	muskelspezifische Konstante
$d_{pe} = 6$	muskelspezifische Konstante

Tabelle 7.4: *Simulationsparameter für das Hill'sche Modell*

7.2.3 Entwurf des Identifikationsverfahrens und des Beobachters in der Simulation

In diesem Kapitel soll ein Verfahren zur Identifikation des Hill'schen Muskelmodells entwickelt werden. Das Hill'sche Muskelmodell stellt eine dynamische Nichtlinearität dar, die in ihrer Struktur grob bekannt ist. Es besteht aus vier statischen Nichtlinearitäten und einer Übertragungsfunktion, deren genaue Verläufe bzw. Parameter unbekannt sind. Die anatomischen Parameter des Armes (vgl. Tab. 7.3) werden

im Folgenden als bekannt vorausgesetzt¹⁴. Zunächst soll die Identifikation der Rekrutierungskurve und der Aktivierungsdynamik, die zusammen ein Hammerstein-Modell bilden, untersucht werden. Dies erfolgt mit dem Hintergrund, dass am Versuchsaufbau die Möglichkeit besteht, eine Kraftmessung bei fixiertem Unterarm durchzuführen. Anschließend wird das gesamte Hill'sche Modell in einer Beobachterstruktur mit dem Winkelsignal φ als Ausgangsgröße identifiziert.

7.2.3.1 Identifikation der Rekrutierung und der Aktivierungsdynamik

Die Rekrutierungskurve $f_{rek}(u)$ bildet zusammen mit der Aktivierungsdynamik $F(s)$ ein Hammerstein-Modell. Die Rekrutierungskurve ist eine typische Sättigungscharakteristik, die, wie in Kapitel 5.2.2 gezeigt wurde, nur bedingt durch ein Polynom identifiziert werden kann. Aus diesem Grund kommt der Funktionsapproximatoransatz aus Kapitel 5.2 zum Einsatz. Die Eingangsgröße der Rekrutierungskurve ist das Stimulationssignal u . Für den Identifikationsalgorithmus ergibt sich somit entsprechend Gl. (5.7):

$$\widehat{F}_{st} = \widehat{\mathcal{N}}\mathcal{L}_{dyn,st}(u) = \widehat{\underline{\theta}}_{st}^T \cdot \underline{\mathcal{A}}_{dyn,st}[k] \quad (7.22)$$

$$\underline{\mathcal{A}}_{dyn,st}^T[k] = \left[\underline{\mathcal{A}}_1^T[k] \cdot \widetilde{\mathbf{R}}^T, \underline{\mathcal{A}}_2^T[k] \cdot \widetilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \widetilde{\mathbf{R}}^T \right]$$

$$\underline{\mathcal{A}}_i^T[k] = \left[\mathcal{A}_i(u[k-1]), \mathcal{A}_i(u[k-2]), \dots, \mathcal{A}_i(u[k-m]) \right]$$

Aufgrund der Tatsache, dass es sich bei der Identifikation um einen Menschen handelt, spielt die Identifikationsdauer eine sehr entscheidende Rolle, da die Stimulation nicht beliebig lange erfolgen kann. Aus diesem Grund werden bei der folgenden Identifikation des Hammerstein-Modells zwei Lernverfahren miteinander verglichen. In einer ersten Simulation wird der rekursive Least-Squares-Algorithmus verwendet, der für diese Identifikationsaufgabe aufgrund der Linearität der unbekannt Parameter im Ausgangssignal angewandt werden darf. Im Gegensatz dazu wird in einer zweiten Simulation das Gradientenabstiegsverfahren betrachtet, das als Standardlernverfahren im Neuronalen Beobachter (vgl. Fehlermodell 4 in Kapitel 6.2) eingesetzt wird. Für die Identifikation werden die Einstellwerte nach Tabelle 7.5 verwendet. Die Simulationsdauer beträgt insgesamt 30 Sekunden. Nach 25 Sekunden wird die Parameteradaption eingestellt, so dass die identifizierte dynamische Nichtlinearität als reines Parallelmodell betrieben wird. Das System wird durch ein Zufallssignal mit einer Frequenz von 20 Hz im Bereich von $u \in [0; 1]$ angeregt. In Abb. 7.12 sind die Identifikationsverläufe bei Verwendung des rekursiven Least-Squares-Algorithmus und des Gradientenabstiegsverfahrens einander gegenüber gestellt.

¹⁴Inwieweit sich diese Annahme bei der Validierung halten lässt, muss sich zeigen.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 0.6$	normierte Standardabweichung des GRNN
$m = 40$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.0$	Formfaktor
$p = m_r \cdot q = 210$	Parameteranzahl
$\eta = 8$	Lernschrittweite bei Gradientenabstiegsverfahren
$h = 0.01s$	Abtastzeit

Tabelle 7.5: Einstellwerte für die Identifikation der Rekrutierungskurve und der Aktivierungsdynamik

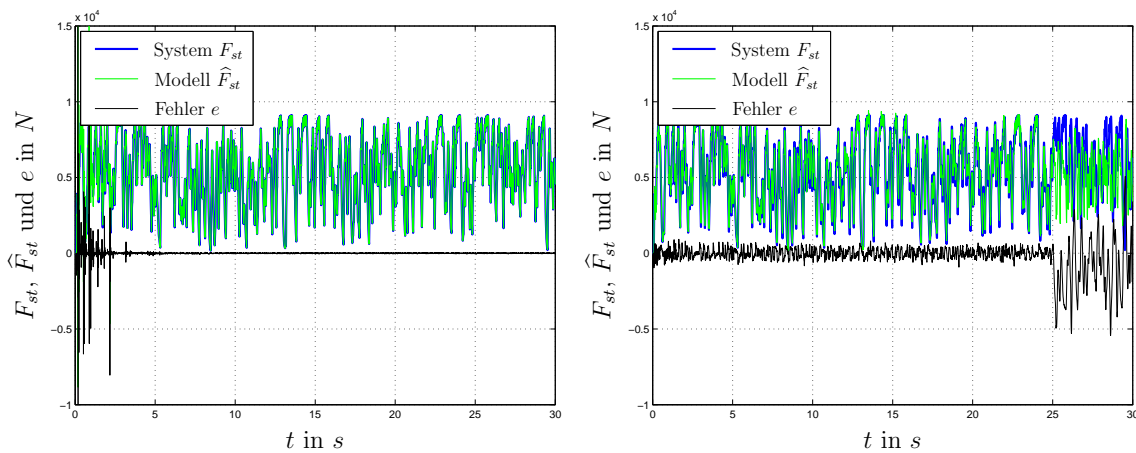


Abb. 7.12: Identifikationsverlauf mit RLS-Algorithmus (links) und Gradientenabstiegsverfahren (rechts)

Es ist zu erkennen, dass der Fehler e beim RLS-Algorithmus sehr schnell klein wird und auch im Parallellauf nicht ansteigt. Im Gegensatz dazu wird der Fehler bei Verwendung des Gradientenabstiegsverfahrens kaum kleiner und steigt im Parallellauf stark an¹⁵. Dieses Verhalten spiegelt sich auch bei der Konvergenz der Parameter in Abb. 7.13 wider. Während die Parameter bei Verwendung des RLS-Algorithmus bereits nach wenigen Sekunden konvergieren, sind die Parameter bei Verwendung des Gradientenabstiegsverfahrens noch nicht annähernd konvergiert. Dies ist im Prinzip des Gradientenabstiegsverfahrens begründet, das lediglich in einer lokalen Umgebung nach einem Minimum des Gütefunktionalen sucht und somit nur vergleichsweise langsam konvergiert und nicht zwangsweise das globale Minimum findet. Der Vergleich der identifizierten Rekrutierungskurven in Abb. 7.14 verdeutlicht ebenfalls das schlechte Lernverhalten des Gradientenabstiegsverfahrens. An dieser Stelle muss somit festgehalten werden, dass der RLS-Algorithmus dem Gradientenabstiegsverfahren deutlich überlegen ist.

¹⁵Die Lernschrittweite wurde so groß gewählt, dass die Identifikation noch stabil blieb.

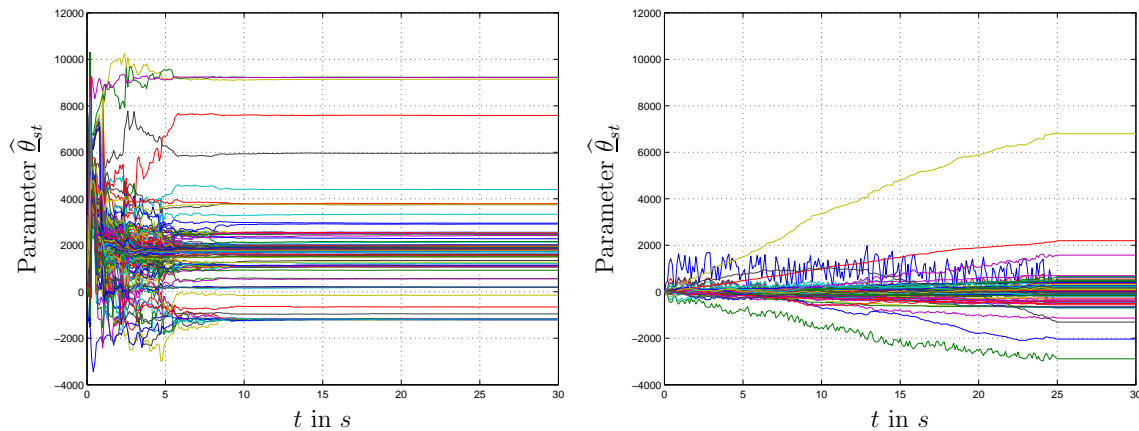


Abb. 7.13: Konvergenz der Parameter mit RLS-Algorithmus (links) und Gradientenabstiegsverfahren (rechts)

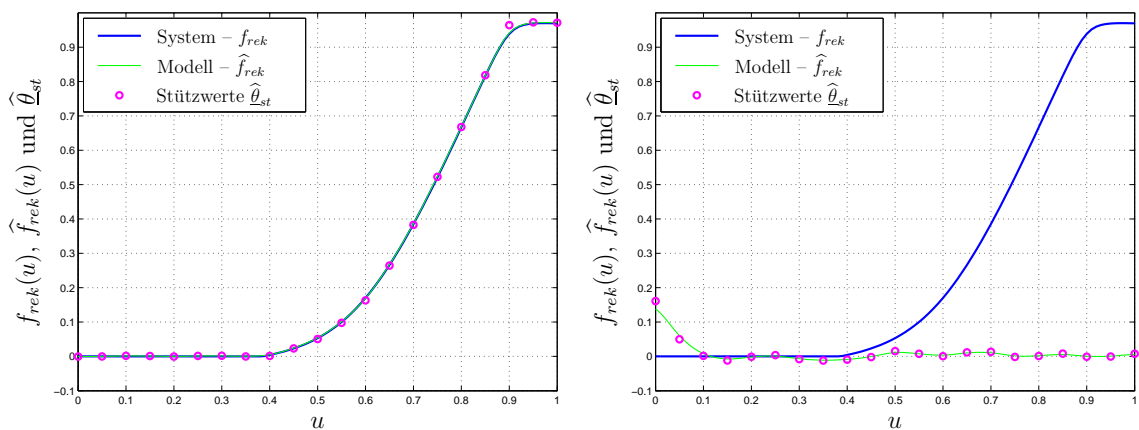


Abb. 7.14: Vorgegebene und identifizierte Rekrutierungskurve mit RLS-Algorithmus (links) und Gradientenabstiegsverfahren (rechts)

7.2.3.2 Identifikation des Hill'schen Modells in Beobachterstruktur

Die eigentliche Aufgabe besteht darin, das Hill'sche Muskelmodell zu identifizieren. Das Problem dabei ist jedoch, dass zur Identifikation der lage- und geschwindigkeitsabhängigen statischen Nichtlinearitäten im Hill'schen Modell der Unterarm nicht fixiert sein darf und somit keine Kraftmessung möglich ist. Jedoch ist eine Messung des Winkels φ , den der Unterarm mit dem Oberarm einschließt, möglich. Das bedeutet, dass das Fehlersignal zur Identifikation nicht direkt gemessen werden kann, sondern erst am Ausgang des Gesamtsystems gebildet werden kann. Im Folgenden soll schrittweise die Identifikation des Hill'schen Modells in einer Beobachterstruktur vorgestellt werden.

Zunächst wird das System entsprechend Gl. (6.1) dargestellt. Mit den Systemzuständen $\underline{x} = [\dot{\varphi}, \varphi]^T$ und unter Berücksichtigung der unbekannt dynamischen Nicht-

linearität der Strecke, ergibt sich folgende Zustandsdarstellung:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{B_e}{J} & -\frac{K_e}{J} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \mathcal{N}_{dyn}(x_2, u) + \begin{bmatrix} \frac{K_e}{J} \cdot \frac{\pi}{2} \\ 0 \end{bmatrix} \quad (7.23)$$

$$y = [0 \quad 1] \cdot \underline{x}$$

Die Ellbogendämpfung B_e , die Ellbogensteifigkeit K_e und das Massenträgheitsmoment J werden als bekannt und konstant vorausgesetzt. Der konstante additive Term in der Zustandsdarstellung, der aufgrund der Ellbogensteifigkeit entsteht, stellt keine Einschränkung der Anwendbarkeit der Beobachtertheorie dar. In der dynamischen Nichtlinearität \mathcal{N}_{dyn} sind neben dem unbekanntem Hill'schen Modell auch die als bekannt vorausgesetzten statischen Nichtlinearitäten zur Berechnung des Hebelarmes und der Muskellänge sowie des Gewichtsmomentes mit eingerechnet, so dass sich für die dynamische Nichtlinearität $\mathcal{N}_{dyn}(x_2, u)$ ergibt:

$$\mathcal{N}_{dyn}(x_2, u) = F \cdot f_h - M_G = [f_v \cdot f_l \cdot F(s) \cdot f_{rek} + F_{pe}] \cdot f_h - M_G \quad (7.24)$$

Die Beobachtbarkeit des linearen Teilsystems sowie die Sichtbarkeit der dynamischen Nichtlinearität kann leicht nachgewiesen werden. Somit kann unter der Annahme eines messbaren Eingangsraumes, der Beobachter wie folgt im Zustandsraum dargestellt werden:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} -\frac{B_e}{J} & -\frac{K_e}{J} \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \widehat{\mathcal{N}}_{dyn}(x_2, u) - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \cdot e + \begin{bmatrix} \frac{K_e}{J} \cdot \frac{\pi}{2} \\ 0 \end{bmatrix} \quad (7.25)$$

$$\hat{y} = [0 \quad 1] \cdot \hat{\underline{x}}$$

Der Ausgang des Identifikationsalgorithmus $\widehat{\mathcal{N}}_{dyn}(x_2, u)$ wird durch die Verknüpfung der Ausgangssignale der unbekanntem und bekannten Nichtlinearitäten berechnet. Dabei wird angenommen, dass die unbekanntem statischen Nichtlinearitäten jeweils durch ein GRNN approximiert werden. Für das approximiertem Ausgangssignal des Hill'schen Modells gilt somit:

$$\hat{F} = \hat{f}_v \cdot \hat{f}_l \cdot \hat{F}_{st} + \hat{F}_{pe} \quad (7.26)$$

Als Eingangssignale für die Identifikation des Hill'schen Modells dienen die Muskellänge l_m und die daraus durch Differentiation berechnete Änderung der Muskellänge v_m . Werden die einzelnen GRNNs durch ihre unbekanntem Parameter und ihre Aktivierungsfunktionen beschrieben, ergibt sich:

$$\hat{F} = \sum_{i_v=1}^{q_v} \hat{\theta}_{v,i_v} \mathcal{A}_{v,i_v}(v_m) \cdot \sum_{i_l=1}^{q_l} \hat{\theta}_{l,i_l} \mathcal{A}_{l,i_l}(l_m) \cdot \sum_{i_s=1}^{q_s} \hat{\theta}_{st,i_s} \mathcal{A}_{dyn,st,i_s}(u) + \sum_{i_p=1}^{q_p} \hat{\theta}_{pe,i_p} \mathcal{A}_{pe,i_p}(l_m) \quad (7.27)$$

Wird Gl. (7.27) ausmultipliziert und in eine vektorielle Form gebracht, gilt:

$$\hat{F} = \underline{\hat{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn} \quad (7.28)$$

$$\text{mit } \underline{\mathcal{A}}_{dyn} = \begin{bmatrix} \mathcal{A}_{v,1} \cdot \mathcal{A}_{l,1} \cdot \mathcal{A}_{dyn,st,1} \\ \mathcal{A}_{v,1} \cdot \mathcal{A}_{l,1} \cdot \mathcal{A}_{dyn,st,2} \\ \vdots \\ \mathcal{A}_{v,1} \cdot \mathcal{A}_{l,1} \cdot \mathcal{A}_{dyn,st,q_s} \\ \mathcal{A}_{v,1} \cdot \mathcal{A}_{l,2} \cdot \mathcal{A}_{dyn,st,1} \\ \mathcal{A}_{v,1} \cdot \mathcal{A}_{l,2} \cdot \mathcal{A}_{dyn,st,2} \\ \vdots \\ \mathcal{A}_{v,q_v} \cdot \mathcal{A}_{l,q_l} \cdot \mathcal{A}_{dyn,st,q_s} \\ \mathcal{A}_{pe,1} \\ \vdots \\ \mathcal{A}_{pe,q_p} \end{bmatrix} \quad \text{und } \underline{\hat{\theta}} = \begin{bmatrix} \hat{\theta}_{v,1} \cdot \hat{\theta}_{l,1} \cdot \hat{\theta}_{st,1} \\ \hat{\theta}_{v,1} \cdot \hat{\theta}_{l,1} \cdot \hat{\theta}_{st,2} \\ \vdots \\ \hat{\theta}_{v,1} \cdot \hat{\theta}_{l,1} \cdot \hat{\theta}_{st,q_s} \\ \hat{\theta}_{v,1} \cdot \hat{\theta}_{l,2} \cdot \hat{\theta}_{st,1} \\ \hat{\theta}_{v,1} \cdot \hat{\theta}_{l,2} \cdot \hat{\theta}_{st,2} \\ \vdots \\ \hat{\theta}_{v,q_v} \cdot \hat{\theta}_{l,q_l} \cdot \hat{\theta}_{st,q_s} \\ \hat{\theta}_{pe,1} \\ \vdots \\ \hat{\theta}_{pe,q_p} \end{bmatrix} \quad (7.29)$$

In obiger Gleichung ist zu erkennen, dass der Identifikationsalgorithmus linear in den Parametern ist¹⁶. An dieser Stelle soll kurz auf die Parameteranzahl eingegangen werden, die sich mit

$$p = q_v \cdot q_l \cdot q_s + q_p \quad (7.30)$$

als sehr hoch ergibt. Dies ist als Preis für die Linearität der Parameter im Ausgangssignal anzusehen. Bei der Parameteranzahl muss somit ein Kompromiss zwischen Genauigkeit und zur Verfügung stehender Rechenleistung gefunden werden.

Das approximierte Kraftsignal \hat{F} stellt jedoch noch nicht die dynamische Nichtlinearität und somit das Eingangssignal des linearen Teilsystems dar. Zur Berechnung von $\hat{\mathcal{N}}\mathcal{L}_{dyn}(x_2, u)$ müssen noch die bekannten statischen Nichtlinearitäten zur Berechnung der Muskellänge, des Hebelarmes und des Gewichtsmomentes berücksichtigt werden. Die Berechnung von $\hat{\mathcal{N}}\mathcal{L}_{dyn}(x_2, u)$ ist in Abb. 7.15 veranschaulicht.

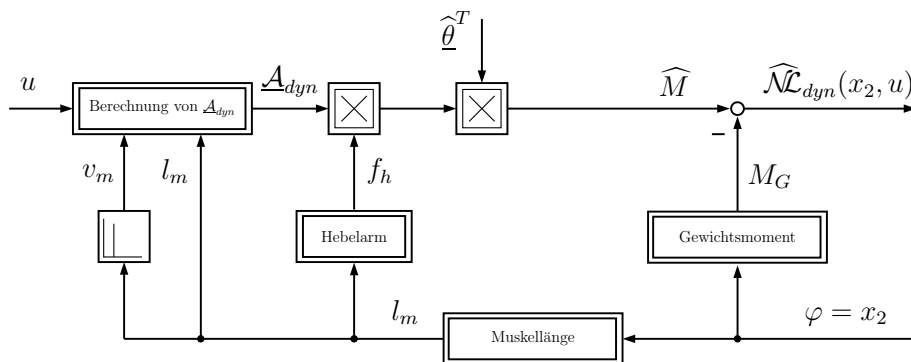


Abb. 7.15: Berechnung von $\hat{\mathcal{N}}\mathcal{L}_{dyn}(x_2, u)$

¹⁶Mit Ausnahme der Passiven Elastizität sind die unbekannt Parameter aufgrund der multiplikativen Verknüpfung der einzelnen Nichtlinearitäten im Hill'schen Modell aus dem späteren Identifikationsergebnis nicht mehr direkt ablesbar.

Die Adaption der unbekannt Parameter $\hat{\underline{\theta}}$ muss mit dem Winkelfehler bzw. Beobachterfehler $e = \hat{\varphi} - \varphi$ erfolgen. Das bedeutet, dass im Lerngesetz die Fehlerübertragungsfunktion $H(s)$ berücksichtigt werden muss. Nach Gl. (6.11) ergibt sich die Fehlerübertragungsfunktion zu:

$$H(s) = \frac{\frac{1}{J}}{s^2 + s\left(\frac{B_e}{J} + l_2\right) + \left(\frac{K_e}{J} + l_1 + \frac{B_e}{J} \cdot l_2\right)} \quad (7.31)$$

Aus Gl. (7.31) wird deutlich, dass die Fehlerübertragungsfunktion nicht die SPR-Bedingung erfüllt und somit Fehlermodell 4 verwendet werden muss. Das Problem, das sich an dieser Stelle ergibt ist, dass das vorgestellte Fehlermodell 4 auf dem Gradientenabstiegsverfahren basiert. Wie in Kapitel 7.2.3.1 gezeigt wurde, ist das Gradientenabstiegsverfahren aufgrund seiner langsamen Konvergenz in diesem Anwendungsfall nicht einsetzbar. Abhilfe für dieses Problem schafft die Verwendung des rekursiven Least-Squares-Algorithmus als Basis für das Fehlermodell 4. Hierzu soll noch einmal die Kernaussage des Fehlermodells aus Kapitel 6.2.1 zusammengefasst werden. Wird das eigentliche Fehlersignal $e_{\mathcal{N}}$ durch eine Fehlerübertragungsfunktion $H(s)$, die nicht die SPR-Bedingung erfüllt, verzögert, dann bleibt das Adaptionsgesetz stabil, wenn anstatt des Beobachterfehlers e , der erweiterte Fehler e_e und anstatt der dynamischen Aktivierung $\underline{\mathcal{A}}_{dyn}$, die verzögerte dynamische Aktivierung $H(s) \cdot \underline{\mathcal{A}}_{dyn}$ verwendet wird. Wird diese Aussage auf den zeitdiskreten RLS-Algorithmus angewandt, ergibt sich für die Adaptionsgleichung:

$$\hat{\underline{\theta}}[k] = \hat{\underline{\theta}}[k-1] + \underline{h}[k] \cdot e_e[k] \quad (7.32)$$

mit

$$\underline{h}[k] = \frac{\mathbf{P}[k-1] \cdot \mathcal{Z}^{-1} \{H(z) \cdot \mathcal{Z} \{ \underline{\mathcal{A}}_{dyn}[k] \} \}}{1 + \mathcal{Z}^{-1} \left\{ \left[H(z) \cdot \mathcal{Z} \{ \underline{\mathcal{A}}_{dyn}[k] \} \right]^T \right\} \cdot \mathbf{P}[k-1] \cdot \mathcal{Z}^{-1} \{H(z) \cdot \mathcal{Z} \{ \underline{\mathcal{A}}_{dyn}[k] \} \}} \quad (7.33)$$

Der erweiterte Fehler wird analog zu Gl. 6.18 wie folgt angesetzt¹⁷:

$$e_e(z) = e(z) + e_z(z) \quad (7.34)$$

mit dem Beobachterfehler¹⁸

$$e(z) = H(z) \cdot \mathcal{Z} \left\{ \left(\hat{\underline{\theta}}^T[k-1] - \underline{\theta}^T \right) \cdot \underline{\mathcal{A}}_{dyn}[k] \right\} \quad (7.35)$$

und dem zusätzlichen Fehler

$$e_z(z) = \hat{\underline{\theta}}^T[k-1] \cdot H(z) \cdot \mathcal{Z} \{ \underline{\mathcal{A}}_{dyn}[k] \} - H(z) \cdot \mathcal{Z} \{ \hat{\underline{\theta}}^T[k-1] \cdot \underline{\mathcal{A}}_{dyn}[k] \} \quad (7.36)$$

¹⁷Es gilt $e_e(z) = \mathcal{Z} \{ e_e[k] \}$.

¹⁸Es wird angenommen, dass sich der Parametervektor $\hat{\underline{\theta}}$ im Vergleich zum System nur sehr langsam ändert.

Bei der rekursiven Berechnung der Kovarianzmatrix ist die zeitdiskrete Fehlerübertragungsfunktion $H(z)$ ebenfalls zu berücksichtigen:

$$\mathbf{P}[k] = \mathbf{P}[k-1] - \underline{h}[k] \cdot \mathcal{Z}^{-1} \left\{ [H(z) \cdot \mathcal{Z} \{ \underline{A}_{dyn}[k] \}]^T \right\} \cdot \mathbf{P}[k-1] \quad (7.37)$$

Für die Anregung des Systems wird eine Lageregelung in Form eines P-Reglers realisiert. Aufgrund der Tatsache, dass als Forderung an die Lageregelung nur das Anheben und Absenken des Unterarmes und keine Genauigkeit zwischen Soll- und Istwertverlauf zu erfüllen ist, reicht diese einfache Regelung zunächst aus. Als Sollwertverlauf wird eine Sinusfunktion mit einem positiven Offset vorgegeben. Für die Identifikation des Hill'schen Modells in der Simulationsumgebung werden die Einstellwerte nach Tabelle 7.6 verwendet.

Einstellwerte	Erläuterung
$q_{st} = q_v = q_l = q_p = 5$	Stützstellenanzahl aller GRNNs
$\sigma_{norm} = 0.8$	normierte Standardabweichung aller GRNNs
$m = 80$	Antwortlänge
$m_r = 6$	Basisfunktionenanzahl
$\zeta = 20.0$	Formfaktor
$\underline{l}^T = [86.3, 13.1]$	Beobachterrückführungen
$p = q_{st} \cdot m_r \cdot q_v \cdot q_l + q_p = 755$	Parameteranzahl
$h = 0.005s$	Abtastzeit

Tabelle 7.6: Einstellwerte für die Identifikation des Hill'schen Modells in der Beobachterstruktur

Die Simulationsdauer beträgt insgesamt 1000 Sekunden. Nach 950 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. In Abb. 7.16 ist der Identifikationsverlauf zu Beginn (links) und nach Abschluss (rechts) der Parameteradaption dargestellt.

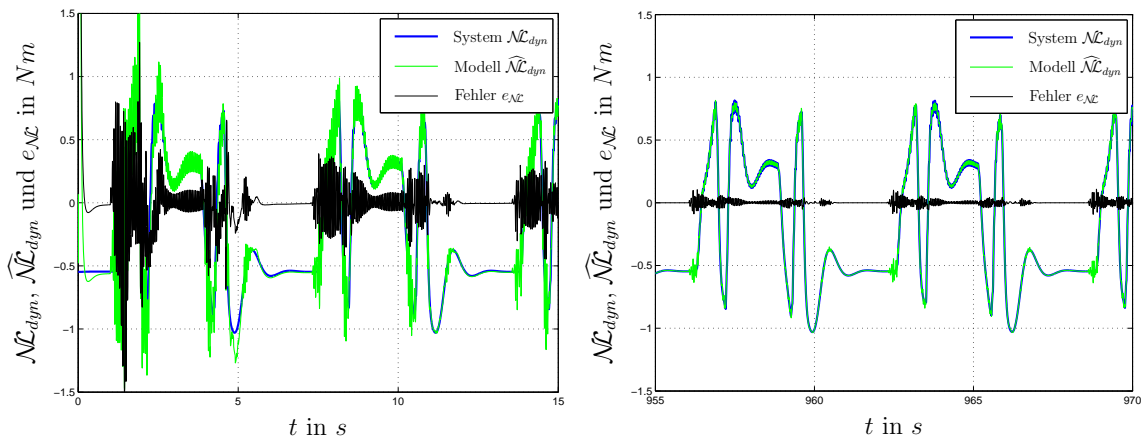


Abb. 7.16: Identifikationsverlauf zu Beginn (links) und nach Abschluss (rechts) der Parameteradaption

Zur weiteren Veranschaulichung ist in Abb. 7.17 der Fehlerverlauf von $e_{\mathcal{N}}$ und das Konvergenzverhalten der Parameter während der gesamten Identifikation gezeigt.

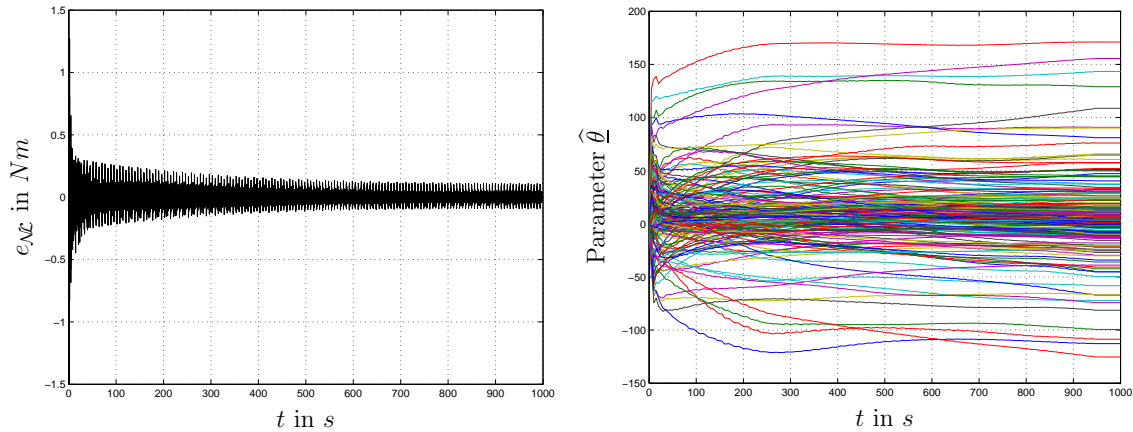


Abb. 7.17: Fehlerverlauf von $e_{\mathcal{N}}$ (links) und Konvergenzverhalten der Parameter (rechts)

Aus den Abbildungen ist zu erkennen, dass der Fehler $e_{\mathcal{N}}$ im Verlauf der Identifikation deutlich kleiner wird und im Parallelbetrieb nicht ansteigt. Allerdings bleibt ein Restfehler, der durch die begrenzte Parameteranzahl bedingt ist, die jedoch aufgrund der Komplexität der Identifikationsaufgabe schon sehr hoch ist. Die hohe Parameteranzahl führt in Verbindung mit der begrenzt dynamischen Systemanregung¹⁹ zu einer noch zu langsamen Konvergenz der Parameter. In der Praxis ist eine Stimulationsdauer von 1000 Sekunden für den Patienten nicht akzeptabel. An dieser Stelle muss zukünftig noch gearbeitet werden, um die Konvergenz der Parameter zu beschleunigen. Hierzu muss in erster Linie die Anzahl der Parameter weiter reduziert werden²⁰ und eine geschickte Vorbelegung²¹ der Parameter zu Beginn der Identifikation erfolgen. Dennoch ist das Identifikationsergebnis vielversprechend, wie der Vergleich der Identifikationsverläufe zu Beginn und nach Abschluss der Parameteradaption, d.h. bei Betrieb als reines Parallelmodell, in Abb. 7.16 zeigt. Noch deutlicher wird die Güte des Identifikationsergebnisses bei einem Vergleich der eigentlich interessierenden Winkelsignale zu Beginn und nach Abschluss der Parameteradaption in Abb. 7.18. Nach Abschluss der Parameteradaption bleibt im Parallelbetrieb des Modells lediglich ein vernachlässigbarer Fehler e bestehen. Für einen praktischen Einsatz muss das Identifikationsverfahren jedoch noch in der angedeuteten Weise optimiert werden.

¹⁹Ein besseres Anregesignal wäre ein Zufallssignal, dies kann jedoch dem Patienten nicht zugemutet werden.

²⁰Es könnte beispielsweise überlegt werden, ob für jede statische Nichtlinearität der Einsatz eines GRNNs notwendig ist, oder ob vielleicht auch Polynome mit einer geringeren Parameteranzahl ausreichend sind.

²¹In der Simulation waren alle Parameter mit Null vorgelegt.

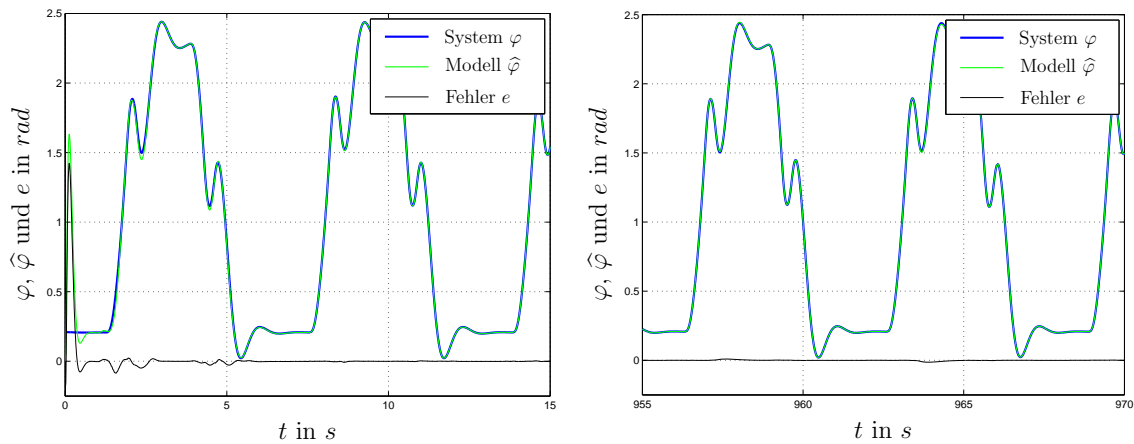


Abb. 7.18: Identifikationsverlauf des Winkels zu Beginn (links) und nach Abschluss (rechts) der Parameteradaption

7.2.4 Versuchsdurchführung am Klinikum Rechts der Isar

In diesem Kapitel wird die vorgestellte Identifikation der Rekrutierungskurve und der Aktivierungsdynamik mit Hilfe von Messdaten, die in einer Versuchsreihe am Klinikum Rechts der Isar aufgenommenen wurden, validiert. In den folgenden Abschnitten wird der Versuchsaufbau sowie die Versuchsdurchführung kurz erläutert und die Identifikationsergebnisse vorgestellt.

7.2.4.1 Versuchsaufbau zur Messung der Kraft

In diesem Abschnitt werden die verschiedenen Elemente des Versuchsaufbaus kurz beschrieben. Der gesamte Versuchsaufbau ist in Abb. 7.19 dargestellt.

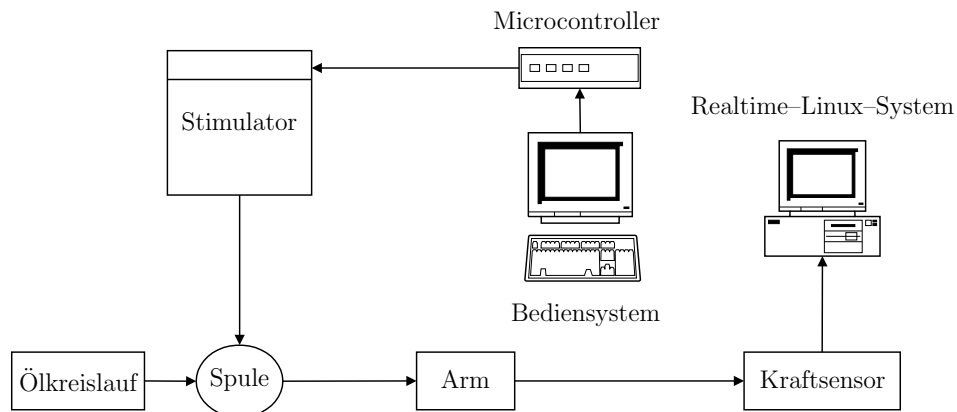


Abb. 7.19: Prinzipieller Versuchsaufbau zur Stimulation des Muskels

Das Bediensystem ermöglicht es zwischen diversen Stimulationsvorgängen²² auszuwählen. Das Bediensystem ist über eine serielle Schnittstelle mit einem Microcontroller verbunden. Der Microcontroller gibt die vorgegebene Stimulationsintensität als analoges Steuerungssignal über eine zweite serielle Schnittstelle an den Magnetstimulator weiter. Mittels dieser Steuerungssignale erzeugt der Stimulator die gewünschten Stimulationsintensitäten an einer ölgekühlten Spule. Diese Kühlung erfolgt durch einen Ölkreislauf und ist notwendig, da sich die Spule während dem Stimulationsvorgang erhitzt und in direktem Kontakt mit dem Oberarm des Patienten steht. Über die Spule werden elektrische Impulse im Muskel induziert, welche je nach Intensität Kräfte erzeugen, die mit einem kapazitiven Kraftaufnehmer gemessen werden. Zur Erfassung der Daten der Kraftmessung, des Stimulationsmusters sowie der Zeit steht ein Realtime-Linux-System zur Verfügung. Einige Bilder zum Versuchsaufbau sind im Anhang C zu finden.

Damit eine plausible Kraftmessung möglich ist, müssen bei der Positionierung des Armes diverse Vorkehrungen getroffen werden. Die Einspannvorrichtung für den Arm ist in Abb. 7.20 dargestellt.

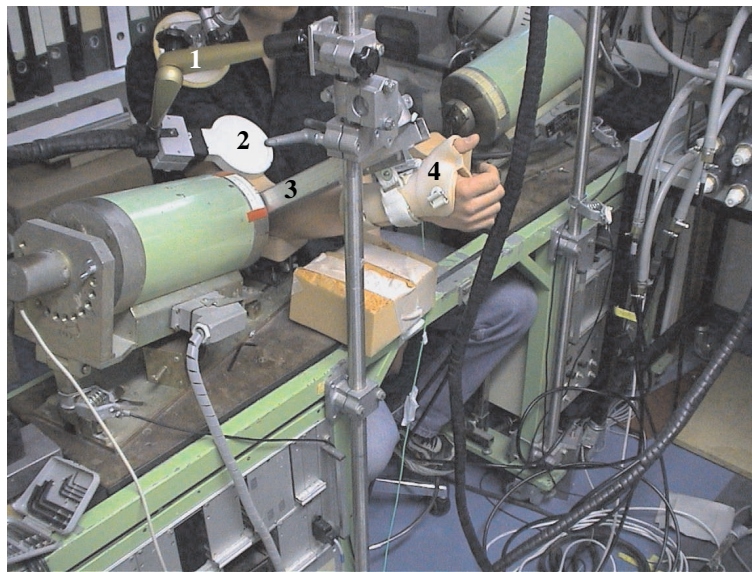


Abb. 7.20: *Einspannvorrichtung für den Arm mit folgenden Elementen: Schulterstütze (1), Spule zur Stimulation (2), Schiene zur Fixierung des Armes sowie der Anpassung der Position des Kraftsensors auf die Unterarmlänge (3), Prothese für die Hand und Kraftsensor (4)*

Für eine optimale Position des Armes muss gewährleistet sein, dass nur der Ellbogen aufliegt und die Schulter in ständigem Kontakt mit der Schulterstütze ist. Damit die Stimulation des Muskels keine Bewegung des Unterarmes erzeugt, ist für die Hand eine Prothese vorgesehen, welche die Hand am Versuchsaufbau fixiert. An der

²²Die Stimulationsvorgänge reichen von manueller Vorgabe bis hin zu längeren vorgefertigten Stimulationsverläufen.

Prothese ist eine Metallplatte in der Höhe des Handgelenks angebracht, welche den Kontakt des stimulierten Armes zum Kraftaufnehmer während des Stimulationsvorgangs gewährleistet. Die Lage des Kraftsensors kann entlang einer Schiene an die Länge des Unterarmes der Versuchsperson angepasst werden.

7.2.4.2 Versuchsdurchführung

Vor jeder Versuchsdurchführung muss der Arm des Patienten unter den im vorherigen Abschnitt erläuterten Bedingungen optimal fixiert werden. Ist der Arm fixiert, muss die Person mit vollem Krafteinsatz den Arm nach oben ziehen. Diese gemessene Kraft wird dann zur Normierung aller weiteren gemessenen Kraftverläufe herangezogen. Ist dies erfolgt, wird anschließend die optimale Position der Spule am Bizepsmuskel ermittelt. Die Spule muss so angeordnet werden, dass sie genau über der Innovationszone des langen Bizepskopfes liegt²³. Liegt die Spule nicht sauber auf dessen Innovationszentrum, werden die Messergebnisse verfälscht, da ein Teil der Kraft durch eine seitliche Drehung des Armes verloren geht und somit nicht durch den Sensor erfasst wird. Die richtige Position der Spule wird durch die Vorgabe von einzelnen Impulsen auf den Muskel und durch manuelles Verschieben der Spule entlang des Muskels bis zu einer optimalen Reaktion des Armes ermittelt. Sind diese Vorbereitungen abgeschlossen, kann mit dem eigentlichen Stimulationszyklus begonnen werden. Die Dauer der Stimulation wird auf 150 Sekunden begrenzt, da eine zu starke Beeinflussung der Krafterzeugung durch die Ermüdung des Muskels vermieden werden soll. Während der Stimulation werden Intensitäten bis zu 80 % des maximalen Wertes durchfahren. Als Abtastrate werden 10 *ms* eingesetzt. Während des Stimulationsvorgangs werden die durch den Muskel erzeugte Kraft sowie das Stimulationsmuster aufgenommen.

In Abb. 7.21 sind die aufgenommenen normierten Kraftverläufe von zwei Versuchspersonen dargestellt. Es werden zwei Effekte deutlich, die bei der Identifikation eine wichtige Rolle spielen werden. Bei den beiden Versuchspersonen handelt es sich um gesunde Personen, die aufgrund ihres eigenen Willens eine Anhebung ihres Unterarmes durchführen können. Dies bedeutet, dass eine nicht entspannte Versuchsperson durch eine Verkrampfung während der Stimulation eine Kraft aufgrund ihres eigenen Willens aufbringen kann. Diese willkürliche Kraft ist nicht durch die Stimulation verursacht und wird im Kraftverlauf als Störsignal, welches im Frequenzbereich des Nutzsignals liegt, mit gemessen. Wird der Kraftverlauf der Versuchsperson 1 mit dem Kraftverlauf der Versuchsperson 2 verglichen, kann festgestellt werden, dass Versuchsperson 1 während der gesamten Stimulation einen mehr oder weniger hohen willkürlichen Kraftanteil aufgrund von Verkrampfung aufbringt. Im Gegensatz dazu ist Versuchsperson 2 wesentlich entspannter, wie an den minimalen Kraftwerten über der Stimulationsdauer zu erkennen ist. Somit ist der Kraftverlauf von

²³Der Bizepsmuskel hat zwei Köpfe – deshalb auch der Name Bi-zeps. Der kurze Bizepskopf ist für eine seitliche Drehung des Armes oder der Hand verantwortlich, der lange Bizepskopf zieht den Unterarm hoch.

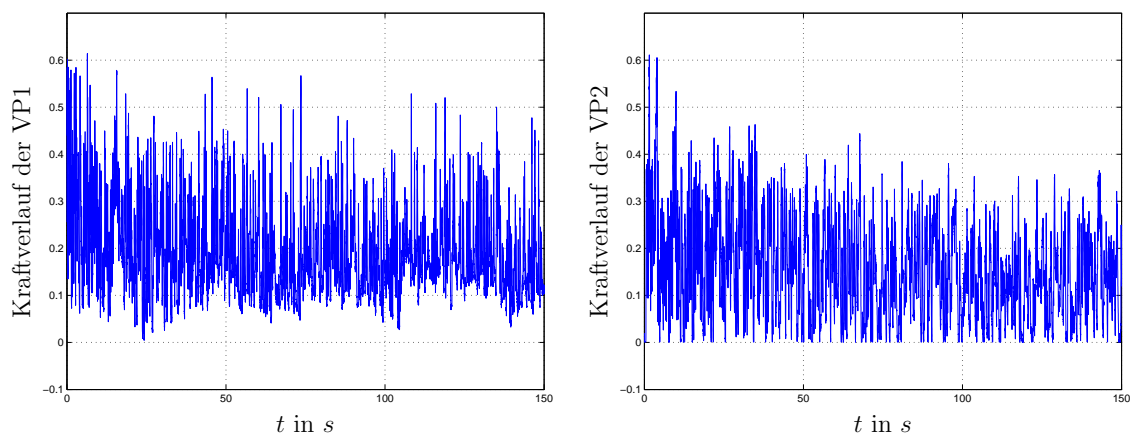


Abb. 7.21: Aufgenommene normierte Kraftverläufe — Versuchsperson 1 (links) und Versuchsperson 2 (rechts)

Versuchsperson 2 aufgrund des wesentlich geringeren willkürlichen Kraftanteils besser zur Identifikation geeignet.

Im Kraftverlauf von Versuchsperson 2 wird ein weiterer Effekt deutlich, der für die Identifikation entscheidend ist. Die maximalen Kraftsignale werden aufgrund der Ermüdung des Muskels während der Stimulationsdauer immer geringer. Die Ermüdung ist bereits nach 60 Sekunden nicht mehr vernachlässigbar, so dass sie bei einer Identifikation zu berücksichtigen sein wird.

7.2.4.3 Identifikationsergebnisse

Im Folgenden soll die Identifikation der Rekrutierungskurve und der Aktivierungsdynamik anhand des gemessenen Kraftverlaufes von Versuchsperson 2 durchgeführt werden. Für die Identifikation dieses Hammerstein-Modells wird das in Kapitel 5.2 vorgestellte Verfahren eingesetzt. Zur Parameteradaption dient der RLS-Algorithmus. In Tabelle 7.7 sind die Einstellungen für die Identifikation zusammengefasst.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 0.6$	normierte Standardabweichung des GRNN
$m = 60$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.0$	Formfaktor
$p = m_r \cdot q = 210$	Parameteranzahl
$h = 0.01s$	Abtastzeit

Tabelle 7.7: Einstellwerte für die Identifikation der Rekrutierungskurve und der Aktivierungsdynamik

In Abb. 7.22 sind die Identifikationsergebnisse für die Rekrutierungskurve und den Parameterverlauf dargestellt.

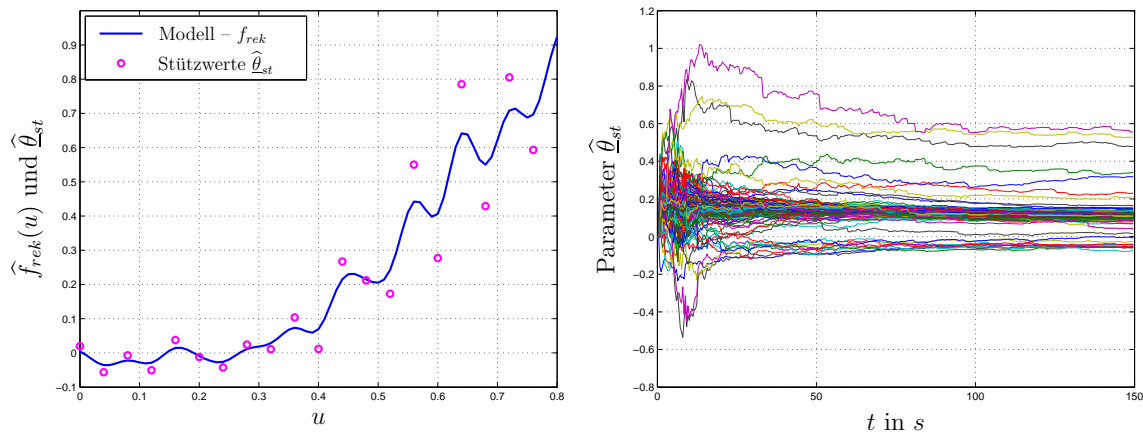


Abb. 7.22: Identifikationsergebnis — Rekrutierungskurve (links) und Konvergenz der Parameter (rechts)

Trotz der scheinbaren Konvergenz der Parameter zeigt die identifizierte Rekrutierungskurve kein befriedigendes Lernergebnis. Die Rekrutierungskurve hat einen stark schwingenden und teilweise negativen Verlauf. Der Verdacht liegt nahe, dass die Ermüdung für diesen Verlauf der Rekrutierungskurve verantwortlich ist. Zur Bekräftigung dieser Aussage ist in Abb. 7.23 das Blockschaltbild des Systems mit Berücksichtigung der Ermüdung und des willkürlichen Kraftanteils veranschaulicht.

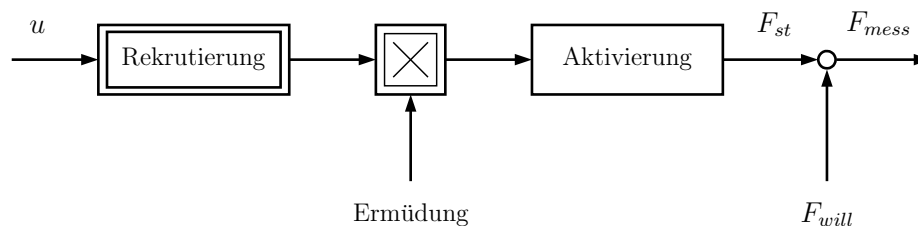


Abb. 7.23: Blockschaltbild mit Einfluss von Ermüdung und willkürlichem Kraftanteil

Durch die Ermüdung wird die Rekrutierungskurve zeitvariant. Mit dem RLS-Algorithmus kann zwar der willkürliche Kraftanteil, der als Störsignal auf den Systemausgang wirkt, sehr gut eliminiert werden, allerdings ist der RLS-Algorithmus nicht in der Lage zeitvariante Systeme zu identifizieren. Hierzu ist die Einführung eines sog. Vergessensfaktors λ erforderlich [24, 25]. Der Vergessensfaktor λ ermöglicht eine Gewichtung des Beitrags vergangener Fehlersignale zur Berechnung der aktuellen Parameter und ermöglicht somit das Lernen von zeitvarianten Systemen. Allerdings hat die Einführung des Vergessensfaktors zur Folge, dass die Parameteradaptation

erheblich empfindlicher gegenüber Störeinflüssen wird. Das bedeutet, dass die Parameteradaptation entsprechend Abb. 7.23 erheblich vom willkürlichen Kraftanteil beeinflusst wird, wie Abb. 7.24 zeigt.

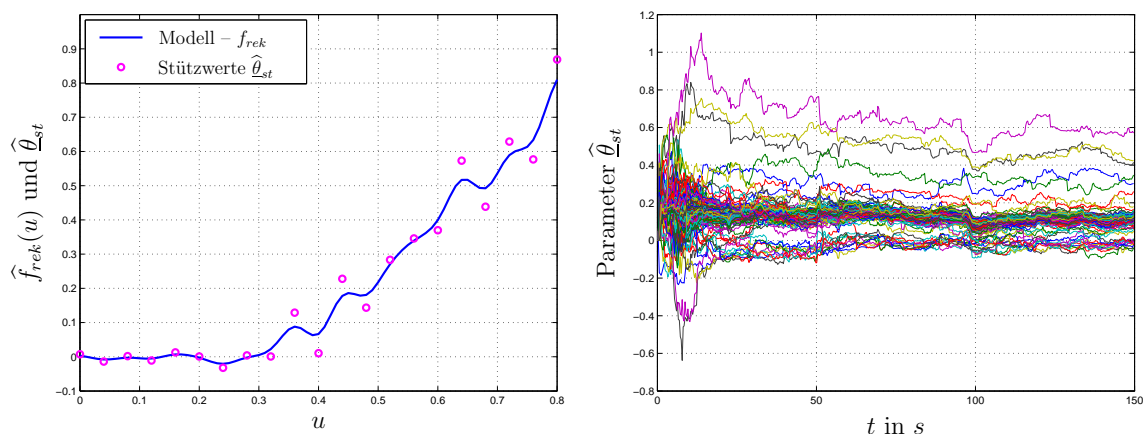


Abb. 7.24: Identifikationsergebnis mit Vergessensfaktor — Rekrutierungskurve (links) und Konvergenz der Parameter (rechts)

In Abb. 7.24 links ist zu erkennen, dass sich der schwingende Verlauf der Rekrutierungskurve erheblich verbessert hat. Außerdem ist die Rekrutierungskurve aufgrund des Ermüdungseinflusses in ihrer Höhe reduziert. Es wird jedoch deutlich, dass die Konvergenz der Parameter (vgl. Abb. 7.24 rechts) durch die Einführung des Vergessensfaktors aufgrund des willkürlichen Kraftanteils erheblich gestört ist und somit kein optimales Identifikationsergebnis möglich ist. Die beschriebenen Effekte aufgrund der Ermüdung und des willkürlichen Kraftanteils konnten auch in der Simulation nachvollzogen werden, so dass aus diesen Erkenntnissen folgt, dass das Problem im gemeinsamen Auftreten von zwei Störfaktoren liegt. Einerseits tritt während des Stimulationsvorgangs eine Ermüdung des Muskels auf, welche einen welligen Verlauf der identifizierten Rekrutierungskurve hervorruft. Andererseits verhindert der willkürliche Kraftanteil, welcher als Rauschen auftritt, die Erweiterung des RLS-Algorithmus mit einem Vergessensfaktor zur Reduktion des Einflusses der Ermüdung, da ansonsten die Konvergenz der Parameter nicht mehr gewährleistet ist.

Um ein eindeutiges Identifikationsergebnis zu erhalten, muss mindestens einer der beiden Störfaktoren ermittelt und kompensiert werden. Bezüglich der Bestimmung des willkürlichen Kraftanteils laufen am Klinikum Rechts der Isar gegenwärtig vielversprechende Versuche, bei denen durch Messungen auf der Haut mittels Elektromyogramm (EMG) der Störeinfluss aufgrund des willkürlichen Kraftanteils eliminiert werden soll. Für die folgende Identifikation wird die Ermüdung in der Rekrutierungskurve berücksichtigt. Aufgrund der Tatsache, dass die Ermüdung über der Zeit auftritt, wird neben der Stimulation u als weitere Eingangsgröße die Zeit t verwendet. Somit ergibt sich für die Identifikation das in Abb. 7.25 veranschaulichte Hammerstein-Modell mit einer zweidimensionalen statischen Nichtlinearität, die als Eingangsgrößen die Stimulation u sowie die Zeit t hat.

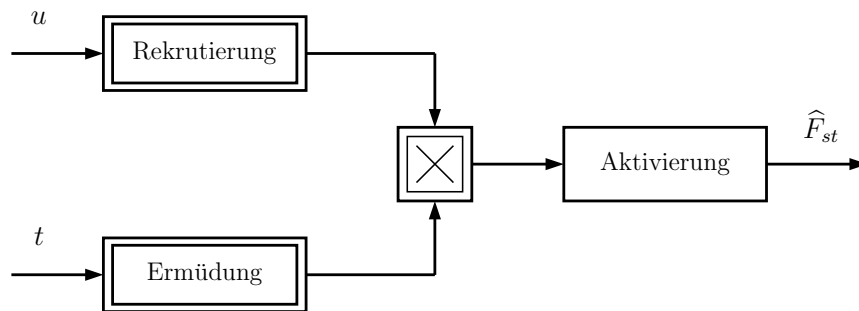


Abb. 7.25: Hammerstein-Modell mit zweidimensionaler statischer Nichtlinearität zur Identifikation der zeitvarianten Rekrutierungskurve

Die Identifikation erfolgt mit dem Ansatz aus Kapitel 5.4.1. Es werden die Einstellwerte nach Tabelle 7.8 verwendet.

Einstellwerte	Erläuterung
$q = 121$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 1.0$	normierte Standardabweichung des GRNN
$m = 60$	Antwortlänge
$m_r = 6$	Basisfunktionenanzahl
$\zeta = 10.0$	Formfaktor
$p = m_r \cdot q = 726$	Parameteranzahl
$h = 0.01s$	Abtastzeit

Tabelle 7.8: Einstellwerte für die Identifikation der Rekrutierungskurve mit Ermüdungseinfluss und der Aktivierungsdynamik

In Abb. 7.26 ist die identifizierte Rekrutierungskurve mit dem Einfluss der Ermüdung und die Konvergenz einiger Parameter dargestellt. Es ist zu erkennen, dass sich der Verlauf der Rekrutierungskurve durch die Berücksichtigung der Ermüdung um ein vielfaches verbessert hat. Über der Stimulationsachse wird die Arcustangens-Form der Rekrutierungskurve deutlich. Der wellige Verlauf ist fast vollständig verschwunden. Über der Zeitachse ist die Ermüdung gut zu erkennen. Nach einer Ermüdungsphase bis ca. 120 Sekunden zeichnet sich eine leichte Erholung während des letzten Zeitabschnitts der Identifikation ab. Dieses Verhalten des Muskels erscheint plausibel. Der Muskel besteht aus schnellen und langsamen Fasern. Die schnellen Fasern sind für eine schnelle Kraftentwicklung verantwortlich, ermüden aber auch sehr schnell. Die langsamen Fasern ermüden über einen längeren Zeitraum sehr viel langsamer als die schnellen Fasern. Da der Einfluss der schnellen Fasern auf die Kraftbildung bereits nach kurzer Zeit abnimmt, tritt die in Abb. 7.26 links zu erkennende Stabilisierung der Ermüdung aufgrund der dominierenden langsamen Fasern auf. Es ist noch anzumerken, dass auch dieses Ergebnis durch den Störeinfluss des willkürlichen Kraftanteils beeinflusst ist. Durch die Wahl der Zeit t als Eingangsgröße für den Ermüdungseinfluss bleibt für die Parameter der einzelnen Rekrutierungskurven nur ein begrenztes Zeitfenster zur Adaption. Der RLS-Algorithmus ist zwar in der

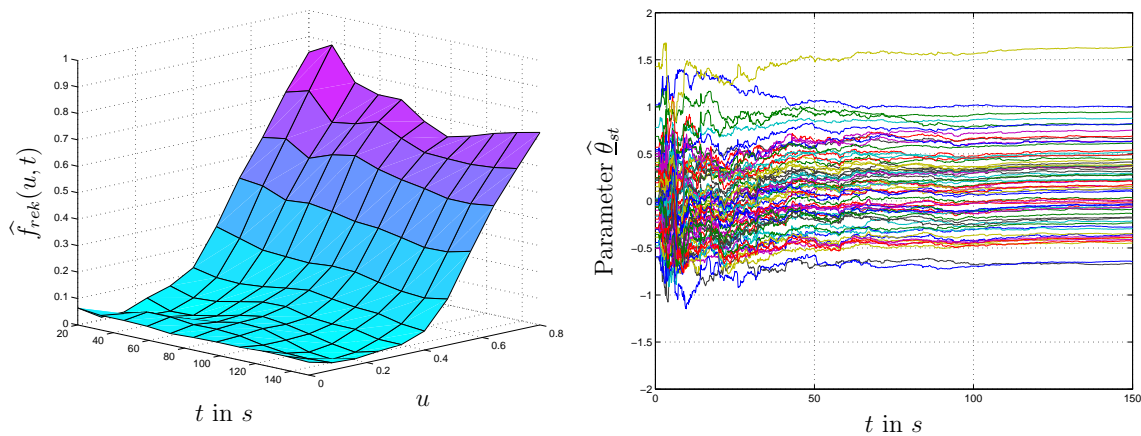


Abb. 7.26: *Identifikationsergebnis — Rekrutierungskurve mit Ermüdungseinfluss (links) und Konvergenz ausgewählter Parameter (rechts)*

Lage Störeinflüsse zu eliminieren, allerdings ist hierzu eine genügend lange Adaptionsdauer nötig. Wie in Abb. 7.26 rechts zu erkennen ist, ist diese Adaptionsdauer nicht lange genug, damit die Parameter vollständig konvergieren können. Somit ist die Eliminierung des Störeinflusses durch den willkürlichen Kraftanteil für ein optimales Ergebnis eine zwingende Voraussetzung.

Die in diesem Kapitel vorgestellten Forschungsarbeiten sind noch nicht abgeschlossen und werden im Rahmen eines Forschungsprojektes der DFG fortgesetzt.

7.3 Kurzzusammenfassung

In diesem Kapitel wurde anhand von zwei Anwendungsbeispielen die Einsatzfähigkeit des vorgestellten Identifikationsverfahrens in der Praxis veranschaulicht. Im ersten Beispiel wurde die entwickelte Theorie zur Identifikation eines mechatronischen Antriebssystems angewandt. Hierbei stellte die Mechanik das bekannte lineare Teilsystem dar. Die dynamische Nichtlinearität bestand aus der nichtlinearen Umrichterdynamik sowie der nichtlinearen Reibkennlinie der Maschinenlager. Die Ergebnisse zeigten, dass das Verfahren praxistauglich ist.

Das zweite Anwendungsbeispiel behandelte einen biomechanischen Aktor. Dieses System ist auf den ersten Blick völlig unterschiedlich. Es zeigte sich jedoch, dass auch die Identifikation des Muskel–Sehne–Komplexes am menschlichen Oberarm mit der vorgestellten Theorie erfolgen kann. Eine Schwierigkeit, die sich schon beim ersten Beispiel angedeutet hat, ist die langsame Konvergenz der Parameter mit dem Fehlermodell 4 basierend auf dem Gradientenabstiegsverfahren. Dies war bei der Identifikation des Antriebssystems noch hinnehmbar, im Falle der Identifikation am Menschen jedoch nicht akzeptabel. Zu diesem Zweck wurde das Fehlermodell 4 basierend

auf dem RLS-Algorithmus entwickelt und in der Simulationsumgebung erfolgreich getestet. Eine erste Validierung des Verfahrens zeigte, dass die Rekrutierungskurve aufgrund der Muskelermüdung zeitvariant ist und zusätzlich ein willkürlicher Kraftanteil der Patienten als Störsignal auftritt. Durch die Erweiterung des Modells um den Ermüdungseinfluss konnten gute Ergebnisse erzielt werden. Eine weitere Verbesserung kann durch eine zusätzliche messtechnische Bestimmung des willkürlichen Kraftanteils erzielt werden. Hierzu finden weitere Forschungsarbeiten am Klinikum Rechts der Isar statt.

8 Identifikation von dynamischen Nichtlinearitäten in strukturierten rekurrenten Neuronalen Netzen

In dieser Arbeit wurde bisher immer davon ausgegangen, dass die Parameter des bekannten linearen Teilsystems vorliegen. Diese Annahme ist in der Praxis jedoch nicht immer erfüllt. In Abb. 6.1 wurde das strukturierte rekurrente Neuronale Netz als eine Erweiterung des Neuronalen Beobachters kurz vorgestellt. Mit dem SRNN ist es möglich die linearen Parameter und die statischen Nichtlinearitäten in einem System, dessen Struktur exakt bekannt ist, gleichzeitig zu identifizieren.

Im Folgenden soll gezeigt werden, dass es durch die Verbindung der Theorie zu den strukturierten rekurrenten Netzen und der in dieser Arbeit vorgestellten Theorie möglich ist, Systeme mit unbekanntem dynamischen Nichtlinearitäten und unbekanntem Parametern, des in seiner Struktur bekannten linearen Teilsystems, zu identifizieren.

8.1 Prinzip der strukturierten rekurrenten Netze

In diesem Kapitel soll kurz das Prinzip der strukturierten rekurrenten Netze erläutert werden. Eine ausführliche Darstellung kann in [2, 16, 23] nachgelesen werden.

Bei der Identifikation eines technischen Prozesses mittels strukturierter rekurrenter Neuronaler Netze wird von einem Modell des Prozesses in Form eines Blockschaltbildes ausgegangen. Die Struktur des Blockschaltbildes wird im rekurrenten Netz übernommen und somit als Vorwissen in die Identifikation eingebracht. Die physikalische Interpretierbarkeit der Netzwerkgewichte ist durch die Analogie zum Blockschaltbild gegeben. In Abb. 8.1 sind die Transformationen der Elemente eines Blockschaltbildes in ein zeitdiskretes strukturiertes rekurrentes Neuronales Netz dargestellt¹. Die Summationspunkte des Blockschaltbildes werden zu den Neuronen und die Parameter zu den Gewichten zwischen den Neuronen des rekurrenten Netzes. Die Integratoren werden entsprechend der Vorwärts-Rechteckapproximation²

$$y_{int}[k] = h \cdot u_{int}[k - 1] + y_{int}[k - 1]$$

¹ h bezeichnet die Abtastzeit des zeitdiskreten SRNN.

²Die Rechteckapproximation wird auch als Euler-Vorwärts-Approximation bezeichnet.

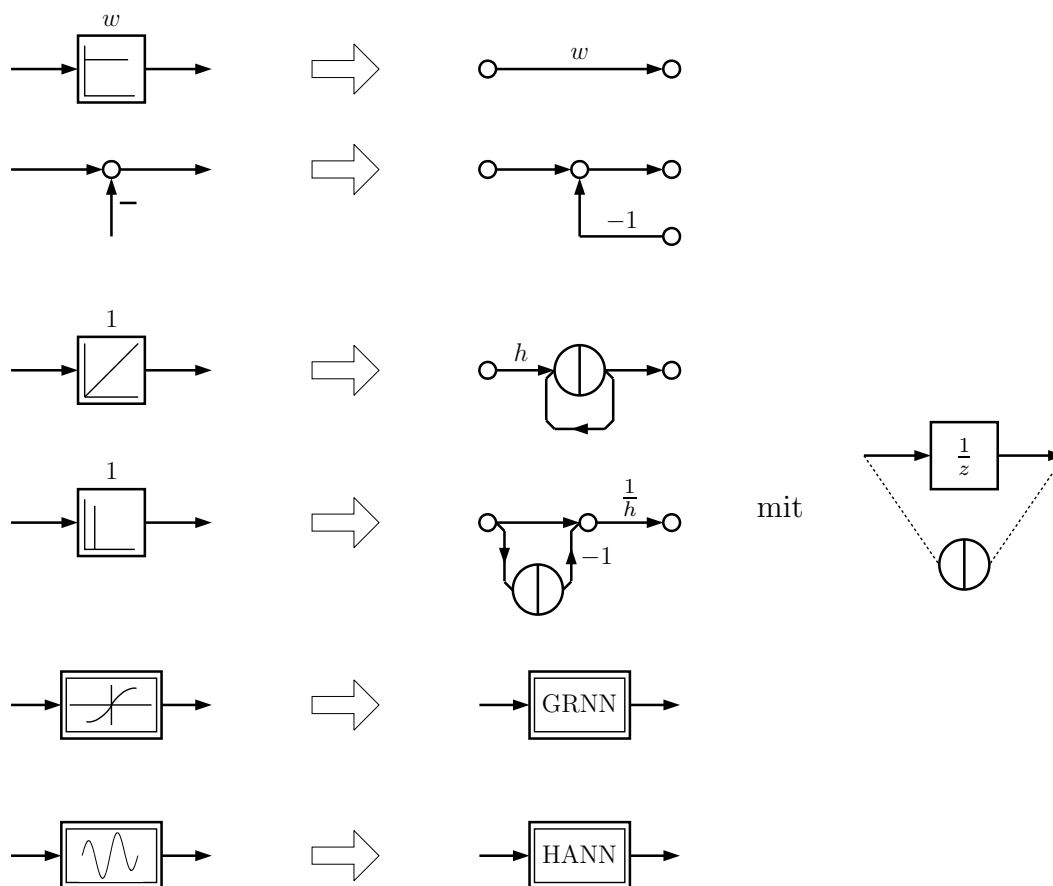


Abb. 8.1: Transformation der Elemente eines Blockschaltbildes in ein strukturiertes rekurrentes Neuronales Netz

mit dem Integratorausgang y_{int} , dem Integratoreingang u_{int} und der Abtastzeit h nach [49] implementiert. Die Rechteckapproximation ist im Vergleich zu anderen Verfahren die einfachste und ungenaueste Form der numerischen Integration. Vorteilhaft ist jedoch die geringe Rechenzeitanforderung. Aus diesem Grund kann die Abtastzeit kleiner gewählt werden, wodurch die Ungenauigkeit der Rechteckapproximation wieder verringert wird. Für die numerische Differentiation wird nach [50] die Form

$$y_{dif}[k] = \frac{1}{h} \cdot (u_{dif}[k] - u_{dif}[k-1])$$

verwendet. Dabei ist u_{dif} der Eingang und y_{dif} der Ausgang des Differentiationsgliedes. Da die Differenz $u_{dif}[k] - u_{dif}[k-1]$ bei kleinen Abtastzeiten sehr klein wird, ist die numerische Differentiation im Vergleich zur numerischen Integration sehr empfindlich gegenüber Rechenungenauigkeiten. Aus diesem Grund werden Signalflusspläne soweit wie möglich aus Integratoren aufgebaut. Statische Nichtlinearitäten werden durch Subnetze repräsentiert. Als Subnetze werden beispielsweise das GRNN oder das HANN eingesetzt, die beide wiederum aus Neuronen und trainierbaren Gewichten bestehen.

Erste vielversprechende Ansätze zu den strukturierten rekurrenten Netzen sind in [8] zu finden. Jedoch stellt in [8] die Korrektur der Zustände des SRNN, die aufgrund der untrainierten Netzwerkgewichte im Allgemeinen nicht den Systemzuständen entsprechen, ein großes Problem dar. In [16] wird dieses Problem gelöst, indem das SRNN in Form eines Beobachters realisiert wird.

Als Lerngesetz wird in [16] das Gradientenabstiegsverfahren verwendet³. Die Herausforderung bei der Implementierung des Lerngesetzes besteht in der Berechnung der partiellen Ableitungen des Ausgangssignals des SRNN nach den unbekannt Parametern des SRNN. Eine entsprechende Vorgehensweise wurde in dieser Arbeit bereits in Kapitel 5.3 bzw. 6.3.2 angewandt.

8.2 Identifikation einer dynamischen Nichtlinearität mit einem strukturierten rekurrenten Netz

Als Grundlage für die Untersuchungen in diesem Kapitel dient die bereits in Kapitel 7.1 vorgestellte Versuchsanlage, deren technische Daten im Anhang B zusammengefasst sind. Für die folgenden Untersuchungen werden die beiden Maschinen durch eine starre Welle miteinander verkoppelt. Eine Verdrehung der Welle kann vernachlässigt werden, so dass die Anlage als Einmassensystem betrachtet werden kann. Die Versuchsanordnung ist in Abb. 8.2 noch einmal verdeutlicht.



Abb. 8.2: *Maschinen mit mechanischer Kopplung (links) und den zugehörigen Umrichtern (rechts)*

Maschine I mit Umrichter I dient als Antriebseinheit, während Maschine II mit Umrichter II als Lasteinheit angesehen werden kann. Im Folgenden wird abhängig von der Drehzahl der Maschinen durch eine statische Nichtlinearität ein Sollbremsmoment, das durch ein zusätzliches PT_1 -Glied mit einer Zeitkonstanten T_V verzögert

³Es werden verschiedene Varianten vorgeschlagen, beispielsweise auch die Einführung eines Momentumterms zur schnelleren Konvergenz der Parameter.

wird, auf den Eingang des zweiten Umrichters gegeben. Durch den Umrichter II mit der Zeitkonstanten T_b wird somit ein Luftspaltmoment M_B an der Maschine II erzeugt, das dem vom Umrichter I hervorgerufenen Luftspaltmoment M_A an der Maschine I entgegenwirkt. Der Aufbau des Antriebsmomentes M_A an der Maschine I erfolgt durch den Umrichter I mit einer Zeitkonstanten T_a ⁴. Die statische Nichtlinearität \mathcal{NL} und $F_1(s)$, in Form eines PT₁-Gliedes, werden als Software implementiert. Für das Gesamtsystem ergibt sich somit das in Abb. 8.3 dargestellte Blockschaltbild⁵:

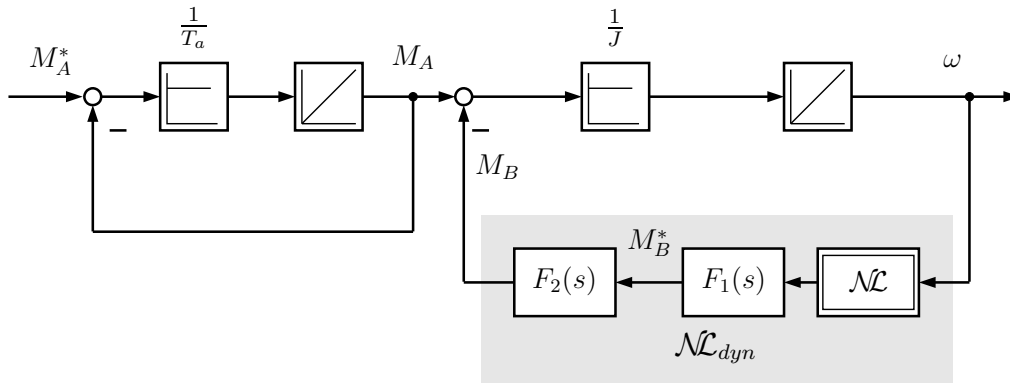


Abb. 8.3: Blockschaltbild des Einmassensystems

Wird das System aus Abb. 8.3 entsprechend Gl. (6.1) dargestellt, ergibt sich:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_a} & 0 \\ \frac{1}{J} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{T_a} \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} \cdot \mathcal{NL}_{dyn}(x_2) \quad (8.1)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{mit} \quad \underline{x}^T = [M_A, \omega] \quad \text{und} \quad u = M_A^*$$

Die dynamische Nichtlinearität $\mathcal{NL}_{dyn}(x_2)$ stellt ein Hammerstein-Modell dar. Für die Übertragungsfunktion des Hammerstein-Modells gilt:

$$F(s) = F_1(s) \cdot F_2(s) = \frac{1}{s \cdot T_V + 1} \cdot \frac{1}{s \cdot T_b + 1} = \frac{1}{T_V \cdot T_b \cdot s^2 + (T_V + T_b) \cdot s + 1} \quad (8.2)$$

Der entscheidende Unterschied im Vergleich zu den vorangegangenen Untersuchungen ist an dieser Stelle, dass neben der dynamischen Nichtlinearität auch die Parameter des in seiner Struktur bekannten linearen Teilsystems unbekannt sein sollen. Anschaulich heißt das, dass die genauen Werte von T_a und J sowie die dynamische Nichtlinearität unbekannt sind. Im Folgenden wird gezeigt, dass es mit Hilfe

⁴Es wird angenommen, dass das Übertragungsverhalten von Soll- zu Istmoment bei beiden Maschinen als ein PT₁-Glied modelliert werden kann. Dies entspricht den Erfahrungen in der Realität [2].

⁵Der Einfluss der Reibung sowie der in Kapitel 7.1 festgestellten periodischen Nichtlinearität soll zunächst vernachlässigt werden.

eines SRNN möglich ist, die linearen Parameter sowie die dynamische Nichtlinearität gleichzeitig zu identifizieren. Somit wird die Einschränkung aus Kapitel 6.2, dass die linearen Parameter des Systems bekannt sein müssen, aufgehoben.

8.2.1 Identifikation in der Simulationsumgebung

Der Entwurf des strukturierten rekurrenten Neuronalen Netzes erfolgt in Beobachterform und stützt sich auf die Zustandsdarstellung des Systems. Ein SRNN in Beobachterform und mit messbarem Eingangsraum der dynamischen Nichtlinearität kann analog zu Gl. (6.4) wie folgt angesetzt werden [2, 54]:

$$\begin{aligned}\dot{\hat{x}} &= \hat{\mathbf{A}} \cdot \hat{x} + \hat{\mathbf{b}} \cdot u + \hat{\mathbf{k}} \cdot \widehat{\mathcal{N}}_{dyn}(\underline{x}_{\mathcal{N}}, u) - \underline{l} \cdot e \\ \hat{y} &= \hat{\mathbf{c}}^T \cdot \hat{x}\end{aligned}\quad (8.3)$$

Der Beobachterfehler ist in Gl. (8.3) definiert zu:

$$e = \hat{y} - y \quad (8.4)$$

Die Struktur des SRNN entspricht der Struktur des Systems und ist in Form von $\hat{\mathbf{A}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}^T$ und $\hat{\mathbf{k}}$ gegeben. Aufgrund der Annahme, dass die Parameter des linearen Teilsystems unbekannt sind, enthalten $\hat{\mathbf{A}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}^T$ und $\hat{\mathbf{k}}$ jedoch zu identifizierende lineare Parameter. Für das vorliegende System ergibt sich das SRNN zu:

$$\begin{aligned}\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} &= \begin{bmatrix} -\hat{\Psi}_1 & 0 \\ \hat{\Psi}_2 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \hat{\Psi}_1 \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -\hat{\Psi}_2 \end{bmatrix} \cdot \widehat{\mathcal{N}}_{dyn}(x_2) - \underline{l} \cdot e \\ \hat{y} &= [0 \quad 1] \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \quad \text{mit} \quad \hat{\underline{x}}^T = [\widehat{M}_A, \widehat{\omega}] \quad \text{und} \quad u = M_A^*\end{aligned}\quad (8.5)$$

Die zu identifizierenden linearen Parameter wurden in Gl. (8.5) mit

$$\hat{\Psi}_1 = \frac{1}{\widehat{T}_a} \quad \text{und} \quad \hat{\Psi}_2 = \frac{1}{\widehat{J}} \quad (8.6)$$

abgekürzt. Für die Identifikation der dynamischen Nichtlinearität kommt an dieser Stelle der Funktionsapproximatoransatz aus Kapitel 5.2 zum Einsatz. Für die dynamische Nichtlinearität gilt allgemein:

$$\widehat{\mathcal{N}}_{dyn}(x_2) = \hat{\underline{\theta}}^T \cdot \underline{\mathcal{A}}_{dyn}(x_2) \quad (8.7)$$

Die linearen Parameter und die Parameter der dynamischen Nichtlinearität werden aus Gründen einer kompakteren Darstellung zu einem Parametervektor $\hat{\underline{w}}$ zusammengefasst [16, 54]:

$$\hat{\underline{w}}^T = [\hat{w}_1, \dots, \hat{w}_p] = [\hat{\Psi}_1, \hat{\Psi}_2, \hat{\theta}_1, \dots, \hat{\theta}_{q \cdot m_r}] \quad (8.8)$$

Aufgrund der Tatsache, dass die unbekannt Parameter \hat{w} in der Regel nichtlinear in das Ausgangssignal des SRNN eingehen, kommt als Lerngesetz das Gradientenabstiegsverfahren zum Einsatz. Für das Gradientenabstiegsverfahren müssen die partiellen Ableitungen des Ausgangssignals des SRNN nach den unbekannt Parametern berechnet werden. Für das Adaptionsgesetz in zeitkontinuierlicher Form gilt somit⁶:

$$\dot{\hat{w}}_i = -\eta_i \cdot e \cdot \frac{\partial \hat{y}}{\partial \hat{w}_i} \quad i = 1 \dots p \quad (8.9)$$

In der Praxis werden für die einzelnen linearen Parameter und die Parameter der dynamischen Nichtlinearität aus Konvergenzgründen eigene Lernschrittweiten eingeführt. Der Ausgangspunkt für die Berechnung der partiellen Ableitungen ist die Zustandsdarstellung des rekurrenten Netzes nach Gl. (8.3). Allgemein ergeben sich die partiellen Ableitung nach den unbekannt Parametern zu:

$$\frac{\partial \hat{y}}{\partial \hat{w}_i} = \underline{\hat{c}}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}_i} \quad (8.10)$$

mit

$$\frac{\partial \hat{x}}{\partial \hat{w}_i} = \hat{\mathbf{A}} \cdot \frac{\partial \hat{x}}{\partial \hat{w}_i} + \frac{\partial \hat{\mathbf{A}}}{\partial \hat{w}_i} \cdot \hat{x} + \frac{\partial \hat{b}}{\partial \hat{w}_i} \cdot u + \hat{k} \cdot \frac{\partial \widehat{\mathcal{N}}_{dyn}(x_2)}{\partial \hat{w}_i} + \frac{\partial \hat{k}}{\partial \hat{w}_i} \cdot \widehat{\mathcal{N}}_{dyn}(x_2) - \underline{l} \cdot \frac{\partial \underline{\hat{c}}^T \hat{x}}{\partial \hat{w}_i} \quad (8.11)$$

Für das betrachtete System berechnet sich somit die partielle Ableitung nach dem linearen Parameter $\hat{w}_1 = \hat{\Psi}_1$ zu⁷:

$$\frac{\partial \hat{y}}{\partial \hat{w}_1} = \underline{\hat{c}}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}_1} \quad (8.12)$$

mit

$$\frac{\partial \hat{x}}{\partial \hat{w}_1} = \begin{bmatrix} -\hat{w}_1 & -l_1 \\ \hat{w}_2 & -l_2 \end{bmatrix} \cdot \frac{\partial \hat{x}}{\partial \hat{w}_1} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot u \quad (8.13)$$

Analog dazu gilt für die partielle Ableitung nach dem linearen Parameter $\hat{w}_2 = \hat{\Psi}_2$:

$$\frac{\partial \hat{y}}{\partial \hat{w}_2} = \underline{\hat{c}}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}_2} \quad (8.14)$$

mit

$$\frac{\partial \hat{x}}{\partial \hat{w}_2} = \begin{bmatrix} -\hat{w}_1 & -l_1 \\ \hat{w}_2 & -l_2 \end{bmatrix} \cdot \frac{\partial \hat{x}}{\partial \hat{w}_2} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \cdot \hat{x} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \cdot \widehat{\mathcal{N}}_{dyn}(x_2) \quad (8.15)$$

⁶Eine ausführliche Herleitung des Lerngesetzes beim SRNN ist in [16] zu finden.

⁷Es gilt $\underline{\hat{c}}^T = \underline{c}^T$.

Die partiellen Ableitungen nach den Parametern $\hat{\theta}$ der dynamischen Nichtlinearität können bestimmt werden nach:

$$\frac{\partial \hat{y}}{\partial \hat{w}_j} = \underline{c}^T \cdot \frac{\partial \hat{x}}{\partial \hat{w}_j} \quad j = 3 \dots p \quad (8.16)$$

mit

$$\frac{\partial \hat{x}}{\partial \hat{w}_j} = \begin{bmatrix} -\hat{w}_1 & -l_1 \\ \hat{w}_2 & -l_2 \end{bmatrix} \cdot \frac{\partial \hat{x}}{\partial \hat{w}_j} + \begin{bmatrix} 0 \\ -\hat{w}_2 \end{bmatrix} \cdot \frac{\partial \hat{\mathcal{N}}_{dyn}(x_2)}{\partial \hat{w}_j} \quad (8.17)$$

Die partiellen Ableitungen der dynamischen Nichtlinearität nach den Parametern \hat{w}_j ergeben sich einfach zu:

$$\frac{\partial \hat{\mathcal{N}}_{dyn}(x_2)}{\partial \hat{w}_j} = \mathcal{A}_{dyn,j}(x_2) \quad (8.18)$$

Die Gleichungen für die partiellen Ableitungen beschreiben jeweils ein Zustandsmodell, das als Ausgang die entsprechende partielle Ableitung hat. In Abb. 8.4 ist das gesamte Prinzip der Identifikation noch einmal grafisch veranschaulicht.

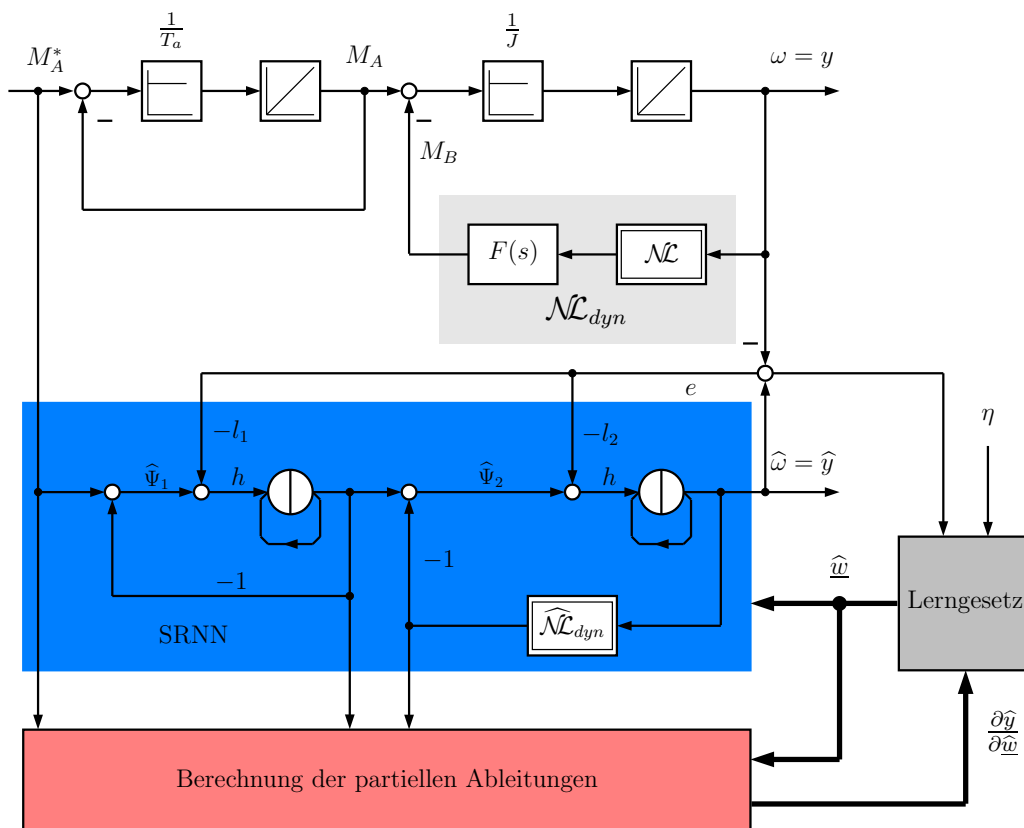


Abb. 8.4: Funktionsprinzip und Aufbau des SRNN

Für die Identifikation in der Simulationsumgebung werden die Annahmen getroffen, dass für die Zeitkonstanten der Umrichter und die Verzögerungszeitkonstante gilt:

$$T_a = T_b = 0.003 \text{ s} \quad T_V = 0.01 \text{ s} \quad (8.19)$$

Das Massenträgheitsmoment ergibt sich aus der Addition der Trägheitsmomente beider Maschinen zu:

$$J = 0.498 \text{ kg m}^2 \quad (8.20)$$

Die statische Nichtlinearität wird als Sättigungscharakteristik wie folgt festgelegt:

$$\mathcal{NL}(x_2) = M_{\mathcal{NL}} \cdot \arctan(x_2) \quad \text{mit} \quad M_{\mathcal{NL}} = 6.5 \text{ Nm} \quad (8.21)$$

Die linearen Parameter T_a und J sowie die dynamische Nichtlinearität, bestehend aus \mathcal{NL} und $F(s)$ sollen im Folgenden unbekannt sein und mit Hilfe des SRNN identifiziert werden. Für die Identifikation werden die Einstellungen nach Tabelle 8.1 verwendet.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 1.0$	normierte Standardabweichung des GRNN
$m = 60$	Antwortlänge
$m_r = 6$	Basisfunktionenanzahl
$\zeta = 13.0$	Formfaktor
$\underline{l}^T = [0, 63.8]$	Beobachterrückführungen
$p = 2 + q \cdot m_r = 128$	Parameteranzahl
$\eta_1 = 4.8$	Lernschrittweite für $\hat{\Psi}_1$
$\eta_2 = 0.005$	Lernschrittweite für $\hat{\Psi}_2$
$\eta_3 = 0.08$	Lernschrittweite für $\hat{\underline{\theta}}$
$h = 0.001 \text{ s}$	Abtastzeit

Tabelle 8.1: *Einstellwerte für die Identifikation mit dem SRNN*

Die Simulationsdauer beträgt insgesamt 1200 Sekunden. Nach 1100 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird mit einem Drehzahlregler, der als Solldrehzahl ein Zufallssignal erhält, im Bereich von $x_2 \in [-10 \frac{\text{rad}}{\text{s}}; 10 \frac{\text{rad}}{\text{s}}]$ angeregt. Die linearen Parameter sind mit $\hat{T}_a[0] = 0.00375 \text{ s}$ und $\hat{J}[0] = 0.619 \text{ kg m}^2$ vorbelegt. Dadurch, dass für die dynamische Nichtlinearität kein Vorwissen eingebracht wird, wird zunächst nur die dynamische Nichtlinearität gelernt und der Lernvorgang für die linearen Parameter nach 35 s bzw. 15 s begonnen⁸.

⁸Dies verhindert, dass die linearen Parameter aufgrund der nicht trainierten dynamischen Nichtlinearität zunächst falsch gelernt werden.

In den Abbildungen 8.5 und 8.6 ist der Fehlerverlauf in der Winkelgeschwindigkeit sowie die Konvergenz der Parameter der dynamischen Nichtlinearität und der linearen Parameter dargestellt.

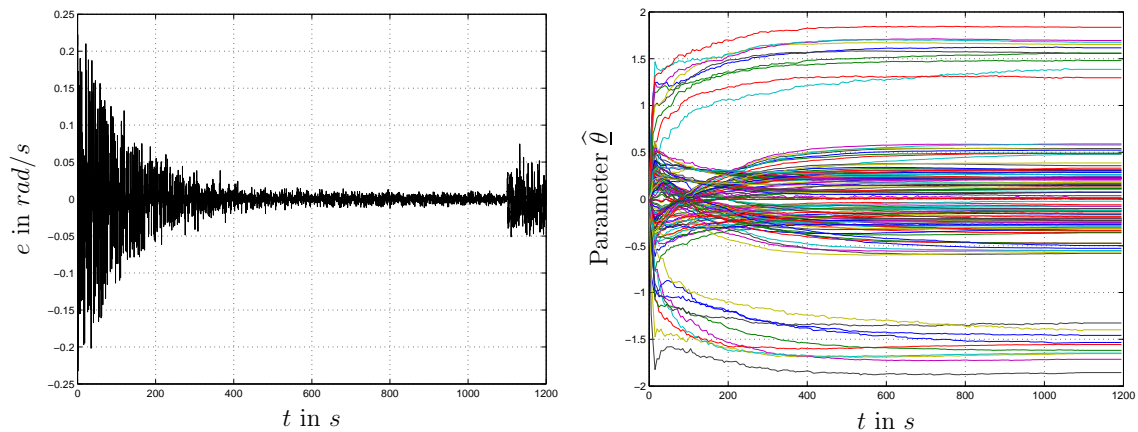


Abb. 8.5: Fehlerverlauf (links) und Konvergenz der Parameter $\hat{\theta}$ (rechts)

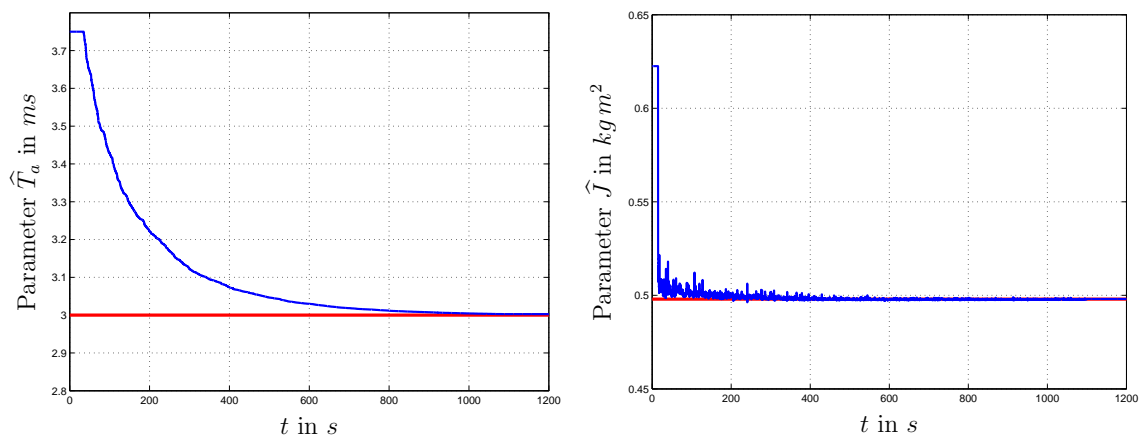


Abb. 8.6: Konvergenz der linearen Parameter \hat{T}_a (links) und \hat{J} (rechts)

Es ist zu erkennen, dass der Fehler e klein wird und die Parameter $\hat{\theta}$ der dynamischen Nichtlinearität konvergieren. Der Fehler steigt jedoch im Parallelbetrieb wieder leicht an. Dies ist darauf zurück zu führen, dass einige Parameter noch nicht vollständig konvergiert haben und die Basisfunktionenanzahl relativ klein gewählt wurde. Zusätzlich wirkt sich der im Vergleich zu den Zeitkonstanten relativ große Abtastzeitschritt negativ aus. Dennoch ist das Ergebnis als sehr gut zu bewerten, da der relative Fehler in der Winkelgeschwindigkeit, bezogen auf eine maximale Winkelgeschwindigkeit von $\omega = 10 \frac{\text{rad}}{s}$, im Parallellauf kleiner als 1% ist. Dies verdeutlicht auch das Konvergenzverhalten der linearen Parameter \hat{T}_a und \hat{J} , die exakt

auf ihre vorgegebenen Werte konvergieren. In Abb. 8.7 ist zur weiteren Veranschaulichung des Identifikationsergebnisses die identifizierte statische Nichtlinearität sowie die identifizierte Impulsantwort dargestellt.

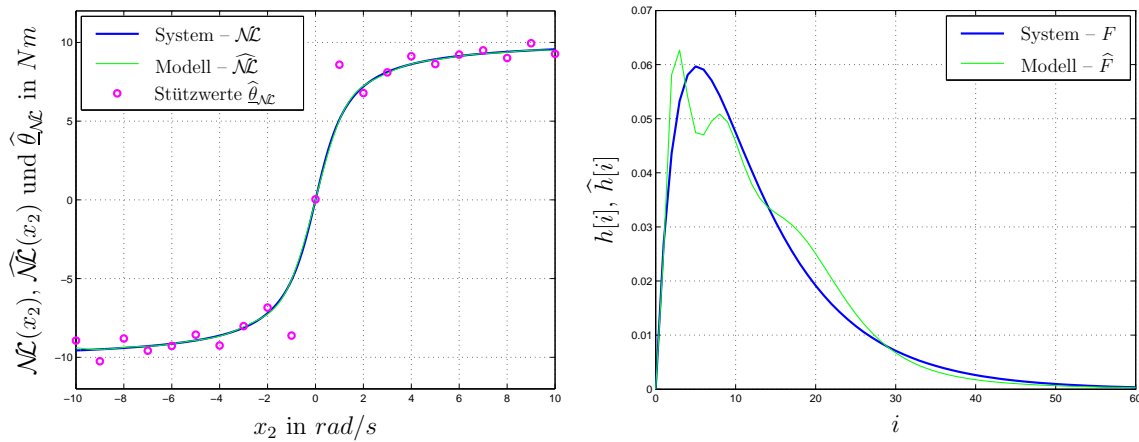


Abb. 8.7: Vergleich zwischen vorgegebener und identifizierter statischer Nichtlinearität (links) sowie Impulsantwort (rechts)

Die statische Nichtlinearität wird durch das GRNN sehr gut identifiziert. Die Impulsantwort weicht geringfügig von der Vorgabe ab. Hierauf ist auch der Fehler im Parallelbetrieb zurück zu führen. Die Gründe wurden bereits genannt.

Die Simulation hat somit gezeigt, dass es mit Hilfe des SRNN möglich ist, lineare Parameter und dynamische Nichtlinearitäten eines Systems gleichzeitig zu identifizieren. In [54] wurde auch der Fall eines nicht messbaren Eingangsraumes der dynamischen Nichtlinearität untersucht. Hierbei können sich bei den partiellen Ableitungen deutlich komplexere Terme ergeben. Die Identifikation ist jedoch prinzipiell genauso möglich.

8.2.2 Validierung am realen System

In den bisherigen Betrachtungen wurden die Reibung und die periodische Nichtlinearität (vgl. Kapitel 7.1), die an der realen Anlage auftreten, vernachlässigt. Versuche an der Anlage zeigen jedoch, dass die beiden Kennlinien einen deutlichen Einfluss auf das Systemverhalten haben. Aus diesem Grund wird das Modell an dieser Stelle um beide Effekte erweitert. In Abb. 8.8 ist das System mit den zusätzlich modellierten Nichtlinearitäten als Blockschaltbild veranschaulicht. Sowohl die Reibkennlinie als auch die periodische Nichtlinearität werden aus Rechenzeitgründen an dieser Stelle nicht identifiziert. Es werden bereits bekannte Identifikationsergebnisse zur Modellierung verwendet (vgl. Kapitel 7.1.2 bzw. [2]). Im Lerngesetz muss jedoch der Beitrag der Reibkennlinie und der periodischen Nichtlinearität zu den partiellen Ableitungen berücksichtigt werden. Die Bildung der partiellen Ableitungen erfolgt in analoger Weise zu Kapitel 8.2.1 und soll an dieser Stelle nicht wiederholt werden.

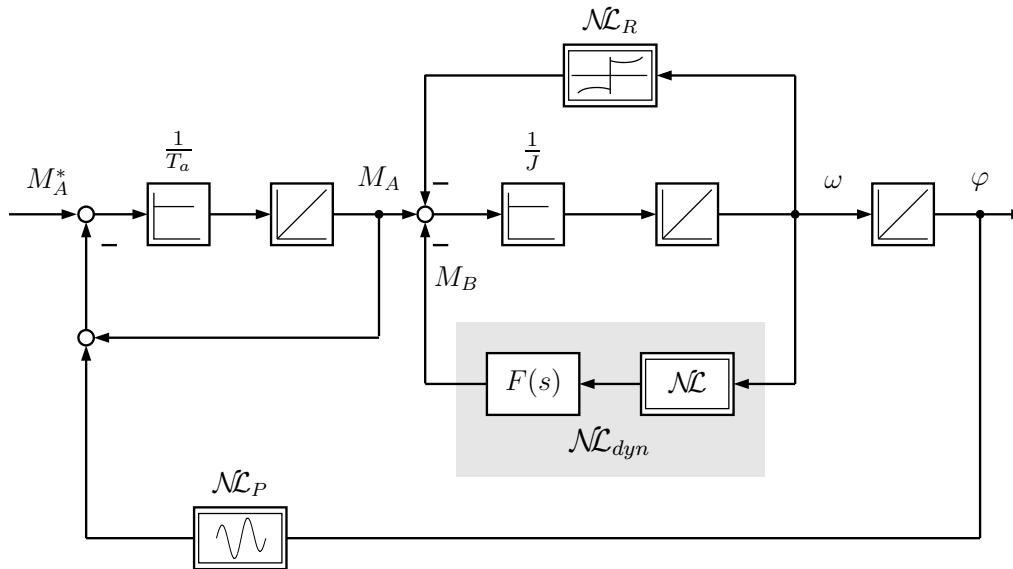


Abb. 8.8: Erweitertes Blockschnittbild des Einmassensystems

Mit dem erweiterten SRNN erfolgt die Identifikation der linearen Parameter T_a und J sowie der dynamischen Nichtlinearität, bestehend aus \mathcal{NL} und $F(s)$ an der realen Anlage. Die Identifikation würde jedoch bereits mit der Parameteranzahl aus Tabelle 8.1 nicht mehr die Echtzeitbedingung erfüllen, so dass ein Datensatz aufgenommen wird und die Identifikation quasi-online am Rechner erfolgt. Die Parameteranzahl kann dadurch zu Gunsten eines besseren Lernergebnisses erhöht werden. Für die Identifikation werden die Einstellungen nach Tabelle 8.2 verwendet.

Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 1.0$	normierte Standardabweichung des GRNN
$m = 60$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 13.0$	Formfaktor
$l^T = [0, 63.8]$	Beobachterrückführungen
$p = 2 + q \cdot m_r = 212$	Parameteranzahl
$\eta_1 = 6.2$	Lernschrittweite für $\hat{\Psi}_1$
$\eta_2 = 0.001$	Lernschrittweite für $\hat{\Psi}_2$
$\eta_3 = 0.08$	Lernschrittweite für $\hat{\theta}$
$h = 0.001s$	Abtastzeit

Tabelle 8.2: Einstellwerte für die Identifikation mit dem erweiterten SRNN

Die Identifikationsdauer beträgt insgesamt 1000 Sekunden. Nach 900 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird mit einem Drehzahlregler, der als Solldrehzahl ein Zufallssignal

erhält, im Bereich von $x_2 \in [-10 \frac{\text{rad}}{\text{s}} ; 10 \frac{\text{rad}}{\text{s}}]$ angeregt. Die linearen Parameter sind mit $\hat{T}_a[0] = 0.00431 \text{ s}$ und $\hat{J}[0] = 0.716 \text{ kg m}^2$ vorgelegt. Dadurch, dass für die dynamische Nichtlinearität kein Vorwissen eingebracht wird, wird analog zur Simulation zunächst nur die dynamische Nichtlinearität gelernt und der Lernvorgang für die linearen Parameter nach 25 s bzw. 10 s gestartet. In den Abbildungen 8.9 und 8.10 ist der Fehlerverlauf in der Winkelgeschwindigkeit sowie die Konvergenz der Parameter der dynamischen Nichtlinearität und der linearen Parameter dargestellt.

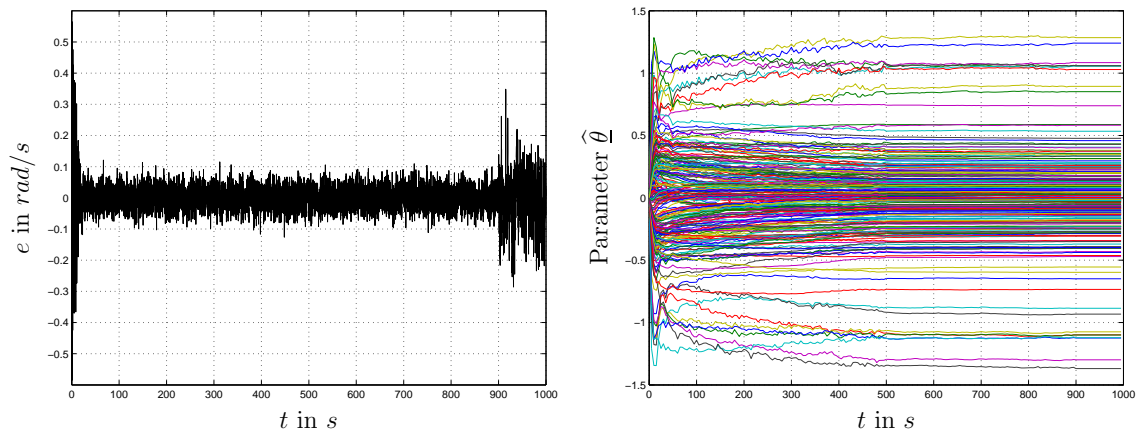


Abb. 8.9: Fehlerverlauf (links) und Konvergenz der Parameter $\hat{\theta}$ (rechts)

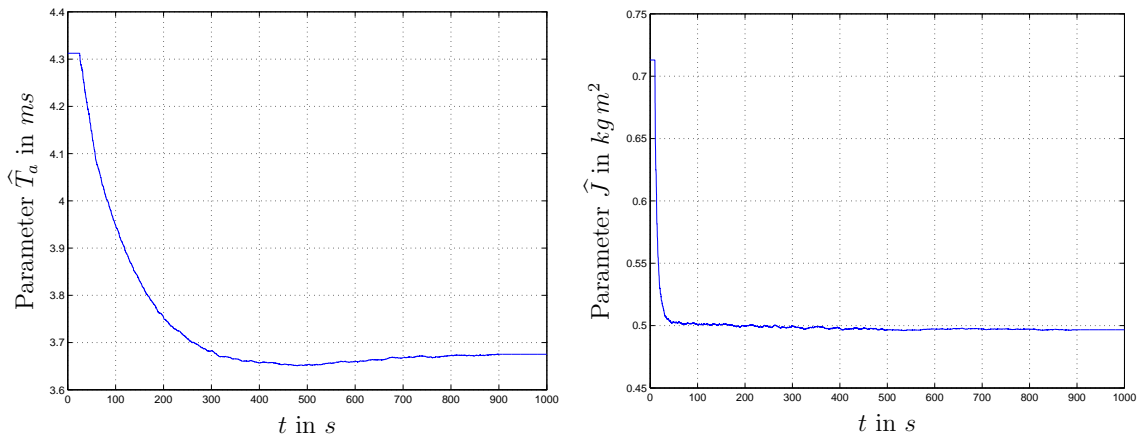


Abb. 8.10: Konvergenz der linearen Parameter \hat{T}_a (links) und \hat{J} (rechts)

Es ist zu erkennen, dass der Fehler e bereits nach ca. 30 Sekunden auf das Messrauschen zurück geht. Im Parallellauf steigt der Fehler wieder leicht an. Dennoch ist das Ergebnis als sehr gut zu bewerten, da der relative Fehler in der Winkelgeschwindigkeit, bezogen auf eine maximale Winkelgeschwindigkeit von $\omega = 10 \frac{\text{rad}}{\text{s}}$, im Parallellauf kleiner als 4 % ist. Dies verdeutlicht auch das Konvergenzverhalten der Parameter $\hat{\theta}$ der dynamischen Nichtlinearität sowie der linearen Parameter \hat{T}_a und \hat{J} . Die

Zeitkonstante \hat{T}_a konvergiert gegen einen Wert von ca. 3.68 ms , was den Erfahrungen an der Anlage entspricht. Das Trägheitsmoment \hat{J} konvergiert gegen einen Wert von ca. 0.496 kg m^2 . Dieser Wert erreicht fast exakt das errechnete Trägheitsmoment von 0.498 kg m^2 und stellt somit ein sehr gutes Identifikationsergebnis dar. In Abb. 8.11 ist zur weiteren Veranschaulichung des Identifikationsergebnisses die identifizierte statische Nichtlinearität sowie die identifizierte Impulsantwort dargestellt.

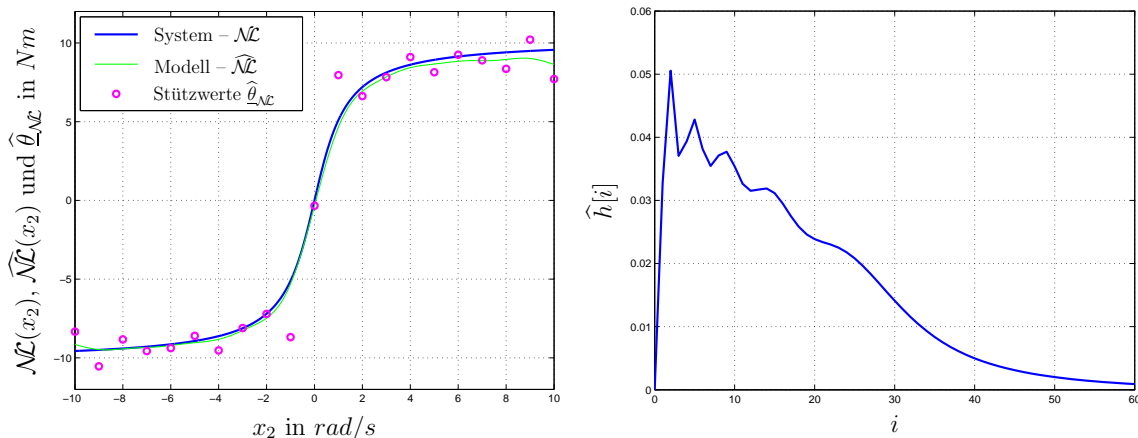


Abb. 8.11: Vergleich zwischen vorgegebener und identifizierter statischer Nichtlinearität (links) sowie Impulsantwort (rechts)

Die statische Nichtlinearität wird durch das GRNN gut identifiziert. Die Impulsantwort zeigt einen vergleichbaren Verlauf wie in der Simulation. Somit hat sich auch bei der Validierung an der Versuchsanlage gezeigt, dass es mit Hilfe des SRNN möglich ist, lineare Parameter und dynamische Nichtlinearitäten eines Systems gleichzeitig zu identifizieren.

8.3 Kurzzusammenfassung

In diesem Kapitel wurde gezeigt, dass es durch die Verbindung der Theorie zu den strukturierten rekurrenten Netzen und der in dieser Arbeit vorgestellten Theorie möglich ist, Systeme mit unbekanntem dynamischen Nichtlinearitäten und unbekanntem Parametern, des in seiner Struktur bekannten linearen Teilsystems, zu identifizieren. Somit wurde die Annahme aufgehoben, dass bei der Identifikation von dynamischen Nichtlinearitäten die Parameter des linearen Teilsystems bekannt sein müssen. Das strukturierte rekurrente Netz wurde dabei in Form eines Beobachters realisiert. Aufgrund der Tatsache, dass die unbekanntem Parameter in der Regel nichtlinear in das Ausgangssignal des SRNN eingehen, kam als Lerngesetz das Gradientenabstiegsverfahren zum Einsatz. Die Berechnung der partiellen Ableitungen des Ausgangssignals des SRNN nach den unbekanntem Parametern stellte hierbei die

Hauptaufgabe dar. Die anschließende Identifikation am Beispiel eines Einmassensystems zeigte in der Simulationsumgebung sehr gute Ergebnisse. Sowohl die linearen Parameter als auch die Parameter der dynamischen Nichtlinearität konvergierten und zeigten eine sehr gute Übereinstimmung mit der Vorgabe. Lediglich bei der Impulsantwort ergaben sich Abweichungen, die jedoch auf die geringe Basisfunktionsanzahl zurück zu führen waren. Die Validierung an der Versuchsanlage bestätigte die guten Ergebnisse aus der Simulation. Allerdings konnte die Identifikation an der Versuchsanlage aufgrund der hohen Rechenzeitanforderung des Identifikationsverfahrens nicht mehr in Echtzeit durchgeführt werden. Die Identifikation erfolgte mit Hilfe eines aufgenommenen Datensatzes quasi-online am Rechner.

9 Zusammenfassung und Ausblick

Als Abschluss der vorliegenden Arbeit sollen die wichtigsten Ergebnisse noch einmal zusammengefasst und ein Ausblick auf weitere Forschungsaktivitäten gegeben werden.

Ausgehend von bekannten Methoden zur Approximation nichtlinearer statischer Funktionen sowie nichtlinearer dynamischer Systeme wurde ein neues Verfahren zur Identifikation mechatronischer Systeme vorgestellt. Die physikalische Modellierung mechatronischer Systeme ist aufgrund der Komplexität dieser Systeme in der Regel sehr aufwändig und schwierig, da die zu modellierenden Systeme in ihrer Struktur nur bedingt bekannt sind. Charakteristisch für mechatronische Systeme ist, dass einerseits die Struktur des mechanischen Teils aufgrund der Wirkungskette Moment – Geschwindigkeit – Position sehr gut bekannt ist, andererseits jedoch die Struktur von nichtlinearen Stellgliedern in der Regel nur mit hohem Aufwand physikalisch exakt modelliert werden kann.

Im Umfeld der Modellierung von mechatronischen Systemen ist der Neuronale Beobachter entstanden, der auf einer nichtlinearen Zustandsdarstellung des Systems beruht und somit zur Klasse der White-Box-Modelle gezählt werden kann. Der Neuronale Beobachter erlaubt die Identifikation von statischen Nichtlinearitäten, wie z.B. von Reib- und Federkennlinien, innerhalb eines in seiner Struktur und in seinen Parametern bekannten dynamischen Modells. Der Neuronale Beobachter stellt die Grundlage für das vorgestellte Identifikationsverfahren dar.

Das Gegenteil zu den White-Box-Modellen stellen Black-Box-Modelle dar, die das Ein-/Ausgangsverhalten eines Systems beschreiben und nur wenig strukturelles Vorwissen benötigen. Jedoch haben die Parameter von Black-Box-Modellen keinerlei physikalische Bedeutung mehr und der Zugriff auf interne Zustände ist nicht möglich.

Eine Alternative zu Black-Box-Modellen stellt die Volterra-Funktionalpotenzreihe dar, die die Beschreibung von blockorientierten Systemen erlaubt, deren Struktur in Form von nichtlinearen statischen Funktionen und dynamischen Übertragungsfunktionen bekannt ist. Sie bietet somit die Möglichkeit Strukturvorwissen in das Identifikationsverfahren einzubringen, so dass das spätere Identifikationsergebnis eine gewisse Transparenz aufweist. Aufgrund der Tatsache, dass es sich bei den Identifikationsalgorithmen auf Basis der Volterra-Funktionalpotenzreihe um Verfahren ohne Ausgangsrückkopplung handelt, weisen sie prinzipielle Vorteile gegenüber Ver-

fahren mit Ausgangsrückkopplung auf. Die Ausgangsfehlerstruktur erlaubt beispielsweise das Führen eines Stabilitätsbeweises nach Lyapunov. Das Problem der hohen Parameteranzahl kann durch die Einführung von orthonormalen Basisfunktionen entschärft werden. Die Identifikation auf Basis der Volterra-Funktionalpotenzreihe stellte sich somit als vielversprechendes Verfahren für eine Erweiterung des Neuronalen Beobachters dar. Die Identifikation beschränkte sich jedoch bisher auf nichtlineare dynamische Systeme mit einem Eingang und einem Ausgang, so dass eine Erweiterung auf Mehrgrößensysteme erfolgen musste.

Die Untersuchungen zur Erweiterung der Volterra-Funktionalpotenzreihe auf Mehrgrößensysteme erfolgten anhand von linear und nichtlinear verkoppelten Hammerstein- und Wiener-Modellen. Bei diesen Untersuchungen wurde der unsymmetrische Volterra-Kern entdeckt, der charakteristisch für nichtlinear verkoppelte Mehrgrößensysteme mit unterschiedlichen Übertragungsfunktionen ist. Die Identifikation der statischen Nichtlinearitäten erfolgte jedoch auf der Basis von Polynomen, was die Anwendbarkeit der abgeleiteten Identifikationsverfahren einschränkte. Aus diesem Grund wurden die Identifikationsverfahren mittels eines Funktionsapproximatoransatzes verbessert. Die Verbesserungen machten sich bei der Approximation der statischen Nichtlinearitäten sowohl beim Interpolationsverhalten als auch beim Extrapolationsverhalten deutlich bemerkbar. Das grundlegende Prinzip ist, dass statische Nichtlinearitäten innerhalb des blockorientierten Systems nicht mehr durch Polynome, sondern durch universelle Funktionsapproximatoren beschrieben werden. Hier eignete sich speziell das GRNN am besten, da dadurch die Linearität der Parameter bei den Identifikationsverfahren weitestgehend erhalten werden konnte. Lediglich bei Systemen mit Wiener-Strukturen zeigte sich, dass die Parameter durch den Funktionsapproximatoransatz nichtlinear in den Modellausgang eingehen. Es konnte jedoch gezeigt werden, dass durch ein verändertes Lerngesetz eine Identifikation solcher Systeme trotzdem möglich ist.

Die entwickelten Identifikationsalgorithmen auf der Basis von Volterra-Reihen wurden im nächsten Schritt in den Neuronalen Beobachter integriert. Mit dem erweiterten Neuronalen Beobachter ist es nun möglich, Systeme mit dynamischen Nichtlinearitäten zu identifizieren. Als dynamische Nichtlinearitäten werden dabei blockorientierte nichtlineare Systeme bezeichnet. Anschaulich bedeutet das, dass der erweiterte Neuronale Beobachter die Möglichkeit bietet, nichtlineare Teilsysteme, die in ihrer Struktur und ihren Parametern weitestgehend unbekannt sind, innerhalb eines bekannten linearen Teilsystems zu identifizieren.

An zwei Anwendungsbeispielen wurde die Einsatzfähigkeit des erweiterten Neuronalen Beobachters in der Praxis veranschaulicht. Im ersten Beispiel wurde die entwickelte Theorie zur Identifikation eines mechatronischen Antriebssystems angewandt. Hierbei stellte die Mechanik das bekannte lineare Teilsystem dar. Die dynamische Nichtlinearität bestand aus der nichtlinearen Umrichterndynamik sowie der nichtlinearen Reibkennlinie der Maschinenlager. Das zweite Anwendungsbeispiel behandelte einen biomechanischen Aktor. Es zeigte sich, dass auch die Identifikati-

on des Muskel–Sehne–Komplexes am menschlichen Oberarm mit der vorgestellten Theorie erfolgen kann. Aufgrund einer schnelleren Konvergenz der Parameter wurde das verwendete Fehlermodell auf den RLS–Algorithmus erweitert.

Durch die Verbindung der Theorie zu den strukturierten rekurrenten Netzen und der in dieser Arbeit vorgestellten Theorie wurde es in einem weiteren Schritt möglich, Systeme mit unbekanntem dynamischen Nichtlinearitäten und unbekanntem Parametern, des in seiner Struktur bekannten linearen Teilsystems, zu identifizieren. Somit wurde die Annahme aufgehoben, dass bei der Identifikation von dynamischen Nichtlinearitäten die Parameter des linearen Teilsystems bekannt sein müssen. Die Berechnung der partiellen Ableitungen des Ausgangssignals des strukturierten rekurrenten Netzes nach den unbekanntem Parametern stellte bei der Ableitung des Lerngesetzes die Hauptaufgabe dar. Die Anwendung des Verfahrens zeigte an einer Versuchsanlage sehr gute Ergebnisse.

Die vorliegende Arbeit stellt ein flexibles Identifikationsverfahren zur Verfügung, bei dem der Anwender entscheidet, wie umfangreich die exakte physikalische Modellierung des Gesamtsystems sein soll. Es baut somit eine Brücke zwischen der exakten White–Box–Modellierung und der Modellierung des Ein–/Ausgangsverhaltens durch Black–Box–Verfahren. Dabei bleibt es dem Anwender überlassen, auf welche Seite er in Anbetracht der Aufgabenstellung den Schwerpunkt legt.

Das Ziel jeglicher Ansätze zur Identifikation eines Systems ist es, eine verbesserte Regelung für dieses System zu entwerfen. Erste vielversprechende Ansätze die gewonnenen Identifikationsergebnisse in eine prädiktive Regelung zu implementieren wurden bereits in [45] durchgeführt. Diese Forschungsaktivitäten werden zukünftig noch vertieft. Desweiteren besteht aufgrund der zunehmenden Komplexität der Systeme und der steigenden Parameteranzahl ein Bedarf an effizienten Lernverfahren, die schnell und stabil zum globalen Minimum des Gütefunktional führen.

A Definitionen

A.1 Definition des $*$ -Operators

Gegeben ist ein Vektor $\underline{a} \in \mathbb{R}^{n \times 1}$. Der $*$ -Operator ist wie folgt definiert:

$$\begin{aligned}\underline{a} * \underline{a} &= \underline{a}^{*2} = [a_1, a_2, \dots, a_n]^T * [a_1, a_2, \dots, a_n]^T \\ &= [a_1 \cdot a_1, a_2 \cdot a_1, \dots, a_n \cdot a_1, a_2 \cdot a_2, \dots, a_n \cdot a_2, \dots, a_n \cdot a_n]^T\end{aligned}$$

Der $*$ -Operator führt zu einem Vektor der alle Kombinationen der Elemente von \underline{a} enthält, wobei die Reihenfolge der Elemente keine Bedeutung hat. Somit gilt:

$$\underline{a}^{*j} \in \mathbb{R}^{\binom{n+j-1}{j} \times 1}$$

A.2 Definition des \otimes -Operators

Gegeben sind zwei Vektoren $\underline{a} \in \mathbb{R}^{n \times 1}$ und $\underline{b} \in \mathbb{R}^{m \times 1}$. Der \otimes -Operator ist wie folgt definiert:

$$\begin{aligned}\underline{a} \otimes \underline{b} &= [a_1, a_2, \dots, a_n]^T \otimes [b_1, b_2, \dots, b_m]^T \\ &= [a_1 \cdot b_1, a_2 \cdot b_1, \dots, a_n \cdot b_1, a_1 \cdot b_2, a_2 \cdot b_2, \dots, a_n \cdot b_2, \dots, a_n \cdot b_m]^T\end{aligned}$$

Der \otimes -Operator führt zu einem Vektor der alle Kombinationen der Elemente von \underline{a} und \underline{b} enthält. Die Dimension des Ergebnisvektors ergibt sich aus dem Produkt der Dimensionen der einzelnen Vektoren.

B Technische Daten der Testanlage

Der gesamte Aufbau besteht aus zwei Systemen, die über eine mechanische Welle miteinander verkoppelt sind. System I, bestehend aus Umrichter I und Maschine I, wird als Antriebseinheit verwendet. System II, bestehend aus Umrichter II und Maschine II, wird zur Belastung der Antriebseinheit eingesetzt. Abbildung B.1 zeigt die beiden Maschinen sowie die beiden Umrichter.



Abb. B.1: *Maschinen mit mechanischer Kopplung (links) und den zugehörigen Umrichtern (rechts)*

Der Schaltplan der Anlage ist in Abb. B.2 dargestellt. Die zwei baugleichen Umrichter der Firma *Siemens* vom Typ *Simovert Master Drives SC 6SE7022-6EC30* weisen folgende Grunddaten auf:

Netzspannung:	3 AC 380 V bis 460 V ($\pm 15\%$)
Zwischenkreisspannung:	DC 510 V bis 620 V ($\pm 15\%$)
Ausgangsspannung:	3 AC 0 V bis 0.86 x Netzspannung
Netzfrequenz:	50/60 Hz ($\pm 15\%$)
Ausgangsfrequenz:	0 Hz bis 400 Hz
Pulsfrequenz:	5 kHz bis 7.5 kHz
Ausgangsbemessungsstrom:	25.2 A
Grundlaststrom:	23.2 A
Kurzzeitstrom:	40.8 A
Verlustleistung:	0.43 kW (bei 5 Hz)
Wirkungsgrad:	96 bis 98 %

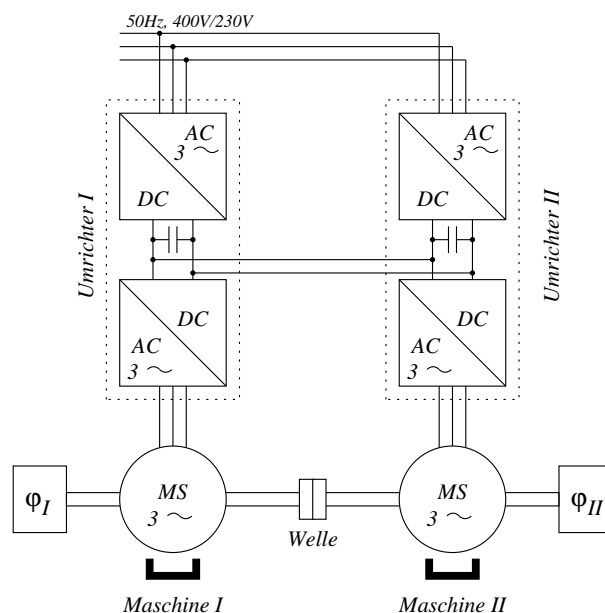


Abb. B.2: Elektrischer Aufbau der Testanlage

Bei den beiden Maschinen handelt es sich um permanenterrechte Synchronmaschinen der Firma *Siemens* vom Typ *1FT6068-8AC71-1CD3* mit integrierten Encodern zur Rotorlageerfassung. Nachfolgende Tabelle zeigt die technischen Grunddaten der Maschinen:

Nennzahl:	2000 min^{-1}
Nennmoment:	23 Nm
Nennstrom:	10.9 A
Nennleistung:	4.8 kW
Trägheitsmoment:	$6.65 \cdot 10^{-3} \text{ kg m}^2$

Die Massenträgheitsmomente der beiden Maschinen, einschließlich Schwungradscheiben, einer Distanzscheibe und der Kupplungskonstruktion errechnen sich zu:

- $J_I = 0.166 \text{ kg m}^2$
- $J_{II} = 0.336 \text{ kg m}^2$

Die Ansteuerung der Umrücker bzw. das Auslesen der Messdaten erfolgt mit einem Standard-PC (Intel Pentium III, 650 MHz Taktfrequenz, 128 MB Hauptspeicher und Echtzeitbetriebssystem *xPC-Target V 1.0* von Mathworks). Die Analogausgabe wird mit dem 12 Bit DA-Wandler *DAS-1602* von Keithley Instruments realisiert. Die Auswertung der Positionsgeber erfolgt mit der Interpolationskarte *Heidenhain IK121V*.

C Bilder des Versuchsaufbaus am Klinikum Rechts der Isar



Abb. C.1: *Bediensystem mit Microcontroller*



Abb. C.2: Stimulator

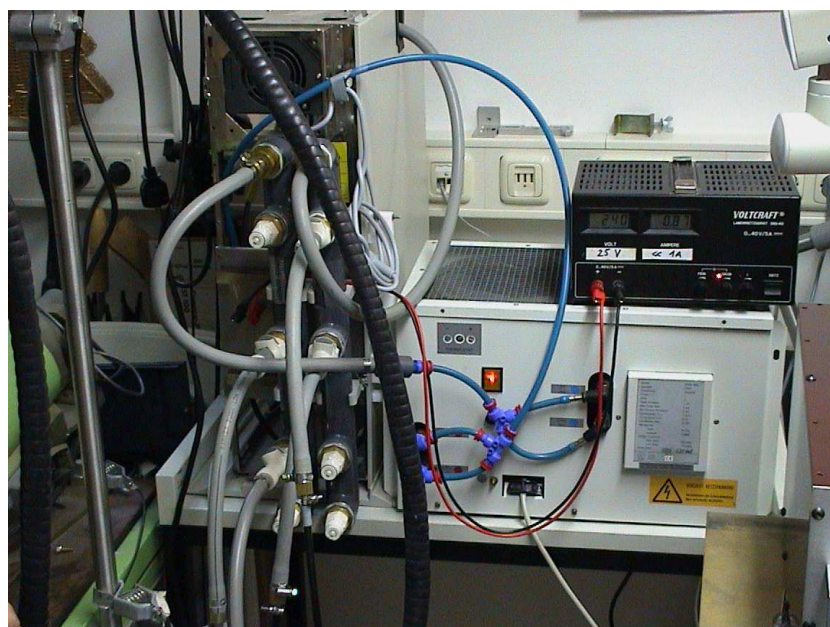


Abb. C.3: Ölkreislauf für die Spule

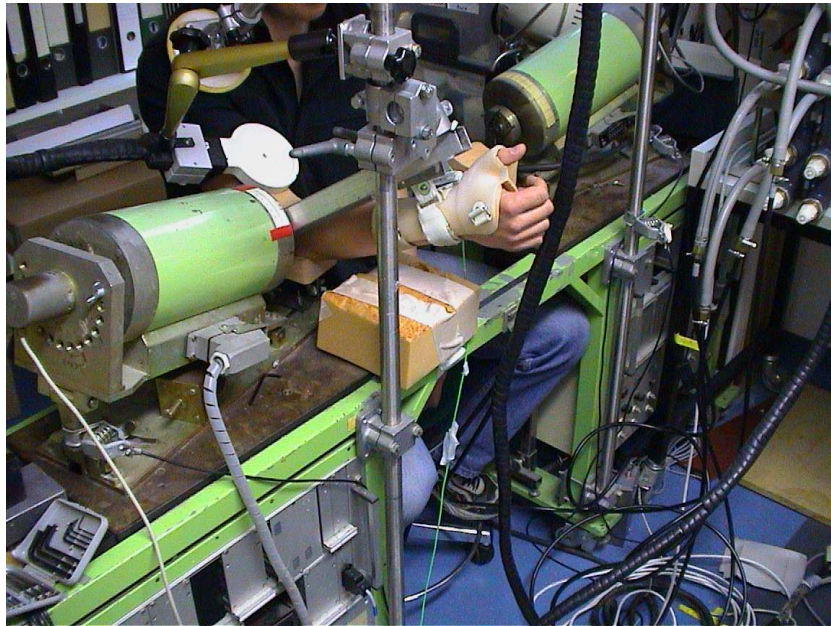


Abb. C.4: Einspannvorrichtung für den Arm mit Spule und Kraftsensor

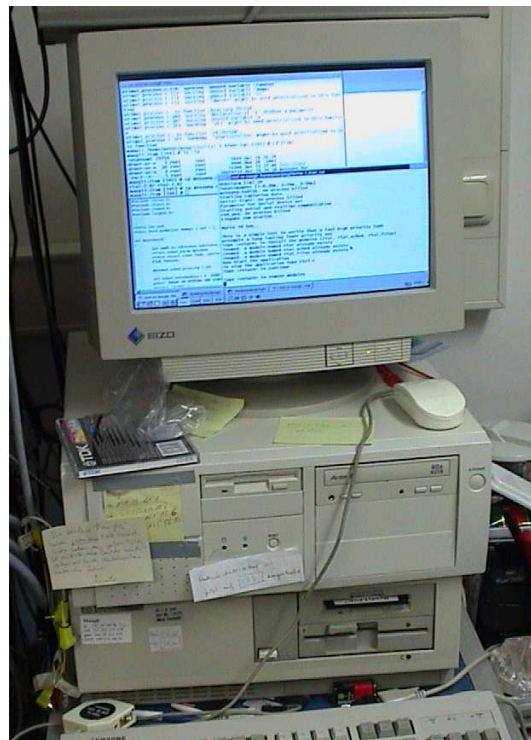


Abb. C.5: Realtime-Linux-System

Bezeichnungen

Variablen

a	muskelspezifische Konstante der Ruhe–Dehnungskurve
a_i	Polynomkoeffizienten Koeffizienten von $A(z)$
$A(z)$	gemeinsames Nennerpolynom
\mathbf{A}	Systemmatrix
α	reziproke Zeitkonstante der Aktivierungsdynamik
$\underline{\mathbf{A}}$	Vektor der Aktivierungsfunktionen
$\underline{\mathbf{A}}_V$	Vektor der verzögerten Aktivierungen
$\underline{\mathbf{A}}_{dyn}$	Vektor der dynamischen Aktivierungen
$\underline{\mathbf{A}}_{V\,dyn}$	Vektor der verzögerten dynamischen Aktivierungen
b	muskelspezifische Konstante der Ruhe–Dehnungskurve
b_i	Koeffizienten von $B(z)$
$B(z)$	Zählerpolynom der Eingangsübertragungsfunktion
B_e	Ellbogendämpfung
$\underline{\mathbf{b}}$	Einkopplungsvektor
β	Einspannwinkel des Oberarmes
c	Steigung einer linearen Transferfunktion
c_{pe}	muskelspezifische Konstante der Passiven Elastizität
$C(z)$	Zählerpolynom der Rauschübertragungsfunktion
$\underline{\mathbf{c}}$	Auskopplungsvektor
\mathbf{C}	Matrix der Cholesky–Zerlegung der Basisfunktionenmatrix
χ_i	Stützstellen der radialsymmetrischen Basisfunktionen
d	Durchgriff Dicke des Unterarmknochens
d_{pe}	muskelspezifische Konstante der Passiven Elastizität
D	Dämpfungsgrad
$D(z)$	Nennerpolynom der Rauschübertragungsfunktion
δ	exponentiell abklingendes Signal

e	Ausgangsfehler bzw. Beobachterfehler
e_e	erweiterter Fehler
$e_{\mathcal{NL}}$	Ausgangsfehler der Nichtlinearität
e_{post}	A-posteriori-Ausgangsfehler
e_z	zusätzlicher Fehler
\underline{e}_Z	Vektor der Beobachterfehler in den Zuständen
E	Gütefunktional
\mathbf{E}	Einheitsmatrix
f	Stimulationsfrequenz
$f(\cdot)$	allgemeine nichtlineare dynamische Funktion
f_h	Hebelarm
f_l	Ruhe-Dehnungskurve
f_{rek}	Rekrutierungskurve
f_v	Kraft-Geschwindigkeitskurve
F	Kraft
$F(z)$	Nennerpolynom der Eingangsübertragungsfunktion
	diskrete Übertragungsfunktion
$F(s)$	Übertragungsfunktion im Laplace-Bereich
	Aktivierungsdynamik
F_G	Gewichtskraft
F_{mess}	gemessene Kraft
F_{pe}	Passive Elastizität
F_{st}	Muskelkraft aufgrund der RPMS
F_{will}	willkürliche Muskelkraft
$\underline{\Phi}$	Parameterfehlervektor
$g(\cdot)$	allgemeine nichtlineare dynamische Funktion
g_0	Beharrungswert der allgemeinen Volterra-Reihe
$g[i_1, \dots, i_q]$	Volterra-Kern q -ten Grades der allgemeinen Volterra-Reihe
g_0	Beharrungswert der vereinfachten Volterra-Reihe
$g[i_1, \dots, i_q]$	Volterra-Kern q -ten Grades der vereinfachten Volterra-Reihe
\underline{g}_r	reduzierter Gewichtsfolgenvektor
$G(z)$	diskrete Übertragungsfunktion
h	Abtastzeit
$h[i]$	Elemente einer Impulsantwort
\underline{h}	Korrekturvektor
$H(s)$	Fehlerübertragungsfunktion im Laplace-Bereich
$H(z)$	Fehlerübertragungsfunktion im \mathcal{Z} -Bereich
$H_S(s)$	Übertragungsfunktion für Sichtbarkeit der Nichtlinearität
η	Lernschrittweite
$\eta_{\mathcal{NL}}$	Lernschrittweite für die Parameter der statischen Nichtlinearität

η_{lin}	Lernschrittweite für die Parameter der linearen Funktion
i	Laufindex
j	Laufindex
J	Massenträgheitsmoment
φ	Winkel
k	Laufindex aktueller Zeitpunkt Konstante
K	Anzahl der harmonischen Basisfunktionen
K_e	Ellbogensteifigkeit
\underline{k}	Einkopplungsvektor der Nichtlinearität
κ_{sat}	Kurvenkrümmung im Sättigungsbereich von f_{rek}
κ_{thr}	Kurvenkrümmung im Schwellbereich von f_{rek}
l	Laufindex
l_{biceps}	Muskellänge des Bizeps bei rechtem Winkel
l_m	Muskellänge des Bizeps
l_{ob}	Länge des Oberarmes
l_s	Abstand zwischen Oberarm und oberem Muskelansatz
l_{ua}	Länge des Unterarmes
$L_i(z)$	orthonormale Filter
\underline{l}	Vektor der Beobachterrückführkoeffizienten
λ	Vergessensfaktor
m	Antwortlänge
m_r	Basisfunktionenanzahl
m_{ua}	Masse des Unterarmes
M	Anzahl der Teilmodelle Moment
M^*	Sollmoment
M_A	antreibendes Moment
M_B	bremsendes Moment
M_{Be}	Moment aufgrund Ellbogendämpfung
M_c	Coulombscher Reibanteil
M_G	Gewichtsmoment
M_h	Haftreibungsmoment
M_{Ke}	Moment aufgrund Ellbogensteifigkeit
M_n	viskoser Reibanteil
M_P	periodisches Moment
M_R	Reibmoment

n	Systemordnung
N	Eingangsdimension
N_S	Losbrechdrehzahl
\mathcal{NL}	statische Nichtlinearität
\mathcal{NL}_{dyn}	dynamische Nichtlinearität
\mathcal{NL}_P	periodische Nichtlinearität
\mathcal{NL}_R	nichtlineare Reibcharakteristik
p	Parameteranzahl Laufindex
P	Anzahl der Trainingspaare
\mathbf{P}	Schätzwert der Kovarianzmatrix
q	Grad eines Polynoms Stützwerteanzahl eines GRNN
\mathbf{Q}_{obs}	Beobachtbarkeitsmatrix
$\underline{\theta}$	Vektor der optimalen Parameter
$\underline{\theta}_{\mathcal{NL}}$	optimaler Stützwertevektor des GRNN bei Funktionsapproximatoransatz
$\underline{\theta}_{lin}$	Vektor der optimalen linearen Parameter bei Funktionsapproximatoransatz
r	Anzahl der verkoppelten Hammerstein- und Wiener-Modelle
r_a	Abstand zwischen Ellbogengelenk und Muskelansatz
\underline{r}	Basisfunktion
$\tilde{\underline{r}}$	orthonormierte Basisfunktion
\mathbf{R}	Basisfunktionenmatrix
$\tilde{\mathbf{R}}$	orthonormierte Basisfunktionenmatrix
s	Laplace-Operator Summe der gewichteten Eingänge eines Perzeptrons
σ , σ_{norm}	Glättungsfaktor, normiert
t	Zeit
T	Zeitkonstante
T_a	Umrichterzeitkonstante
T_b	Umrichterzeitkonstante
T_V	Verzögerungszeitkonstante
\mathcal{T}	Transferfunktion eines Perzeptrons
u , \underline{u}	Systemeingang bzw. Vektor der Systemeingänge
u_{sat}	maximale Pulsintensität von f_{rek}
u_{thr}	minimale Pulsintensität von f_{rek}

v	weißes Rauschsignal Zwischengröße beim Hammerstein- und Wiener-Modell
v_m	zeitliche Änderung der Muskellänge des Bizeps
V	Lyapunov-Funktion Verstärkungsfaktor
\underline{w}	Parametervektor des SRNN
ω	Winkelgeschwindigkeit
\underline{x}	Zustandsvektor
$\underline{x}_{\mathcal{NL}}$	Vektor der Zustandsgrößen von denen die Nichtlinearität abhängt
\mathbf{X}	Regressionsmatrix
ξ	Winkel am oberen Muskelansatz
y, \underline{y}	Systemausgang bzw. Vektor der Systemausgänge
$y_{\mathcal{NL}}$	Ausgangssignal der Nichtlinearität
y_u	unverrauschter Systemausgang
Ψ	linearer Parameter des SRNN
z	Rauschsignal Operator der \mathcal{Z} -Transformation: $z = u + jv = e^{sT}$
ζ	Formfaktor für Basisfunktionen

Indizes

A	gerade Anteile des HANN
B	ungerade Anteile des HANN
dyn	dynamisch
l	Ruhe-Dehnungskurve
lin	linear
\mathcal{NL}	nichtlinear
pe	Passive Elastizität
rek	Rekrutierungskurve
st	aufgrund der Magnetstimulation
v	Kraft-Geschwindigkeitskurve

Symbole

$\hat{}$	geschätzte Größe z.B. $\hat{\theta}$
$\tilde{}$	virtuelle Größe z.B. $\tilde{\theta}$
$\tilde{}$	orthonormierte Größe z.B. \tilde{r}
$\dot{}$	erste zeitliche Ableitung z.B. $\dot{\varphi}$
$\ddot{}$	zweite zeitliche Ableitung z.B. $\ddot{\varphi}$
$\frac{\partial y}{\partial x}$	partielle Ableitung von y nach x z.B. $\frac{\partial y}{\partial \theta}$
*	Sollwert z.B. M^*
$\mathcal{L}\{\dots\}$	Laplace-Transformation
$\mathcal{Z}\{\dots\}$	\mathcal{Z} -Transformation
$\mathcal{L}^{-1}\{\dots\}$	Laplace-Rücktransformation
$\mathcal{Z}^{-1}\{\dots\}$	\mathcal{Z} -Rücktransformation

Abkürzungen

ARMAX	AutoRegressive Moving Average with eXogenous input
ARX	AutoRegressive with eXogenous input
FIR	Finite Impulse Response
GRNN	General-Regression-Neural-Network
HANN	Harmonic-Activation-Neural-Network
LMS	Least-Mean-Squares-Algorithmus
LOLIMOT	LOcal-LInear-MOdel-Tree
LS	Least-Squares-Algorithmus
MIMO	Multiple Inputs – Multiple Outputs
MISO	Multiple Inputs – Single Output
MLP	Multi-Layer-Perceptron
NARX	Nonlinear AutoRegressive with eXogenous input
NFIR	Nonlinear Finite Impulse Response
NLS	Nonlinear-Least-Squares-Algorithmus
NOBF	Nonlinear Orthonormal Basis Function
NOE	Nonlinear Output Error
OBF	Orthonormal Basis Function
OE	Output Error
PRN	partiell rekurrentes Netz
RBF	Radial-Basis-Funktion
RLS	rekursiver Least-Squares-Algorithmus
RPMS	repetitive periphere Magnetstimulation
SISO	Single Input – Single Output
SPR	streng positiv reell
SRNN	strukturiertes rekurrentes Neuronales Netz
VRN	voll rekurrentes Netz

Literaturverzeichnis

- [1] ANGERER, B.: *Algorithmus zur Stützwertanhebung von 2D-GRNN-Netzen*. Interner Bericht, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2001.
- [2] ANGERER, B.: *Online Identifikation mechatronischer Systeme mit Hilfe rekurrenter Netze*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2000.
- [3] ARMSTRONG-HÉLOUVRY, B., P. DUPONT und C. CANUDAS DE WIT: *A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction*. Automatica, Vol. 30, No. 7, pp. 1083–1138, 1994.
- [4] AYOUBI, M.: *Nonlinear System Identification Based on Neural Networks with Locally Distributed Dynamics and Application to Technical Processes*. Dissertation, TU Darmstadt, 1996.
- [5] BECKER, T. T.: *Methoden kleinster Fehlerquadrate zur parametrischen Identifikation dynamischer Systeme*. VDI Verlag, Düsseldorf, 1989.
- [6] BEUSCHEL, M.: *Neuronale Netze zur Diagnose und Tilgung von Drehmoment-schwingungen am Verbrennungsmotor*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2000.
- [7] BORIC, I.: *Regelstreckenanalyse und Konzepte zur nichtlinearen adaptiven Regelung von Ellbogenbewegungen mittels repetitiver peripherer Magnetstimulation*. Diplomarbeit, Lehrstuhl für Messsystem- und Sensortechnik, TU München, 2001.
- [8] BRYCHCY, T.: *Modellierung dynamischer Systeme mit vorstrukturierten Neuronalen Netzen*. Dissertation, Institut für Informatik, TU München, 2000.
- [9] DOYLE, F. J., R. K. PEARSON und B. A. OGUNNAIKE: *Identification and Control Using Volterra Models*. Springer-Verlag, 2002.
- [10] FELGEN, L.: *Identifikation der biomechanischen Eigenschaften des Muskel-Sehne-Komplexes am Oberarm*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2002.

- [11] FRENZ, T.: *Stabile Neuronale Online Identifikation und Kompensation statischer Nichtlinearitäten am Beispiel von Werkzeugvorschubantrieben*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 1998.
- [12] FROSCHAMMER, F.: *Identifikation und Kompensation von Sensorungenauigkeiten für die drehzahlgestützte On-Board-Diagnose von Verbrennungsmotoren*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2002.
- [13] HAVEL, P. M.: *Geregelte Induktion von Reich- und Greifbewegungen am Menschen mittels repetitiver peripherer Magnetstimulation*. Dissertation, Lehrstuhl für Messsystem- und Sensortechnik, TU München, 2002.
- [14] HEUBERGER, P. S., P. M. VAN DEN HOF und O. H. BOSGRA: *A Generalized Orthonormal Basis for Linear Dynamical Systems*. IEEE Transactions on Automatic Control, Vol. 40, pp. 451–465, 1995.
- [15] HINTZ, C., M. RAU und D. SCHRÖDER: *Combined Identification of Parameters and Nonlinear Characteristics based on Input-Output Data*. Proceedings of 6th International Workshop on Advanced Motion Control AMC 2000, pp. 175–180, Nagoya, Japan, 2000.
- [16] HINTZ, C.: *Identifikation nichtlinearer mechatronischer Systeme mit strukturierten rekurrenten Netzen*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2003.
- [17] HOFFMANN, N.: *Neuronale Netze – Kleines Handbuch*. Vieweg Verlag, Braunschweig, 1993.
- [18] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *Identification and Observation of Mechatronic Systems with unknown Nonlinear Dynamic Functions*. Proceedings of the 10th International Power Electronics and Motion Control Conference EPE-PEMC, Cavtat & Dubrovnik, Croatia, 2002.
- [19] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *Identification and Observation of Mechatronic Systems including Multidimensional Nonlinear Dynamic Functions*. Proceedings of the 7th International Workshop on Advanced Motion Control AMC 2002, pp. 285–290, Maribor, Slovenia, 2002.
- [20] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *A new Approach for the Identification and Observation of Complex Mechatronic Systems*. Proceedings of the 2nd IFAC Conference on Mechatronic Systems, Berkeley, USA, 2002.
- [21] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *Identification of Nonlinear Dynamic MISO Systems on Fundamental Basics of the Volterra Theory*. Proceedings of 23th International Conference on Power Electronics, Intelligent Motion and Power Quality PCIM 2002, pp. 231–236, Nuremberg, Germany, 2002.

- [22] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *Identification of Nonlinear Dynamic MISO Systems with Orthonormal Base Function Models*. IEEE International Symposium on Industrial Electronics ISIE 2002, pp. 337–342, L’Aquila, Italy, 2002.
- [23] HOFMANN, M.: *Identifikation von linearen Parametern und nichtlinearen Kennlinien einer Drosselklappe und Entwurf von Regelungskonzepten*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2001.
- [24] ISERMANN, R.: *Identifikation dynamischer Systeme – Band 1*. Springer-Verlag, 1992.
- [25] ISERMANN, R.: *Identifikation dynamischer Systeme – Band 2*. Springer-Verlag, 1992.
- [26] JUNGE, T. F.: *„On-line“-Identifikation und lernende Regelung nichtlinearer Regelstrecken mittels neuronaler Netze*. VDI Verlag, Düsseldorf, 1999.
- [27] KILLICH, A.: *Prozeßidentifikation durch Gewichtsfolgeschätzung*. Dissertation, VDI Verlag, Düsseldorf, 1991.
- [28] KNOHL, T.: *Anwendung künstlicher neuronaler Netze zur nichtlinearen adaptiven Regelung*. VDI Verlag, Düsseldorf, 2001.
- [29] KORTMANN, M.: *Die Identifikation nichtlinearer Ein- und Mehrgrößensysteme auf der Basis nichtlinearer Modellansätze*. VDI Verlag, Düsseldorf, 1989.
- [30] KURTH, J.: *Identifikation nichtlinearer Systeme mit komprimierten Volterra-Reihen*. Dissertation, VDI Verlag, Düsseldorf, 1995.
- [31] LANG, M.: *Signaldarstellung*. Vorlesungsskriptum Lehrstuhl für Mensch-Maschine-Kommunikation der TU München, 1999.
- [32] LENZ, U.: *Lernfähige neuronale Beobachter für eine Klasse nichtlinearer dynamischer Systeme und ihre Anwendung zur intelligenten Regelung von Verbrennungsmotoren*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 1998.
- [33] LJUNG, L. und T. SÖDERSTRÖM: *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, Massachusetts, 1987.
- [34] LUDYK, G.: *Theoretische Regelungstechnik 2*. Springer-Verlag, 1995.
- [35] MCCULLOCH, W. S. und W. PITTS: *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics 5, pp. 115–133, 1943.
- [36] MURRAY-SMITH, R.: *A Local Linear Model Network Approach to Nonlinear Modelling*. Ph.D. Thesis, University of Strathclyde, UK, 1994.

- [37] MURRAY-SMITH, R. und T. A. JOHANSEN: *Multiple Model Approaches to Modelling and Control*. Taylor & Francis, 1997.
- [38] NARENDRA, K. S. und A. M. ANNASWAMY: *Stable Adaptive Systems*. Prentice Hall, 1989.
- [39] NELLES, O., O. HECKER und R. ISERMANN: *Automatische Strukturselektion für Fuzzy-Modelle zur Identifikation nichtlinearer, dynamischer Prozesse*. at – Automatisierungstechnik, Nr. 6, pp. 302–312, 1998.
- [40] NELLES, O.: *Nonlinear System Identification with Local Linear Neuro-Fuzzy Models*. Dissertation, TU Darmstadt, 1999.
- [41] NELLES, O.: *Nonlinear System Identification*. Springer-Verlag, 2001.
- [42] NOCEDAL, J. und S. J. WRIGHT: *Numerical Optimization*. Springer-Verlag, 1999.
- [43] PREUSS, H. P.: *Methoden der nichtlinearen Modellierung — vom Interpolationspolynom zum Neuronalen Netz*. at – Automatisierungstechnik, Nr. 10, pp. 449–457, 1994.
- [44] RAHM, J.: *Identifikation nichtlinearer dynamischer Systeme unter Verwendung der Volterra-Reihe mit Hilfe orthonormaler Basisfunktionen*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2000.
- [45] REITER, T.: *Prädiktive Regelung für nichtlineare dynamische Systeme mit Hammerstein- bzw. Wiener-Modellen*. Bachelorarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2002.
- [46] REKLAITIS, G. V., A. RAVINDRAN und K. M. RAGSDALL: *Engineering Optimization – Methods and Applications*. John Wiley & Sons, 1983.
- [47] SCHÄFFNER, C.: *Analyse und Synthese neuronaler Regelungsverfahren*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 1996.
- [48] SCHETZEN, M.: *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley & Sons, 1980.
- [49] SCHMIDT, G.: *Regelungs- und Steuerungstechnik 1*. Vorlesungsskriptum Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, 1996.
- [50] SCHMIDT, G.: *Regelungs- und Steuerungstechnik 2*. Vorlesungsskriptum Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, 1996.
- [51] SCHMIDT, G.: *Optimierungsverfahren in der Automatisierungstechnik*. Vorlesungsskriptum Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, 1999.

- [52] SCHRÖDER, D.: *Intelligent Observer and Control Design for Nonlinear Systems*. Springer-Verlag, 2000.
- [53] SCHRÖDER, D.: *Elektrische Antriebe – Regelung von Antriebssystemen*. Springer-Verlag, 2001.
- [54] SEYDEL, R.: *Identifikation dynamischer Nichtlinearitäten mit Hilfe strukturierter rekurrenter Netze*. Bachelorarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2002.
- [55] SOEDERSTROEM, T. und P. STOICA: *System Identification*. University Press Cambridge, 1989.
- [56] SPECHT, D.: *A General Regression Neural Network*. IEEE Transactions on Neural Networks, Vol. 2, pp. 568–576, 1991.
- [57] STRAUB, S.: *Entwurf und Validierung neuronaler Beobachter zur Regelung nichtlinearer dynamischer Systeme im Umfeld antriebstechnischer Problemstellungen*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 1998.
- [58] STROBL, D.: *Identifikation nichtlinearer mechatronischer Systeme mittels neuronaler Beobachter*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 1999.
- [59] TREICHL, T., S. HOFMANN und D. SCHRÖDER: *Identification of Nonlinear Dynamic Systems with Multiple Inputs and Single Output using discrete-time Volterra Type Equations*. 15th International Symposium on Mathematical Theory of Networks and Systems MTNS 2002, Notre Dame, USA, 2002.
- [60] TREICHL, T.: *Identifikation nichtlinearer dynamischer MISO Systeme auf Grundlage der Volterra-Funktionalpotenzreihe*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2001.
- [61] UNBEHAUEN, H. und G. P. RAO: *Identification of Continuous Systems*. North-Holland Systems and Control Series, Vol. 10, 1987.
- [62] UNBEHAUEN, H.: *Regelungstechnik I*. Vieweg Verlag, 1992.
- [63] UNBEHAUEN, H.: *Regelungstechnik II*. Vieweg Verlag, 1993.
- [64] UNBEHAUEN, H.: *Regelungstechnik III*. Vieweg Verlag, 1995.
- [65] VACHENAUER, P. und K. MEYBERG: *Höhere Mathematik 1*. Springer-Verlag, 1995.
- [66] VOLTERRA, V.: *Theory of Functionals*. Blackie and Sons, 1930.

- [67] WAHLBERG, B.: *System Identification using Laguerre Models*. IEEE Transactions on Automatic Control, Vol. 36, pp. 551–562, 1991.
- [68] WAHLBERG, B.: *System Identification using Kautz Models*. IEEE Transactions on Automatic Control, Vol. 39, pp. 1276–1282, 1994.
- [69] WALTER, W.: *Identifikation nichtlinearer dynamischer Systeme mit Hilfe des Wiener-Modells*. Bachelorarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2002.
- [70] WELLERS, M. und N. KOSITZA: *Identifikation nichtlinearer Systeme mit Wiener- und Hammerstein-Modellansätzen auf Basis der Volterra-Reihe*. at – Automatisierungstechnik, Nr. 5, pp. 209–216, 1999.
- [71] WILLIAMS, R. J. und D. ZIPSER: *Gradient-Based Learning Algorithms for Recurrent Connectionist Networks*. Tech. Rep. NU-CSS90-9, College of Computer Science, Northeastern University, Boston, MA, 1990.
- [72] ZELL, A.: *Simulation Neuronaler Netze*. Addison–Wesley, 1994.