

Dierk Schröder



# Intelligente Verfahren

Identifikation und Regelung  
nichtlinearer Systeme

II

# Intelligente Verfahren

Dierk Schröder

# Intelligente Verfahren

Identifikation und Regelung nichtlinearer  
Systeme



Prof. Dr.-Ing. Dr.-Ing h.c. Schröder  
Technische Universität München  
Lehrstuhl für Elektrische Antriebssysteme und Leistungselektronik  
Arcisstr. 21  
80333 München  
Deutschland  
[dierk.schroeder@tum.de](mailto:dierk.schroeder@tum.de)

ISBN 978-3-642-11397-0      e-ISBN 978-3-642-11398-7  
DOI 10.1007/978-3-642-11398-7  
Springer Heidelberg Dordrecht London New York

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

© Springer-Verlag Berlin Heidelberg 2010

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zu widerhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

*Einbandentwurf:* WMXDesign GmbH

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media ([www.springer.com](http://www.springer.com))

## **Empfehlung**

Das Buch „Intelligente Verfahren“ führt exemplarisch in das Gebiet der computationel intelligence ein. Hervorzuheben ist die sehr gute Strukturierung der Themen, die aufeinander aufbauend zur Lösung für zunehmend komplexere Aufgabenstellungen einsetzbar sind. Es ist dem Autor in beeindruckender und hervorragender Art und Weise gelungen, das Buch sehr nah am Puls der Zeit in der Forschung und gleichzeitig didaktisch ideal zu gestalten. Wesentlich sind auch die ausführlich durchgerechneten Beispiele zu den Aufgabenstellungen, die von Leserinnen und Lesern nachgerechnet werden können und somit eine deutliche Verbesserung der Verständlichkeit der Verfahren erzielen. Das Buch ist damit sowohl für die Ausbildung im wissenschaftlichen Bereich als auch für den Anwender hervorragend geeignet.

**Professor Dr.-Ing./Univ.Tokyo M. Buss**  
**Technische Universität München**  
**Lehrstuhl für Steuerungs- und Regelungstechnik**

# Vorwort

In dem vorliegenden Beitrag werden ausgewählte Verfahren der künstlichen Intelligenz anwendungsnah vorgestellt. Die Ziele sind

- die Identifikation unbekannter nichtlinearer Systeme
- die Modellbildung dieser nichtlinearen Systeme sowie
- die Regelung derartiger Systeme.

Wesentlich ist, dass zu Beginn methodische Ansätze gewählt werden, die die ingenieurtechnischen Aspekte — wie Vorkenntnisse der Struktur des nichtlinearen Systems und damit die physikalische Interpretierbarkeit — berücksichtigen. Die Voraussetzung der Vorkenntnisse wird von Kapitel zu Kapitel fortschreitend reduziert bis hin zur Regelung von unbekannten zeitvarianten nichtlinearen Systemen, die außerdem unbekannt gestört sind. Wichtig ist weiterhin, dass die erarbeiteten Methoden ausführlich und anschaulich an praktischen Beispielen vorgestellt werden, um ihre Vor- und Nachteile zu erkennen.

Es hat sich inzwischen gezeigt, dass die vorgestellten Methoden generell, d.h. von technischen Fragestellungen ausgehend — wie sie in der Mechatronik, bei technologischen Verfahren sowie der Adaptronik auftreten — zusätzlich auch in der Medizintechnik und im Bankensektor verwendet werden können.

Das Buch „Intelligente Verfahren“ ist Teil einer fünfbandigen Buchreihe. Das erste Buch erläutert ausführlich die Grundlagen der elektrischen Antriebe, d.h. beispielsweise der Aktorik in mechatronischen Systemen. Im zweiten Buch werden die Regelverfahren für alle Arten der elektrischen Antriebe dargestellt, um einen Einblick in die realisierbaren dynamischen Eigenschaften und die Regelgenauigkeit zu erhalten. Zur Abrundung dieser Fragestellung werden zusätzlich die mechatronischen Regelungen eines elastischen Mehrmassensystems, verschiedene Verfahren der Schwingungsdämpfung, Simulationsverfahren sowie die Regelung kontinuierlicher Produktionsanlagen vorgestellt. Die obigen Darstellungen werden im dritten Buch „Leistungselektronische Stellglieder“ und im vierten Buch „Leistungshalbleiter“ vertieft, da die Stellglieder und deren Leistungshalbleiter mit ihrem dynamischen Verhalten das erreichbare dynamische Verhalten des Strom- bzw. Momentenregelkreises bestimmen. Das nun vorliegende fünfte Buch ergänzt diese Buchserie regelungstechnisch um das Gebiet Identifikation, Modellbildung und Regelung unbekannter nichtlinearer Systeme.

Bedanken möchte ich mich bei allen wissenschaftlichen Mitarbeitern des Lehrstuhls, die mit mir gemeinsam mit großer Begeisterung und hoher Motivation das Gebiet der Intelligenten Verfahren erforscht haben.

Dies gilt insbesondere für die Herren Dr. Endisch und Dr. Feiler, die mir bei der endgültigen Fassung dieses wissenschaftlichen Beitrags außerordentlich geholfen haben.

Dank gebührt auch Herrn Dipl.-Ing.(FH) Ebert für die Korrekturen aller kleinen Tippfehler und der Erstellung des Layouts.

Großer Dank gebührt auch meiner Frau, die mich stets unterstützt hat.

D. Schröder

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	1
<b>2</b>	<b>Einführung in mechatronische Systeme</b>	9
2.1	Zweimassensystem . . . . .	9
2.2	Regelung der Arbeitsmaschinendrehzahl . . . . .	12
2.3	Regelung der Antriebsmaschinendrehzahl . . . . .	13
2.4	Proportionale Zustandsregelung . . . . .	18
2.5	Integrale Zustandsregelung . . . . .	20
2.6	Nichtlineares Zweimassensystem . . . . .	22
2.7	Dreimassensystem . . . . .	25
2.8	Zusammenfassung: Mechatronische Systeme . . . . .	30
2.9	Kontinuierliche Produktionsanlagen . . . . .	32
2.10	Zusammenfassung: Technologische Systeme . . . . .	32
<b>3</b>	<b>Statische Funktionsapproximatoren</b>	37
3.1	Übersicht: Neuronale Netze . . . . .	38
3.2	Statische nichtlineare Funktionen . . . . .	39
3.3	Methoden der Funktionsapproximation . . . . .	40
3.4	Kriterien zur Beurteilung künstlicher neuronaler Netze . . . . .	42
3.5	Funktionsapproximation mit lokalen Basisfunktionen . . . . .	43
3.6	Radial Basis Function (RBF) Netz . . . . .	45
3.7	General-Regression-Neural-Network (GRNN) . . . . .	48
3.7.1	Ursprüngliche Anwendung des GRNN . . . . .	49
3.7.2	Optimierung bei mehrdimensionalem Eingangsraum . . . . .	49
3.7.3	Beispiele . . . . .	51
3.8	Harmonisch Aktiviertes Neuronales Netz (HANN) . . . . .	58
3.8.1	Grundstruktur . . . . .	58
3.8.2	Erweiterung . . . . .	60
3.9	LOLIMOT — LOcal LInear MOdel Tree . . . . .	62
3.9.1	Grundlegende Idee . . . . .	62
3.9.2	Parameter- und Strukturoptimierung . . . . .	63

3.9.2.1	Parameterberechnung . . . . .	64
3.9.2.2	Strukturoptimierung . . . . .	66
3.9.3	Beispiele . . . . .	68
3.10	Multi-Layer-Perceptron (MLP) Netze . . . . .	77
3.10.1	Einleitung . . . . .	77
3.10.2	Technische Abstraktion . . . . .	77
3.10.3	Transferfunktionen . . . . .	78
3.10.4	Mehrschichtiges MLP-Netz . . . . .	79
3.10.5	Auslegung von feedforward Netzen . . . . .	81
3.10.6	Beispiele . . . . .	82
3.11	Bewertung und Vergleich der Funktionsapproximatoren . . . . .	87
3.11.1	Bewertung der Eigenschaften des GRNN und RBF-Netzwerks . .	87
3.11.2	Bewertung HANN . . . . .	88
3.11.3	Bewertung LOLIMOT . . . . .	88
3.11.4	Bewertung MLP Netz . . . . .	88
3.11.5	Einsatzbereich der Netztypen . . . . .	89
<b>4</b>	<b>Lernen bei statischer Funktionsapproximation: Grundlagen</b>	<b>91</b>
4.1	Gradientenabstiegsverfahren . . . . .	94
4.1.1	Lerngesetz für das RBF und GRNN-Netz . . . . .	96
4.1.1.1	Gradientenverfahren mit Momentum Term . . . . .	97
4.1.1.2	Stabilität nach Lyapunov . . . . .	97
4.1.1.3	Parameterkonvergenz . . . . .	98
4.1.1.4	Fehlermodell 1 für das Gradientenabstiegsverfahren . . . . .	99
4.1.2	Lerngesetz für das HANN . . . . .	99
4.1.3	Lerngesetz für mehrschichtige Netze . . . . .	102
4.1.3.1	Herleitung der Backpropagation-Regel . . . . .	102
4.1.3.2	Zusammenfassung des BP-Algorithmus . . . . .	107
4.1.3.3	Gradientenverfahren bei mehrschichtigen Netzen . . . . .	108
4.1.3.4	Gradientenverfahren mit Momentum Term . . . . .	109
4.1.3.5	Beispiele . . . . .	109
4.1.4	Probleme beim einfachen Gradientenabstieg . . . . .	115
4.1.4.1	Lokale Minima der Fehlerfläche . . . . .	116
4.1.4.2	Flache Plateaus . . . . .	117
4.1.4.3	Oszillationen in steilen Schluchten . . . . .	118
4.1.4.4	Verlassen guter Minima . . . . .	118
4.2	Lerngesetz: Least-Squares-Verfahren . . . . .	119
4.2.1	Nichtrekursiver Least-Squares-Algorithmus (LS) . . . . .	119
4.2.2	Rekursiver Least-Squares-Algorithmus (RLS) . . . . .	120
4.2.3	Weighted-Least-Squares-Algorithmus (WLS) . . . . .	123
4.2.4	Anwendung des Least-Squares-Algorithmus zur Parameteroptimierung bei RBF-Netzen . . . . .	124
4.2.5	Bewertung des Least-Squares-Algorithmus . . . . .	125

<b>5</b>	<b>Lernfähiger Beobachter</b>	127
5.1	Strecken mit isolierter Nichtlinearität . . . . .	127
5.2	Beobachterentwurf bei messbarem Eingangsraum . . . . .	129
5.2.1	Voraussetzungen . . . . .	129
5.2.2	Beobachterentwurf zur Identifikation der Nichtlinearität . . . . .	130
5.2.3	Parameterkonvergenz . . . . .	137
5.3	Beobachterentwurf bei nicht messbarem Eingangsraum . . . . .	138
5.3.1	Zusätzliche Voraussetzung . . . . .	138
5.3.2	Beobachteransatz analog Luenberger . . . . .	138
5.4	Zweimassensystem mit Reibung . . . . .	141
5.5	Identifikation mehrerer Nichtlinearitäten . . . . .	147
5.6	Ergänzung: Fehlermodelle . . . . .	150
5.6.1	Fehlermodell 1 . . . . .	150
5.6.2	Fehlermodell 2 . . . . .	152
5.6.3	Fehlermodell 3 . . . . .	152
5.6.4	Fehlermodell 4 . . . . .	154
5.7	Anwendung auf einen Vorschubantrieb . . . . .	156
5.7.1	Modellbildung . . . . .	157
5.7.2	Identifikation der Reibungskennlinie . . . . .	160
5.7.3	Kompensation . . . . .	165
5.8	Identifikation von Hysterese . . . . .	171
5.8.1	Modellierung der Hysterese . . . . .	171
5.8.2	Physikalisch motiviertes Modell der Hysterese . . . . .	172
5.8.3	Parametrierung . . . . .	173
5.8.4	Verallgemeinertes Hysteresemodell . . . . .	174
5.8.5	Der allgemeingültige Signalflußplan . . . . .	176
5.8.6	Identifikation von Hysterese . . . . .	177
5.9	Zusammenfassung und Bewertung . . . . .	179
<b>6</b>	<b>Identifikation nichtlinearer Systeme mit vorstrukturierten rekurrenten Netzen</b>	181
6.1	Strukturierte rekurrente Netze . . . . .	182
6.1.1	Anwendung der Transformation . . . . .	183
6.1.2	Parameteradaption . . . . .	184
6.1.3	Zustandsdarstellung . . . . .	186
6.2	Erweiterung zum Luenberger-Beobachter . . . . .	192
6.2.1	Partielle Ableitungen . . . . .	193
6.2.2	Implementierung der statischen Neuronalen Netze . . . . .	196
6.2.3	Anwendung der Beobachterstruktur . . . . .	197
6.2.4	Durchführung der Identifikation . . . . .	198
6.3	Beurteilung des Identifikationsverfahrens . . . . .	203
6.4	Anwendungsbeispiel . . . . .	205
6.4.1	Losemodellierung und Approximation . . . . .	205
6.4.2	Approximation der Reibungskennlinie . . . . .	208

6.4.3	Identifikation des losebehafteten Zweimassensystems . . . . .	209
6.4.4	Identifikation . . . . .	213
<b>7</b>	<b>Identifikation linearer dynamischer Systeme</b>	217
7.1	Grundlagen der Identifikation . . . . .	217
7.1.1	Parametrische und nichtparametrische Identifikationsverfahren .	217
7.1.2	Identifikation . . . . .	218
7.2	Lineare dynamische Modellstrukturen . . . . .	220
7.2.1	Modelle mit Ausgangsrückkopplung . . . . .	222
7.2.1.1	Autoregressive with Exogenous Input Model . . . . .	222
7.2.1.2	Output Error Model . . . . .	225
7.2.2	Modelle ohne Ausgangsrückkopplung . . . . .	227
7.2.2.1	Finite Impulse Response Model . . . . .	228
7.2.2.2	Orthonormal Basis Function Model . . . . .	230
7.3	Identifikationsbeispiele . . . . .	234
7.3.1	ARX–Modell . . . . .	234
7.3.2	OE–Modell . . . . .	236
7.3.3	FIR–Modell . . . . .	238
7.3.4	OBF–Modell . . . . .	240
7.4	Zusammenfassung . . . . .	242
<b>8</b>	<b>Identifikation nichtlinearer dynamischer Systeme</b>	245
8.1	Klassifikation nichtlinearer dynamischer Systeme . . . . .	247
8.1.1	Nichtlineare Zustandsdarstellung . . . . .	247
8.1.2	Blockorientierte nichtlineare Modelle . . . . .	247
8.1.3	Allgemeine nichtlineare Systembeschreibung . . . . .	248
8.2	Verfahren zur Identifikation nichtlinearer dynamischer Systeme	249
8.2.1	Nichtlineare Modelle mit Ausgangsrückkopplung . . . . .	253
8.2.1.1	Time Delay Neural Network . . . . .	255
8.2.1.2	Local Linear Model Tree . . . . .	256
8.2.2	Nichtlineare Modelle ohne Ausgangsrückkopplung . . . . .	260
8.2.2.1	Volterra–Funktionalpotenzreihe . . . . .	260
8.2.2.2	Hammerstein–Modell und Wiener–Modell im Ansatz der Volterra–Funktionalpotenzreihe . . . . .	262
8.2.2.3	Eigenschaften der Volterra–Funktionalpotenzreihe . . . . .	265
8.2.2.4	Volterra–Funktionalpotenzreihe mit Basisfunktionen . . . . .	266
8.2.2.5	Allgemeiner Ansatz für Wiener– und Hammerstein–Modelle .	270
8.2.2.6	Erweiterung des Identifikationsansatzes . . . . .	271
8.2.2.7	Rekonstruktion der blockorientierten Modellstruktur . . . . .	272
8.2.3	Anregungssignale zur Identifikation . . . . .	276
8.3	Zusammenfassung . . . . .	279

<b>9</b>	<b>Beobachterentwurf bei dynamischen Nichtlinearitäten</b>	281
9.1	Systeme mit dynamischen Nichtlinearitäten . . . . .	283
9.2	Beobachterentwurf . . . . .	286
9.2.1	Beobachterentwurf bei messbarem Eingangsraum der dynamischen Nichtlinearität . . . . .	287
9.2.2	Beobachterentwurf bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität . . . . .	291
9.3	Identifikation von global integrierenden Systemen . . . . .	294
9.4	Simulationsbeispiel für Beobachterentwurf . . . . .	300
9.5	Identifikation eines mechatronischen Antriebssystems . . . . .	307
9.5.1	Identifikation in der Simulationsumgebung . . . . .	309
9.5.2	Validierung am realen System . . . . .	314
9.6	Zusammenfassung . . . . .	317
<b>10</b>	<b>Nichtlineare Optimierung in der Systemidentifikation</b>	319
10.1	Optimierungsverfahren 0. Ordnung . . . . .	321
10.1.1	Die Simplex-Methode . . . . .	321
10.1.2	Das Hooke-Jeeves-Tastverfahren . . . . .	324
10.2	Verfahren zur Liniensuche . . . . .	326
10.2.1	Ein klassisches Liniensuchverfahren mit Intervallsuchphase und Intervallverkleinerungsphase . . . . .	328
10.2.1.1	Die Intervallsuchphase . . . . .	329
10.2.1.2	Die Intervallverkleinerungsphase . . . . .	330
10.2.2	Adaptives Liniensuchverfahren mit Lagrange-Interpolation . . . . .	337
10.3	Optimierungsverfahren 1. Ordnung . . . . .	345
10.3.1	Gradientenabstieg mit Momentumterm . . . . .	349
10.3.2	Gradientenabstieg mit variabler Lernschrittweite . . . . .	351
10.4	Optimierungsverfahren 2. Ordnung . . . . .	353
10.4.1	Das Nichtlineare Konjugierte Gradientenverfahren . . . . .	354
10.4.2	Das Skalierte Konjugierte Gradientenverfahren . . . . .	358
10.4.3	Das Newton-Verfahren . . . . .	365
10.4.3.1	Konvergenz des Newton-Verfahrens . . . . .	366
10.4.3.2	Hessematrixberechnung beim Newton-Verfahren . . . . .	367
10.4.4	Die Quasi-Newton-Verfahren . . . . .	368
10.4.4.1	Die Quasi-Newton-Bedingung . . . . .	369
10.4.4.2	Die Aufdatierungsformel von Broyden . . . . .	371
10.4.4.3	Die DFP-Aufdatierungsformel . . . . .	371
10.4.4.4	Die BFGS-Aufdatierungsformel . . . . .	372
10.4.5	Levenberg-Marquardt-Algorithmus . . . . .	377
10.5	Zusammenfassung: Deterministische Optimierungsverfahren . . . . .	380
10.5.1	Konvergenz der Parameter . . . . .	381
10.5.2	Rechen- und Speicheraufwand . . . . .	382
10.5.3	Aufwand bei der Implementierung . . . . .	382

10.5.4	Ergebnisse des Optimierungsbeispiels . . . . .	382
10.6	Identifikationsbeispiele . . . . .	383
10.6.1	Identifikation einer statischen Reibkennlinie . . . . .	383
10.6.2	Identifikation von stark verrauschten Messdaten . . . . .	384
<b>11</b>	<b>Stochastische Optimierungsverfahren</b>	<b>387</b>
11.1	Simulated Annealing . . . . .	387
11.2	Evolutionsstrategien . . . . .	392
11.3	Particle Swarm Optimization . . . . .	400
11.4	Stochastische Optimierungsverfahren bei der Systemidentifikation mit Neuronalen Netzen . . . . .	406
<b>12</b>	<b>Verfahren zur Regelung nichtlinearer Systeme</b>	<b>409</b>
12.1	Relativgrad und Ordnung . . . . .	409
12.2	Nulldynamik . . . . .	411
12.2.1	Nulldynamik bei linearen Systemen . . . . .	412
12.2.2	Auswirkung von Nullstellen auf die Impulsantwort . . . . .	413
12.2.3	Auswirkung von Nullstellen auf den geschlossenen Regelkreis . . . . .	415
12.2.4	Unterdrückung von Eingangssignalen durch Nullstellen . . . . .	417
12.2.5	Nulldynamik im nichtlinearen System . . . . .	420
12.2.6	Analogie zwischen Kompensation der Nulldynamik und Pol-Nullstellen-Kürzung . . . . .	421
12.2.7	Zusammenfassung . . . . .	426
12.3	Nichtlineare Regelungsnormalform . . . . .	426
12.3.1	Beispiel zur NRNF . . . . .	430
12.4	Exakte Ein-/Ausgangslinearisierung . . . . .	432
12.4.1	Beispiel zur exakten Ein-Ausgangslinearisierung . . . . .	434
12.4.2	Beispiel zur exakten Ein-Ausgangslinearisierung mit Reglerentwurf . . . . .	439
12.5	Regelung auf ein Referenzsignal (Tracking) . . . . .	448
12.5.1	Beispiel zur Regelung auf ein Referenzsignal . . . . .	449
12.5.2	Wahl des Referenzsystems . . . . .	460
12.6	Der Einsatz von neuronalen Beobachtern . . . . .	467
12.6.1	Anwendung auf das nichtlineare System 2. Ordnung . . . . .	467
12.6.2	Simulationsergebnis . . . . .	468
12.6.3	Anwendung auf Regelung mit NRNF . . . . .	469
12.6.4	Simulationsergebnisse . . . . .	471
12.6.5	Kurzzusammenfassung: NRNF und lernfähiger Beobachter . . . . .	472
12.7	Ein-Ausgangslinearisierung zur neuronalen Regelung . . . . .	473
12.7.1	Erlernen der Input-Output Linearisierung mit Neuronalen Netzen . . . . .	474
12.7.2	Regelung einer nichtlinearen Strecke zweiter Ordnung . . . . .	476
12.8	Stabile referenzmodellbasierte Neuroregelung (SRNR) . . . . .	479
12.8.1	Parameteradaption . . . . .	483

<b>Inhaltsverzeichnis</b>	<b>XV</b>
12.8.2 Stellgrößen-Beschränkung . . . . .	484
12.8.3 Aufteilung in Teilfunktionen . . . . .	485
<b>13 Modellbasierte Adaptive Regelung</b>	<b>487</b>
13.1 ARMA-Modell als Prädiktionsmodell . . . . .	490
13.2 Systemidentifikation . . . . .	498
13.2.1 Projektionsalgorithmus . . . . .	501
13.2.2 Rekursiver Least-Squares-Algorithmus (RLS) . . . . .	512
13.3 Entwurf des adaptiven Regelkreises . . . . .	517
13.3.1 Inverser Regler mit integrierter Systemidentifikation . . . . .	518
13.3.2 Stabilitätsuntersuchung des geschlossenen Regelkreises . . . . .	521
13.4 Anwendung des adaptiven Reglers auf ein reales Zwei-Massen-System . . . . .	525
<b>14 Disturbance Rejection</b>	<b>533</b>
14.1 Linear Disturbance Rejection . . . . .	534
14.1.1 Deterministic Disturbances . . . . .	534
14.1.2 Stochastic Disturbances . . . . .	538
14.1.2.1 A Qualitative Analysis . . . . .	539
14.1.2.2 Stochastic Adaptive Control . . . . .	544
14.2 Nonlinear Disturbance Rejection . . . . .	549
14.3 Time-Varying Disturbances . . . . .	554
14.3.1 Multi-Model Adaptive Control . . . . .	555
14.3.1.1 General Methodology . . . . .	556
14.3.1.2 Models . . . . .	558
14.3.1.3 Switching and Tuning . . . . .	558
14.3.1.4 Control . . . . .	559
14.3.1.5 Benefits . . . . .	560
14.3.2 Proof of Stability . . . . .	562
14.3.2.1 Case (i): All adaptive models . . . . .	563
14.3.2.2 Case (ii): One adaptive model and one fixed model . . . . .	564
14.3.2.3 Case (iii): (N-2) fixed models and 2 adaptive models . . . . .	565
14.4 Mathematical background . . . . .	566
14.4.1 Nonlinear Differential Equations . . . . .	566
14.4.2 Concepts from Analysis . . . . .	566
14.4.3 Existence and uniqueness . . . . .	573
14.4.4 Lyapunov's direct method . . . . .	577
14.4.5 LTI Systems and Lyapunov Stability . . . . .	583
14.4.6 Barbalat's Lemma . . . . .	585
<b>15 Lernende Automaten</b>	<b>589</b>
15.1 Einleitung . . . . .	589
15.2 Mathematische Grundlagen . . . . .	590

15.2.1	Stochastische Prozesse . . . . .	590
15.2.2	Markov-Ketten . . . . .	591
15.2.3	Konvergenzbegriffe . . . . .	597
15.3	Automaten . . . . .	598
15.3.1	Automat und Umgebung . . . . .	599
15.3.2	Nützlichkeit und Optimalität . . . . .	602
15.3.3	Stochastische Automaten veränderlicher Struktur . . . . .	604
15.3.3.1	Ein ergodisches Lerngesetz: <i>Linear Reward-Penalty</i> . . . . .	606
15.3.3.2	Ein absolut nützliches Lerngesetz: <i>Linear Reward-Inaction</i> . . . . .	610
15.3.3.3	Ein Kompromiss: Der $L_{R-\varepsilon P}$ -Algorithmus . . . . .	614
15.3.4	Ein deterministischer Automat mit fester Struktur . . . . .	616
15.4	Prognose stochastischer Parameterwechsel . . . . .	620
15.4.1	Regelung mit multiplen Modellen . . . . .	621
15.4.2	Stochastische Parameterwechsel . . . . .	623
15.4.3	Erweiterte Regelungsstruktur . . . . .	624
15.4.4	Quantifizierung der erreichten Regelgüte . . . . .	626
15.4.5	Simulationsbeispiel . . . . .	627
<b>16</b>	<b>Hochverstärkungsbasierte Regelung</b>	629
16.1	Grundidee der hochverstärkungsbasierten Regelung . . . . .	630
16.2	Auswirkung großer Verstärkungen im Regelkreis . . . . .	632
16.3	Empfindlichkeit gegenüber hohen Relativgraden . . . . .	637
16.4	Funnel-Control . . . . .	640
16.5	Hochverstärkungsbasierte Regelung mit zeitvarianter, nicht-monotoner Verstärkung . . . . .	648
16.6	Anwendung am Beispiel Einmassensystem . . . . .	649
16.7	Internes Modell für die Realisierung einer stationär genauen Regelung . . . . .	649
16.8	Allgemeines zur Regelung von Zweimassensystemen mit Funnel-Control . . . . .	656
16.9	Funnel-Regelung für das lineare Zweimassensystem . . . . .	657
16.9.1	Antriebsdrehzahl als Regelgröße . . . . .	658
16.9.2	Arbeitsmaschinendrehzahl als Regelgröße . . . . .	661
16.9.3	Zustandsregler mit Funnel . . . . .	662
16.10	Funnel-Regelung für das nichtlineare Zweimassensystem . . . . .	676
16.11	Ergebnisse mit Filter und Integralanteil . . . . .	693
<b>17</b>	<b>Funnel-Control: Implementierung, Erweiterung und Anwendung</b>	697
17.1	Funnel-Control (FC) . . . . .	699
17.1.1	Trichterentwurf: Trichterfunktion und Trichterrand . . . . .	702
17.1.2	Referenz- bzw. Sollwertsignale . . . . .	708
17.1.3	Systemklasse $\mathcal{S}$ . . . . .	710

17.1.3.1	LTI SISO Systeme der Klasse $\mathcal{S}$ . . . . .	713
17.1.3.2	Beispielsysteme . . . . .	717
17.1.4	Regelziel . . . . .	723
17.2	Kundenanforderungen . . . . .	723
17.3	Skalierung der Reglerverstärkung . . . . .	727
17.4	Minimale zukünftige Distanz (MD) . . . . .	731
17.4.1	Analytischer Ansatz (aMD) . . . . .	734
17.4.2	Numerischer Ansatz (nMD) . . . . .	735
17.4.3	Differenzierender Ansatz (dMD) . . . . .	736
17.4.4	Simulationsbeispiele . . . . .	739
17.5	Error Reference Control (ERC) . . . . .	741
17.6	Anwendung . . . . .	746
17.6.1	Nichtlineares Zwei-Massen-System (2MS) . . . . .	746
17.6.2	Aktive Dämpfung durch statische Zustandsrückführung . . . . .	748
17.6.3	Erweiterung des 2MS für Zugehörigkeit in $\mathcal{S}$ . . . . .	751
17.6.4	Messung und Bewertung der Ergebnisse . . . . .	755
<b>18</b>	<b>Einführung in die Fuzzy–Regelung</b>	761
18.1	Grundlagen der Theorie der unscharfen Mengen . . . . .	762
18.1.1	Definition der unscharfen Menge . . . . .	762
18.1.2	Weitere Definitionen . . . . .	763
18.1.3	Grundlegende Mengenoperationen für unscharfe Mengen . . . . .	767
18.1.4	Modifikatoren . . . . .	771
18.2	Grundlagen der unscharfen Logik . . . . .	773
18.2.1	Einführung . . . . .	773
18.2.2	Grundbegriffe der unscharfen Logik . . . . .	775
18.2.3	Fuzzyfizierung und logisches Schließen . . . . .	776
18.2.4	Logische Operatoren . . . . .	778
18.3	Grundlagen der Fuzzy–Regelung . . . . .	793
18.3.1	Fuzzyfizierung . . . . .	795
18.3.2	Inferenz . . . . .	797
18.3.3	Defuzzyfizierung . . . . .	799
18.4	Anhang: Die „theoretische“ Darstellungsweise der unscharfen Logik	804
18.4.1	Grundlagen des plausiblen Schließens . . . . .	806
18.4.2	Implikationsoperatoren . . . . .	810
18.4.3	Berücksichtigung von Verbundaussagen und mehreren Regeln .	811
18.4.4	Berücksichtigung zusätzlicher Unsicherheiten . . . . .	811
18.4.5	Die „anwendungsorientierte“ unscharfe Logik als Spezialfall des plausiblen Schließens . . . . .	814
<b>Literaturverzeichnis</b>		817
<b>Stichwortverzeichnis</b>		831

# 1 Einführung

In dem vorliegenden Buch *Intelligente Verfahren – Systemidentifikation und Regelung nichtlinearer Systeme* soll ausgehend von der linearen Regelungstheorie das weite Gebiet der nichtlinearen Systeme, deren Identifikation, d.h. damit einhergehend die nichtlinearen Modellbildungen der betrachteten nichtlinearen Systeme sowie deren Regelungen dargestellt werden. Die vollständige Darstellung dieses umfangreichen Gebiets ist im Rahmen eines einzigen Buchs nicht möglich, da bereits Einzelaspekte außerordentlich vielfältig sind. Aufgrund dieser Schwierigkeit erfolgt eine Auswahl der behandelten Themen, der den systemtechnischen Aspekt Vorwissen berücksichtigt. Die Berücksichtigungen des systemtechnischen Aspekts Vorwissen bedeutet, daß ein Grundwissen über das zu untersuchende bzw. zu regelnde System vorhanden ist. Dieses Grundwissen kann beispielsweise bedeuten, daß die systemtechnische Struktur zum Teil oder vollständig bekannt ist. Eine weitere Vereinfachung der Identifikation bzw. Modellbildung besteht, wenn ein Teil der Parameter ebenso bekannt ist. Von weiterer Bedeutung ist, ob interne Zustände des zu untersuchenden Systems zugänglich, d.h. meßbar, sind. Diese Meßbarkeit von internen Zuständen bedeutet eine Chance zur Unterteilung des zu untersuchenden Systems und damit zu einer im allgemeinen vereinfachten Identifikation.

Um den Zugang zu dem komplexen Gebiet der intelligenten Strategien zu vereinfachen und die Notwendigkeit der Berücksichtigung der nichtlinearen Effekte in Strecken allgemein zu begründen, wird im zweiten Kapitel exemplarisch die Regelung eines linearen elastischen Zweimassensystems diskutiert. Ein Ergebnis dieser Diskussion ist, daß die bisher noch überwiegend eingesetzte Kaskadenregelung nur unter der Voraussetzung eines sehr starren Systems akzeptable Regelergebnisse erbringt. Wenn dies nicht gegeben ist, dann muß entweder die Regeldynamik sehr reduziert werden, oder es kann nur die Motordrehzahl geregelt werden. Wesentlich günstigere Regelergebnisse lassen sich mit der Zustandsregelung erreichen; allerdings ist diese Art der Regelung nur bei linearen Strecken und Parameterkenntnis anwendbar. Wenn typische Nichtlinearitäten wie Reibung oder Lose in der Strecke vorhanden sind, dann ist die lineare Zustandsregelung nicht mehr voll zufriedenstellend einsetzbar. Dies bedeutet, daß bereits in diesem einfachsten Fall eines typischen nichtlinearen Systems einschneidende Einschränkungen zu akzeptieren sind, wenn nur die lineare Regelungstheorie eingesetzt wird. Die Betrachtung wird um die Technologie „kontinuierliche Pro-

duktionsanlagen“ wie bei der Herstellung von Papier, Folien oder Stahl erweitert, und es wird als typische Nichtlinearität der Elastizitätsmodul als Technologie-Parameter eingeführt. Es besteht somit die Notwendigkeit, die Nichtlinearitäten in Strecken zu berücksichtigen, diese zu identifizieren und bei der Reglerauslegung zu bedenken.

Ausgehend von diesen Ergebnissen werden im dritten Kapitel verschiedene statische Funktionsapproximatoren wie die neuronalen Netze RBF, GRNN, HANN bzw. MLP vorgestellt und an Beispielen gegenübergestellt. Die Entscheidung, neuronale Netze als statische Funktionsapproximatoren generell einzusetzen, ist damit begründet, dass die physikalische Modellbildung beispielsweise der Reibung durch komplexe Abhängigkeiten wie der Oberflächenrauhigkeit der reibenden Körper, der Temperatur und damit der Viskosität sowie der Dicke des Schmiermittels und der Drehzahl bzw. Geschwindigkeit der beiden Körper zueinander gekennzeichnet ist und diese Einflüsse auch noch zeitvariant berücksichtigt werden müssen. Neuronale Netze können dies durch on-line-Lernen zeitvariant erfassen.

Im vierten Kapitel werden die grundlegenden Lerngesetze besprochen. Vorge stellt werden das Gradientenabstiegs-Verfahren, der Backpropagation-Algorithmus und die Least-Squares-Verfahren. Diese Verfahren können sowohl zur Identifikation von Nichtlinearitäten alleine als auch von Nichtlinearitäten in dynamischen Systemen eingesetzt werden. Bei der Identifikation von Nichtlinearitäten in dynamischen Systemen sind allerdings die Fehlermodelle zu beachten, und es ist zu bedenken, dass die Least-Squares-Verfahren nur dann eingesetzt werden dürfen, wenn der Modellausgang der Strecke linear in den Parametern ist. Die letzte Einschränkung muss bei den beiden anderen Verfahren nicht beachtet werden. Beispiele ergänzen die theoretischen Abhandlungen. Mächtigere Lernverfahren werden in den beiden Kapiteln 10 und 11 abgehandelt.

Im fünften Kapitel wird der neuronale Beobachter vorgestellt. Um einen relativ einfachen Zugang zu diesem Gebiet zu erhalten, soll deshalb zu Beginn angenommen werden, daß die Vorkenntnisse des zu untersuchenden Systems sehr groß sind, d.h. daß beispielsweise sowohl die systemtechnische Struktur als auch die Parameter des linearen Teils bekannt sein sollen und die Nichtlinearität(en) anregenden Zustände des Systems zugänglich sind. In diesem Fall sind dann „nur noch“ die unbekannten Nichtlinearitäten zu identifizieren, um das gewünschte nichtlineare Streckenmodell zu erhalten. Bereits diese Aufgabenstellung ist relativ komplex, denn zur Identifizierung der Nichtlinearität(en) muß (müssen) diese entweder im Ausgangssignal oder falls zugänglich in einem der zugänglichen internen Signale sichtbar sein. Wesentlich ist außerdem, daß erstens der Lernvorgang stabil und zweitens konvergent ist.

Eine weitere wesentliche Rolle bei der Identifikation hat die Fehler-Übertragungsfunktion, die vier unterschiedliche Fehlermodelle zur Folge hat, um die Stabilität der Identifikation sicherzustellen. Im weiteren Verlauf der Darstellungen werden die Anforderungen an die Vorkenntnisse bzw. die Verfügbarkeit der anregenden Zustände der Nichtlinearität verringert. Abschließend wird in diesen

Kapiteln die erfolgreiche Identifikation von Reibung und deren Kompensation detailliert am Beispiel einer Werkzeugmaschine dargestellt.

Im sechsten Kapitel wird die Voraussetzung „Kenntnis der linearen Parameter des Systems“ aus dem vorherigen Kapiteln fallengelassen. Es ist somit nur noch die Struktur des unbekannten nichtlinearen Systems bekannt, und es müssen nun sowohl die linearen Systemparameter als auch die Nichtlinearität identifiziert werden. Um diese Aufgabenstellung auch für on-line Anwendungen zu lösen, werden die Strukturkenntnisse in ein strukturiertes rekurrentes neuronales Netz übertragen und diese neue rekurrente Struktur in eine Beobachterstruktur eingebunden. Ausgehend von einfachen Beispielen wird zuletzt ein nichtlineares elastisches Zwei-Massen-System überzeugend identifiziert, wobei die Nichtlinearitäten jeweils Reibungen bei dem Zwei-Massen-System und eine Lose sind. Die Stoßeffekte am Ende des Losevorgangs werden ebenso erfolgreich identifiziert.

Zur Vorbereitung des achten Kapitels, in dem die Identifikation „dynamischer Nichtlinearitäten“ entwickelt wird, erfolgt im siebten Kapitel eine grundlegende Einführung in die Identifikation linearer Systeme. Es werden die Modelle ARX, OE, FIR und OBF vorgestellt, Gleichungsfehler- und Ausgangsfehler-Modell sowie ihre Empfindlichkeit gegenüber Rauschen beschrieben. Außerdem wird der Zusammenhang dargestellt, wie in Abhängigkeit vom verwendeten Modell die unbekannten Parameter in den Modellausgang eingehen und wie in Abhängigkeit dieses Effekts das Optimierungsverfahren zur Identifikation zu wählen ist. Abschließend werden die theoretisch erarbeiteten Aussagen durch Beispiele validiert. Damit ist die Grundlage für das achte Kapitel geschaffen.

Im achten Kapitel wird angenommen, dass es in einem System Teilmodelle gibt, welche bekannt und andere Teilmodelle, die unbekannt sind. Für diese unbekannten Teilmodelle gibt es außer dem Ein- und Ausgangssignal keine weiteren Informationen, d.h. es kann nur das Ein- zu Ausgangsverhalten identifiziert werden. Um für diese unbekannten Teilmodelle einen grundlegenden Ansatz zu schaffen, wird angenommen, dass diese Teilmodelle statische Nichtlinearitäten und lineare dynamische Strukturen enthalten und somit „dynamische Nichtlinearitäten“ sind. Bekannte „dynamische Nichtlinearitäten“ sind die Hammerstein und Wiener Modelle sowie ihre Kombinationen. Um derartige Systeme abzubilden, gibt es Netze mit externer Dynamik wie NOE, NARX, NFIR und NOBF-Modelle oder Netze mit interner Dynamik, dies sind beispielsweise die rekurrenten neuronalen Netze; alle Strukturen werden detailliert vorgestellt. Ein weiterer Modellsatz ist die Volterra-Funktionalpotenzreihe, die allerdings eine hohe Parameterzahl erfordert. Aufgrund der hohen Parameterzahl wird der Volterra-Ansatz um den OBF-Ansatz erweitert, der zu einer deutlichen Reduzierung der Parameteranzahl führt. Beide Ansätze werden auf das Hammerstein, das Wiener-Modell und ihre Kombinationen angewendet. Damit liegen verschiedene Strukturen und Verfahren zur Identifizierung dynamischer Nichtlinearitäten prinzipiell vor.

Nachdem im siebten Kapitel die Identifikations-Verfahren für lineare Systeme und im achten Kapitel für nichtlineare Systeme prinzipiell vorgestellt wurden, stellt sich die Frage, ob die Struktur und Verfahren des achten Kapitels direkt

genutzt werden könnten. Wie schon in Kapitel fünf und sechs werden in Kapitel neun die in Kapitel acht dargestellten SISO- und gekoppelten MISO-Strukturen der unbekannten Teilstrecke „dynamische Nichtlinearitäten“ in eine Beobachterstruktur integriert. Die vorgestellten Verfahren sind sowohl bei meßbarem als auch nicht meßbarem Eingangsraum geeignet. Der Abschluss sind wiederum zwei prägnante Beispiele.

Das zehnte Kapitel geht ausführlich auf die Parameteroptimierung ein. Während das vierte Kapitel mit dem Gradientenabstieg lediglich die Grundlagen der nichtlinearen Optimierung behandelt, folgt nun im zehnten Kapitel ein umfangreicher Überblick zu den deterministischen Parameteroptimierungsverfahren. Neben dem bereits bekannten Gradientenabstiegsverfahren kommen vor allem die leistungsfähigen Optimierungsverfahren 2. Ordnung zum Einsatz, zu denen die Methoden der Konjugierten Gradienten und die Newton-Verfahren zählen. Damit die Algorithmen stabil arbeiten, behandelt das Kapitel die beiden Ansätze der Skalierung und der Liniensuche. Eine Skalierung gebrauchen die beiden Verfahren Skalierter Konjugierter Gradientenabstieg und Levenberg-Marquardt. Eine Liniensuche verwenden die Nichtlinearen Konjugierten Gradienten-Algorithmen und die Quasi-Newton-Algorithmen. Neben einem klassischen Liniensuchalgorithmus beschreibt das Kapitel auch einen adaptiven Suchweitenalgorithmus, der eine Lagrange-Interpolation nutzt, sich selbst auf die Gegebenheiten der Fehlerfläche einstellt und so ohne anwendungsbezogene Daten auskommt. Die leistungsfähigen Optimierungsverfahren 2. Ordnung sind besonders bei der Parameteroptimierung von Neuronalen Netzen von essentieller Bedeutung, da der einfache Gradientenabstieg derartige Optimierungsprobleme meist nicht lösen kann. Das zehnte Kapitel enthält viele Beispiele zum Nachrechnen und Simulieren. Diese ermöglichen einen anschaulichen Einstieg in die Thematik.

Bei den deterministischen Optimierungsverfahren des zehnten Kapitels besteht das Problem, dass die Optimierung vorzeitig in einem schlechten lokalen Minimum enden kann. Interessant ist deshalb zu untersuchen, welche Möglichkeiten es gibt, eines der globalen Minima des Optimierungsproblems zu finden. Aus diesem Grund wird im Kapitel elf die Gruppe der stochastischen Optimierungsverfahren untersucht. Die stochastischen Optimierungsverfahren setzen gezielt Zufallsgrößen ein, um global die beste Lösung zu finden. Im Vergleich zu den in Kapitel zehn beschriebenen lokalen Optimierungsverfahren sind die Prozesse bei den stochastischen Verfahren nicht reproduzierbar. Stochastische Optimierungsverfahren sind wegen der einfachen Mathematik leicht zu implementieren. Aufgrund der vielen Kostenfunktionsauswertungen ist die Konvergenzgeschwindigkeit jedoch meist niedrig. Ziel des elften Kapitels ist, die grundlegende Arbeitsweise stochastischer Verfahren zu verstehen, um die Grenzen dieser Algorithmen bei der Systemidentifikation mit Neuronalen Netzen herausarbeiten zu können. Die im Kapitel elf beschriebenen Algorithmen orientieren sich alle an Phänomenen der Natur: Die Grundidee beim Simulated Annealing ist der Abkühlungsprozess von geschmolzenen Festkörpern. Bei den Evolutionären Algorithmen handelt es sich um mathematische Modelle, welche die Evolution des Lebens simulieren.

Unter Swarm Intelligence versteht man Verfahren, die das Schwarmverhalten von Tieren nachahmen. Stellvertretend für diese Gattung wird das Particle Swarm Optimization Verfahren vorgestellt.

In den Kapiteln drei bis elf wurden detailliert verschiedene Identifikationsstrukturen und -Verfahren dargestellt, und es wird nun vorausgesetzt, die Identifikation war erfolgreich, da die Modellparameter zu den realen Streckenparametern konvergiert sind. Es folgt in Kapitel zwölf eine sehr ausführliche Darstellung der Verfahren zur Regelung nichtlinearer Strecken. Da dieser Bereich außerordentlich komplex ist, werden zuerst an linearen Strukturen die Kompensation der systemeigenen Dynamik, die Auswirkungen von Nullstellen wie beispielsweise auf die Impulsantwort, auf die Stabilität beim geschlossenen Regelkreis oder die Unterdrückung des Eingangssignals sowie die Nulldynamik bei Relativgraden ungleich der Ordnung des Nennerpolynoms vorgestellt. Es folgen zwei theoretisch ausgerichtete Kapitel, in denen die Transformation in die nichtlineare Regelungsnormalform NRNF (Byrnes-Isidori-Normalform, BINF) und die exakte Eingangs-Ausgangslinearisierung erläutert werden. Um das Verständnis für diese beiden wichtigen theoretischen Verfahren zu erhöhen, werden anschließend sehr ausführlich mehrere Beispiele durchgerechnet. Zuerst wird die exakte Einzu Ausgangslinearisierung anhand eines Beispiels mit isolierter Nichtlinearität erarbeitet. Es folgt ein zweites Beispiel, bei dem die Nichtlinearität nicht isoliert ist. Nach der Linearisierung erfolgt die Erarbeitung der Regelstruktur. Anschließend wird an Beispielen die Nulldynamik vorgestellt und die Anforderungen an die Nulldynamik beim Reglerentwurf abgeleitet. Ein weiteres Beispiel zeigt ausführlich durchgerechnet den Reglerentwurf bei der Ein- Ausgangslinearisierung mit Nulldynamik sowie die Entwurfsschritte bei der asymptotischen Modellfolgeregelung. Um gleichzeitig Aussagen über die Wirksamkeit dieser Methoden zu erhalten, wird durchgängig ein Beispiel für die verschiedenen Methoden simuliert und mit einem linearen Vergleichsregelkreis verglichen. Die Entwurfsmethoden für das durchgängige Beispiel werden nicht mehr so ausführlich dokumentiert, um dem Leser die Möglichkeit zu geben, die Ergebnisse selbst zu überprüfen.

Bisher wurde angenommen, dass alle Systemzustände einer Messung zugänglich sind. Es besteht die Frage, ob bei nichtlinearen Systemen die on-line lernfähigen Beobachter eingesetzt werden können, um die nicht messbaren Zustände für die Regelung zu erhalten. An einem Beispiel mit der NRNF-Regelung wird erfolgreich die Funktion des on-line lernfähigen Beobachters nachgewiesen. Es müssen allerdings gewisse Einschränkungen wie die Differenzierbarkeit der Nichtlinearität gegeben sein.

Es besteht nun die weitergehende Frage, ob eine Trennung von Identifikation und Reglerentwurf überhaupt notwendig ist. Im Kapitel „Erlernen der Input-Output Linearisierung mit Neuronalen Netzen“ wird nachgewiesen, dass die Parameter  $\alpha$  und  $\beta$  der E-A-Linearisierung on-line während des geregelten Betriebs gelernt werden können und der Istwert den Sollwert entsprechend einem vorgegebenen Modell folgt. Dieser Ansatz wird im nachfolgenden Kapitel „Modellbasierte Adaptive Regelung“ weiterverfolgt.

Das Kapitel 13 befasst sich mit der modellbasierten adaptiven Regelung. Hierbei ist das Ziel, ein unbekanntes System ohne vorherigen Identifikationslauf sofort regeln zu können. Dazu findet parallel zum Regelvorgang die Identifikation der durch das Sollsignal angeregten Systemdynamik statt; somit liegt stets ausreichend Systeminformation vor, um das Regelziel durch Anpassung der Reglerparameter zu erreichen. Der Grundgedanke dieses adaptiven Konzeptes besagt, nur so viel Systeminformation zu sammeln bzw. zu identifizieren, wie momentan für das Erreichen der Solltrajektorie bzw. des Regelziels notwendig ist.

Der Vorteil dieses Konzeptes besteht darin, das System nicht erst in Identifikationsläufen vollständig identifizieren zu müssen, bevor ein konventioneller fest eingestellter Regler stabil ausgelegt werden kann. Der adaptive Regler kann ohne Wissen über das System dieses regeln, da der Regler in kurzen Lernvorgängen immer nur das momentan notwendige noch nicht bekannte Systemwissen identifiziert. Zudem ist beim konventionellem seriellen Vorgehen von Identifikation und Regelung ungewiss, ob die wahren Systemparameter nach einer bestimmten Zeit überhaupt gefunden werden, so dass die Gefahr eines schlechten bzw. instabilen Regelvorgangs besteht. Beim adaptiven Regler ermöglicht hingegen die ständige aktive Identifikation eine Adaption des Reglers, so dass stets ein gutes Regelergebnis gewährt wird, selbst bei Parameteränderungen im System.

Es ist jedoch zu bedenken, dass durch das parallele Vorgehen von Identifikation und Regelung ein zeitvariantes System entsteht, wodurch die Kombination aus einer stabilen Identifikation und einem stabilen Regler nicht zwingend in einen stabilen adaptiven Regelvorgang resultiert. Die im geschlossenen Regelkreis entstehende Dynamik auf Grund der Zeitvarianz muss daher mit nichtlinearen Stabilitätsmethoden untersucht werden, damit garantiert werden kann, dass alle Systemsignale bei der gewählten Kombination aus Regler und Identifikation für alle Zeiten beschränkt bleiben. In Kapitel dreizehn wird zunächst ein intuitives Verständnis für einen modellbasierten adaptiven Regler gegeben, bevor am Beispiel eines adaptiven Referenzmodell-Reglers dessen Eigenschaft sowie Stabilität bewiesen wird. Hierzu dient sowohl ein Lyapunov-Stabilitätsbeweis als auch ein Widerspruchsbeweis.

Ausgangspunkt für Kapitel 14 ist die Fragestellung, wie Systeme mit unbekannten Parametern adaptiv geregelt werden können, wenn sie gleichzeitig Störgrößen unterworfen sind. Es werden Verfahren vorgestellt, die den Einfluss der Störgrößen auf den Adaptionsvorgang unterdrücken, so dass dieser wie im ungestörten Fall ablaufen kann und die Stabilisierung des geschlossenen Regelkreises herbeiführt. Grundidee ist dabei, das Systemmodell um ein Störmodell zu erweitern, das linear oder nichtlinear entworfen werden kann. Dies führt zu einer Erhöhung der Systemordnung (order augmentation) und schafft die strukturelle Voraussetzung für die Konvergenz der Parameter. Es wird gezeigt, wie deterministische, stochastiche und sprungförmige Störgrößen unterdrückt werden können. Im letzten Fall liegt eine stochastiche Störung vor, deren statistische Eigenschaften unbekannt sind. Das Verfahren der adaptiven Multi-Modellregelung, das am Ende des Kapitels vorgestellt wird, macht auch solche Störungen beherrschbar.

Kapitel 15 zeigt, wie komplexe Lernaufgaben in diskrete Schritte zerlegt und dadurch vereinfacht werden können. Als Beispiel wird das Erlernen einer sich zufällig und sprunghaft ändernden Störgrösse betrachtet. Solche Störungen sind mit den in Kapitel 14 eingeführten multiplen Modellen zwar beherrschbar, es bleibt jedoch ein Regelfehler, da ein Sprung erst erkannt werden muss. Mit Hilfe sogenannter lernender Automaten können wahrscheinliche Sprünge vorhergesagt werden und so der verbleibende Fehler reduziert werden. Wesentlich für das Verfahren ist, dass alle Entwurfsschritte im Wahrscheinlichkeitsraum (nicht im Zustandsraum) erfolgen. Die dazu notwendigen Werkzeuge werden zu Beginn des Kapitels 15 eingeführt.

Das sechzehnte Kapitel behandelt die Zustandsregelung von nichtlinearen Strecken, deren Parameter unbekannt sind. Am Beispiel eines reibungsbehafteten — und damit nichtlinearen — Zweimassensystems wird ein zeitvarianter Zustandsregler entworfen, der gänzlich ohne Identifikation der Streckenparameter auskommt. Weder die linearen Parameter (Massenträgheitsmomente, Federhärte, Dämpfung) noch die nichtlinearen Parameter (Reibkennlinie) werden in irgendeiner Weise identifiziert oder gelernt. Selbes gilt für unbekannte Störgrößen wie Lastmomente, Messrauschen oder unmodellierte Verfälschung der Stellgröße durch den Aktor. Es kommt ein hochverstärkungsbasierter Regler zum Einsatz, der den Aufwand für die Implementierung aufwändiger und komplexer Lerngesetze vermeidet und dadurch eine sehr einfache Struktur erhalten kann. Neben dem Entwurf und der theoretischen Analyse eines solchen Reglers wird gezeigt, dass eine vorgegebene Fehlertoleranzgrenze eingehalten werden kann und eine aktive Bedämpfung von Oszillationen in der Antriebswelle erfolgt.

Aufbauend auf den Erkenntnissen aus Kapitel 16 wird in Kapitel 17 das hochverstärkungsbasierte zeit-variante adaptive Regelkonzept ‘Funnel Control’ in größerem Detail beschrieben und sinnvolle Erweiterungen als auch mögliche Implementierungsvarianten vorgestellt. Es werden Maßnahmen diskutiert, die z.B. eine Bedämpfung des transienten Verhaltens durch Skalierung der Reglerverstärkung (‘Dämpfungsskalierung’) bzw. durch ‘Führen’ entlang eines Wunschfehlerverlaufs (‘Error Reference Control’) ermöglichen oder z.B. einen effektiveren Nutzen der Stellgröße durch Einsatz der ‘zukünftigen Distanz’ erzielen. Alle Erweiterungen werden anschaulich und ausführlich an einfachen Beispielsystemen getestet, simuliert und besprochen. Abschließend wird das nichtlineare Zwei-Massen-System aufgegriffen. Das Modell wird um den Einfluss von (z.B. negativen) Übersetzungswöhnltnissen erweitert und eine vom Regler unabhängige Methode zur Bedämpfung des Verdrehwinkels der Antriebswelle vorgestellt. Das Kapitel endet mit Messergebnissen, die den erzielbaren Nutzen der vorgestellten Maßnahmen in der Anwendung veranschaulichen.

Abschließend erfolgt im Kapitel siebzehn eine Einführung in die Fuzzy-Logik und Fuzzy-Regelung. Diesem Ansatz wurde anfangs eine große Chance vorhergesagt, der sich allerdings nicht in dem Umfang realisiert hat [37]. Aufgrund dieser Situation wird nur eine Einführung gegeben. Es werden die Definitionen zu unscharfen Mengen, die Mengenoperationen, die Grundlagen und Grundbe-

griffe der unscharfen Logik, logische Operatoren und für ein Regelbeispiel die Fuzzyfizierung, Inferenz und Defuzzyfizierung anhand von Beispielen dargestellt. Im Anhang folgt eine heuristische Darstellung der unscharfen Logik.

## 2 Einführung in mechatronische Systeme

In diesem Kapitel wird zuerst die Modellbildung eines linearen, elastisch gekoppelten Zwei- und Dreimassensystems und deren Regelung dargestellt. Dies ist bereits sehr ausführlich im Buch „Elektrische Antriebe – Regelung von Antriebsystemen“ 3. Auflage, Springer–Verlag [207] erfolgt, so daß in diesem Buch der Text nur die wesentlichen Punkte aufzeigt.

### 2.1 Zweimassensystem

Die folgende Abbildung 2.1 zeigt eine typische, einfache Anordnung eines mechatronischen Antriebsstrangs, ein geregelter elektromagnetischer Energiewandler treibt über ein Getriebe und eine Welle eine Arbeitsmaschine an. Vom geregelten elektromagnetischen Energiewandler ist nur der Rotor (Trägheitsmoment  $J_1$ ) dargestellt. Es wird weiterhin angenommen, daß die Kupplungen zwischen der elektrischen Maschine und dem Getriebe sowie zwischen dem Getriebe und der Welle bzw. der Welle und Arbeitsmaschine losefrei sind, so daß nur die Getriebelose zu berücksichtigen ist. Außerdem sei nur bei der Arbeitsmaschine die Reibung relevant.

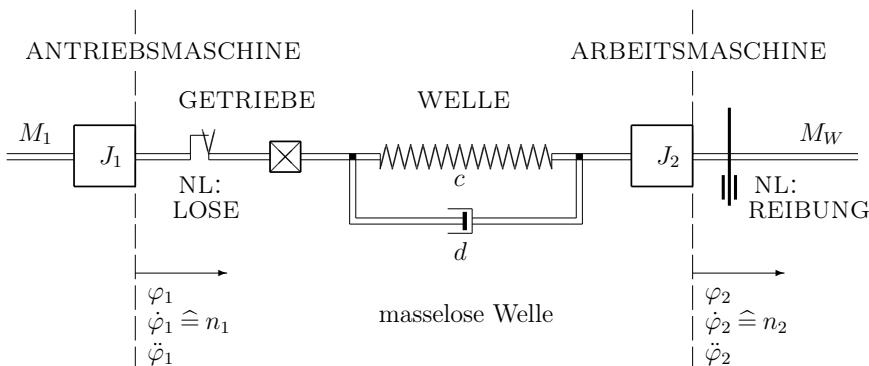


Abb. 2.1: Elastische Verbindung Antriebsmaschine – Arbeitsmaschine

In den folgenden Ableitungen wird angenommen, daß die Nichtlinearitäten Lose und Reibung nicht berücksichtigt werden. Es ergeben sich die folgenden Gleichungen:

$$\text{Beschleunigungsmoment der Masse} : M_B = J \cdot \ddot{\varphi} \quad (2.1)$$

$$\text{Übertragungsmoment der Feder} : M_C = c \cdot \Delta\varphi \quad (2.2)$$

$$\text{Übertragungsmoment durch Dämpfung} : M_D = d \cdot \Delta\dot{\varphi} \quad (2.3)$$

Darin bedeuten:	$\ddot{\varphi}$	Winkelbeschleunigung
	$\dot{\varphi}$	Winkelgeschwindigkeit
	$\varphi$	Drehwinkel
	$c$	Drehfedersteifigkeit
	$d$	mechanische Dämpfung
		Wellendaten

#### Antriebsmaschinenmasse:

$$\ddot{\varphi}_1 = \frac{1}{J_1} \cdot M_1 - \frac{1}{J_1} \cdot (M_C + M_D) \quad (2.4)$$

#### Welle:

$$\Delta\varphi = \varphi_1 - \varphi_2 \quad (2.5)$$

$$\Delta\dot{\varphi} = \dot{\varphi}_1 - \dot{\varphi}_2 \quad (2.6)$$

#### Arbeitsmaschinenmasse:

$$\ddot{\varphi}_2 = \frac{1}{J_2} \cdot (M_C + M_D) - \frac{1}{J_2} \cdot M_W \quad (2.7)$$

#### Bestimmungsgleichungen:

$$M_C = c \cdot \Delta\varphi \quad (2.8)$$

$$M_D = d \cdot \Delta\dot{\varphi} \quad (2.9)$$

Aus den Gleichungen (2.8) und (2.9) ergibt sich das Rückwirkungsmoment  $M_C + M_D$  zu:

$$M_C + M_D = c \cdot \Delta\varphi + d \cdot (\dot{\varphi}_1 - \dot{\varphi}_2) \quad (2.10)$$

Dieses Rückkoppelmoment  $M_C + M_D$  kann nun in die Gleichungen (2.4) und (2.7) eingesetzt werden. Geordnet nach den Zustandsgrößen kann man folgende Zustandsgleichungen angeben:

$$\ddot{\varphi}_1 = -\frac{d}{J_1}\dot{\varphi}_1 - \frac{c}{J_1}\Delta\varphi + \frac{d}{J_1}\dot{\varphi}_2 + \frac{1}{J_1}M_1 \quad (2.11)$$

$$\Delta\dot{\varphi} = \dot{\varphi}_1 - \dot{\varphi}_2 \quad (2.12)$$

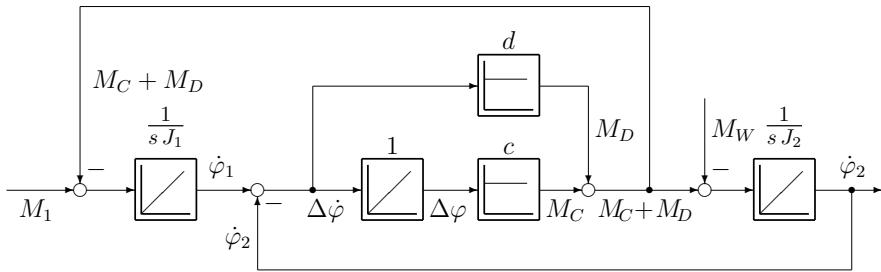
$$\ddot{\varphi}_2 = \frac{d}{J_2}\dot{\varphi}_1 + \frac{c}{J_2}\Delta\varphi - \frac{d}{J_2}\dot{\varphi}_2 - \frac{1}{J_2}M_W \quad (2.13)$$

Aus diesen drei Gleichungen lässt sich eine Matrizendarstellung angeben mit  $u = M_1$  und  $z = M_W$ . **Matrizendarstellung:**

$$\underline{\dot{x}} = \mathbf{A} \cdot \underline{x} + \underline{b} \cdot u + \underline{v} \cdot z \quad (2.14)$$

$$\begin{pmatrix} \ddot{\varphi}_1 \\ \Delta\dot{\varphi} \\ \ddot{\varphi}_2 \end{pmatrix} = \begin{pmatrix} -\frac{d}{J_1} & -\frac{c}{J_1} & \frac{d}{J_1} \\ 1 & 0 & -1 \\ \frac{d}{J_2} & \frac{c}{J_2} & -\frac{d}{J_2} \end{pmatrix} \cdot \begin{pmatrix} \dot{\varphi}_1 \\ \Delta\varphi \\ \dot{\varphi}_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{J_1} \\ 0 \\ 0 \end{pmatrix} \cdot M_1 + \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{J_2} \end{pmatrix} \cdot M_W \quad (2.15)$$

Mit den Gleichungen (2.1) bis (2.13) lässt sich der Signalflußplan des linearen, elastischen Zweimassensystems zeichnen.



**Abb. 2.2:** Unnormierter Signalflußplan des Zweimassensystems

Zur Ermittlung der beiden Übertragungsfunktionen  $G_{S1}(s) = \dot{\varphi}_2(s)/M_1(s)$  und  $G_{S2}(s) = \dot{\varphi}_1(s)/M_1(s)$  kann der Signalflußplan in Abb. 2.2 umgezeichnet werden (Abb. 2.3):

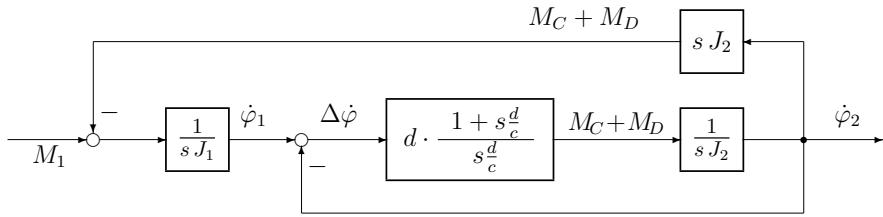
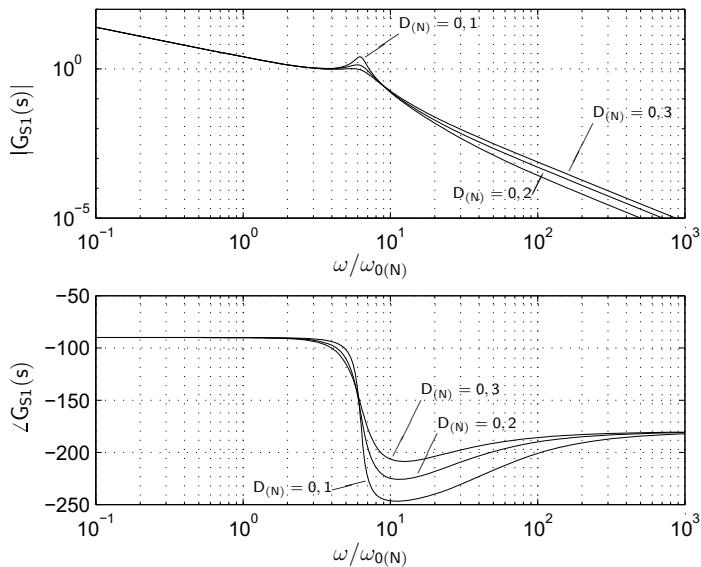


Abb. 2.3: Umgeformter Signalflußplan

## 2.2 Regelung der Arbeitsmaschinendrehzahl

$$G_{S1}(s) = \frac{\dot{\varphi}_2(s)}{M_1(s)} = \underbrace{\frac{1}{s(J_1 + J_2)}}_{\text{starre Verbindung}} \cdot \underbrace{\frac{1 + s\frac{d}{c}}{1 + s\frac{d}{c} + s^2 \frac{J_1 J_2}{(J_1 + J_2) \cdot c}}}_{\text{Einfluß der elastischen Welle}} \quad (2.16)$$

Abb. 2.4: Bode-Diagramm zu  $G_{S1}(s)$ : Variation des Dämpfungsgrades  $D_{(N)}$

Diskussion von  $G_{S1}(s)$ :

In der Gleichung von  $G_{S1}(s)$  sind zwei Anteile zu separieren, der integrale Anteil wie beim Einmassensystem (starre Verbindung) und der Einfluß der elastischen Welle. Dieser zweite Teil hat ein Zählerpolynom erster Ordnung und ein Nennerpolynom zweiter Ordnung. Damit ist auch das Bode-Diagramm in Abb. 2.4 prinzipiell wie folgt zu erläutern:

- bei tiefen Frequenzen wirkt der integrale Anteil; 1 : 1 Amplitudenabfall und 90° Phasenabsenkung
- bei hohen Frequenzen wirkt der zweite Gleichungsanteil der elastischen Welle; damit zusätzlicher 1 : 1 Abfall und 90° Phasenabsenkung, resultierend bei Frequenzen im Resonanzbereich des Nennerpolynoms 1 : 2 Amplitudenabfall und 180° Phasenabsenkung
- Im Resonanzbereich insbesondere sehr steiler Abfall der Phase von 90° auf Tendenz 270° (Nennerpolynom zweiter Ordnung) und danach Wirkung des Zählerpolynom. Wenn nun ein Stromregelkreis und ein Drehzahlregelkreis in Kaskadenstruktur für diese Strecke realisiert wird und wie im Buch „Elektrische Antriebe – Regelung von Antriebssystemen“ 3. Auflage, Springer–Verlag [207] dargestellt, für die Optimierung des Strom-(Momenten-)Regelkreises das Betragsoptimum und für den Drehzahlregelkreis das symmetrische Optimum verwendet werden, dann ergibt sich für ein Einmassensystem (starres Zweimassensystem) das Bode-Diagramm des offenen Drehzahlregelkreises mit der durchgezogenen Linie in Abb. 2.5. (Beachte: Strom-Regelkreis in  $G_{S1}(s)$  nicht enthalten!) Wird allerdings eine elastische Verbindung angenommen, dann gilt die gestrichelte Linie. Wesentlich ist der Phasenverlauf in Kombination mit dem Amplitudenverlauf. Bei einer harten Kopplung wird zwar die 180° Stabilitätsgrenze überschritten, die Verstärkung ist aber wesentlich kleiner als eins, d.h. das System ist stabil. Dies gilt nicht mehr bei einer weichen Kopplung. Die erste Aussage ist allerdings kritisch zu beurteilen. In Abb. 2.5 ist die Resonanz-Kreisfrequenz  $\omega_{0(N)}$  in Relation zu  $\omega_d$  bzw. zu  $1/T_{\sigma n}$  gesetzt. Die Zeitkonstante  $T_{\sigma n}$  entspricht der Ersatz-Zeitkonstanten des optimierten Stromregelkreises und wird bei dynamischen Antrieben kleiner 2 ms sein. Mechanische Eigenfrequenzen sind aber sehr häufig im Gebiet „einige Hertz“, d.h. die Dynamik des Stromregelkreises und darausfolgend des Drehzahlregelkreises muß drastisch verringert werden.

## 2.3 Regelung der Antriebsmaschinendrehzahl

In gleicher Weise wie in Kapitel 2.2 kann die Übertragungsfunktion  $G_{S2}(s)$  ermittelt werden.

$G_{S2}(s)$  hat wiederum einen rein integralen Anteil und einen zweiten Teil aus dem

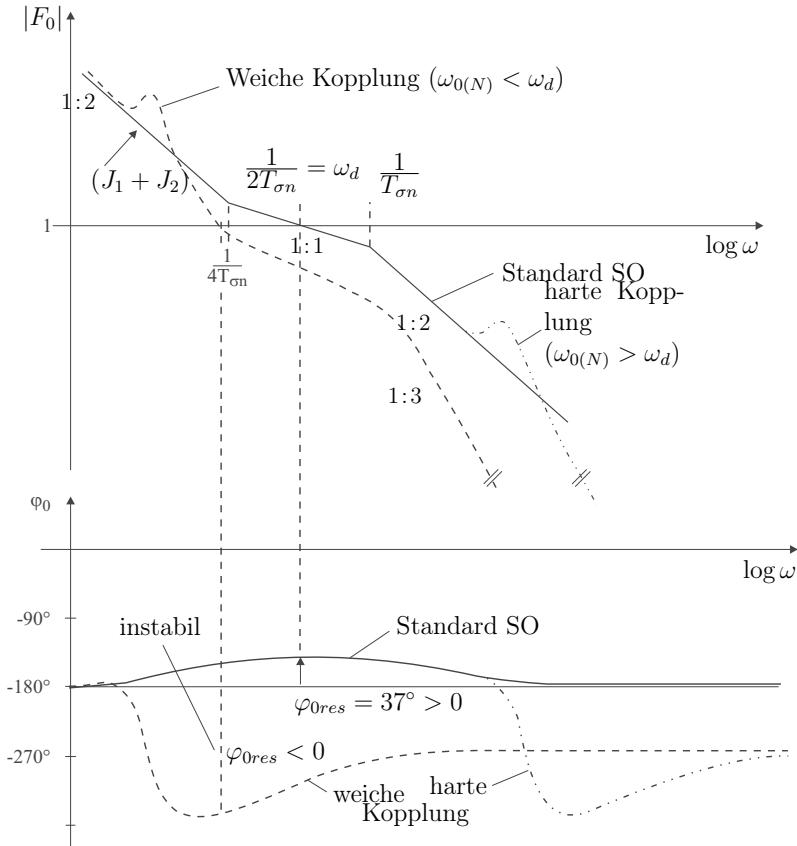


Abb. 2.5: Frequenzgänge des offenen  $\dot{\varphi}_2$ -Regelkreises

Einfluß der elastischen Welle kommend. Allerdings ist nun sowohl das Zähler- und das Nennerpolynom zweiter Ordnung.

$$G_{S2}(s) = \frac{\dot{\varphi}_1(s)}{M_1(s)} = \underbrace{\frac{1}{s(J_1 + J_2)}}_{\text{starre Verbindung}} \cdot \underbrace{\frac{1 + s\frac{d}{c} + s^2\frac{J_2}{c}}{1 + s\frac{d}{c} + s^2\frac{J_1 J_2}{(J_1 + J_2)c}}}_{\text{Einfluß der elastischen Welle}} \quad (2.17)$$

Tiefe Frequenzen ( $s \rightarrow 0$ ):

$$G_{S2}(s) \approx \frac{1}{s(J_1 + J_2)} = \frac{1}{s T'} \quad (2.18)$$

**Hohe Frequenzen ( $s \rightarrow \infty$ ):**

$$G_{S2}(s) \approx \frac{1}{s(J_1 + J_2)} \cdot \frac{\frac{J_2}{c}}{\frac{J_1 J_2}{(J_1 + J_2)c}} = \frac{1}{s J_1} = \frac{1}{s T''} \quad (2.19)$$

Aus den Ableitungen ist zu entnehmen, daß nun sowohl bei tiefen als auch bei sehr hohen Frequenzen ein rein integrales Verhalten resultiert, dies ist auch in Abb.2.6 dargestellt.

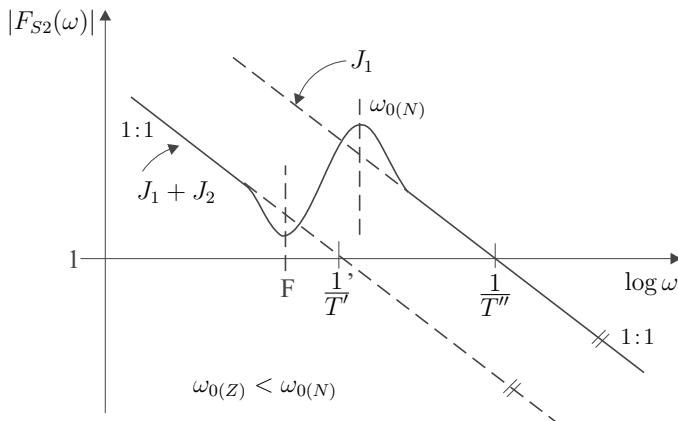


Abb. 2.6: Amplitudengang von  $G_{S2}(s)$

Wichtig ist nun wiederum das Verhalten im Resonanzbereich. Hierzu dienen die folgenden Gleichungen für das Nennerpolynom zweiter Ordnung:

**Normpolynom 2. Ordnung:**

$$N(s) = 1 + s \cdot \frac{2D}{\omega_0} + s^2 \cdot \frac{1}{\omega_0^2} \quad (2.20)$$

$$x = \frac{J_1}{(J_1 + J_2)} \quad (2.21)$$

$$J_{ges} = J_1 + J_2 \quad (2.22)$$

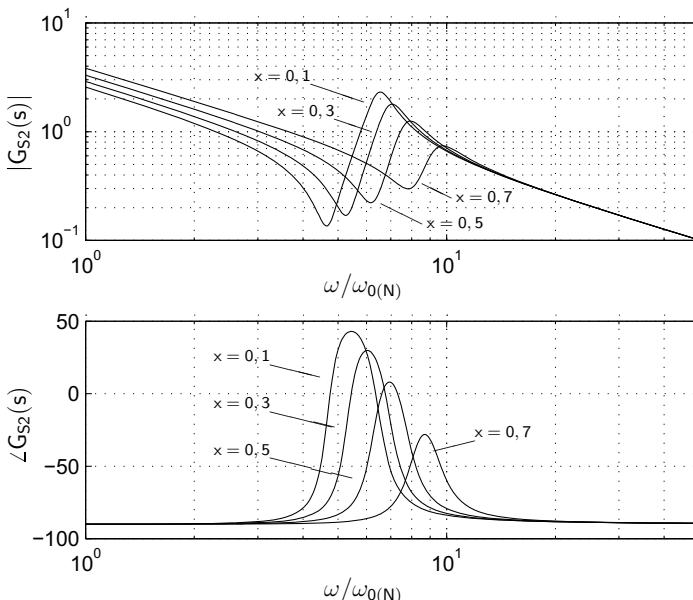
$$J_1 = x \cdot J_{ges} \quad (2.23)$$

$$J_2 = (1 - x) J_{ges} \quad (2.24)$$

$$\omega_{0(N)} = \sqrt{\frac{(J_1 + J_2) \cdot c}{J_1 J_2}} = \sqrt{\frac{c}{x(1-x) \cdot J_{ges}}} \quad (2.25)$$

$$D_{(N)} = \frac{d}{2} \cdot \sqrt{\frac{(J_1 + J_2)}{J_1 J_2 \cdot c}} = \frac{d}{2} \cdot \sqrt{\frac{1}{x(1-x) \cdot J_{ges}}} \quad (2.26)$$

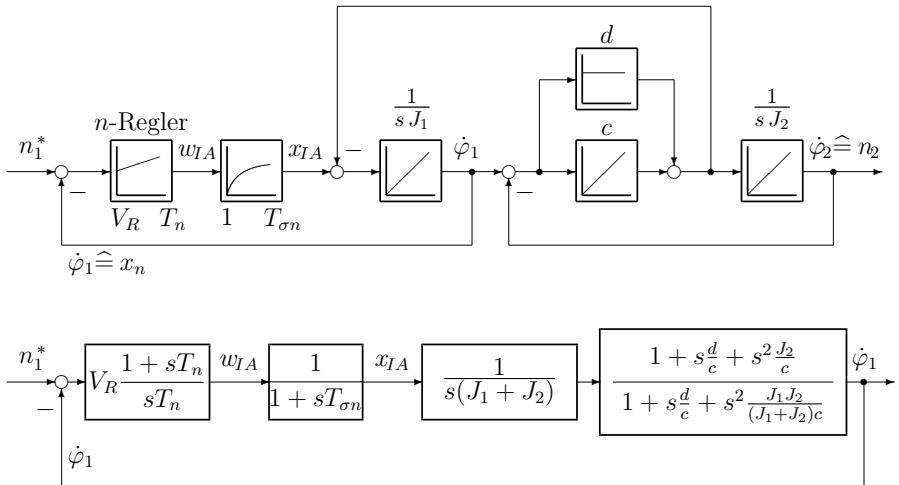
Werden diese Ansätze auf das Zählerpolynom übertragen, dann ergibt sich  $\omega_{0(N)} = \omega_{0(Z)}/\sqrt{x}$  mit  $x < 1$ , d.h.  $\omega_{0(N)} > \omega_{0(Z)}$ ! Dies bedeutet, daß die Phasenverringerung aufgrund des Zählerpolynoms zweiter Ordnung bei einer tieferen Kreisfrequenz wirksam ist als die Phasenerhöhung durch das Nennerpolynom zweiter Ordnung. Dies ist in der Abb. 2.7 dargestellt.



**Abb. 2.7:** Bode-Diagramm zu  $G_{S2}(s)$ : Variation des Trägheitsmomentenverhältnis  $x$

Wenn statt der Phasenvergrößerung im Resonanzbereich bei  $G_{S1}(s)$  nun bei  $G_{S2}(s)$  eine Phasenverringerung und dann eine gleiche Phasenvergrößerung erfolgt, dann werden bei einer Optimierung nach dem symmetrischen Optimum für den Drehzahlregelkreis der Antriebsmaschine keine grundlegenden Stabilitätsprobleme für die Antriebsmaschinendrehzahl zu erwarten sein (Abb. 2.9).

Die Abb. 2.8 zeigt den Signalfußplan der Regelung für die Drehzahl der Antriebsmaschine. Der unterlagerte Stromregelkreis ist durch die PT<sub>1</sub>-Übertragungsfunktion mit der Zeitkonstanten  $T_{σn}$  bzw.  $T_{ersi}$  approximiert.



**Abb. 2.8:** Regelung der Antriebsmaschinendrehzahl

Die Abb. 2.9 zeigt die Bode-Diagramme des offenen Drehzahlregelkreises der Antriebsmaschine bei harter und weicher Kopplung und Abb. 2.10 sowie Abb. 2.11 das Bode-Diagramm des geschlossenen Drehzahlregelkreises sowie die Führungssprungantwort (ohne Führungsglättung). Zu sehen ist bei der Sprungantwort das dynamische Verhalten für die Drehzahl der Antriebsmaschine; das dynamische Verhalten der Arbeitsmaschinendrehzahl ist, insbesondere bei weicher Kopplung, völlig unbefriedigend, es zeigt ein schwach gedämpftes Schwingverhalten.

In Abb. 2.10 ist deutlich das prinzipielle Übergangsverhalten bei SO-Optimierung (ohne Führungsglättung) zu erkennen. Allerdings schwingen die beiden Drehzahlen im Gegensinn. Dieses gegensinnige Schwingen wird wesentlich gravierender bei zunehmend weicher Ankopplung (Abb. 2.12 und Abb. 2.11). Die Erklärung für dieses unerwünschte Verhalten ist relativ einfach. Die Antriebsmaschinendrehzahl ist geregt. Das System elastische Welle und Trägheitsmomente  $J_2$  verhalten sich wie ein passiver Dämpfer (siehe auch Kapitel 20.1 im Buch „Elektrische Antriebe – Regelung von Antriebssystemen“ 3. Auflage, Springer-Verlag [207]). Zusammenfassend lässt sich feststellen, daß die Regelung der Antriebsmaschinendrehzahl wesentlich unproblematischer ist als die Drehzahlregelung der Arbeitsmaschine.

Allerdings müssen auch — wie schon vorher diskutiert — deutliche Reduzierungen bei der Dynamik der Drehzahl- und Stromregelkreise in Kauf genommen werden, um die Schwingung nicht anzuregen.

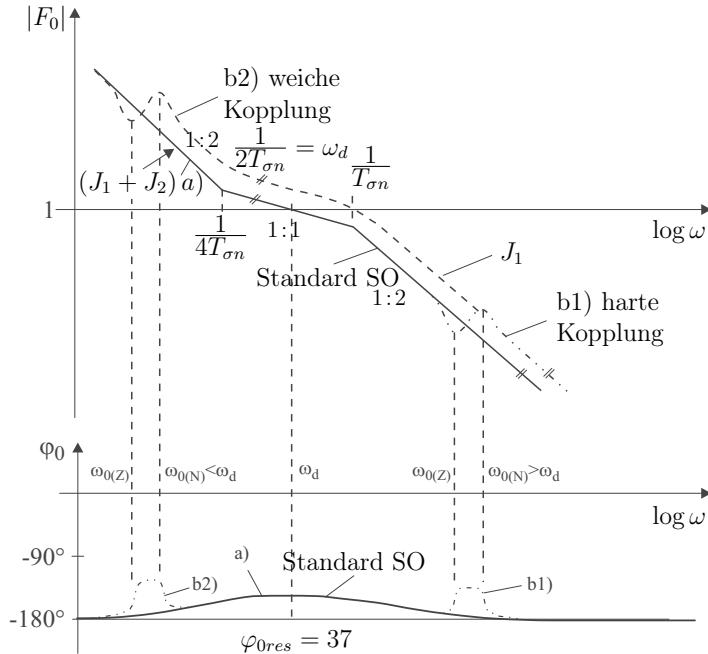


Abb. 2.9: Bode-Diagramm des offenen  $\dot{\varphi}_1$ -Regelkreises

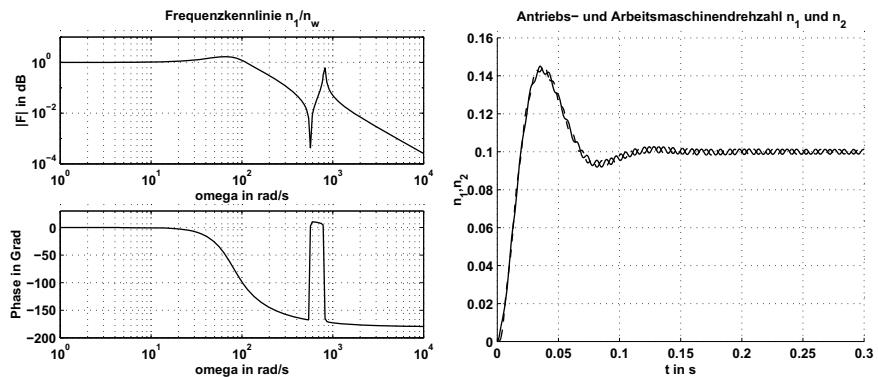
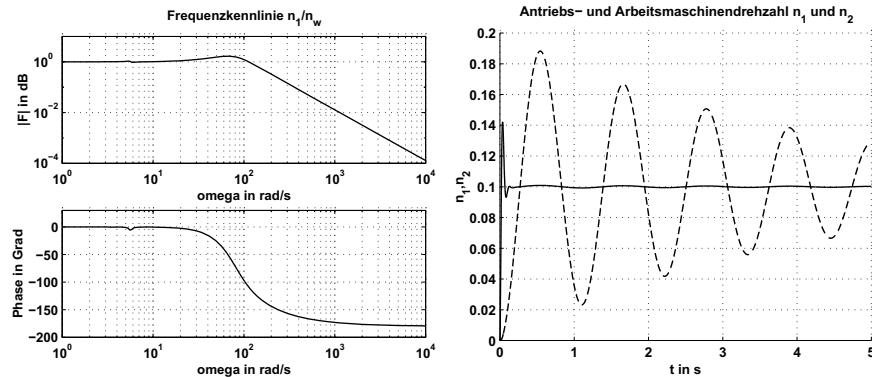


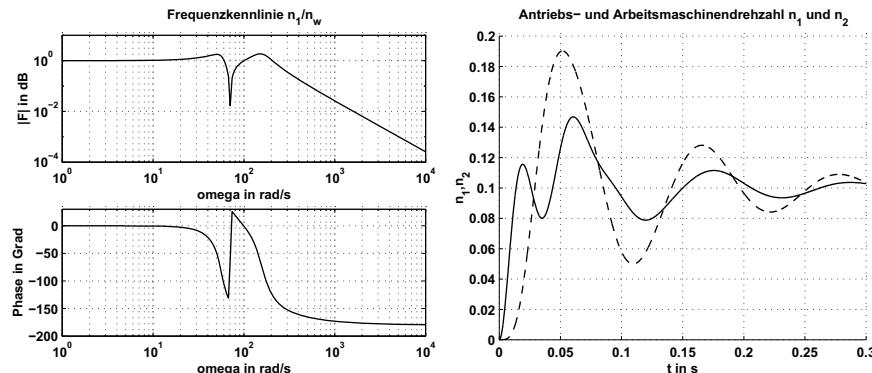
Abb. 2.10: Bodediagramm des geschlossenen Drehzahlregelkreises und Führungssprungantwort bei harter Ankopplung; ( $\omega_{0(N)} = 10 \omega_d$ ) (—  $n_1$ , - - -  $n_2$ )

## 2.4 Proportionale Zustandsregelung

Die Zustandsregelung ist eine seit langem bekannte Variante, die nun aufgrund der Schwierigkeiten bei der Kaskadenregelung eingesetzt werden soll. Die



**Abb. 2.11:** Bodediagramm des geschlossenen Drehzahlregelkreises (Antriebsmaschine) und Führungssprungantwort bei weicher Ankopplung; ( $\omega_{0(N)} = 0,1 \omega_d$ ) (—  $n_1$ , - - -  $n_2$ )



**Abb. 2.12:** Bodediagramm des geschlossenen Drehzahlregelkreises (Antriebsmaschine) und Führungssprungantwort bei Lage der Eigenfrequenzen im Nutzfrequenzbereich; ( $\omega_{0(N)} = \omega_d$ ) (—  $n_1$ , - - -  $n_2$ )

Abb. 2.13 zeigt die prinzipielle Struktur, wobei die drei Zustände  $n_1$ ,  $n_2$  und  $\Delta\varphi$  proportional mit den Verstärkungen  $r_1$ ,  $r_2$  und  $r_3$  zurückgeführt werden. (Beachte: Strom-Regelkreis vernachlässigt!)

Zusätzlich ist noch ein Korrekturfaktor  $K_V$  notwendig, um im ungestörten Bereich ( $M_W = 0$ ) den Regelfehler Null sicherzustellen. Die Auslegung der Faktoren  $r_1$ ,  $r_2$ ,  $r_3$  und  $K_V$  ist beispielsweise aus dem Buch „Elektrische Antriebe – Regelung von Antriebssystemen“ 3. Auflage, Springer-Verlag [207] zu entnehmen.

Die Regelergebnisse für harte, mittlere und weiche Kopplung werden in den Abbildungen 2.14, 2.15 und 2.16 gezeigt. Grundsätzlich ist festzustellen, daß die

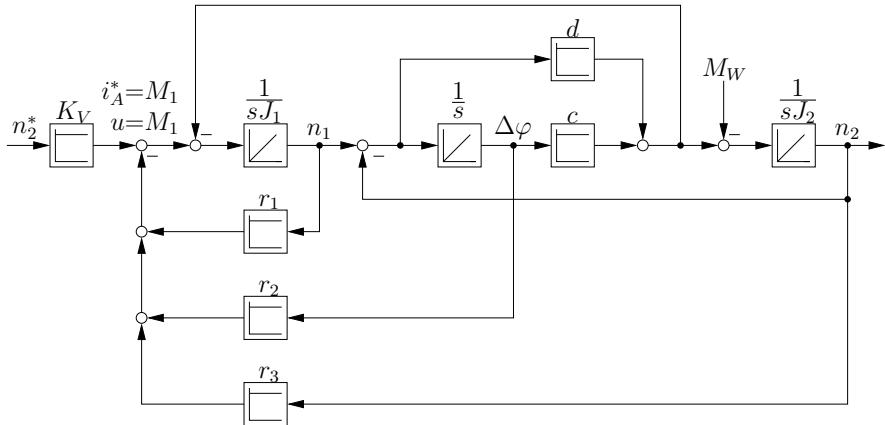


Abb. 2.13: Zustandsregelung für  $n_2$  ohne I-Anteil

Regelung der Arbeitsmaschine nun dynamisch nahezu perfekt erfolgt. Zu beachten ist, daß mit abnehmender Steifigkeit der Kopplung sich die erreichbare Dynamik verringert und daß bei Störgrößen ungleich Null eine bleibende Regelabweichung auftritt. Weiterhin ist zu bedenken, daß in den Abbildungen nur ein Führungssprung von 1 % vorgegeben wurde, daß aber in Abb. 2.14 bereits das Nennmoment überschritten wurde. Dies bedeutet, daß bei größeren Führungs sprungen generell die Stellgrößenbeschränkung wirksam wird und damit dieser nichtlineare Effekt zu berücksichtigen ist.

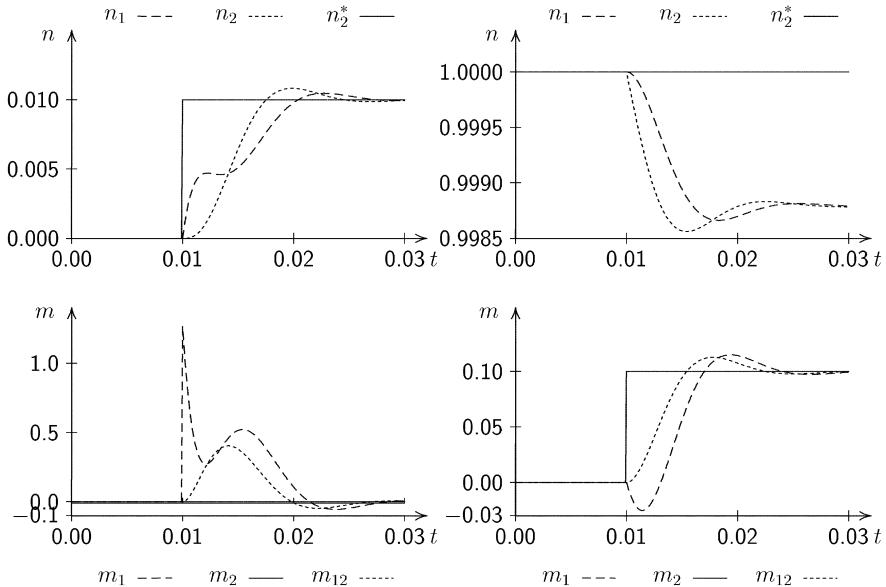
## 2.5 Integrale Zustandsregelung

Aufgrund der bleibenden Regelabweichung bei Störgrößen ungleich Null und proportionaler Zustandsregelung, wird nun ein Führungsintegrator zur Ausregelung der Regelfehler der Arbeitsmaschinendrehzahl  $n_2$  zusätzlich eingeführt.

Den Signalflußplan zeigt Abb. 2.17. (Beachte: Strom-Regelkreis vernachlässigt!)

Für die Zustandsdarstellung der Strecke gilt wieder:

$$\begin{pmatrix} \dot{n}_1 \\ \Delta\dot{\varphi} \\ \dot{n}_2 \end{pmatrix} = \begin{pmatrix} -\frac{d}{J_1} & -\frac{c}{J_1} & \frac{d}{J_1} \\ 1 & 0 & -1 \\ \frac{d}{J_2} & \frac{c}{J_2} & -\frac{d}{J_2} \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ \Delta\varphi \\ n_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{J_1} \\ 0 \\ 0 \end{pmatrix} \cdot M_1 \quad (2.27)$$



**Abb. 2.14:** Zustandsregelung ohne I-Anteil bei  $\omega_{0(N)} = 628.32 \text{ s}^{-1}$  (starre Kopplung)

Das Regelgesetz lautet in diesem Fall

$$u = m_1 = -\underline{r}^T \cdot \underline{x} + r_4 \cdot x_I \quad \text{mit} \quad \underline{r}^T = (r_1 \ r_2 \ r_3) \quad (2.28)$$

und beinhaltet den Reglerzustand  $x_I$ , für den gilt:

$$\dot{x}_I = \frac{n_1^* - n_1}{T_N} \quad (2.29)$$

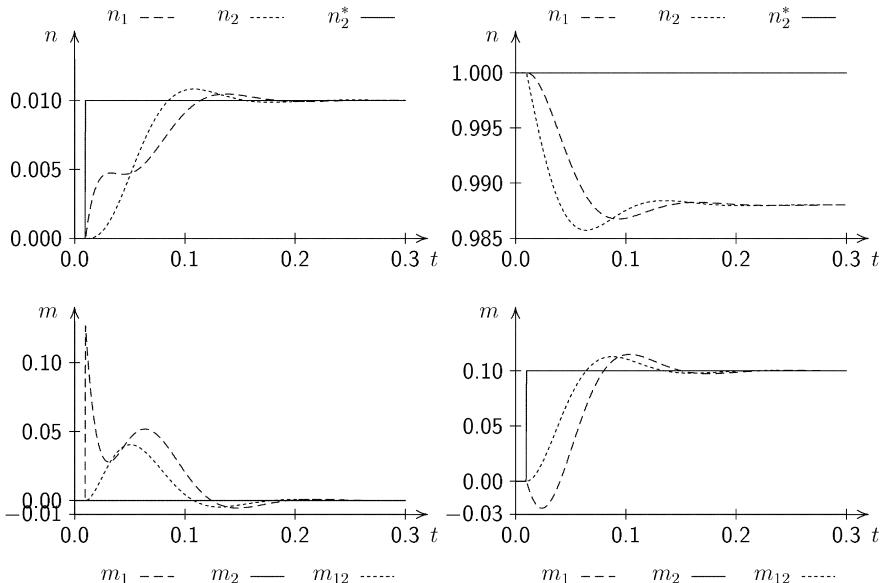
Damit ergibt sich die Zustandsdarstellung des geschlossenen Kreises zu:

$$\dot{\underline{x}}_I = \begin{pmatrix} \dot{\underline{x}} \\ \dot{x}_I \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{A} - \underline{b} \cdot \underline{r}^T & \underline{b} \cdot r_4 \\ -\frac{1}{T_N} \cdot \underline{c}^T & 0 \end{pmatrix}}_{\mathbf{A}_{ZRI}} \cdot \underline{x}_I + \underbrace{\begin{pmatrix} 0 \\ \frac{1}{T_N} \end{pmatrix}}_{\underline{b}_{ZRI}} \cdot n_1^* \quad (2.30)$$

mit

$$\underline{c}^T = (1 \ 0 \ 0) \quad (2.31)$$

Die Abbildungen 2.18 bis 2.20 zeigen die Regelergebnisse (Führung und Störung), die Ergebnisse sind sehr zufriedenstellend.

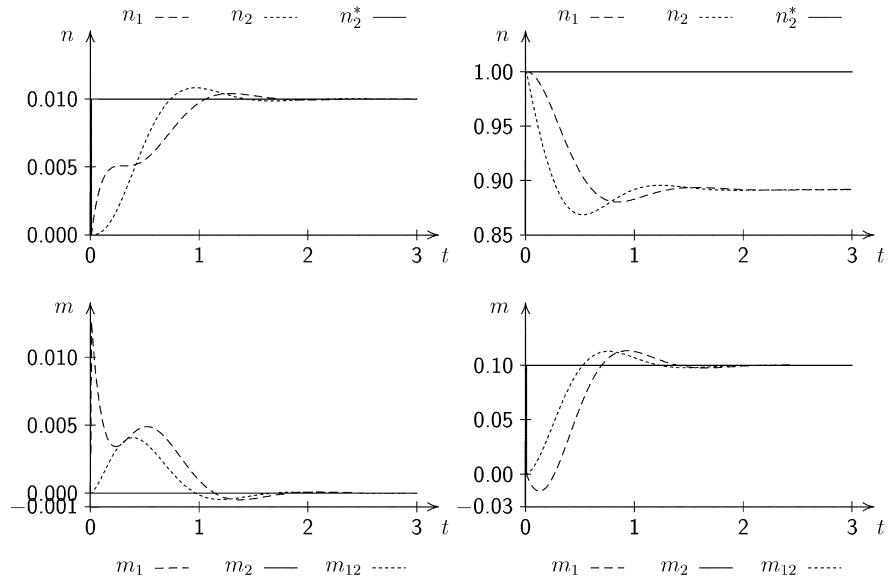


**Abb. 2.15:** Zustandsregelung ohne I-Anteil bei  $\omega_0(N) = 62.832 \text{ s}^{-1}$  (mittlere Kopplung)

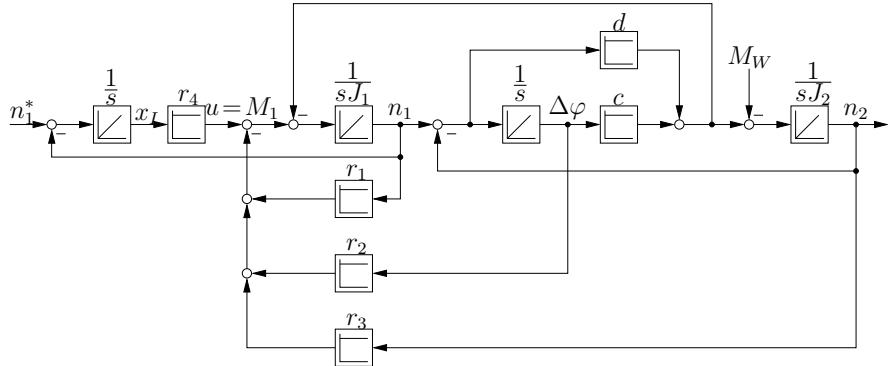
## 2.6 Nichtlineares Zweimassensystem

Bisher wurde in den Untersuchungen ein lineares, elastisch gekoppeltes Zweimassensystem angenommen. Bei den folgenden Überlegungen wollen wir annehmen, daß die Reibung als Haft- und Gleitreibung bei der Arbeitsmaschine relevant sei (siehe auch Abb. 2.1).

Wenn sich das mechanische System im Stillstand befindet, dann ist die Haftreibung wirksam und die elastische Welle sowie der Rotor der elektrischen Maschine (Trägheitsmoment  $J_1$ ) müssen um einen Winkel  $\alpha_{12}$  verdreht werden, so daß  $M_C = c\Delta\varphi = M_{\text{Haftreibung}}$  ist. Erst wenn die Haftreibung überschritten wird, wird  $n_2 \neq 0$  sein und dann ist die Gleitreibung wirksam. Diese beiden Effekte können in Abb. 2.21 sehr gut in ihrer Auswirkung beobachtet werden. In der Abb. 2.21 wird einmal ein dreieckförmiger bzw. ein sinusförmiger Verlauf der Position angenommen. Die durchgezogenen Linien sind Meßwerte aus einem realen System (Werkzeugmaschine), die gestrichelten Linien sind Simulationsergebnisse. Bei der Simulation wurde ein lineares elastisches Zweimassensystem angenommen. Dieses Simulationsmodell hatte sich aufgrund einer Ordnungsreduktion eines FE-Moduls mit sehr hoher Ordnung (516. Ordnung) ergeben. Beim dreieckförmigen Verlauf der Position und der daraus resultierenden Motordrehzahl sind keine sehr großen Unterschiede zwischen der Simulation und der Messung

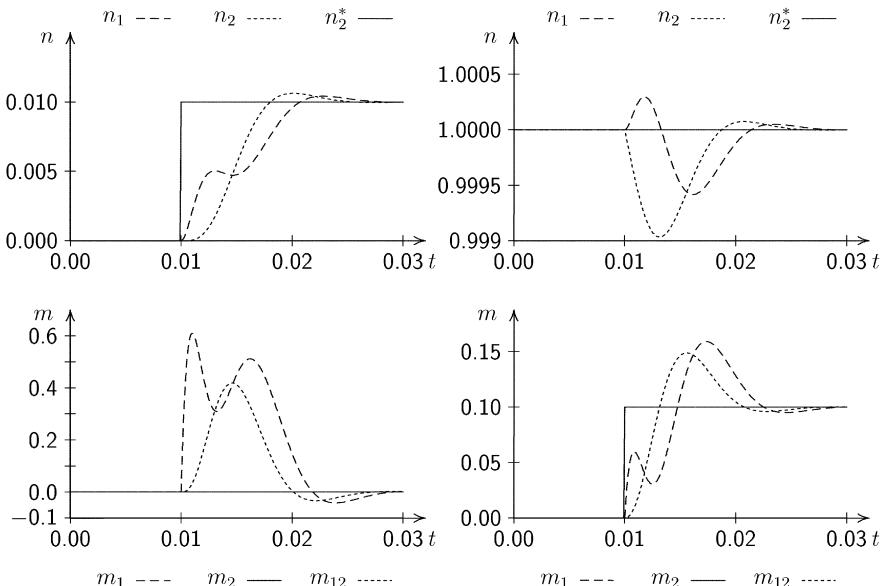


**Abb. 2.16:** Zustandsregelung ohne I-Anteil bei  $\omega_{0(N)} = 6.2832 \text{ s}^{-1}$  (weiche Kopplung)



**Abb. 2.17:** Signalflußplan der Zustandsregelung mit I-Anteil

festzustellen. Beim Motorstrom sind allerdings deutliche Unterschiede zu erkennen, denn bei der Simulation ist bei konstanter Motordrehzahl das notwendige Motormoment (Ankerstrom) Null (ideal, keine Gleitreibung) bei der Messung aber ungleich Null.

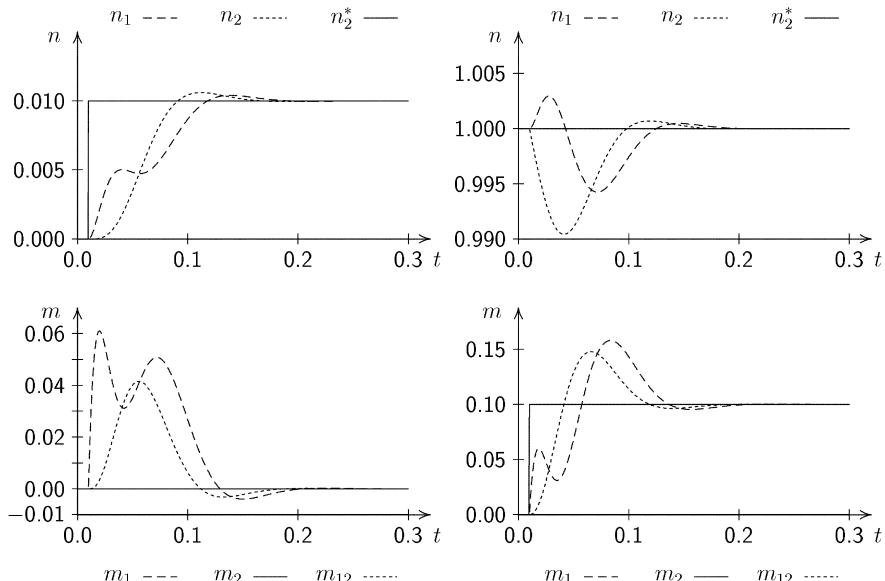


**Abb. 2.18:** Zustandsregelung mit I-An teil bei  $\omega_{0(N)} = 628.32 \text{ s}^{-1}$  (starre Kopplung)

Bei der sinusförmigen Positionsvariation sind bereits bei den Motordrehzahlen Unterschiede festzustellen, denn beim realen System wird beim Nulldurchgang der Drehzahl die Haftreibung wirksam, und es tritt ein Drehzahlstillstand auf (stick-slip-Effekt). Noch deutlicher ist der Einfluß der Haftreibung beim Moment (Ankerstrom) zu erkennen.

Dies bedeutet, es sind sehr deutliche Unterschiede im Verlauf des Moments (Ankerstrom), von der Betriebsweise auch bei den Drehzahlen und damit letztendlich auch bei den Positionen festzustellen. Es ist einsichtig, daß das lineare Modell nicht für eine genaue und dynamische Regelungsauslegung geeignet ist, da sich die Streckenmodelle bereits deutlich unterscheiden. Die Abb. 2.22 zeigt die gleichen Betriebsfälle wie in Abb. 2.21. Allerdings ist nun bei der Simulation die Reibung (Haft- und Gleitreibung) als geschätzter Effekt, d.h. nicht genau parametriert, berücksichtigt. Es ist nun eine grundsätzliche Übereinstimmung zwischen Simulation und Messung festzustellen.

Ein weiterer durch die Reibung bedingter Effekt ist der hunting-Effekt, der bei der Lageregelung zu beobachten ist. Wenn — bei der Geschwindigkeit Null — die Sollposition die Istposition überschreitet, dann entsteht ein positiver Lage-Regelfehler und damit ein geringes Motormoment, welches die Haftreibung unterschreitet, d.h. es wird keine Positionsänderung auftreten. Wenn nun der Lageregler einen Integralanteil hat, dann wird der positive Regelfehler aufintegriert und somit das Motormoment kontinuierlich erhöht. Zu dem Zeitpunkt, zu dem

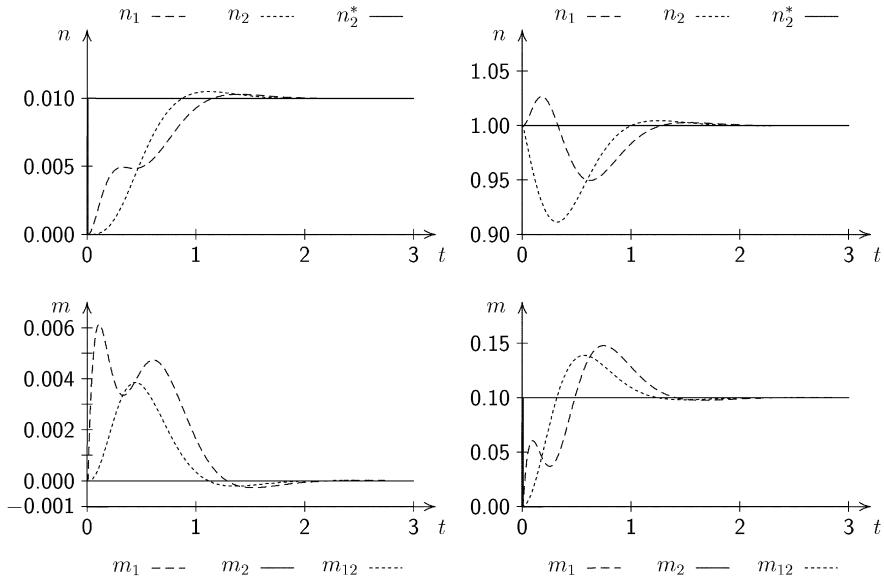


**Abb. 2.19:** Zustandsregelung mit I-Anteil bei  $\omega_{0(N)} = 62.832 \text{ s}^{-1}$  (mittlere Kopplung)

das Motormoment die Haftreibung erreicht bzw. geringfügig überschreitet, erfolgt der notwendige Geschwindigkeitsaufbau von Null aus. Dadurch bedingt fällt das Reibungsmoment von der Haftreibung auf die normalerweise geringere Gleitreibung (bei geringen Geschwindigkeiten) ab, die Geschwindigkeit wird deutlich erhöht und die Istposition überschreitet die Sollposition; dies bedeutet einen negativen Regelfehler, das Motormoment geht in den Bremsbetrieb, die Geschwindigkeit wird zu Null abgebaut — bei negativem Lage-Regelfehler. Der Vorgang wiederholt sich analog zum positiven Lage-Regelfehler. Dies ist der Ablauf beim hunting-Effekt, der immer dann auftritt, wenn der Lageregler einen Integralanteil hat und die Gleitreibung geringer ist als die Haftreibung.

## 2.7 Dreimassensystem

In den vorherigen Kapiteln wurde ein elastisch gekoppeltes Zweimassensystem angenommen. Die Abb. 2.23 zeigt den Signalflußplan eines elastisch gekoppelten Dreimassensystems. Aus diesem Signalflußplan ist die Erweiterungsmaßgabe bei Mehrmassensystemen erkennbar. Es ist somit nicht allzu schwierig, ein physikalisch orientiertes Modell und den zugehörigen Signalflußplan zu ermitteln. Dies hat den Vorteil, daß gegenüber der Ein–Ausgangs–Darstellung ein physikalisches



**Abb. 2.20:** Zustandsregelung mit I-Anteil bei  $\omega_0(N) = 6.2832 \text{ s}^{-1}$  (weiche Kopplung)

Verständnis des Systems erhalten bleibt. Der gleiche Vorteil besteht gegenüber den ordnungsreduzierten Modellen.

Es gelten die Differentialgleichungen:

### Differentialgleichungen

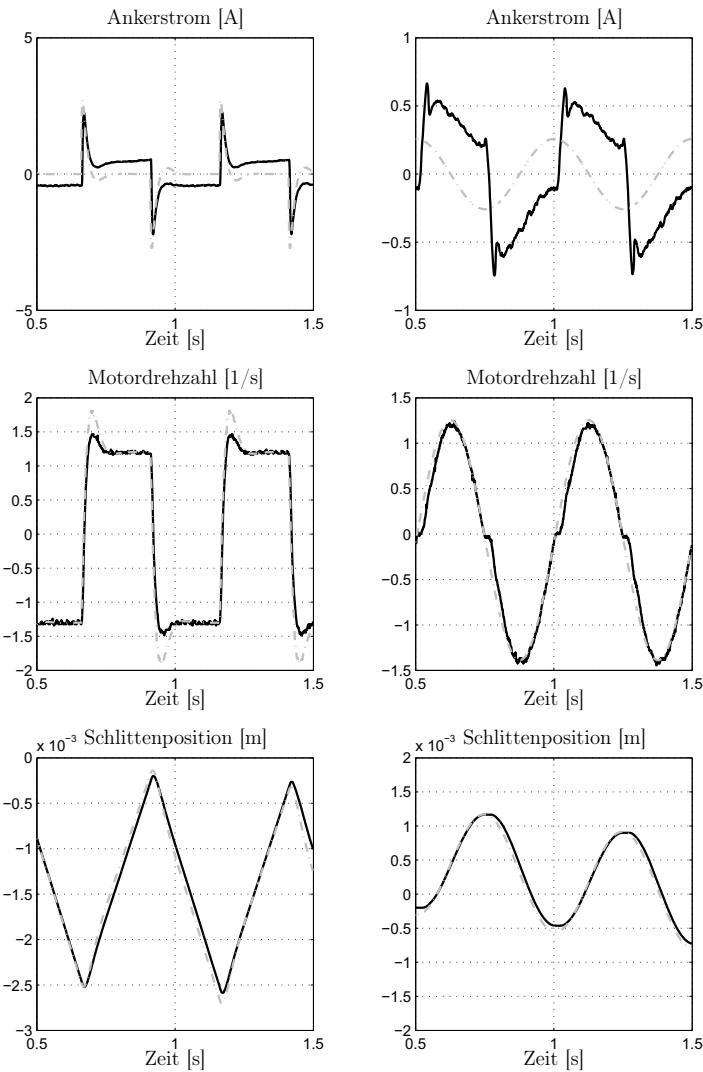
$$\ddot{\varphi}_3 = \frac{1}{J_3} \left( c_2(\varphi_2 - \varphi_3) + d_2(\dot{\varphi}_2 - \dot{\varphi}_3) - M_W \right) \quad (2.32)$$

$$\ddot{\varphi}_2 = \frac{1}{J_2} \left( c_1(\varphi_1 - \varphi_2) + d_1(\dot{\varphi}_1 - \dot{\varphi}_2) - c_2(\varphi_2 - \varphi_3) - d_2(\dot{\varphi}_2 - \dot{\varphi}_3) \right) \quad (2.33)$$

$$\ddot{\varphi}_1 = \frac{1}{J_1} \left( -c_1(\varphi_1 - \varphi_2) - d_1(\dot{\varphi}_1 - \dot{\varphi}_2) + M_1 \right) \quad (2.34)$$

Als Übertragungsfunktion ergibt sich:

$$G_{S3}(s) = \frac{\dot{\varphi}_3}{M_1} = \frac{b_2 s^2 + b_1 s + b_0}{a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s} \quad (2.35)$$



**Abb. 2.21:** Vergleich von simulierten linearem Modell und gemessenen Daten (schwarz: Gemessene Daten; grau: Simulations Daten)

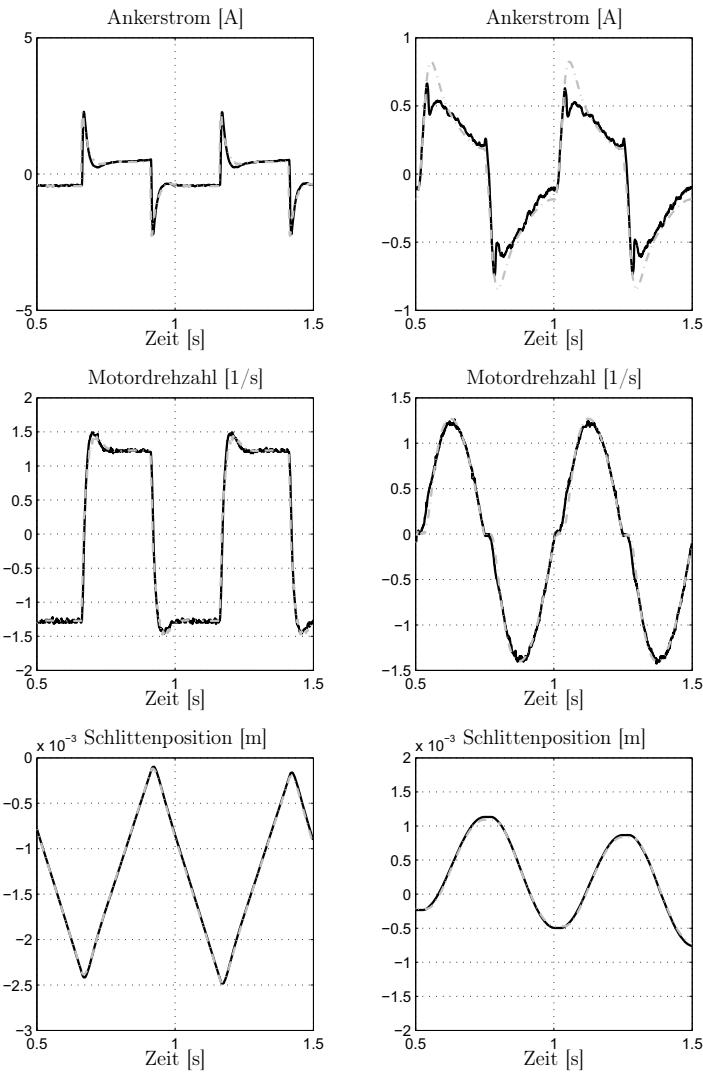
mit den Parametern

$$b_2 = d_1 d_2$$

$$b_1 = c_1 d_2 + c_2 d_1$$

$$b_0 = c_1 c_2$$

$$a_5 = J_1 J_2 J_3$$



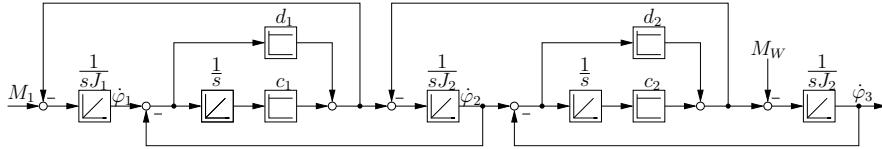
**Abb. 2.22:** Vergleich von Simulation des nichtlinearen Zweimassenmodells mit gemessenen Daten (schwarz: Gemessene Daten; grau: Simulations Daten)

$$a_4 = d_1 J_3 (J_1 + J_2) + d_2 J_1 (J_2 + J_3)$$

$$a_3 = c_1 J_3 (J_1 + J_2) + c_2 J_1 (J_2 + J_3) + d_1 d_2 (J_1 + J_2 + J_3)$$

$$a_2 = (d_1 c_2 + d_2 c_1) (J_1 + J_2 + J_3)$$

$$a_1 = c_1 c_2 (J_1 + J_2 + J_3)$$



**Abb. 2.23:** Unnormierter Signalflußplan des linearen Dreimassensystems

Es ergibt sich ein Zählerpolynom dritter und ein Nennerpolynom fünfter Ordnung. Um wie vorher dieses System zu einem rein integralen Teil (starres System) und einem zweiten Teil (elastische Einflüsse) einfach aufteilen zu können, wird  $d_1 = d_2 = 0$  gesetzt, eine allgemein zulässige Vereinfachung; damit werden die Parameter  $b_1 = b_2 = a_2 = a_4 = 0$ . Mit  $d_1 = d_2 = 0$  ergibt sich das Nennerpolynom zu:

1. Rein integraler Anteil:

$$\frac{1}{(J_1 + J_2 + J_3)s} \quad (2.36)$$

$$\text{Pol: } s_1 = 0 \quad (2.37)$$

2. Biquadratischer Anteil:

$$\text{Pole: } s_{2,3} = \pm j\sqrt{q_1 + q_2} \quad (2.38)$$

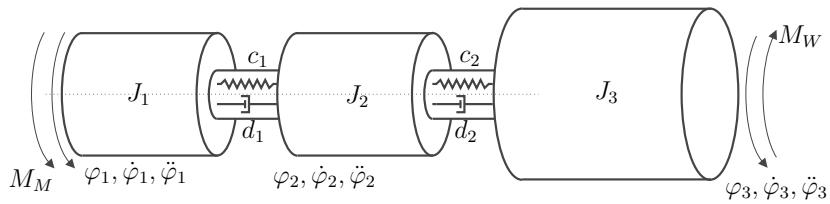
$$s_{4,5} = \pm j\sqrt{q_1 - q_2} \quad (2.39)$$

mit den Parametern  $q_1$  und  $q_2$

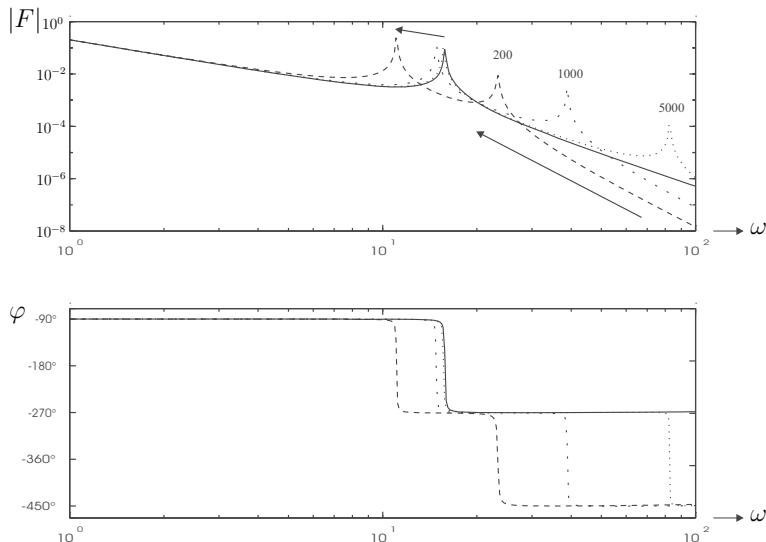
$$q_1 = -\frac{c_1 J_2 (J_1 + J_3) + c_2 J_1 (J_2 + J_3)}{2 J_1 J_2 J_3} \quad (2.40)$$

$$q_2 = \frac{\sqrt{[c_1 J_2 (J_1 + J_3) + c_2 J_1 (J_2 + J_3)]^2 - 4 c_1 c_2 J_1 J_2 J_3 (J_1 + J_2 + J_3)}}{2 J_1 J_2 J_3} \quad (2.41)$$

Mit diesem vereinfachten Modell lassen sich sehr leicht — insbesondere auch anschaulich mit Matlab/Simulink — die Einflüsse bei unterschiedlichen Trägheitsmoment- oder Elastizitätskoeffizient-Variationen der drei Massen studieren. In gleicher Weise kann auch die Ordnungsreduktion von einem Drei- auf ein Zweimassensystem geübt werden. Dies zeigen die folgenden Abbildungen 2.24 und 2.25. Wenn beispielsweise  $c_2$  variiert wird, dann ändern sich die höheren Kreisfrequenzen sehr deutlich, die niedrige Kreisfrequenz aber nur in einem begrenzten Bereich.



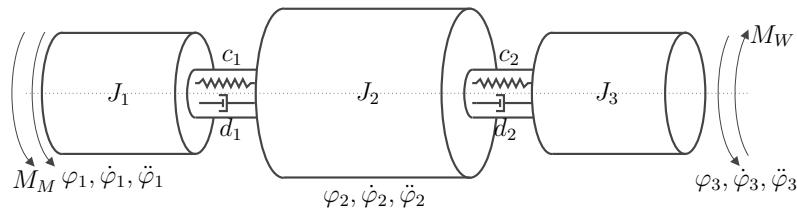
$$\begin{aligned}
 J_1 &= 1; & c_2 &= 5000; \\
 J_2 &= 1; & c_2 &= 1000; \\
 J_3 &= 3; & c_2 &= 200; \\
 c_1 &= 200; \\
 d_1 &= 0.1; \\
 d_2 &= 0.1;
 \end{aligned}$$



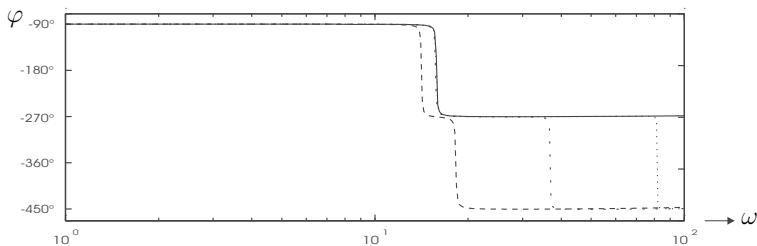
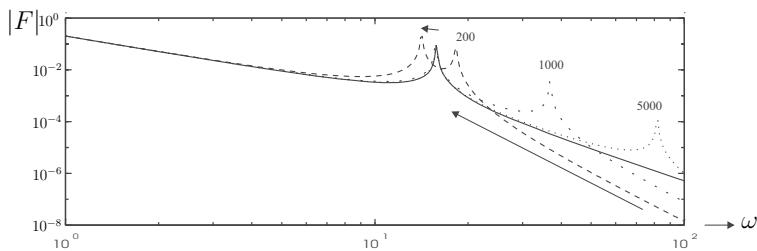
**Abb. 2.24:** Dreimassensystem und Bode-Diagramm bei  $J_2 + J_3 = J_L$  und  $J_2 < J_3$  (- - - Originalsystem, — reduziertes System)

## 2.8 Zusammenfassung: Mechatronische Systeme

Die Analysen in den vorhergehenden Kapiteln haben deutlich die Einschränkungen linearer Ansätze bei der Modellbildung der Strecke und darausfolgend bei der Regelung dargestellt. Ein weiterer Aspekt ist, daß eine physikalisch interpretier-



$$\begin{aligned}
 J_1 &= 1; & c_2 &= 5000; \\
 J_2 &= 3; & c_2 &= 1000; \\
 J_3 &= 1; & c_2 &= 200; \\
 c_1 &= 200; \\
 d_1 &= 0.1; \\
 d_2 &= 0.1;
 \end{aligned}$$



**Abb. 2.25:** Dreimassensystem und Bode-Diagramm bei  $(J_2 + J_3) = J_L$  sowie  $J_2 > J_3$  (- - - Originalsystem, — reduziertes System)

bare Darstellung eine allgemeineren und tieferen Einblick in die physikalischen Zusammenhänge und damit ein besseres Verständnis ermöglichen.

Dies sind die Gründe, warum in den kommenden Kapiteln die beiden obigen Aspekte nichtlineare Strecke und physikalische Interpretierbarkeit als Ausgangsbasis aller Überlegungen dienen.

## 2.9 Kontinuierliche Produktionsanlagen

Bisher wurde nur der Aspekt „mechanische Verbindung zwischen Antriebs- und Arbeitsmaschine“ behandelt. Im allgemeinen ist die Arbeitsmaschine aber zur Erfüllung einer technologischen Aufgabenstellung vorgesehen, z.B. die Werkzeugmaschine zur Bearbeitung von Metallen oder eine kontinuierliche Produktionsanlage zur Herstellung von Papier, Folien oder Stahl oder eine Verpackungsmaschine. Alle diese Arbeitsmaschinen haben im allgemeinen mehrere Antriebe, die abgestimmt zueinander arbeiten müssen, um die technologische Aufgabenstellung zu erfüllen. Die Abb. 2.26 zeigt das Blockschaltbild einer kontinuierlichen Produktionsanlage mit mehreren elektrischen Antrieben, die über das Material miteinander gekoppelt sind. Es ist angenommen, daß die elektrischen Antriebe eine Kaskadenregelung (Strom  $i$  und Drehzahlen  $n$ ) haben. Überlagert ist das technologische Führungssystem. Wesentlich ist nun erstens, daß das zu bearbeitende Material sehr häufig eine nichtlineare Charakteristik aufweist (siehe Abb. 2.27), die teilweise plastisch, teilweise elastisch sein kann. Diese Nichtlinearität kann zusätzlich abhängig sein von weiteren äußeren Einflüssen wie Wassergehalt beim Papier oder Temperatur bei Folien oder Stahl. Dies bedeutet, es gibt mehrdimensionale Nichtlinearitäten, die zeitlich variieren können.

Es soll nun nicht auf die weitere Behandlung derartiger System eingegangen werden. Im Buch „Elektrische Antriebe – Regelung von Antriebssystemen“ 3. Auflage, Springer-Verlag [207] werden die diesbezüglichen Informationen gegeben. Zum Abschluß soll nur noch in Abb. 2.28 der nichtlineare Signalflußplan eines Teilsystems (siehe Abb. 2.29) und der linearisierte Signalflußplan des Gesamtsystems aus Abb. 2.26 dargestellt werden. Es ist einsichtig, daß die Regelung eines derartig gekoppelten Systems hohe Anforderungen an die Regelungstechnischen Kenntnisse voraussetzt — insbesondere dann, wenn mehrdimensionale Nichtlinearitäten entsprechend Abb. 2.27 und die mechanische Kopplung mit ihren Nichtlinearitäten berücksichtigt werden.

## 2.10 Zusammenfassung: Technologische Systeme

Ausgehend von der Zusammenfassung für mechatronische Systeme ist bei technologischen Systemen eine im allgemeinen noch höhere Komplexität festzustellen. Diese höhere Komplexität ist u. a. durch die Technologien bedingt, bei der im allgemeinen mehrere mechatronische Antriebssysteme auf die Arbeitsmaschine mit dem technologischen Prozeß einwirken, dies kann beispielsweise über das zu bearbeitende Material zu nichtlinearen Verkopplungen führen, die außerdem zeitvariant sein können. Ein weiterer Punkt sind die mehrdimensionalen Nichtlinearitäten. Die Kernaspekte der zu erarbeitenden Lösungsansätze verbleiben somit.

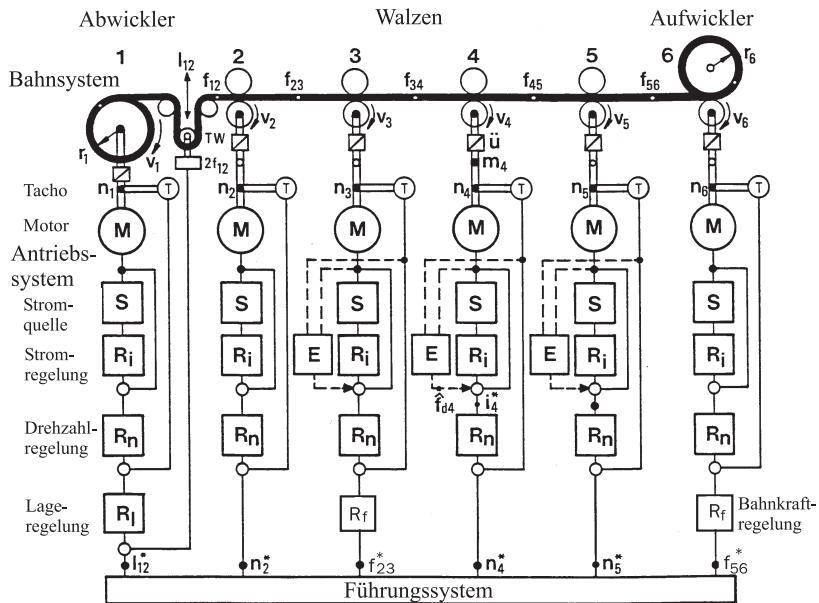
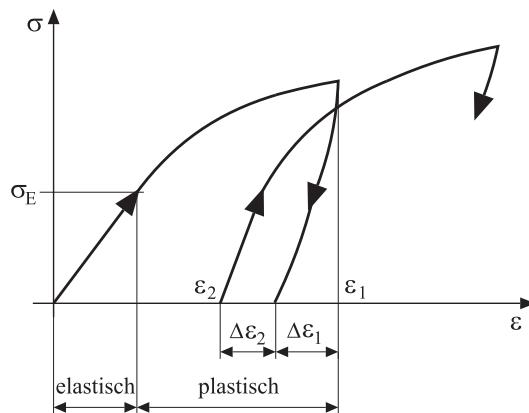


Abb. 2.26: Beispiel einer kontinuierlichen Fertigungsanlage

Abb. 2.27: Prinzipielles Spannungs-  $\sigma$  Dehnungsdiagramm  $\varepsilon$  für Papier

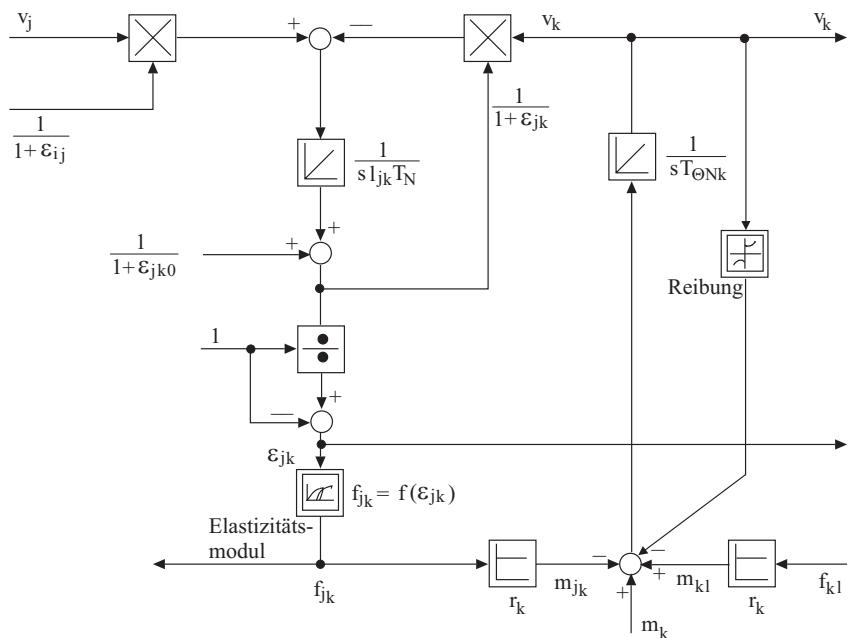


Abb. 2.28: Nichtlinearer Signalflussplan eines Teilsystems

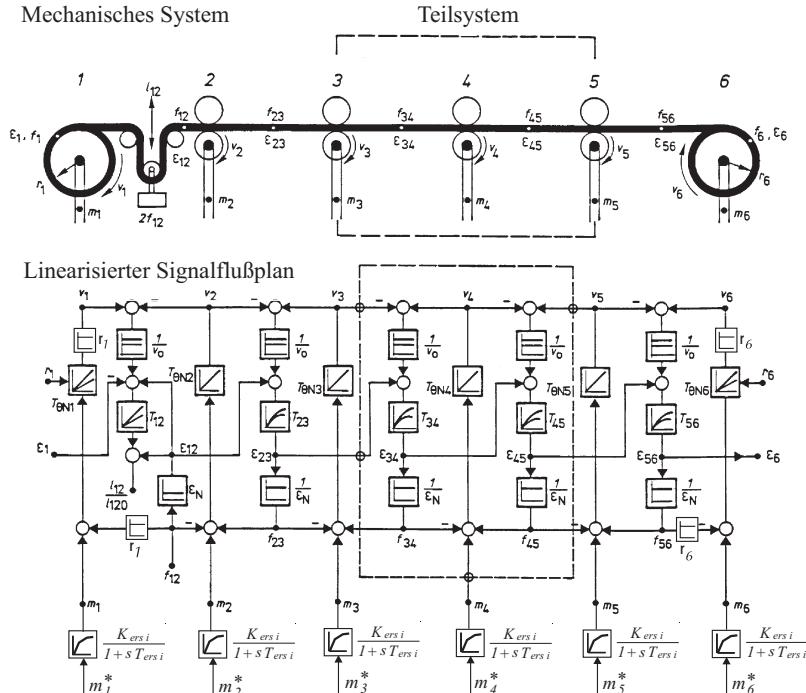


Abb. 2.29: Linearisierter Signalflußplan des Gesamtsystems

### 3 Statische Funktionsapproximatoren

In Kapitel 2 wurden zwei verschiedene nichtlineare Strecken diskutiert, die „mechatronische“ und die „technologische Strecke“. Bei beiden Fällen waren relevante Nichtlinearitäten wie u. a. die Reibung, die Lose oder das Elastizitätsmodul zu beachten. Als weiterer Kernaspekt der folgenden Überlegungen war die physikalische Interpretierbarkeit des Streckenmodells genannt worden.

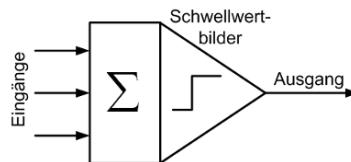
Es besteht nun die Frage, wie diese Nichtlinearitäten zu modellieren sind. Beispielsweise setzt sich die Reibung grundsätzlich aus der Haftreibung und der Gleitreibung zusammen. Die Haftreibung ist prinzipiell vom Gewicht der beiden sich reibenden Körper, der Oberflächenbeschaffenheit der beiden Körper, der Materialien und der Schmierstoffe sowie der Temperatur der Schmierstoffe (Viskosität) abhängig. Solange das von außen angreifende Moment bzw. die Kraft geringer ist als die Haftreibung — beispielsweise bei einem elektrischen Motor — ist die Geschwindigkeit der Motorwelle gleich Null. Wenn die Geschwindigkeit der Motorwelle dagegen ungleich Null ist, dann ist die Gleitreibung — Coulombsches Reibungsmodell — wirksam, die eine Funktion der Geschwindigkeit ist und der Bewegung entgegenwirkt. Im Allgemeinen ist das Haftreibungsmoment größer als das Gleitreibungsmoment, der Übergang von dem Haftreibungsmoment zum Gleitreibungsmoment ist stetig (Stribeck-Effekt). Weitere Einflussgrößen sind die Schmierstofftemperatur, die die Viskosität des Schmiermittels ändert und weitere Einflüsse wie der zeitliche Verschleiß. Zu beachten ist weiterhin, dass durch „harte“ Nichtlinearitäten wie die Haftreibung sowie der Lose eine Strukturänderung im Signalflussplan eintritt, im vorliegenden Fall ist der Regelkreis offen.

Es ist ersichtlich, dass eine derartige „physikalische“ Modellierung außerordentlich schwierig sein kann. Aus diesem Grunde werden mit diesem Kapitel beginnend statische Funktionsapproximatoren eingesetzt. Diese Funktionsapproximatoren sind zur Nachbildung ein- und mehrdimensionaler Nichtlinearitäten geeignet, wenn die Nichtlinearitäten entweder überhaupt nicht oder nur sehr schwierig analytisch beschreibbar oder aber nicht unmittelbar zugänglich sind. Damit sind derartige statische Funktionsapproximatoren außerordentlich vorteilhaft, denn es wird mittels Trainingsdaten die Parametrierung erzielt.

Prinzipiell können für die statischen Funktionsapproximatoren Polynome, Spline-Ansätze, Tabellen und auch neuronale Netze, aber auch „Fuzzy-Logik“ verwendet werden. In diesem Kapitel werden wir uns auf die neuronalen Netze konzentrieren.

### 3.1 Übersicht: Neuronale Netze

Statische Neuronale Netze zur Identifikation unbekannter statischer Funktionen existierten bereits vor der ersten Blütezeit der Neuronalen Netze (1955-1969). Das wohl erste statische Neuronale Netz wurde 1943 von W. McCULLOCH und W. PITTS basierend auf den McCulloch-Pitts-Neuronen entwickelt [150]. Bei diesem Neuron werden die Eingangssignale addiert, ein Schwellwert-Gatter erzeugt anschließend ein binäres Ausgangssignal. Eine schematische Darstellung ist in Abbildung 3.1 zu sehen. Beim ursprünglichen McCulloch-Pitts-Neuron war kein Lernmechanismus vorgesehen. Dies war erst der nächsten Generation von neuronalen Modellen vorbehalten, wie beispielsweise dem Perceptron, welches in Abbildung 3.47 eingeführt wird.



**Abb. 3.1:** Schematische Darstellung eines McCulloch-Pitts-Neuron

Die erste Blütezeit der Neuronalen Netze begann mit dem ersten erfolgreichen Neurocomputer (Mark I Perceptron), der in den Jahren 1957 - 1958 am MIT entwickelt wurde [189], und endete 1969 mit M. MINSKYS und S. PAPERTS Buch "Perceptrons" [153].

In den stillen Jahren (1969-1982) fand das Gebiet der Neuronalen Netze kaum Aufmerksamkeit, allerdings wurden in dieser Zeit viele Grundlagen für die neuen Entwicklungen gelegt. So entwickelte P. WERBOS 1974 das Backpropagation-Verfahren [238].

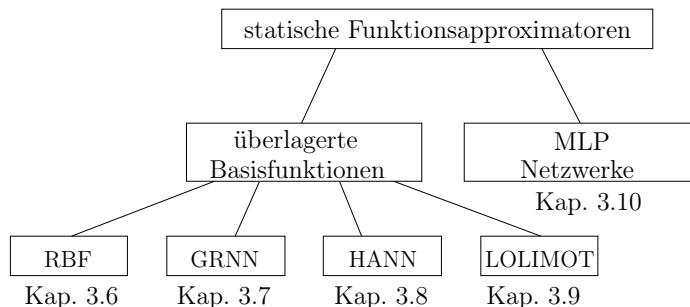
Mit der Wiederentdeckung des Backpropagation Algorithmus durch D. RUMELHART, G. HINTON und R.J. WILLIAMS 1986 begann die Renaissance Neuronaler Netze [190], die bis heute andauert.

Seit Beginn der neunziger Jahre des 20. Jahrhunderts haben sich mehrere statische Neuronale Netze durchgesetzt. Es gibt Netze, die ihre Approximationsfähigkeit durch Überlagerung gewichteter lokaler Basisfunktionen erlangen, wie das Radial-Basis-Function-Netz (RBF) und die beiden daraus abgeleiteten Funktionsapproximatoren General-Regression-Neural-Network (GRNN) und Harmonic-Activated-Neural-Network (HANN). Beim Local-Linear-Model-Tree (LOLIMOT) können mehrere unterschiedliche Basisfunktionen vorhanden sein, welche jeweils über Zugehörigkeitsfunktionen bestimmten Bereichen zugeordnet sind. Die größte Bedeutung haben allerdings die Multi-Layer-Perzeptronen (MLP) Netzwerke erlangt. Sie kommen sowohl in der wissenschaftlichen Literatur als auch in der Praxis mit Abstand am häufigsten zum Einsatz [44]. Eine Übersicht über die in diesem Buch behandelten statischen Funktionsapproximatoren

zeigt Abbildung 3.2. Unterhalb der Neuronalen Netze ist jeweils das zugehörige Buchkapitel angegeben.

Eine interessante Anwendung der statischen Funktionsapproximatoren behandelt das Kapitel 5 des Neuronalen Beobachters: Falls sich die unbekannte Nichtlinearität in einem sonst bekannten linearen System befindet, kann ein Neuronaler Beobachter zum Einsatz kommen. Durch die Implementierung statischer Neuronaler Netze in einen Luenberger-Beobachter [143] gelingt mit der Identifikation eine Modellverfeinerung.

Außer den statischen Neuronalen Netzen existieren noch viele dynamische Neuronale Netze zur Ein-/Ausgangsidentifikation nichtlinearer dynamischer Systeme. Hierbei kann man zwischen Neuronalen Netzen mit interner und externer Dynamik unterscheiden. Zu der Klasse der Systeme mit externer Dynamik gehören das Time-Delay-Neural-Network (TDNN) [168] in den beiden Grundstrukturen Nonlinear-Output-Error (NOE) und Nonlinear-Autoregressiv-with-exogenous-Inputs (NARX) sowie die Ansätze basierend auf der Volterra-Funktionalpotenzreihe in der Nonlinear-Finite-Impulse-Response (NFIR)-Form bzw. in der Nonlinear-Orthonormal-Basis-Functions (NOBF)-Form [93]. Zu der Klasse der Systeme mit interner Dynamik gehören die vollrekurrenten Netze [243] sowie die partiell rekurrenten Netze wie z. B. das Elman-Netz [43]. Eine gute Übersicht über dynamische Neuronale Netze finden man beispielsweise in [44]. Die Identifikation mit dynamischen Neuronalen Netzen behandelt das Kapitel 8.



**Abb. 3.2:** Übersicht über wichtige statische Funktionsapproximatoren — RBF, GRNN, HANN und LOLIMOT überlagern jeweils Basisfunktionen, MLP-Netze bestehen aus vielen abbildenden Schichten

## 3.2 Statische nichtlineare Funktionen

Als Grundlage der nachfolgenden Ausführungen wird zunächst die Klasse der darstellbaren nichtlinearen Funktionen definiert:

**Definition:** Eine stetige, beschränkte und zeitinvariante Funktion  $\mathcal{NL} : \mathbb{R}^N \rightarrow \mathbb{R}$ , die einen  $N$ -dimensionalen Eingangsvektor  $\underline{u}$  auf einen skalaren Ausgangswert  $y$  abbildet, sei eine *Nichtlinearität*  $\mathcal{NL}$  mit  $\underline{u} = [u_1 \ u_2 \ \dots \ u_N]^T$ .

$$y = \mathcal{NL}(\underline{u})$$

Für den Fall mehrdimensionaler Nichtlinearitäten wird entsprechend der Dimension der Ausgangsgröße  $y$  die entsprechende Anzahl skalarer Nichtlinearitäten kaskadiert. Daher soll es im folgenden genügen, lediglich den skalaren Fall zu betrachten.

Für eine klare Darstellung der Eigenschaften und der Anwendung neuronaler Netze werden an dieser Stelle zunächst einige häufig verwendete Begriffe definiert.

Die hier betrachteten neuronalen Netze bestehen aus einer oder mehreren Schichten, die jeweils *Neuronen* enthalten. Der Ausgang der Neuronen ist jeweils mit dem Eingang wenigstens eines Neurons einer nachfolgenden Schicht verbunden. Die *Aktivierung* jedes Neurons wird aus seinen Eingangsgrößen berechnet. Mittels *Gewichte* werden die Verbindungen zwischen den Neuronen skaliert; die Gewichte stellen die variablen Parameter des neuronalen Netzes dar.

Die Begriffe *Lernen*, *Adaption* und *Identifikation* werden im weiteren synonym verwendet und beschreiben unterschiedliche Aspekte der Anwendung neuronaler Netze. Dabei kann der Schwerpunkt auf die Analogie der neuronalen Netze zu ihren biologischen Vorbildern gelegt werden oder auf ihre technische Funktion als Algorithmen zur Nachbildung funktionaler Zusammenhänge.

### 3.3 Methoden der Funktionsapproximation

Zur Approximation einer Nichtlinearität stehen verschiedene Methoden zur Verfügung, die sich jeweils in ihren Einsatzmöglichkeiten und Randbedingungen unterscheiden. Im folgenden werden nach [197] einige Möglichkeiten aufgeführt und anschließend näher erläutert (siehe auch Beispiele in Abb. 3.3):

- Algebraische Darstellung mit Funktionsreihen
- Tabellarische Darstellung mit Stützstellen
  - Interpolation
  - Approximation
- Konnektionistische Darstellung

Bekannte Beispiele für eine **algebraische Darstellung** sind Polynome (wie z.B. die Taylor-Reihe) und Reihenentwicklungen (wie z.B. die Fourier-Reihe). In der Regel hängt die Ausgangsgröße linear von einer endlichen Anzahl an Koeffizienten ab. Nachteilig für den Einsatz in adaptiven Verfahren erweist sich dabei

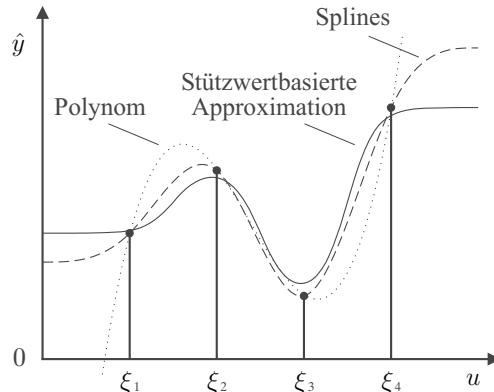


Abb. 3.3: Beispiele zur Funktionsapproximation

meist, dass jeder Koeffizient auf weite Bereiche des Eingangsraums wirkt, also keine lokale Zuordnung zu bestimmten Eingangswerten möglich ist.

Eine **tabellarische Darstellung** kommt häufig zum Einsatz, wenn eine Funktion bereits rasterförmig vermessen vorliegt. Zwischen diesen meist in einer Tabelle (*Look-Up Table*) abgelegten Messwerten wird dann geeignet interpoliert bzw. approximiert. Somit wirken alle Parameter lokal und nachvollziehbar. Ein Sonderfall der Interpolation sind dabei z.B. Splines. Diese stellen eine Zwischenform dar, in der durch die Messwerte eine globale tabellarische Repräsentation vorliegt, die aber lokal durch den algebraischen Zusammenhang der Splines ausgewertet wird. Da eine durch Interpolation nachgebildete Funktion alle Messwerte enthält, wirken sich Messfehler empfindlich aus.

Um diesen Einfluss zu verringern, kann statt der Interpolation auch eine sogenannte **stützwertbasierte Approximation** erfolgen. Im Gegensatz zur Interpolation stehen bei einer Approximation nicht genügend freie Parameter zur Verfügung, um alle Randbedingungen (z.B. in Form vorgegebener Messwerte) zu erfüllen. Stattdessen wird die Abweichung der approximierten Funktion von diesen Vorgaben, der sogenannte Approximationsfehler, minimiert. Beispiele für stützwertbasierte Approximation sind die im weiteren Verlauf behandelten neuronalen **RBF-Netze**. Die dort verwendeten Stützwerte besitzen zwar einen quantitativen Zusammenhang mit dem Wert der Funktion für den zugehörigen Eingangswert, sind selbst aber nicht notwendigerweise Teil des approximierten Funktionsverlaufs. Durch die glättende Wirkung der Approximation können Störungen und Messfehler wirkungsvoll heraus gefiltert werden. Gleichzeitig erlaubt die lokale Zuordnung der Stützwerte zu Bereichen des Eingangsraums eine lokale und schnelle Adaption. In der Regel ist damit auch die Eindeutigkeit der adaptierten Parameter und somit eine Parameterkonvergenz verbunden, die wesentlich für eine Interpretierbarkeit der adaptierten Funktion ist.

Mehrschichtige neuronale Netze, wie z.B. das **Multi Layer Perceptron (MLP) Netz**, gehören zur Gruppe mit **konnektionistischer Darstellung** und können auch Funktionen mit einer Eingangsgröße hoher Dimension nachbilden. Eine Deutung der Parameter ist in aller Regel nicht möglich.

### 3.4 Kriterien zur Beurteilung künstlicher neuronaler Netze

Zur Beurteilung künstlicher neuronaler Netze müssen zunächst einige Kriterien definiert werden, die für die theoretischen Analysen und den praktischen Einsatz dieser Netze von Bedeutung sind. Die nachfolgend aufgeführten Kriterien beziehen sich nicht nur auf den Einsatz künstlicher neuronaler Netze zur Approximation statischer nichtlinearer Funktionen, sondern auch auf den Einsatz zur Identifikation nichtlinearer dynamischer Systeme. Im einzelnen sind folgende Punkte von Bedeutung [197, 91]:

- **Repräsentationsfähigkeit**

Repräsentationsfähigkeit bedeutet, dass der gewählte Netztyp mit der entsprechenden Netztopologie in der Lage ist, die Abbildungsaufgabe zu erfüllen.

- **Lernfähigkeit**

Der Vorgang, bei dem die Parameter eines Netzes so bestimmt werden, dass es in der Reproduktionsphase das gewünschte Verhalten aufweist, wird als Lernen bezeichnet. Lernfähigkeit bedeutet, dass die gewünschte Abbildung mittels eines geeigneten Lernverfahrens in das künstliche neuronale Netz eintrainiert werden kann. In der Praxis spielen dabei Kriterien, wie Lerngeschwindigkeit, Konvergenz und Eindeutigkeit der Parameter, Stabilität und Online-Fähigkeit des Lernverfahrens, eine wichtige Rolle.

- **Verallgemeinerungsfähigkeit**

Die Verallgemeinerungsfähigkeit beschreibt die Fähigkeit des trainierten Netzes 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 Eingangsbereichs der Trainingsdaten charakterisieren.

Das vorrangige Ziel ist es, eine gute Verallgemeinerungsfähigkeit zu erreichen. Die Überprüfung der Verallgemeinerungsfähigkeit darf 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) eintritt, d.h. das künstliche neuronale Netz 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.

### 3.5 Funktionsapproximation mit lokalen Basisfunktionen

Beim Einsatz neuronaler Netze für Regelungstechnische Aufgaben werden in der Regel kurze Adaptionzeiten sowie eine nachweisbare Stabilität und Konvergenz gefordert. Dies wird am besten von neuronalen Netzen erfüllt, die nach der Methode der stützwertbasierten Approximation arbeiten. Ein wesentliches Merkmal dieser Netze ist die Verwendung lokaler Basisfunktionen, die hier wie folgt definiert werden:

**Definition:** Eine zusammenhängende begrenzte und nicht-negative Funktion  $\mathcal{B} : \mathbb{R}^P \rightarrow \mathbb{R}_{0+}$  sei eine lokale Basisfunktion, wenn sie ein globales Maximum bei  $\underline{u} = \xi$  besitzt und wenn für alle Elemente  $u_i$  des  $N$ -dimensionalen Eingangsvektors  $\underline{u}$  gilt

$$\frac{\partial \mathcal{B}(\underline{u})}{\partial u_i} \quad \begin{cases} \geq 0 & \text{für } u_i < \xi_i \\ \leq 0 & \text{für } u_i > \xi_i \end{cases}$$

und

$$\lim_{\|\underline{u}\| \rightarrow \infty} \mathcal{B}(\underline{u}) = 0$$

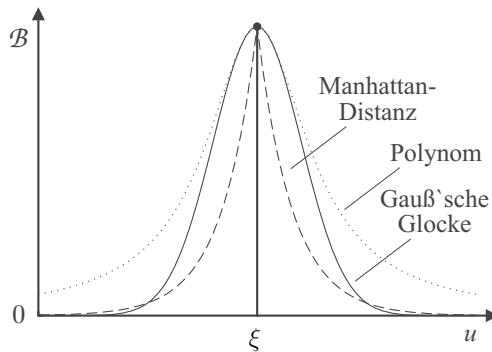


Abb. 3.4: Beispiele lokaler Basisfunktionen

Beispiele lokaler Basisfunktionen sind die Gauß'sche Glockenkurve, die Manhattan-Distanz, Polynome wie z.B.  $1/(1+u^2)$  oder ein Dreieckfenster; dabei

bezeichnet  $\underline{\xi}$  das Zentrum der Basisfunktion (siehe Abb. 3.4). In [20] wird sogar das Rechteckfenster zu den Basisfunktionen gezählt. Ebenso können auch die im Rahmen der Fuzzy-Logik verwendeten Aktivierungsgrade formal als Basisfunktionen behandelt werden.

Um die oben definierten Basisfunktionen zur Funktionsapproximation mit neuronalen Netzen einsetzen zu können, müssen die universelle Einsetzbarkeit zur Nachbildung *beliebiger* Funktionen und eine konvergente Adaption gewährleistet sein.

Im Zusammenhang neuronaler Netze werden lokale Basisfunktionen als Aktivierungsfunktionen der Neuronen bzw. ihrer Gewichte eingesetzt. Eine Aktivierungsfunktion  $\mathcal{A}_i$  sei allgemein eine lokale Basisfunktion  $\mathcal{B}$  der vektoriellen (mehrdimensionalen) Eingangsgröße  $\underline{u}$  und eines Vektors  $\underline{\xi}_i$ , der die Lage des Zentrums und damit des Maximums der Aktivierungsfunktion im Eingangsraum angibt. Die einzelnen Aktivierungsfunktionen  $\mathcal{A}_i$  werden nun zu einem Vektor  $\underline{\mathcal{A}}$  zusammengefasst

$$\underline{\mathcal{A}}(\underline{u}) = \left[ \mathcal{B}(\underline{u}, \underline{\xi}_1) \mathcal{B}(\underline{u}, \underline{\xi}_2) \dots \mathcal{B}(\underline{u}, \underline{\xi}_p) \right]^T \quad (3.1)$$

Wird nun ein Gewichtsvektor  $\underline{\Theta}$  derselben Länge  $p$  aufgestellt,

$$\underline{\Theta} = [\Theta_1 \ \Theta_2 \ \dots \ \Theta_p]^T \quad (3.2)$$

kann eine nichtlineare Funktion  $\mathcal{NL}$  nach obiger Definition als Skalarprodukt aus Gewichts- und Aktivierungsvektor dargestellt werden. Diese Darstellung erscheint zunächst willkürlich und ohne physikalische Entsprechung gewählt. Sie dient aber im weiteren Verlauf der anschaulichen Darstellung der Adaption neuronaler Netze:

$$\mathcal{NL} : \quad y(\underline{u}) = \underline{\Theta}^T \underline{\mathcal{A}}(\underline{u}) + d(\underline{u}) \quad (3.3)$$

Die Gleichung (3.3) beschreibt eine hochparallele Signalverarbeitung, wobei die Gleichung (3.3) sinngemäß von rechts nach links gelesen werden muß. Dies bedeutet, dass Eingangssignal regt einige der sich im allgemeinen überlappenden Aktivierungsfunktionen  $\underline{\mathcal{A}}(\underline{u})$  an und deren Ausgänge werden mit den zugeordneten Gewichten  $\underline{\Theta}$  bewertet. Diese Interpretation und Schreibweise soll grundsätzlich für alle lokal wirkenden Approximatoren gelten, d.h. sowohl für die RBF-, GRNN-, HANN- Netze als auch für den Lolimot-Ansatz. Insbesondere ist diese Interpretation in Kapitel 5.6 zu beachten.

Die Auswahl der Aktivierungsfunktionen, ihrer Anzahl und Parameter muss dabei so möglich sein, dass der inhärente Approximationsfehler  $d(\underline{u})$  des Verfahrens nach (3.3) eine beliebig kleine Schranke nicht überschreitet. Bei Verwendung von z.B. lokalen Basisfunktionen steht damit ein universeller Funktionsapproximator zur Verfügung.

Die gleiche Darstellung kann nun auch für die durch ein neuronales Netz nachgebildete (d.h. „geschätzte“ bzw. gelernte) Funktion  $\widehat{\mathcal{NL}}$  verwendet werden.

$$\widehat{\mathcal{N}} : \quad \hat{y}(\underline{u}) = \hat{\Theta}^T \mathcal{A}(\underline{u}) \quad (3.4)$$

Dabei wird angenommen, dass der Vektor  $\mathcal{A}$  der Aktivierungsfunktionen mit dem Aktivierungsvektor bei der oben eingeführten Darstellung der betrachteten Nichtlinearität identisch ist, d.h. daß  $\underline{u}$  messbar ist und verzögerungsfrei vorliegt; dies soll auch für  $y(\underline{u})$  gelten. Damit kann ein Adaptionsfehler  $e(\underline{u})$  eingeführt werden, der im weiteren auch als Lernfehler bezeichnet wird.

$$e(\underline{u}) = \hat{y}(\underline{u}) - y(\underline{u}) = \hat{\Theta}^T \mathcal{A}(\underline{u}) - \underline{\Theta}^T \mathcal{A}(\underline{u}) = (\hat{\Theta}^T - \underline{\Theta}^T) \mathcal{A}(\underline{u}) \quad (3.5)$$

Die Aufgabe der Adaption lässt sich so auf eine Anpassung der Gewichte reduzieren. Optimale Adaption bedeutet dann, dass der Gewichtsvektor  $\hat{\Theta}$  des neuronalen Netzes gleich dem Gewichtsvektor  $\underline{\Theta}$  der Nichtlinearität ist. Dies ist gleichbedeutend mit einem Verschwinden des Parameterfehlers  $\underline{\Phi}$ , der bei optimaler Adaption zu Null wird.

$$\underline{\Phi} = \hat{\Theta} - \underline{\Theta} \quad (3.6)$$

Der Ausgangsfehler kann nun dargestellt werden als

$$e(\underline{u}) = \underline{\Phi}^T \mathcal{A}(\underline{u}) = \mathcal{A}(\underline{u})^T \underline{\Phi} \quad (3.7)$$

Bei ausreichender Variation des Eingangswertes  $\underline{u}$  ermöglicht die lokale Wirksamkeit der Aktivierungsfunktionen im Eingangsraum die Eindeutigkeit der Adaption. Für Gauß'sche Radiale Basisfunktionen wurde die Eindeutigkeit der Funktionsdarstellung in [130] nachgewiesen.

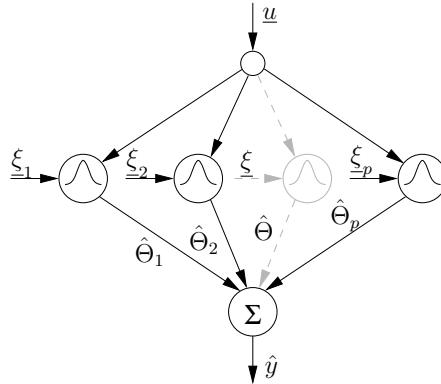
## 3.6 Radial Basis Function (RBF) Netz

Viele in der Regelungstechnik eingesetzten neuronalen Netze gehören zur Familie der Radial Basis Function (RBF) Netze. Im Gegensatz zu anderen neuronalen Ansätzen, wie beispielsweise die Multi Layer Perceptron Netze, weisen sie eine feste und lokale Zuordnung der einzelnen Neuronen zu Bereichen des Eingangsraums auf. Dies ermöglicht insbesondere eine physikalische Interpretierbarkeit der adaptierten Gewichte, die z.B. der Diagnose eines Prozesses dienen kann.

Im folgenden werden die Gewichte der Neuronen, ihre Aktivierungsfunktionen und die zugehörigen Zentren auch unter dem Begriff *Stützwerte* zusammengefasst. Dabei bezeichnet der *Wert* eines Stützwertes das zugeordnete Gewicht  $\Theta_i$  und die *Lage* (oder auch *Koordinate*) eines Stützwertes das Zentrum  $\xi_i$  der zuständigen Aktivierungsfunktion.

Der Ausgang  $\hat{y}$  eines RBF-Netzes mit  $p$  Neuronen kann als gewichtete Summe der Aktivierungsfunktionen gebildet werden, wenn das Skalarprodukt aus Gl. (3.4) in Summendarstellung übergeführt wird.

$$\hat{y}(\underline{u}) = \sum_{i=1}^p \hat{\Theta}_i \mathcal{A}_i(\underline{u}) \quad (3.8)$$



**Abb. 3.5:** Struktur des RBF-Netzes mit  $p$  Stützstellen

Üblicherweise werden als Aktivierungsfunktionen Gauß'sche Glockenkurven verwendet, deren Darstellung an die der Standardverteilung mit der Varianz  $\sigma^2$  angeglichen ist. [218]

$$\mathcal{A}_i = \exp\left(-\frac{\mathcal{C}_i}{2\sigma^2}\right) \quad (3.9)$$

Hier bezeichnet  $\sigma$  einen Glättungsfaktor, der den Grad der Überlappung zwischen benachbarten Aktivierungen bestimmt, und  $\mathcal{C}_i$  das Abstandsquadrat des Eingangsvektors vom  $i$ -ten Stützwert, d.h. vom Zentrum  $\underline{\xi}_i$  der zugehörigen Aktivierungsfunktion.

$$\mathcal{C}_i = \|\underline{u} - \underline{\xi}_i\|^2 = (\underline{u} - \underline{\xi}_i)^T (\underline{u} - \underline{\xi}_i) = \sum_{k=1}^N (u_k - \xi_{ik})^2 \quad (3.10)$$

$N$  gibt die Dimension des Eingangsvektors  $\underline{u}$  an (ein Beispiel für einen mehrdimensionalen Eingangsräum ist in Abb. 3.17 zu sehen). Damit kann die Struktur eines RBF-Netzes auch graphisch umgesetzt werden, wie in Abb. 3.5 gezeigt.

Im Allgemeinen müssen die Stützstellen  $\xi_i$  nicht gleichmäßig verteilt sein. Bei den in diesem Beitrag eingesetzten Funktionsapproximatoren mit lokalen Basisfunktionen sind die Stützstellen im Eingangsbereich jedoch äquidistant verteilt. Dadurch ist der Abstand zwischen zwei Stützstellen  $\Delta\xi$  konstant. Wird der Eingangsbereich, der z. B. im Fall einer Reibungskennlinie durch die maximal zulässigen Drehzahlen gegeben ist, allgemein durch  $u_{max}$  und  $u_{min}$  beschrieben, berechnen sich die Stützstellen zu

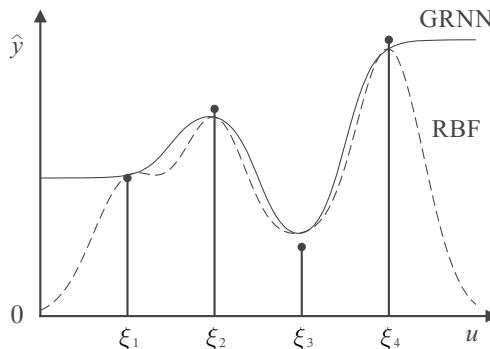
$$\xi_i = u_{min} + (i - 1) \cdot \frac{u_{max} - u_{min}}{p - 1}$$

Damit muss für die Anzahl  $p$  der Stützstellen  $p \geq 2$  gelten.

Zur besseren Vergleichbarkeit wird der Glättungsfaktor  $\sigma$  auf den Abstand zwischen zwei Stützstellen normiert. Damit gilt  $\sigma_{norm} = \frac{\sigma}{\Delta\xi}$ . Mit dem normierten Glättungsfaktor ergibt sich die mathematische Beschreibung des RBF-Netzes zu

$$\hat{y} = \sum_{i=1}^p \hat{\Theta}_i \cdot \mathcal{A}_i(\underline{u}) \quad \text{mit} \quad \mathcal{A}_i(\underline{u}) = e^{-\frac{\|\underline{u} - \xi_i\|^2}{2 \cdot \sigma_{norm}^2 \cdot \Delta\xi^2}} \quad (3.11)$$

Das erzielte Approximationsverhalten ist in Abb. 3.6 an einem Beispiel dargestellt. Dabei fällt allerdings die ungünstige Approximation zwischen den Stützwerten und die ungünstige Extrapolation dieses Netzes aufgrund der fehlenden Monotonie-Erhaltung auf [193]. Dadurch kann der Wert der approximierten Funktion zwischen den Zentren zweier Aktivierungsfunktionen (z.B.  $\xi_1$  und  $\xi_2$ ) auch außerhalb der durch ihre Gewichte begrenzten Bereich liegen, d.h. eine Monotonie der Gewichte bedingt nicht notwendigerweise auch einen monotonen Verlauf der approximierten Funktion. Ein weiterer Punkt ist das ungünstige Extrapolationsverhalten außerhalb des Stützwertebereichs. Hier sinkt der Ausgang des Netzes auf Null ab (Abb. 3.6). Da aber die Erhaltung von Monotonie und ein sinnvolles Extrapolationsverhalten der gelernten Kennlinie wesentliche Forderungen in Regelungstechnischen Anwendungen ist, führt dies zu einer Modifikation des RBF-Netzes.



**Abb. 3.6:** Vergleich der Approximation von RBF-Netz und GRNN

Die Funktionsapproximation bei mehrdimensionalem Eingangsraum behandelt das Kapitel 3.7.2. Identifikationsbeispiele zum RBF-Netz sind im Kapitel 3.7.3 zu finden.

### 3.7 General-Regression-Neural-Network (GRNN)

Das General Regression Neural Network (GRNN) stellt eine Weiterentwicklung des RBF-Netzes aus den oben genannten Gründen dar. Der wesentliche Unterschied besteht in einer Normierung aller Aktivierungsfunktionen auf deren Summe, wie in Abb. 3.7 dargestellt. Als Abstandsfunktion  $\mathcal{C}_n$  kommt dabei das Abstandsquadrat nach Gl. (3.10) zum Einsatz. Die Aktivierungsfunktionen ergeben sich damit zu

$$\mathcal{A}_i = \frac{\exp\left(-\frac{\mathcal{C}_i}{2\sigma^2}\right)}{\sum_{k=1}^p \exp\left(-\frac{\mathcal{C}_k}{2\sigma^2}\right)} \quad (3.12)$$

Damit gilt

$$\sum_{k=1}^p \mathcal{A}_k = 1 \quad (3.13)$$

Durch die Normierung wird sichergestellt, dass der Wert der approximierten Funktion stets innerhalb der durch den Wert der angrenzenden Stützwerte gegebenen Grenzen verläuft und gleichzeitig eine Monotonie der Stützwerte auch einen monotonen Verlauf der approximierten Funktion bewirkt. Diese Eigenschaft führt insbesondere auch zu einer verbesserten Extrapolation, bei der die approximierte Funktion dem jeweils nächstliegenden – und damit wahrscheinlichsten – Stützwert asymptotisch zustrebt.

Zur besseren Vergleichbarkeit unterschiedlich parametrierter GRNN wird ebenfalls ein normierter Glättungsfaktor  $\sigma_{norm}$  eingeführt, der auf den kleinsten bzw. den äquidistanten Abstand  $\Delta\xi$  zweier Stützwerte normiert ist.

$$\sigma_{norm} = \frac{\sigma}{\Delta\xi} \quad (3.14)$$

Die mathematische Beschreibung des GRNN mit der Eingangsdimension  $N = 1$  lautet

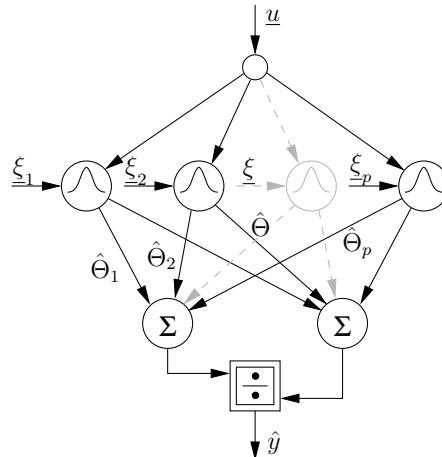
$$\hat{y} = \sum_{i=1}^p \hat{\Theta}_i \cdot \mathcal{A}_i(u) \quad \text{mit} \quad \mathcal{A}_i(u) = \frac{e^{-\frac{\mathcal{C}_i}{2\cdot\sigma_{norm}^2\cdot\Delta\xi^2}}}{\sum_{k=1}^p e^{-\frac{\mathcal{C}_k}{2\cdot\sigma_{norm}^2\cdot\Delta\xi^2}}} \quad (3.15)$$

In kompakter vektorieller Schreibweise lässt sich Gleichung (3.15) darstellen als

$$\hat{y} = \hat{\Theta}^T \cdot \underline{\mathcal{A}}(u) \quad (3.16)$$

mit

$$\hat{\Theta} = [ \hat{\Theta}_1 \ \dots \ \hat{\Theta}_p ]^T \quad \text{und} \quad \underline{\mathcal{A}}(u) = [ \mathcal{A}_1(u) \ \dots \ \mathcal{A}_p(u) ]^T$$



**Abb. 3.7:** Struktur des GRNN mit  $p$  Stützstellen

### 3.7.1 Ursprüngliche Anwendung des GRNN

Die Beschreibung des GRNN geht auf die Arbeit von Specht [218] zurück. Beim GRNN von Specht ist eine Gewichtsadaption nicht vorgesehen. Vielmehr gilt es, durch geeignete Messdaten eine Funktion anzunähern. Für jedes gemessene Ein-/Ausgangspaar kann direkt ein Neuron im GRNN angelegt werden, wobei der Eingangsvektor das Zentrum der Basisfunktion und der gemessene Ausgangsvektor das entsprechende Netzwicht bestimmt. In der Literatur spricht man meist vom normierten RBF-Netz (NRBF-Netz, Normalized Radial Basis Function Network), falls die Gewichte — wie hier bei der statischen Funktionsapproximation — trainiert werden. Da die grundsätzliche mathematische Beschreibung und auch die Approximationseigenschaft beim NRBF und beim GRNN übereinstimmen, soll in diesem Buch ausschließlich die Bezeichnung GRNN Verwendung finden, auch wenn eine Gewichtsadaption erfolgt.

### 3.7.2 Optimierung bei mehrdimensionalem Eingangsraum

Beim Einsatz eines RBF-Netzes bzw. eines GRNN mit einem Eingangsraum der Dimension  $N \geq 2$  kann die verwendete Exponentialfunktion der Aktivierung vorteilhaft faktorisiert werden, um Rechenzeit einzusparen. Dazu müssen die einzelnen Aktivierungsfunktionen  $\mathcal{A}_i$  mit Ausnahme ihrer Zentren  $\underline{\xi}_i$  identisch sein und ihre Zentren in einem gleichförmigen mehrdimensionalen Gitter über den Eingangsraum verteilt sein.

Allgemein kann die zur Berechnung der Aktivierung eines Neurons benötigte Abstandsfunktion  $C_i$  als Summe der Abstandsfunktionen  $c_{ik}$  der einzelnen Kom-

ponenten des Eingangsvektors bestimmt werden. (vgl. Gl. (3.9) und (3.10))

$$\mathcal{C}_i = \sum_{k=1}^N c_{ik} = \sum_{k=1}^N f(u_k - \xi_{ik}) \quad (3.17)$$

Da die Aktivierungsfunktion — beispielsweise wie beim RBF — auf alle möglichen Kombinationen der Summanden angewandt werden muss, wächst so die Zahl der notwendigen Rechenoperationen exponentiell mit der Dimension des Eingangsvektors. Um dies zu umgehen, kann die bei der Aktivierung verwendete Exponentialfunktion wie folgt zerlegt werden.

$$\begin{aligned} \mathcal{A}_i &= \exp\left(-\frac{\mathcal{C}_i}{2\sigma^2}\right) = \exp\left(-\frac{c_{1i} + c_{2i} + \cdots + c_{Ni}}{2\sigma^2}\right) \\ &= \underbrace{\exp\left(-\frac{c_{1i}}{2\sigma^2}\right)}_{\mathcal{A}_{1i}} \underbrace{\exp\left(-\frac{c_{2i}}{2\sigma^2}\right)}_{\mathcal{A}_{2i}} \cdots \underbrace{\exp\left(-\frac{c_{Ni}}{2\sigma^2}\right)}_{\mathcal{A}_{Ni}} \end{aligned} \quad (3.18)$$

Damit kann die Aktivierungsfunktion  $\mathcal{A}_i$  als Produkt von Teil-Aktivierungsfunktionen  $\mathcal{A}_{ki}$  berechnet werden, die jeweils einer Dimension des Eingangsvektors zugeordnet sind. Bei jedem Auswertungsschritt genügt es dann, diese Teil-Aktivierungsfunktionen entlang jeder Koordinatenachse des oben genannten Gitters einmal zu bestimmen und zu speichern. Die Aktivierung der Neuronen wird anschließend durch Multiplikation der zuständigen Teil-Aktivierungen gebildet.

Diese Vereinfachung ist auch im normierten Fall des GRNN anwendbar. Die Herleitung wird der Anschaulichkeit halber lediglich für zwei Eingangsdimensionen und einem Neuronen-Gitter mit  $p_x$  „Zeilen“ und  $p_y$  „Spalten“ durchgeführt. Das GRNN enthalte  $p_x \cdot p_y = p$  Neuronen. Die Aktivierungsfunktionen lassen sich dann folgendermaßen aufspalten:

$$\begin{aligned} \mathcal{A}_i &= \frac{\exp\left(-\frac{\mathcal{C}_i}{2\sigma^2}\right)}{\sum_{k=1}^p \exp\left(-\frac{\mathcal{C}_k}{2\sigma^2}\right)} = \frac{\exp\left(-\frac{c_{1i} + c_{2i}}{2\sigma^2}\right)}{\sum_{x=1}^{p_x} \sum_{y=1}^{p_y} \exp\left(-\frac{c_{1x}}{2\sigma^2}\right) \exp\left(-\frac{c_{2y}}{2\sigma^2}\right)} \\ &= \underbrace{\frac{\exp\left(-\frac{c_{1i}}{2\sigma^2}\right)}{\sum_{x=1}^{p_x} \exp\left(-\frac{c_{1x}}{2\sigma^2}\right)}}_{\mathcal{A}_{1i}} \cdot \underbrace{\frac{\exp\left(-\frac{c_{2i}}{2\sigma^2}\right)}{\sum_{y=1}^{p_y} \exp\left(-\frac{c_{2y}}{2\sigma^2}\right)}}_{\mathcal{A}_{2i}} \end{aligned} \quad (3.19)$$

Die dargestellten Zusammenhänge gelten bei Eingangsgrößen höherer Dimension analog. Da durch die vorgenommene Aufspaltung separate Glättungsfaktoren für jede Dimension verwendet werden, können diese bei Bedarf auch unterschiedlich vorgegeben werden, z.B. um die Skalierung der Eingangsgrößen anzulegen.

### 3.7.3 Beispiele

In diesem Abschnitt sollen die Eigenschaften des RBF- und GRNN Netzes zur Approximation einer ein- und zweidimensionalen Nichtlinearität untersucht werden. Insbesondere soll das Interpolationsverhalten bei unterschiedlichen Stützwertezahlen und das Extrapolationsverhalten bei Überschreitung der Bereichsgrenzen dargestellt werden. Ferner wird der erforderliche Rechenaufwand untersucht.

#### Approximation einer eindimensionalen statischen Nichtlinearität

Als Nichtlinearität wird folgende statische Kennlinie verwendet:

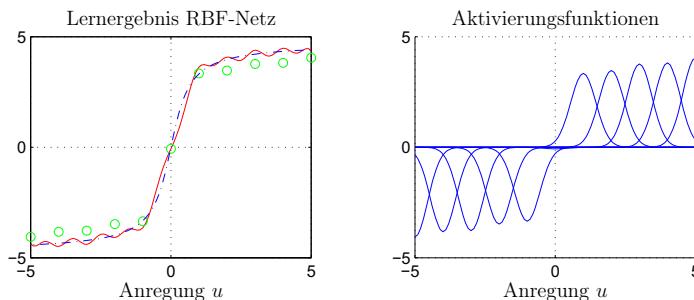
$$\mathcal{NL}(u) = 3 \cdot \arctan(2 \cdot u) \quad (3.20)$$

Als Lerngesetz für das RBF-Netz sowie für das daraus abgeleitete GRNN wird an dieser Stelle das Gradientenabstiegsverfahren verwendet, welches in Kapitel 4 (vgl. Gleichung (4.12)) noch ausführlich beschrieben wird:

$$\frac{d}{dt} \hat{\Theta} = -\eta e \mathcal{A} \quad (3.21)$$

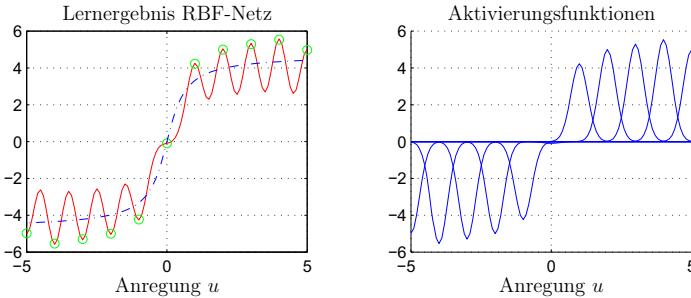
Das Training erfolgt solange, bis keine Verbesserung des Ergebnisses mehr eintritt, d.h. es ist davon auszugehen, dass eine Verlängerung der Lernzeit keine Ergebnisverbesserung bewirkt.

Zunächst wird die Nichtlinearität mit einem RBF-Netzwerk gemäß Gleichung (3.9) approximiert. Die Stützwertezahl ist wie im Fall des GRNN gleich 11. Abbildung 3.8 zeigt die fertig gelernte Kennlinie durch das RBF-Netzwerk bei einem Glättungsfaktor  $\sigma_{norm} = 0.45$ . Es ist bereits eine deutlich erhöhte



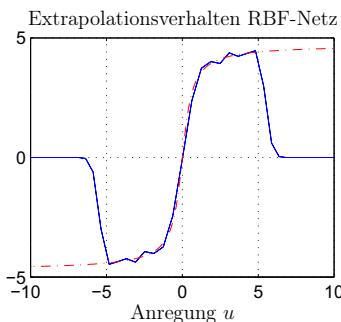
**Abb. 3.8:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (o) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $p = 11$  Stützwerte und  $\sigma_{norm} = 0.45$

Welligkeit der interpolierten Kurve zu erkennen. Eine weitere Charakteristik von



**Abb. 3.9:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (o) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $p = 11$  Stützwerte und  $\sigma_{norm} = 0.3$

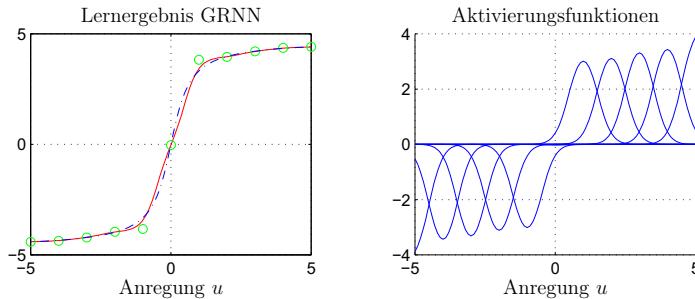
RBF–Netzwerken ist, dass die Stützstellen normalerweise nicht auf der approximierten Kurve liegen. Erniedrigt man den Glättungsfaktor auf  $\sigma_{norm} = 0.3$  so verstärkt sich die angesprochene Welligkeit noch mehr (Abb. 3.9). Die Welligkeit kann entweder durch erhöhte Stützwertezahl (höherer Rechenaufwand) oder größeren Glättungsfaktor  $\sigma$  reduziert werden. Eine Vergrößerung von  $\sigma$  verstärkt jedoch die Tendenz zur Mittelwertbildung durch die Approximationskurve. Eine weitere unerwünschte Eigenschaft für Regelungstechnische Anwendungen ist das Extrapolationsverhalten außerhalb des gelernten Bereichs. Außerhalb des Trainingsbereichs fällt der Ausgang des RBF–Netzwerks auf Null ab. Dieses Verhalten ist in Abb. 3.10 dargestellt.



**Abb. 3.10:** Extrapolationsverhalten des RBF–Netzwerks außerhalb des Eingangsbereichs für  $u < -5$  und  $u > 5$  (.- Vorgabe, - Netzausgabe)

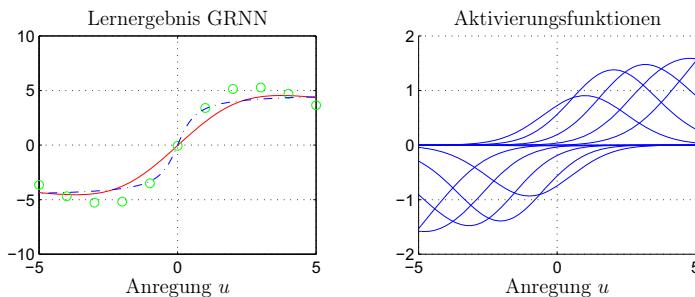
Nun wird die selbe Nichtlinearität mit einem GRNN–Netzwerk mit 11 Stützwerten und einem normierten Glättungsfaktor  $\sigma_{norm} = 0.5$  approximiert. In Abb. 3.11 links erkennt man, dass zwischen Funktionsvorgabe und Identifikationsergebnis nahezu kein Unterschied erkennbar ist. Die Dimensionierung des

Netzes ist also gut gewählt. Das Interpolationsverhalten des GRNN zwischen



**Abb. 3.11:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (○) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $\sigma_{\text{norm}} = 0.5$

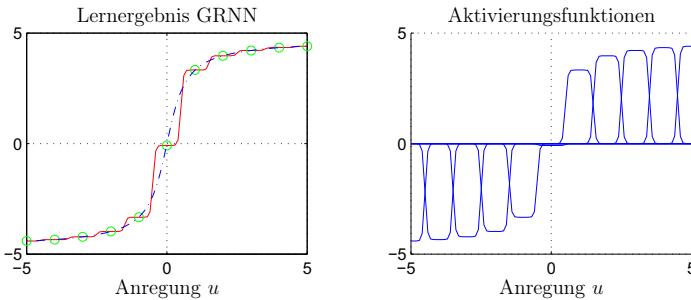
zwei Stützwerten wird durch Gleichung (3.15) bestimmt. Der Glättungsfaktor  $\sigma$  bestimmt die Breite der Aktivierungsfunktionen und damit den Bereich der lokalen Wirkung eines Stützwertes. Eine Erhöhung von  $\sigma$  bedeutet eine Ausweitung der lokalen Stützwertewirkung, so dass für  $\sigma \rightarrow \infty$  schließlich nur noch der Mittelwert der zu approximierenden Funktion nachgebildet werden kann. Abb. 3.12 zeigt das Lernergebnis und die Aktivierungsfunktionen für 11 Stützwerte und  $\sigma_{\text{norm}} = 1.5$ . Die stärkere Überlappung der einzelnen Aktivierungsfunktionen



**Abb. 3.12:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (○) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $\sigma_{\text{norm}} = 1.5$

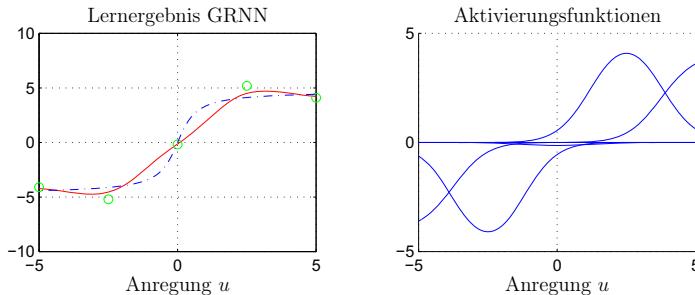
bewirkt eine stärkere Glättung bzw. Mittelwertbildung der gelernten Kennlinie. Im steilen Bereich der arctan-Funktion wird dadurch der Approximationsfehler größer (Abb. 3.12 links).

Eine Verringerung des Glättungsfaktors wirkt sich in einer geringeren Überlappung der lokalen Wirkungsbereiche der jeweiligen Stützstellen aus. In Abb. 3.13 rechts ist fast keine Überlappung mehr vorhanden, so dass sich quasi ein *harte* Umschaltung zwischen den Werten der Stützstellen ergibt.



**Abb. 3.13:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (o) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $\sigma_{\text{norm}} = 0.2$

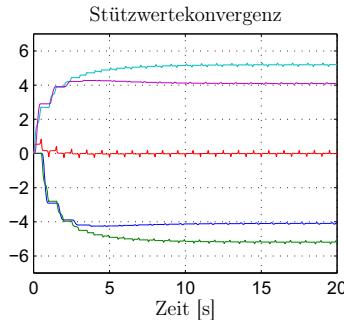
Im folgenden soll nun noch die Auswirkung einer zu geringen Anzahl von Stützstellen untersucht werden. In Bereichen, in denen sich die zu lernende Funktion nur wenig ändert, reichen auch wenige Stützwerte zur Approximation aus. In Bereichen steiler Übergänge (im vorliegenden Beispiel um den Wert  $u = 0$ ) kann die Funktion nur mit verminderter Genauigkeit nachgebildet werden (Abb. 3.14). Eine Steigerung der Approximationsgenauigkeit kann nur durch Erhöhung der Stützwertezahl erreicht werden; durch eine Verlängerung der Lerndauer kann keine Verbesserung erreicht werden. Abb. 3.15 zeigt die Konvergenz der fünf Stütz-



**Abb. 3.14:** Gelernte Kennlinie (-) mit den einzelnen Gewichten (o) und wahre Kennlinie (.-) (links) und Aktivierungsfunktionen (rechts) für  $p = 5$  Stützwerte und  $\sigma_{\text{norm}} = 0.5$

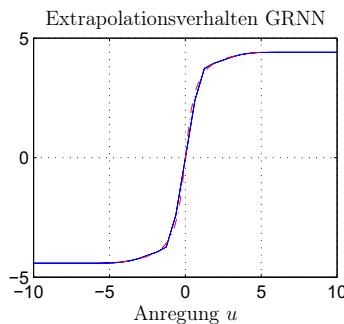
werte im Zeitbereich. Nach etwa zehn Sekunden ergibt sich keine nennenswerte Veränderung der Stützwerte mehr. Die Interpolation zwischen zwei Stützstellen erfolgt hier durch *sanfte* Überlagerung der beiden nächstgelegenen Stützwerte. Eine *harte* Umschaltung findet wegen ausreichend großem  $\sigma$  nicht statt.

Wenn das trainierte Netzwerk zur Reproduktion der Kennlinie betrieben wird, kann es vorkommen, dass Eingangswerte abgefragt werden, die außerhalb des vorgesehenen Eingangsbereichs liegen. Diese Extrapolationseigenschaften des



**Abb. 3.15:** Stützwertekonvergenz für  $p = 5$  und  $\sigma_{\text{norm}} = 0.5$

GRNN werden an einem vollständig trainierten Netzwerk bei abgeschaltetem Lernen veranschaulicht. Innerhalb des Eingangsbereichs (ist identisch mit dem Trainingsbereich) für  $-5 \leq u \leq 5$  wird die Kennlinie exakt wiedergegeben. Außerhalb des Eingangsbereichs erfolgt eine konstante Interpolation, d.h. der Wert des jeweils äußersten Stützwertes wird konstant gehalten. Für Regelungstechnische Anwendungen ist besonders wichtig, dass kein Abfall der Kennlinie auf Null erfolgt. Dies entspricht normalerweise nicht den physikalischen Tatsachen. Das Extrapolationverhalten des GRNN zeigt Abb. 3.16. Außerhalb des Eingangsbereichs (für  $u < -5$  und  $u > 5$ ) wird der Ausgangswert auf dem Wert des nächstliegenden Stützwertes gehalten.



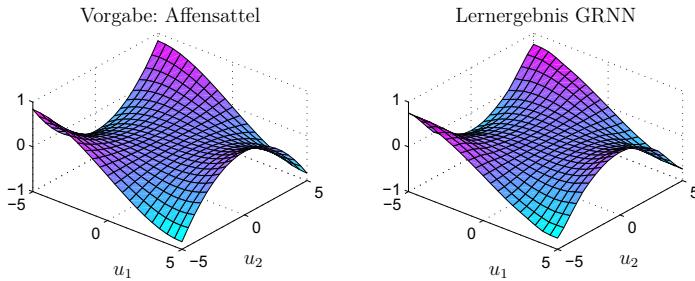
**Abb. 3.16:** Extrapolationsverhalten außerhalb des Eingangsbereichs für  $u < -5$  und  $u > 5$  (.- Vorgabe, - Netzausgabe)

### Approximation einer zweidimensionalen statischen Nichtlinearität

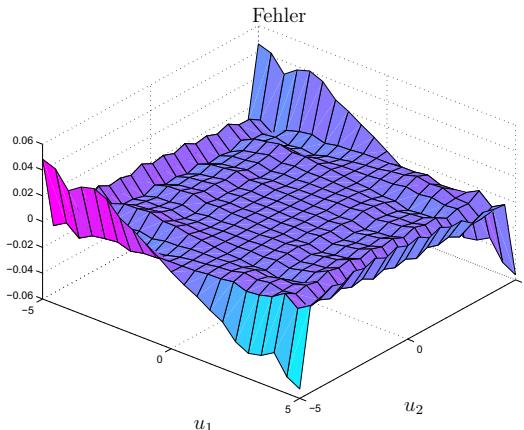
Zur Identifikation einer zweidimensionalen Nichtlinearität soll als Vorgabefunktion der sogenannte Affensattel verwendet werden:

$$\mathcal{NL}(u_1, u_2) = (u_1^3 - 3u_1u_2^2)/300 \quad (3.22)$$

Die Funktion in Gleichung (3.22) soll im Bereich  $-5 \leq u_{1,2} \leq 5$  durch ein GRNN approximiert werden. Dazu wird der Eingangsbereich gleichmäßig über einen Zeitraum von 100 Sekunden abgefahrt. Pro Eingangsdimension wurden 11 Stützwerte verwendet, so dass sich die Gesamtparameterzahl auf 121 beläuft. Das Vorgabekennfeld und die erlernte Nichtlinearität sind in Abb. 3.17 dargestellt. Es ist in allen Bereichen sehr gute Übereinstimmung zu erkennen. Dies zeigt auch die geringe Abweichung zwischen der vorgegebenen Nichtlinearität und der gelernten Funktion in Abbildung 3.18.



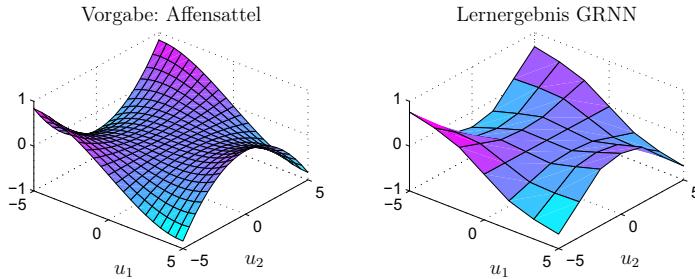
**Abb. 3.17:** Nichtlineare Funktion und Lernergebnis bei  $11 \cdot 11 = 121$  Stützstellen und  $\sigma_{norm} = 0.5$



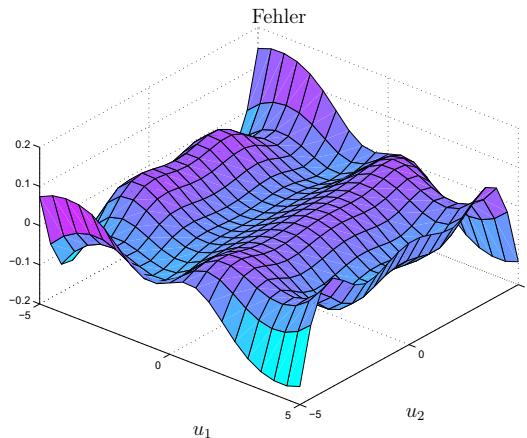
**Abb. 3.18:**  $\mathcal{NL}(u_1, u_2) - \widehat{\mathcal{NL}}(u_1, u_2)$  bei 121 Stützstellen und  $\sigma_{norm} = 0.5$

Es sind grundsätzlich die selben Eigenschaften wie im eindimensionalen Fall vorhanden. Wenn der Eingangsbereich nicht ausreichend angeregt wird, ergeben

sich größere Approximationsfehler in diesen Bereichen. Bei zu geringer Stützstellenanzahl und zu großem  $\sigma$  wird die Kennlinie immer mehr gemittelt. Beispielsweise sei hier der Fall einer zu geringen Anzahl von Stützstellen gezeigt. Die Kennlinie wird dann zwischen den Stützstellen nur noch ungenau angenähert (Abb. 3.19). Dies wird ebenfalls deutlich durch die Abweichung der beiden Oberflächen in Abbildung 3.20.



**Abb. 3.19:** Nichtlineare Funktion und Lernergebnis bei  $4 \cdot 4 = 16$  Stützstellen und  $\sigma_{norm} = 0.5$



**Abb. 3.20:**  $\mathcal{N}\hat{\mathcal{L}}(u_1, u_2) - \widehat{\mathcal{L}}(u_1, u_2)$  bei 16 Stützstellen und  $\sigma_{norm} = 0.5$

In der Praxis muss ein Kompromiss zwischen hoher Stützstellenanzahl und Approximationsgenauigkeit angestrebt werden. Es ist auch der erhöhte Rechenaufwand bei steigender Stützstellenanzahl zu beachten. Die Anzahl  $p$  zu trainierender Gewichte ergibt sich mit der Eingangsdimension  $N$  und der neu definierten Stützwertanzahl  $M$  pro Eingangsdimension zu

$$p = M^N \quad (3.23)$$

Dieser enorme Zuwachs beim Rechenaufwand hat zur Folge, dass das GRNN vorwiegend für Eingangsdimensionen kleiner drei bei heutiger Rechnertechnik Anwendung findet.

### 3.8 Harmonisch Aktiviertes Neuronales Netz (HANN)

Prinzipiell eignet sich das GRNN auch zur Darstellung periodischer Signale. Trotz des insgesamt guten Approximationsverhaltens führt aber die Aufteilung des Eingangsraums auf eine endliche Anzahl lokal wirkender Stützwerte beim GRNN zu einer begrenzten örtlichen Auflösung über dem Eingangsraum.

Für eine überall gleichmäßig hohe Approximationsgenauigkeit wird somit eine große Anzahl an Stützwerten und damit ein hoher Rechenaufwand bei der Adaption und Auswertung des GRNN benötigt. Um die Grenzen des GRNN bezüglich der örtlichen Auflösung und der Effizienz zu überwinden, aber andererseits dessen Vorteile der Echtzeitfähigkeit, Stabilität und Interpretierbarkeit der Parameter weiter zu nutzen, wird im folgenden ein Harmonisch Aktiviertes Neuronales Netz (HANN)[17] vorgestellt.

#### 3.8.1 Grundstruktur

Die betrachtete Nichtlinearität  $\mathcal{NL}$  sei periodisch im Winkel  $\varphi$ ; daher kann sie als endliche, reelle Fourierreihe mit den Fourierreihenkoeffizienten  $a_k$  und  $b_k$  sowie der Ordnung  $k$  dargestellt werden, wobei  $d(\varphi)$  der Restfehler nicht berücksichtigter höherfrequenter Anteile ist. Es werden gerade und ungerade Anteile unterschieden.

$$\mathcal{NL} : \quad y(\varphi) = \frac{a_0}{2} + \underbrace{\sum_{k=1}^K a_k \cos(k\varphi)}_{\text{Gerade Anteile}} + \underbrace{\sum_{k=1}^K b_k \sin(k\varphi)}_{\text{Ungerade Anteile}} + d(\varphi) \quad (3.24)$$

Eine entsprechende Darstellung kann nun auch für die durch ein neuronales Netz nachgebildete („geschätzte“) Funktion  $\widehat{\mathcal{NL}}$  aufgestellt werden, wobei Index  $A$  gerade und Index  $B$  ungerade Spektralanteile bezeichnet.

$$\widehat{\mathcal{NL}} : \quad \hat{y}(\varphi) = \sum_{k=0}^K \hat{\Theta}_{Ak} \mathcal{A}_{Ak}(\varphi) + \sum_{k=1}^K \hat{\Theta}_{Bk} \mathcal{A}_{Bk}(\varphi) \quad (3.25)$$

$$= \hat{\underline{\Theta}}_A^T \underline{\mathcal{A}}_A(\varphi) + \hat{\underline{\Theta}}_B^T \underline{\mathcal{A}}_B(\varphi) \quad (3.26)$$

Unter Vernachlässigung des inhärenten Approximationsfehlers  $d$  können nun die Darstellungen von Gl. (3.24) und (3.25) ineinander übergeführt werden, indem die Koeffizienten  $a_k$  und  $b_k$  zu Vektoren  $\underline{\Theta}_A$  bzw.  $\underline{\Theta}_B$  sowie die Winkelfunktionen

$\cos(k\varphi)$  und  $\sin(k\varphi)$  als Basisfunktionen aufgefasst und ebenfalls zu Vektoren  $\underline{\mathcal{A}}_A$  bzw.  $\underline{\mathcal{A}}_B$  zusammengefasst werden.

Die genannten Vektoren werden nun als Gewichts- und Aktivierungsvektoren eines neuronalen Netzes interpretiert. Der Aktivierungsvektor enthält dabei harmonische Funktionen, weshalb hier die Bezeichnung *Harmonisch Aktiviertes Neuronales Netz (HANN)* verwendet wird.

$$\begin{aligned}\underline{\mathcal{A}}_A^T(\varphi) &= \left[ 1 \ \cos(\varphi) \ \cos(2\varphi) \ \dots \ \cos(K\varphi) \right]^T \\ \underline{\mathcal{A}}_B^T(\varphi) &= \left[ \sin(\varphi) \ \sin(2\varphi) \ \dots \ \sin(K\varphi) \right]^T\end{aligned}\quad (3.27)$$

$$\begin{aligned}\hat{\underline{\Theta}}_A^T &= \left[ \hat{\Theta}_{A0} \ \hat{\Theta}_{A1} \ \hat{\Theta}_{A2} \ \dots \ \hat{\Theta}_{AK} \right]^T \\ \hat{\underline{\Theta}}_B^T &= \left[ \hat{\Theta}_{B1} \ \hat{\Theta}_{B2} \ \dots \ \hat{\Theta}_{BK} \right]^T\end{aligned}\quad (3.28)$$

Damit können Parameterfehler

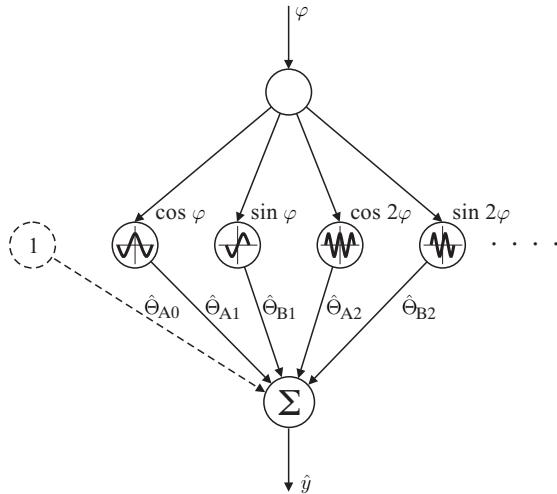
$$\begin{aligned}\underline{\Phi}_A &= \hat{\underline{\Theta}}_A^T - \underline{\Theta}_A^T \\ \underline{\Phi}_B &= \hat{\underline{\Theta}}_B^T - \underline{\Theta}_B^T\end{aligned}\quad (3.29)$$

und ein Adoptionsfehler (bzw. auch Lernfehler)  $e(\varphi)$  eingeführt werden:

$$\begin{aligned}e(\varphi) = \hat{y}(\varphi) - y(\varphi) &= \hat{\underline{\Theta}}_A^T \underline{\mathcal{A}}_A(\varphi) + \hat{\underline{\Theta}}_B^T \underline{\mathcal{A}}_B(\varphi) - \underline{\Theta}_A^T \underline{\mathcal{A}}_A(\varphi) - \underline{\Theta}_B^T \underline{\mathcal{A}}_B(\varphi) \\ &= \underline{\Phi}_A^T \underline{\mathcal{A}}_A(\varphi) + \underline{\Phi}_B^T \underline{\mathcal{A}}_B(\varphi)\end{aligned}\quad (3.30)$$

Aufgrund der formalen Verwandtschaft erlaubt das HANN, eine Struktur der Neuronen ähnlich der eines RBF-Netzes zu verwenden, wie für die Grundstruktur des HANN in Abb. 3.21 gezeigt. Wahlweise kann der Gleichanteil  $\hat{\Theta}_{A0}$  mit der konstanten Aktivierung Eins auch entfallen (gestrichelt gekennzeichnet), sofern er im betrachteten Signal nicht auftritt oder aber bei der Adaption ausgeblendet werden soll.

Im Unterschied zu RBF-Netzen verwendet das HANN in seiner gezeigten Grundform nach der Klassifizierung in Abschnitt 3.3 die Methode der algebraischen Darstellung einer Funktion. Daher können die einzelnen Stützpunkte des HANN nicht mehr einem bestimmten Bereich des Eingangsraums zugeordnet werden. Stattdessen ist eine Zuordnung im Frequenzbereich zu diskreten Frequenzen offensichtlich. Damit lässt das Lernergebnis eine Aussage über die spektrale Zusammensetzung der identifizierten Funktion zu, wodurch eine unmittelbare Diagnose und Überwachung des betrachteten Prozesses möglich wird. Zur Auswertung können Amplitude  $\hat{A}_k$  und Phase  $\hat{\psi}_k$  der gewünschten Frequenzen  $k$  bestimmt werden.



**Abb. 3.21:** Grundstruktur des Harmonisch Aktivierten Neuronalen Netzes

$$\hat{A}_k = \sqrt{\hat{\theta}_{Ak}^2 + \hat{\theta}_{Bk}^2} \quad (3.31)$$

$$\hat{\psi}_k = \arctan \frac{\hat{\theta}_{Bk}}{\hat{\theta}_{Ak}} \quad (3.32)$$

Für die Phasenlage muss dabei zusätzlich der Quadrant anhand der Vorzeichen der Stützwerte ermittelt werden.

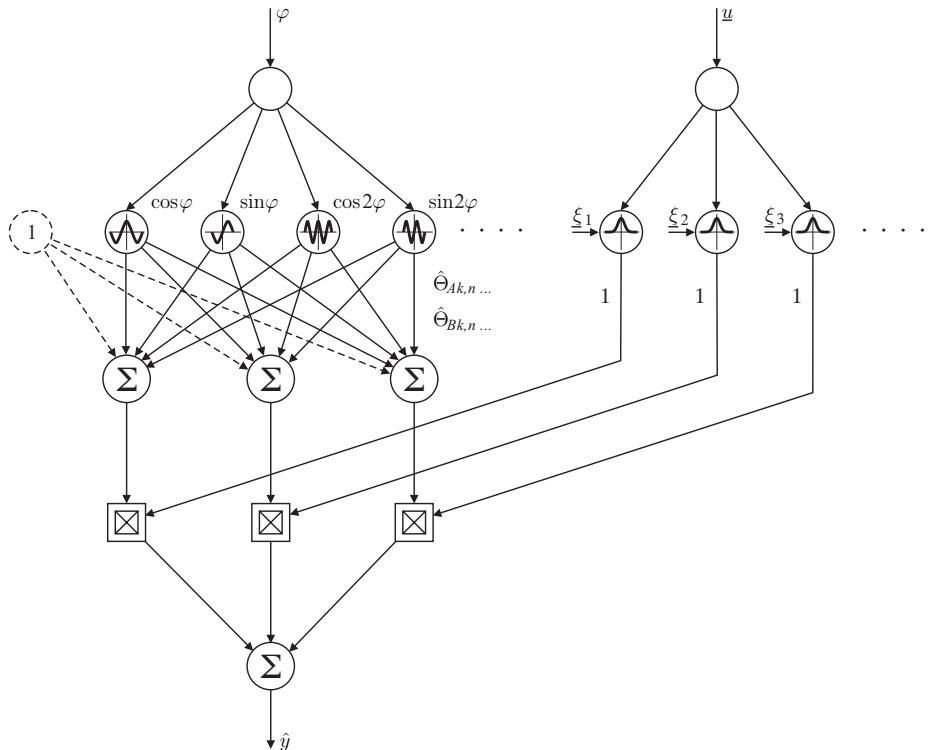
Das HANN wird ebenfalls mit dem in Abschnitt 4 beschriebenen Gradientenabstiegsverfahren gelernt und ist in Abschnitt 4.1.2 dargestellt.

### 3.8.2 Erweiterung

In vielen Anwendungsfällen ist die zu identifizierende Nichtlinearität nicht nur periodischer Natur (d.h. eine Funktion des Winkels  $\varphi$ ) sondern zusätzlich von einer weiteren Eingangsgröße  $\underline{u}$  abhängig (Bsp.: Arbeitspunkte). Damit ist eine Erweiterung des HANN um einen zusätzlichen Eingangsvektor  $\underline{u}$  notwendig. Dabei sollen die bereits nachgewiesenen Eigenschaften der RBF-Netze und der Grundform des HANN, wie Stabilität und Interpretierbarkeit der Parameter, weiterhin genutzt werden können. Daher liegt es nahe, für die Erweiterung des HANN auf die bereits bewährte allgemeine Struktur der RBF-Netze und das zugehörige Lerngesetz zurückzugreifen.

Zur Erläuterung dieser Erweiterung wird zunächst an Abschnitt 3.7.2 (mehrdimensionaler Raum) angeknüpft. Dort wurde die Aktivierung der Stützwerte

in eine multiplikative Überlagerung von Teil-Aktivierungsfunktionen aufgeteilt, die jeweils dem Beitrag einer Eingangsdimension entsprechen. Dieses Verfahren soll hier in umgekehrter Richtung angewandt werden, d.h.  $N$  verschiedene Arbeitspunkte bei gleicher Frequenz entsprechen  $N$  Eingangsdimensionen: Dabei sind sowohl durch die Elemente des Eingangs  $\underline{u}$  die Teil-Aktivierungsfunktionen  $\mathcal{A}_n(\underline{u})$  als auch durch den Winkel  $\varphi$  die zugehörigen Teil-Aktivierungsfunktionen  $\mathcal{A}_{Ak}(\varphi)$  und  $\mathcal{A}_{Bk}(\varphi)$  festgelegt.



**Abb. 3.22:** Erweiterte Struktur des Harmonisch Aktivierten Neuronalen Netzes

Anschließend werden alle möglichen Produkte von Teil-Aktivierungsfunktionen verschiedener Eingangsdimensionen gebildet und je einem Stützwert  $\hat{\Theta}_{Ak,n}$  zugeordnet. Die Stützwerte können dabei in einem mehrdimensionalen Raster angeordnet gedacht werden, von dem eine Dimension dem Winkel  $\varphi$  und die restlichen Dimensionen den Elementen von  $\underline{u}$  entsprechen. Ausgehend von Gl. (3.8) und (3.25) wird der Schätzwert  $\hat{y}(\varphi, \underline{u})$  nun bestimmt zu

$$\hat{y}(\varphi, \underline{u}) = \sum_{i=1}^p \mathcal{A}_i(\underline{u}) \cdot \left( \sum_{k=0}^K \hat{\Theta}_{Ak,i} \mathcal{A}_{Ak}(\varphi) + \sum_{k=1}^K \hat{\Theta}_{Bk,i} \mathcal{A}_{Bk}(\varphi) \right) \quad (3.33)$$

Allgemein können die Teil-Aktivierungsfunktionen  $\mathcal{A}_i(\underline{u})$  wiederum sowohl aus radialen wie auch aus harmonischen Basisfunktionen bestehen. Im Bereich mechatronischer Systeme tritt häufig das Problem auf, last- und drehzahlabhängige Schwingungen zu identifizieren und damit schließlich eine aktive Schwingungsdämpfung zu realisieren. Hierfür eignen sich normierte radiale Basisfunktionen, wie sie beim GRNN verwendet werden, am besten; daher beschränken sich die folgenden Ausführungen auf diesen Fall. Die durch den Eingang  $\underline{u}$  bestimmten Teil-Aktivierungsfunktionen können nach Gl. (3.10) und (3.12) mit

$$\mathcal{A}_i(\underline{u}) = \frac{\exp\left(-\frac{\|\underline{u} - \underline{\xi}_i\|^2}{2\sigma^2}\right)}{\sum_{m=1}^p \exp\left(-\frac{\|\underline{u} - \underline{\xi}_m\|^2}{2\sigma^2}\right)}$$

und den winkelabhängigen Teil-Aktivierungsfunktionen für den behandelten Fall wie folgt festgelegt werden.

$$\begin{aligned} \mathcal{A}_{Ak}(\varphi) &= \cos(k\varphi) & \text{mit} & \quad k \in \mathbb{N}_0 \\ \mathcal{A}_{Bk}(\varphi) &= \sin(k\varphi) & \text{mit} & \quad k \in \mathbb{N} \end{aligned}$$

Abbildung 3.22 zeigt die verwendete Struktur des erweiterten HANN. Dabei ist die Normierung der radialen Basisfunktionen bereits in die Aktivierung der Neuronen in der rechten Hälfte integriert gedacht.

## 3.9 LOLIMOT — LOcal LInear MOdel Tree

In diesem Kapitel soll ein Verfahren vorgestellt werden, das erst seit kurzer Zeit in der Fachliteratur bekannt ist [156, 157, 166, 167]. Der Algorithmus modelliert komplexe nichtlineare Funktionen durch einfachere lokale lineare Modelle, die mittels Zugehörigkeitsfunktionen überlagert werden. Die lokalen linearen Modelle können als Neuronen mit einer linearen Transferfunktion der Steigung Eins (Identitätsfunktion — vgl. Kapitel 3.10) interpretiert werden. Mittels eines Konstruktionsverfahrens wird schrittweise eine Netzstruktur aufgebaut, wobei sowohl die Anzahl der Neuronen als auch deren Gültigkeitsbereich im Eingangsraum optimiert wird.

### 3.9.1 Grundlegende Idee

LOLIMOT ist im Prinzip ein statischer Funktionsapproximator und 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 zugeordnet, welche die Gültigkeit eines Teilmodells innerhalb des Eingangsraumes festlegt, wobei der Übergang zwischen den einzelnen Gültigkeitsbereichen unscharf ist. Als Zugehörigkeitsfunktionen werden in der Regel normierte gaußsche Glockenkurven verwendet. Durch die Normierung ist die Summe der Zugehörigkeitsfunktionen an jeder Stelle des Eingangsraumes gleich Eins. Der Modellausgang  $\hat{y}$  berechnet sich für M Teilmodelle zu

$$\hat{y} = \sum_{i=1}^M (\hat{\Theta}_{0,i} + \hat{\Theta}_{1,i} \cdot u_1 + \dots + \hat{\Theta}_{N,i} \cdot u_N) \cdot \mathcal{A}_i(\underline{u}, \underline{\xi}_i, \underline{\sigma}_i), \quad (3.34)$$

wobei  $u_1 \dots u_N$  die Eingangsgrößen,  $\hat{\Theta}_{0,i} \dots \hat{\Theta}_{N,i}$  die Parameter und  $\mathcal{A}_i$  die Zugehörigkeitsfunktion des i-ten Teilmodells mit den Zentrumskoordinaten  $\underline{\xi}_i$  und den Standardabweichungen  $\underline{\sigma}_i$  sind. Die Zugehörigkeitsfunktionen werden nach folgender Gleichung berechnet:

$$\mathcal{A}_i(\underline{u}, \underline{\xi}_i, \underline{\sigma}_i) = \frac{\mu_i}{\sum_{j=1}^M \mu_j} \quad (3.35)$$

mit

$$\mu_j = \exp \left[ -\frac{1}{2} \left( \frac{(u_1 - \xi_{1,j})^2}{\sigma_{1,j}^2} + \dots + \frac{(u_N - \xi_{N,j})^2}{\sigma_{N,j}^2} \right) \right] \quad (3.36)$$

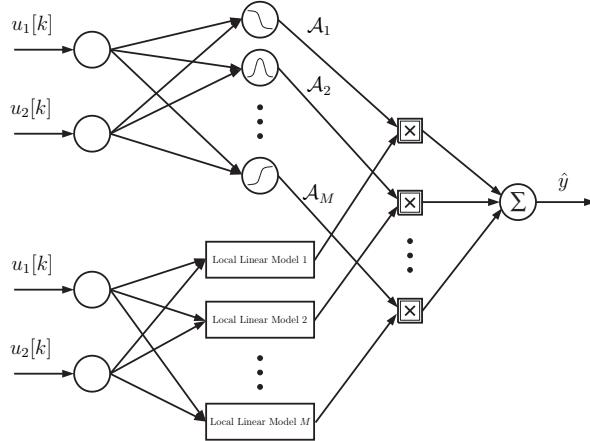
In Abb. 3.23 ist die Netzstruktur zur Approximation einer zweidimensionalen statischen Nichtlinearität dargestellt.

Im eindimensionalen Fall kann man die Teilmodelle als Geraden interpretieren, mittels derer die nichtlineare Funktion angenähert wird. Im zweidimensionalen bzw. N-dimensionalen Eingangsraum können die Teilmodelle als Ebenen bzw. N-dimensionale Hyperebenen angesehen werden. In Abb. 3.24 ist das Prinzip der Überlagerung von Teilmodellen noch einmal veranschaulicht.

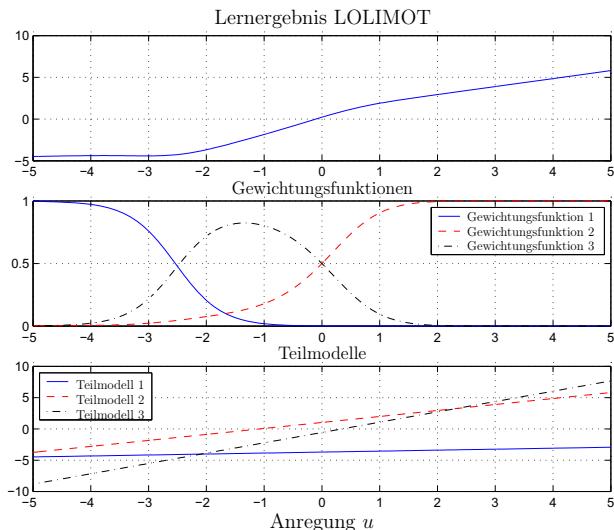
Aus Abb. 3.24 wird deutlich, wie aus den Teilmodellen (Geraden) durch Interpolation mit den Zugehörigkeitsfunktionen eine differenzierbare Funktion entsteht.

### 3.9.2 Parameter- und Strukturoptimierung

Der Algorithmus besteht im Wesentlichen aus zwei Schleifen. In der äußeren Schleife wird die Struktur, d.h. die Bereiche, in denen die Teilmodelle liegen sollen und die zugehörigen Gewichtungsfunktionen, bestimmt. In der inneren Schleife werden die Parameter der einzelnen Teilmodelle berechnet. Im Folgenden soll auf die Parameterberechnung und die Strukturoptimierung genauer eingegangen werden.



**Abb. 3.23:** LOLIMOT – Netzstruktur zur Approximation einer zweidimensionalen statischen Nichtlinearität



**Abb. 3.24:** LOLIMOT – Prinzip der Überlagerung von lokalen, linearen Teilmodellen

### 3.9.2.1 Parameterberechnung

Zunächst soll davon ausgegangen werden, dass die Struktur des Netzes fest vorgegeben ist, d.h. die Zentren  $\underline{c}_i$  und die Standardabweichungen  $\underline{\sigma}_i$  der  $M$  Teilm-

modelle mittels Strukturoptimierung bestimmt wurden (vgl. Kapitel 3.9.2.2). Da der Modellausgang von LOLIMOT linear von den Parametern  $\hat{\underline{\Theta}}_i$  abhängt, kann ein lineares Verfahren zur Berechnung der Parameter angewandt werden. Prinzipiell sind natürlich auch nichtlineare Verfahren, wie das Gradientenabstiegsverfahren anwendbar, allerdings sind die Lernzeiten solcher Verfahren deutlich länger, und sie führen nicht unbedingt zum globalen Minimum der Verlustfunktion. Die Parameter  $\hat{\underline{\Theta}}_i$  werden also vorzugsweise durch die Anwendung eines linearen Optimierungsverfahrens für alle Teilmodelle separat bestimmt. Dies hat den Vorteil, dass die Parameterberechnung sehr schnell ist und immer zum globalen Minimum der Verlustfunktion führt. Zur Anwendung kommt eine gewichtete Least-Squares-Methode (WLS), die eine Erweiterung der normalen Least-Squares-Methode darstellt. Während bei der einfachen Least-Squares-Methode in der Verlustfunktion alle Gleichungsfehler  $e[k]$  gleich gewichtet werden, versieht man bei der gewichteten Least-Squares-Methode die einzelnen Gleichungsfehler mit unterschiedlichen Gewichten [111].

Die Herleitung der Least-Square Algorithmen wird in Kapitel (4.2.2) durchgeführt. Der Vollständigkeit halber sei hier die Berechnungsvorschrift für die rekursive Adaption der Parameter angegeben.

$$\hat{\underline{\Theta}}_i[k+1] = \hat{\underline{\Theta}}_i[k] + \underline{\gamma}_i[k+1] \left( y[k+1] - \underline{x}^T[k+1] \hat{\underline{\Theta}}_i[k] \right) \quad (3.37)$$

$$\underline{\gamma}_i[k+1] = \frac{\mathbf{P}_{wi}[k] \underline{x}[k+1]}{1/q_i[k+1] + \underline{x}^T[k+1] \mathbf{P}_{wi}[k] \underline{x}[k+1]} \quad (3.38)$$

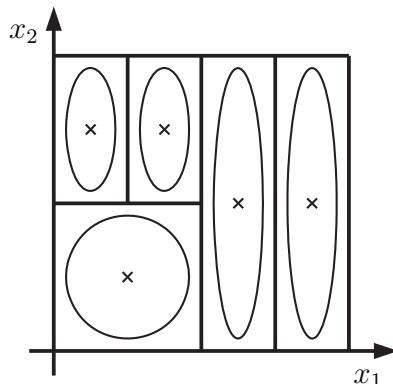
$$\mathbf{P}_{wi}[k+1] = (\mathbf{E} - \underline{\gamma}_i[k] \underline{x}^T[k+1]) \mathbf{P}_{wi}[k] \quad (3.39)$$

wobei  $\hat{\underline{\Theta}}_i$  die Parameter des  $i$ ten Teilmodells sind und  $\underline{\gamma}_i$  der Adaptionsschrittweite entspricht. Näheres zur Berechnung von  $\underline{\gamma}_i$  und den Eigenschaften des Algorithmus findet sich in Kapitel (4.2.2). Hier soll lediglich festgehalten werden, dass die rekursive Berechnung der Parameter für jedes einzelne Teilmodell separat durchgeführt wird. Daher spricht man von einer lokalen Parameterberechnung. Da sich die Zugehörigkeitsfunktionen aber gegenseitig überlappen (vgl. Abb. 3.24), beeinflusst jeder Parameter jedes Teilmodells den Modellausgang (siehe Gleichung (3.34)). Durch die Vernachlässigung der Einflüsse aller anderen Teilmodelle entsteht ein Fehler bei der Parameterberechnung, der Interpolationsfehler genannt wird. Der Interpolationsfehler ist beim scharfen Umschalten der Teilmodelle ( $\sigma = 0$ ) gleich Null und steigt mit zunehmender Überlappung der Zugehörigkeitsfunktionen stetig an. Im Gegensatz dazu spricht man von einer globalen Parameterberechnung, wenn die Parameter aller  $M$  Teilmodelle gleichzeitig berechnet werden. Mit der globalen Parameterberechnung würde man den Interpolationsfehler vermeiden, jedoch steigt der Rechenaufwand mit zunehmender Anzahl der Teilmodelle stark an. Außerdem wird in [156] gezeigt, dass bei verrauschten Messdaten die lokale Parameterberechnung der globalen Parameterberechnung trotz Interpolationsfehler in der Approximationsgüte überlegen ist.

Dies ist anschaulich dadurch zu erklären, dass die lokale Parameterberechnung einen Regularisierungseffekt hat, der die Gefahr vermindert, dass die Trainingsdaten auswendig gelernt werden (Overfitting – siehe Kapitel 3.4). Eine ausführliche Diskussion erfolgt in Kapitel 4.2.2.

### 3.9.2.2 Strukturoptimierung

Bei der Berechnung der Parameter ist von einer gegebenen Netzstruktur ausgegangen worden. Diese Netzstruktur wird schrittweise durch einen Offline-Algorithmus aufgebaut. Der Algorithmus zur Strukturoptimierung teilt den Eingangsraum der nichtlinearen Funktion in sog. Hyperquader auf. Jedem Hyperquader ist ein Teilmodell mit seiner entsprechenden Zugehörigkeitsfunktion zugeordnet. In die Mitte des Hyperquaders wird das Zentrum  $c$  der Zugehörigkeitsfunktion gelegt. Die Standardabweichungen  $\sigma$  werden in jeder Eingangsdimension proportional zur Ausdehnung des Hyperquaders gewählt. In Abb. 3.25 sind diese Zusammenhänge für den zweidimensionalen Fall veranschaulicht.



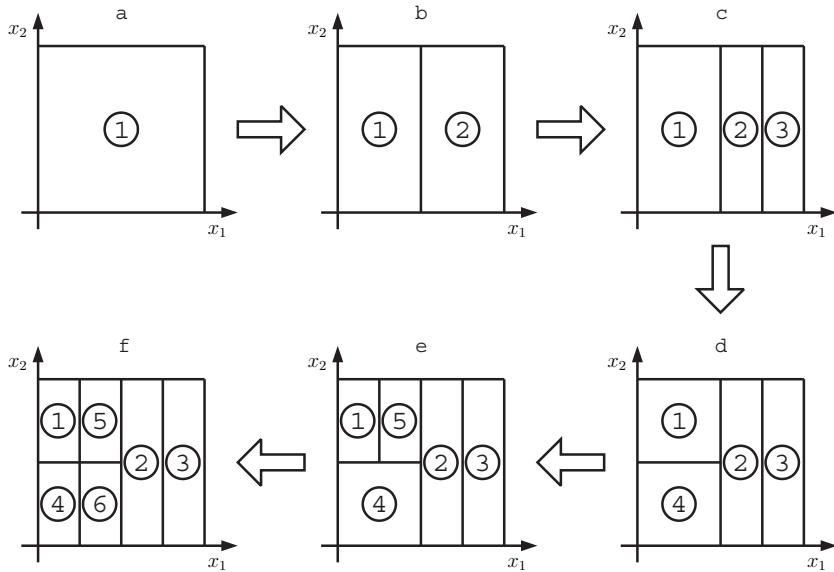
**Abb. 3.25:** Platzierung der Aktivierungsfunktionen im geteilten Eingangsraum

Die Kreuze markieren die Zentren der einzelnen Teilmodelle. Die Kreise und Ellipsen stellen Höhenlinien der nicht normierten Zugehörigkeitsfunktionen dar. In Abb. 3.25 ist zu erkennen, dass ein Teilmodell für den gesamten Eingangsreich von  $x_2$ , aber nur für einen kleinen Bereich von  $x_1$  zuständig ist.

In Abb. 3.26 ist der Algorithmus zur schrittweisen Konstruktion der Teilmodelle veranschaulicht.

Die Konstruktion der Teilmodelle läuft wie folgt ab:

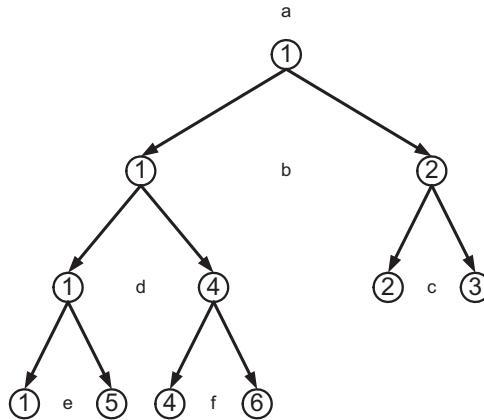
1. Initialisiere das erste Teilmodell so, dass alle  $P$  Vektoren der Datenmatrix enthalten sind und berechne die Parameter des globalen linearen Modells (siehe Abb. 3.26 (a)).



**Abb. 3.26:** Schrittweise Konstruktion der Teilmodelle

2. Für alle Eingangsdimensionen  $j := 1 \dots N$ :
  - (a) Teile das Teilmodell entlang der Dimension  $j$  in zwei Hälften.
  - (b) Berechne die Zugehörigkeitsfunktionen für beide neuen Teilmodelle.
  - (c) Berechne die Parameter für die neuen Teilmodelle.
  - (d) Berechne den Ausgangsfehler des Modells nach Gleichung (3.34).
3. Bestimme, welcher der obigen  $N$  Schnitte den kleinsten Ausgangsfehler aufweist.
4. Führe diesen Schnitt aus und verwende die Zugehörigkeitsfunktionen aus 2 b und die Parameter aus 2 c.
5. Berechne die lokalen Fehlermaße für die lokalen linearen Teilmodelle durch Gewichtung des Ausgangsfehlers mit den entsprechenden Zugehörigkeitsfunktionen.
6. Selektiere das Teilmodell mit dem größten lokalen Fehlermaß zur nächsten Teilung.
7. Wenn das Abbruchkriterium erfüllt ist, dann gehe zu Schritt 2, ansonsten endet der Algorithmus.

In Abb. 3.27 ist die zu Abb. 3.26 gehörende Baumstruktur dargestellt.



**Abb. 3.27:** Baumstruktur

Durch die Teilung des jeweils schlechtesten Teilmodells in jeder Iteration, wird die Komplexität des Modells von der Komplexität der zu approximierenden Funktion bestimmt, d.h. es werden nur Modelle verfeinert, wo dies auch notwendig ist. Durch die schrittweise Konstruktion der Teilmodelle, ausgehend von einem globalen Modell, erhält man durch den Grad der Verbesserung in jedem Iterationsschritt einen Anhaltspunkt für die Wahl der optimalen Modellanzahl. Dadurch, dass die Anzahl der Parameter nur linear mit der Eingangsdimension steigt, ist LOLIMOT auch für hochdimensionale Eingangsräume geeignet, wie dies bei der Identifikation nichtlinearer dynamischer Systeme der Fall ist.

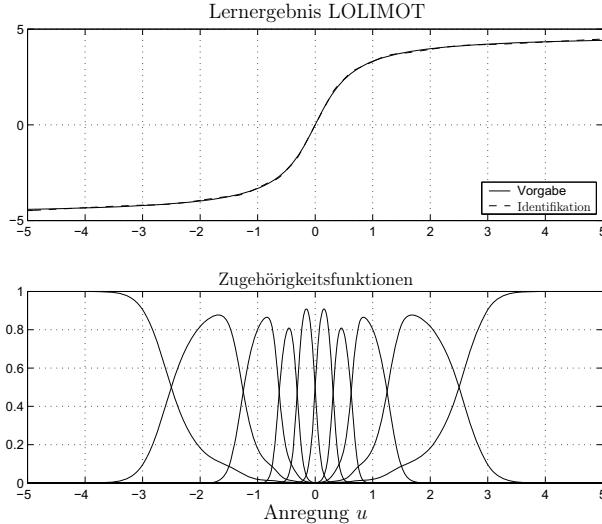
### 3.9.3 Beispiele

Im folgenden Kapitel sollen die speziellen Eigenschaften von LOLIMOT an den Beispielen einer ein- und einer zweidimensionalen statischen Nichtlinearität (siehe Kapitel 3.7.3 und 3.10) untersucht werden. Dabei soll insbesondere auf das Interpolationsverhalten und das Extrapolationsverhalten, sowie auf die Konvergenz der Parameter und den Rechenaufwand eingegangen werden. Diese Ergebnisse sollen dann im folgenden Kapitel zu einer Bewertung der speziellen Netzeigenschaften verwendet werden, woraus abschließend Aussagen über den spezifischen Einsatzbereich abgeleitet werden.

#### Approximation einer eindimensionalen statischen Nichtlinearität

Zunächst soll die eindimensionale statische Nichtlinearität  $\mathcal{N} = 3 \arctan(2u)$  aus Kapitel 3.7.3 identifiziert werden. Für den Netzeingang gilt  $x_1 = u$ . In Abb. 3.28

ist das Identifikationsergebnis mit 10 Teilmodellen sowie die entsprechenden Zugehörigkeitsfunktionen dargestellt.



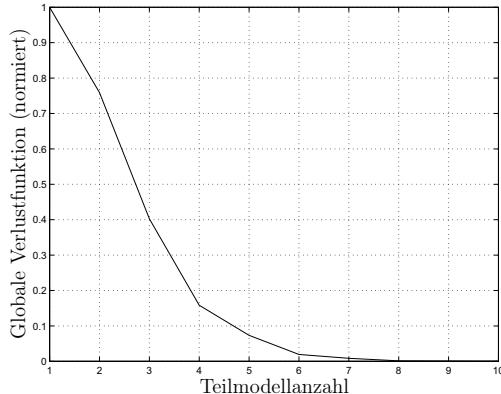
**Abb. 3.28:** Identifikationsergebnis mit 10 Teilmodellen —  $\mathcal{N} = 3 \arctan(2u)$

In Abb. 3.28 erkennt man, dass der Algorithmus den Eingangsraum verstkt im nichtlinearen Bereich der Funktion um den Ursprung teilt. Als normierte Standardabweichung wurde  $\sigma = 0.4$  gewhlt. Mit zunehmender Teilmodellanzahl verbessert sich auch das Identifikationsergebnis. In Abb. 3.29 ist der Verlauf der globalen Verlustfunktion ber 10 Teilmodelle angetragen. Der Verlustfunktion ist auf den Betrag Eins fr ein Teilmodell normiert.

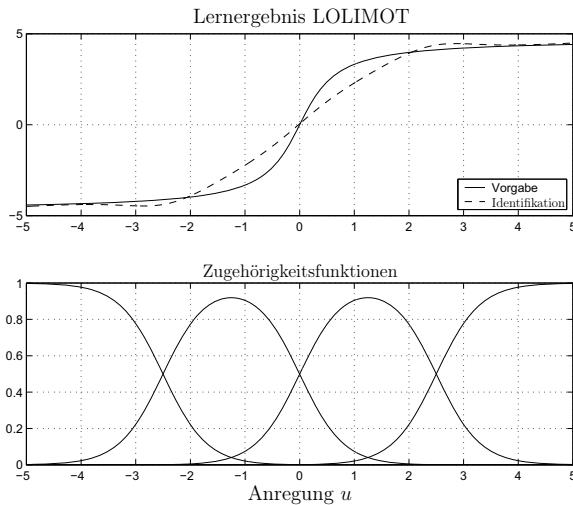
Das Interpolationsverhalten von LOLIMOT wird durch die Zugehrigkeitsfunktionen nach Gleichung (3.35) und (3.36) bestimmt. Die Standardabweichungen  $\sigma_i$  beeinflussen die berlappung der Zugehrigkeitsfunktionen, wobei  $\sigma = 0$  eine scharfe Abgrenzung bedeutet. Zur besseren Veranschaulichung des Interpolationsverhaltens wird ein Modell mit nur 4 Teilmodellen gewhlt. In Abb. 3.30 ist als Referenz das Identifikationsergebnis mit einer normierten Standardabweichung von  $\sigma = 0.4$  dargestellt.

Mit 4 Teilmodellen kann die nichtlineare Funktion selbstverstkt noch nicht mit ausreichender Genauigkeit approximiert werden, jedoch liefert die Wahl von  $\sigma$  ein plausibles Identifikationsergebnis. Whlt man  $\sigma = 0.1$ , erhlt man eine relativ scharfe Umschaltung zwischen den Teilmodellen, was in Abb. 3.31 veranschaulicht ist.

Der bergang zwischen den Teilmodellen ist deutlich zu erkennen. Der approximierte Funktionsverlauf geht schlagartig auf ein neues Teilmodell ber, d.h. auf eine Gerade mit unterschiedlicher Steigung und unterschiedlichem Achsen-



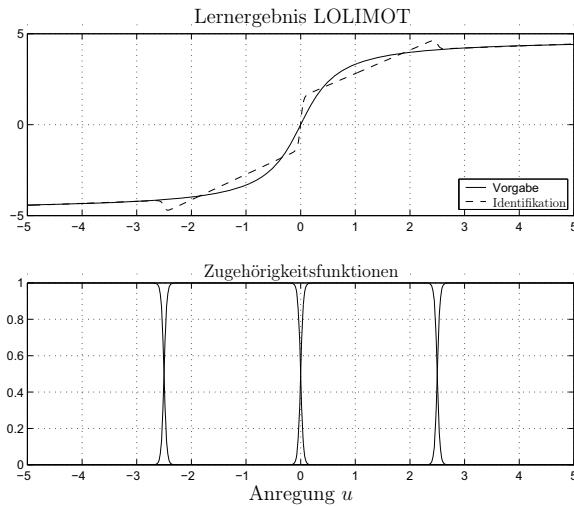
**Abb. 3.29:** Verlauf der globalen Verlustfunktion über 10 Teilmodelle



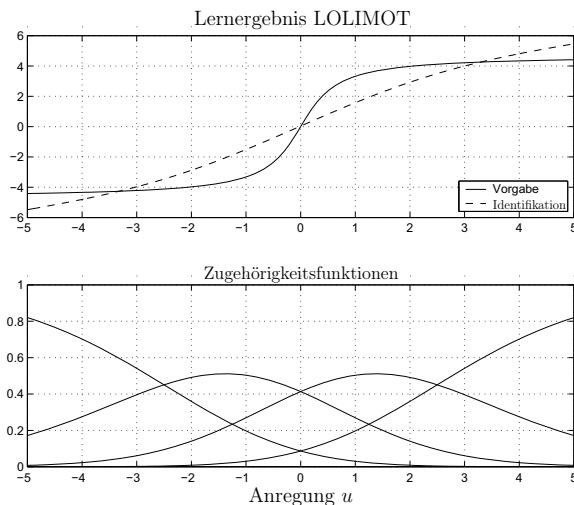
**Abb. 3.30:** Ergebnis mit 4 Teilmodellen und  $\sigma = 0.4$

abschnitt. Wählt man  $\sigma = 0.8$ , überlappen sich die Zugehörigkeitsfunktionen in einem weiten Bereich. In Abb. 3.32 werden die einzelnen Teilmodelle sehr stark verschliffen.

Durch die starke Überlappung der Zugehörigkeitsfunktionen kann die nichtlineare Funktion nur noch sehr schlecht approximiert werden. Die Wahl eines zu großen Wertes für  $\sigma$  hat ein schlechtes Interpolationsverhalten zur Folge und liefert somit ein unbefriedigendes Identifikationsergebnis. Die Wahl eines zu kleinen



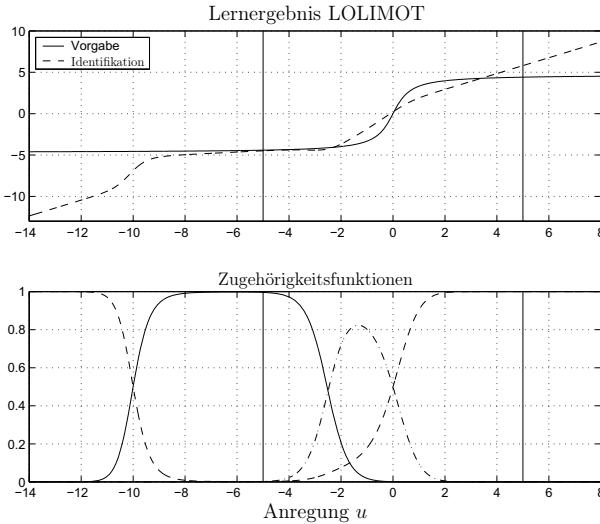
**Abb. 3.31:** Ergebnis mit 4 Teilmodellen bei scharfem Umschalten ( $\sigma = 0.1$ )



**Abb. 3.32:** Ergebnis mit 4 Teilmodellen bei starkem Überlappen der Zugehörigkeitsfunktionen ( $\sigma = 0.8$ )

Wertes führt ebenfalls zu einem schlechten Interpolationsverhalten, dies wirkt sich jedoch mit zunehmender Teilmallanzahl immer weniger aus.

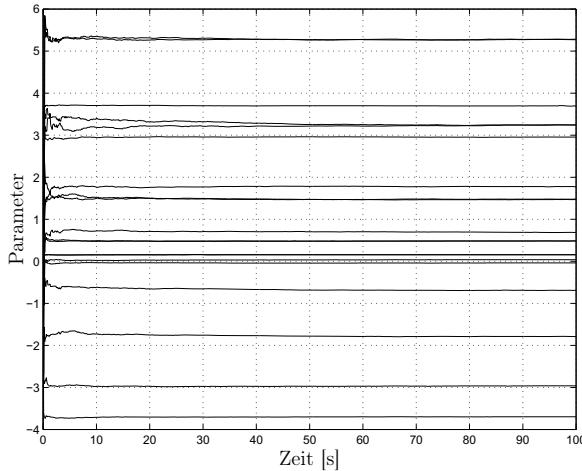
Das Extrapolationsverhalten beschreibt die Approximationsfähigkeit über den Trainingsbereich hinaus. Die nichtlineare Funktion  $\mathcal{N}\mathcal{L} = 3 \arctan(2u)$  wurde im Bereich von  $-5$  bis  $5$  angeregt und trainiert. Für Eingangssignale außerhalb von diesem Bereich muss das Netz extrapoliieren. In Abb. 3.33 ist das Extrapolationsverhalten bei 3 Teilmodellen dargestellt. Die Teilmodellanzahl wurde bewusst sehr klein gewählt, um die Zusammenhänge besser zu veranschaulichen.



**Abb. 3.33:** Extrapolationsverhalten von LOLIMOT

Von LOLIMOT zu erwarten wäre ein lineares Extrapolationsverhalten entsprechend dem linken lokalen linearen Modell bei  $u < -5$  bzw. entsprechend dem rechten lokalen linearen Modell bei  $u > 5$ . Dies ist aber nicht der Fall. Streben die Eingangswerte gegen  $+\infty$  oder  $-\infty$ , so dominiert stets das Teilmodell mit der Zugehörigkeitsfunktion, die die größte Standardabweichung aufweist, d.h. in Abb. 3.33 das rechte Teilmodell. Für  $u > 5$  stimmt dies zufällig mit dem erwarteten Extrapolationsverhalten überein, betrachtet man jedoch Eingangswerte  $u < -5$  so erkennt man, dass bei  $u = -10$  ebenfalls das rechte Teilmodell dominant wird. Um dieses unerwünschte Extrapolationsverhalten zu verhindern, müssen die Werte der Zugehörigkeitsfunktionen an den Grenzen zur Extrapolation, d.h. bei  $u = -5$  und  $u = 5$  festgehalten werden.

Ein wesentlicher Vorteil von LOLIMOT ist, dass zur Parameterberechnung ein lineares Optimierungsverfahren (gewichtetes Least-Squares-Verfahren) zum Einsatz kommt. Dies garantiert eine schnelle Konvergenz der Parameter und das Erreichen des globalen Minimums der Verlustfunktion. In Abb. 3.34 ist das Konvergenzverhalten aller 20 Parameter (2 Parameter je Teilmodell) dargestellt.



**Abb. 3.34:** Konvergenzverhalten der Parameter

Die meisten Parameter erreichen bereits nach 20 s Lernzeit Werte, die sich kaum mehr ändern. Die Parameter, die die Teilmodelle am Rand des Anregebereichs beschreiben, konvergieren etwas langsamer.

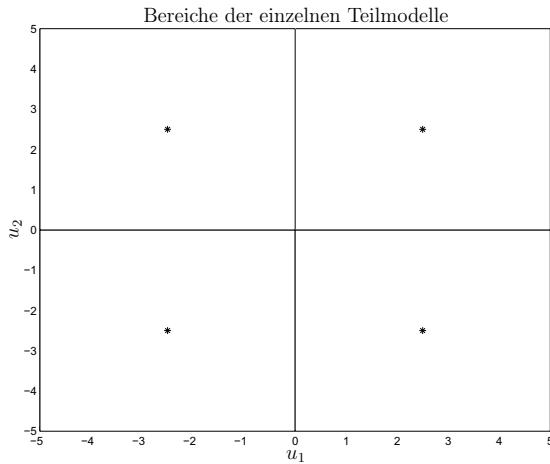
### Approximation einer zweidimensionalen statischen Nichtlinearität

Mit LOLIMOT können auch mehrdimensionale statische Nichtlinearitäten approximiert werden. In diesem Kapitel soll der sog. Affensattel aus Kapitel 3.7.3 identifiziert werden. Die Güte des Identifikationsergebnisses steigt mit der Anzahl der Teilmodelle. Zunächst sollen zur Veranschaulichung der Ergebnisse 4 Teilmodelle betrachtet werden. In Abb. 3.35 ist die Einteilung des zweidimensionalen Eingangsraumes dargestellt, wie sie vom Algorithmus zur Strukturoptimierung vorgenommen wird.

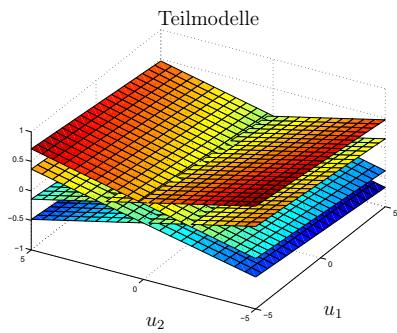
Es wird deutlich, dass bei 4 Teilmodellen der Eingangsraum symmetrisch eingeteilt wird. Die Sterne in Abb. 3.35 markieren die Zentren der Zugehörigkeitsfunktionen, die jeweils in der Mitte der Teilmodelle liegen. Die Teilmodelle mit den zugehörigen Aktivierungsfunktionen sind in Abb. 3.36 und Abb. 3.37 abgebildet.

Die einzelnen Teilmodelle sind nun Ebenen, die entsprechend der Aktivierungsfunktionen überlagert werden. Im höherdimensionalen Fall sind die Teilmodelle als Hyperebenen interpretierbar, die allerdings nicht mehr anschaulich dargestellt werden können.

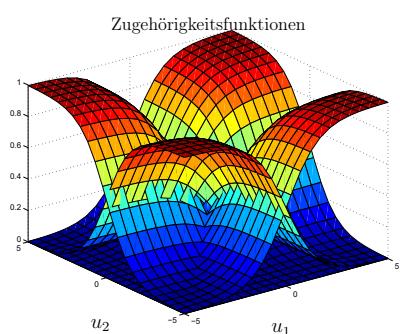
Abb. 3.38 und Abb. 3.39 zeigen die vorgegebene statische Nichtlinearität zusammen mit dem Identifikationsergebnis bei 4 Teilmodellen. Der zugehörige absolute Fehler ist in Abb. 3.40 dargestellt.



**Abb. 3.35:** Teilung des Eingangsraumes bei 4 Teilmodellen



**Abb. 3.36:** vier Teilmodelle

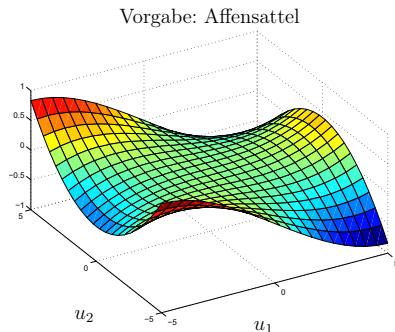
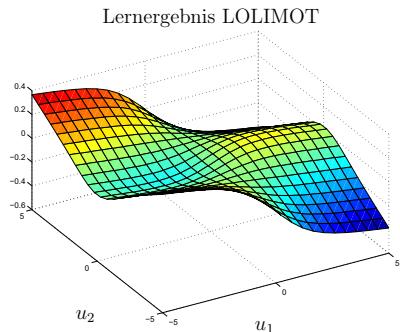
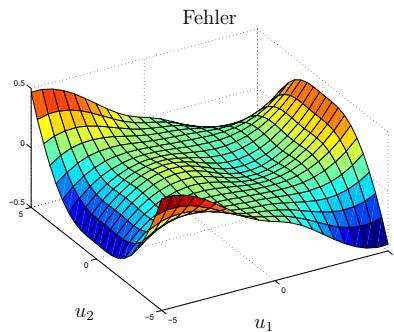
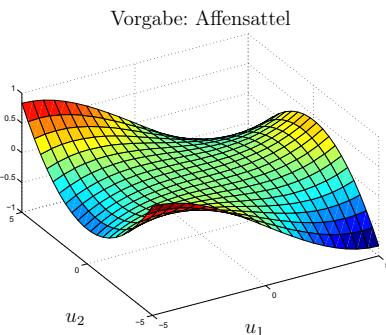
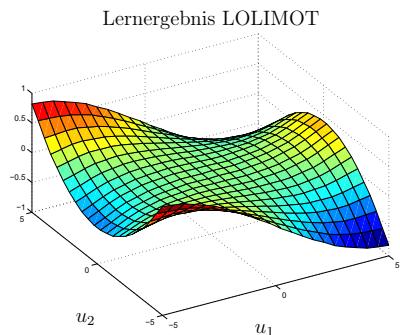


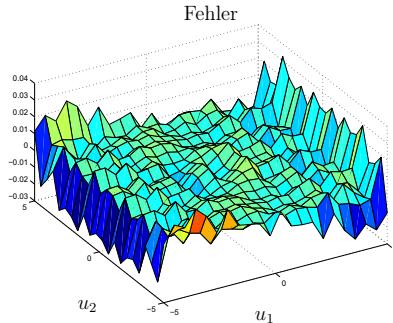
**Abb. 3.37:** vier Zugehörigkeitsfunktionen

In Abb. 3.39 erkennt man, dass 4 Teilmodelle nicht ausreichen, um die statische Nichtlinearität genau zu approximieren. Mit der Teilmodellanzahl erhöht sich die Genauigkeit des Identifikationsergebnisses, allerdings auch die Anzahl der zu berechnenden Parameter. Die Parameteranzahl ist linear von der Teilmödellanzahl abhängig.

In Abb. 3.42 und Abb. 3.43 ist das Identifikationsergebnis und der absolute Fehler mit 64 Teilmodellen abgebildet.

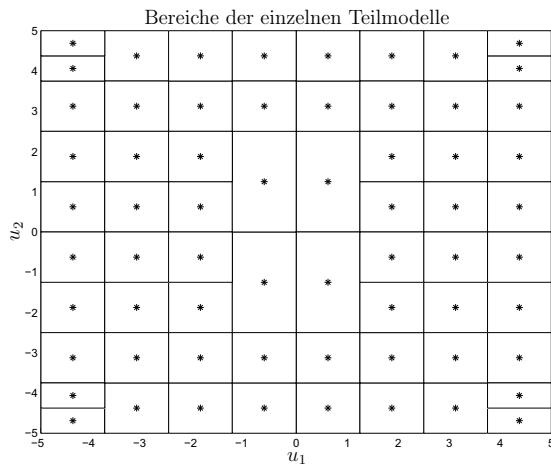
Die identifizierte Nichtlinearität approximiert die vorgegebene Nichtlinearität sehr genau, allerdings ist die Anzahl der Teilmodelle sehr hoch. In der Praxis muss

**Abb. 3.38:** Nichtlinearität**Abb. 3.39:** Identifikationsergebnis**Abb. 3.40:** Absoluter Fehler**Abb. 3.41:** Nichtlinearität**Abb. 3.42:** Identifikationsergebnis



**Abb. 3.43:** Absoluter Fehler (64 Teilmodelle)

hier ein Kompromiss zwischen der gewünschten Genauigkeit und der Anzahl der Parameter gefunden werden. In Abb. 3.44 ist die Bereichseinteilung für 64 Teilmodelle dargestellt.



**Abb. 3.44:** Teilung des Eingangsraumes bei 64 Teilmödellen

Es ist sehr gut zu erkennen, dass der Algorithmus zur Strukturoptimierung den Eingangsraum verstrkt in den „Ecken“ teilt, wo die vorgegebene Funktion sehr stark nichtlinear ist. Im Bereich um den Ursprung, wo die Funktion annhernd linearen Charakter aufweist, werden dagegen tendenziell weniger Teilm- odelle bentigt um die Funktion zu approximieren. Die Anzahl der zu berech- nenden Parameter ergibt sich entsprechend Gleichung (3.40) zu

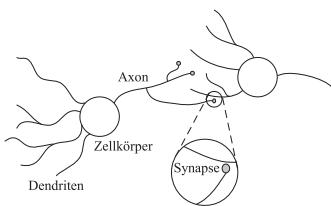
$$p = M \cdot (N + 1). \quad (3.40)$$

Die Anzahl der Parameter steigt linear mit der Teilmodellanzahl und linear mit  $(N + 1)$ . Im vorliegenden Beispiel mussten für die Teilmodelle  $M = 64$  und bei der Dimension  $N = 2$  insgesamt  $p = 192$  Parameter trainiert werden. Dies bedeutet, dass die Parameteranzahl im Vergleich zu RBF-Netzen mit zunehmender Eingangsdimension deutlich weniger ansteigt. Ein weiterer Vorteil bezüglich des Rechenaufwandes ergibt sich aus der lokalen Parameterberechnung, d.h. der separaten Berechnung für jedes einzelne Teilmodell. Bei der lokalen Parameterberechnung steigt der Rechenaufwand linear mit der Teilmodellanzahl.

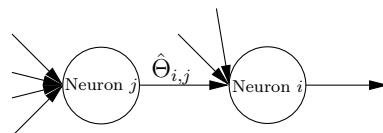
## 3.10 Multi-Layer-Perceptron (MLP) Netze

### 3.10.1 Einleitung

Ein Multi-Layer-Perceptron Netzwerk (MLP) ist ein klassisches künstliches neuronales Netz, welches an dem biologischen Vorbild „menschliches Gehirn“ angelehnt ist. Die einzelnen Neuronen, oft auch als Perzeptoren bezeichnet, bestehen wie das biologische Vorbild aus einem Zellkörper, den Dendriten, welche die Eingabe des Netzes in die Zelle aufsummieren und einem Axon, welches die Ausgabe einer Zelle nach außen weiterleitet. Im biologischen Vorbild verzweigen sich diese Axone und treten über die Synapsen mit den Dendriten nachfolgender Neuronen in Kontakt. Die Stärke der Synapsen wird durch das Verbindungsgewicht  $\hat{\Theta}_{ij}$  dargestellt (Abb. 3.45 und 3.46).



**Abb. 3.45:** Schema biologischer Neuronen

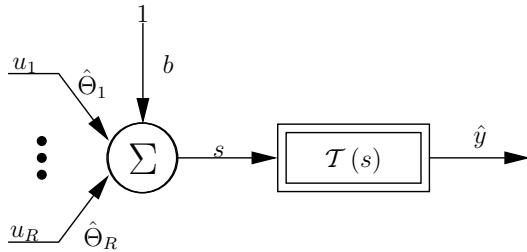


**Abb. 3.46:** Schema technischer Neuronen

### 3.10.2 Technische Abstraktion

Das technische Perzepton, wie es in den MLP Netzen Verwendung findet, wird im folgenden beschrieben.

Abbildung 3.47 zeigt den funktionalen Zusammenhang zwischen dem Ausgang  $\hat{y}$  und den Eingängen  $u_1$  bis  $u_R$  des Perzeptrons. Zunächst werden die



**Abb. 3.47:** Schema eines technischen Perzeptrons

Eingänge mit den zugehörigen Verbindungsgewichten (Kantengewichte)  $\hat{\Theta}_i$  gewichtet und anschließend aufsummiert. Zusätzlich zu dieser Summe wird noch der Bias oder Offset  $b$  hinzu addiert. Die anschließende nichtlineare Transferfunktion (oder auch Entscheidungs- bzw. Aktivierungsfunktion)  $\mathcal{T}(s)$  bildet den Summenausgang  $s$  nichtlinear auf den Perzeptronausgang  $\hat{y}$  ab. Algebraisch lässt sich das einzelne Perzeptron wie folgt beschreiben:

$$\begin{aligned} s &= \hat{\Theta}^T \cdot \underline{u} + b = \sum_{i=1}^R \hat{\Theta}_i \cdot u_i + b \\ \hat{y} &= \mathcal{T}(s) \end{aligned} \quad (3.41)$$

Dabei stellen  $\hat{\Theta} = [\hat{\Theta}_1, \hat{\Theta}_2, \dots, \hat{\Theta}_R]^T$  und  $b$  die trainierbaren Gewichte dar und  $\underline{u} = [u_1, u_2, \dots, u_R]^T$  beschreibt den Eingangsvektor. Hierbei gilt, dass  $\hat{\Theta}_i \in \mathbb{R}$  ist und je nach Anwendung  $u_i \in \mathbb{R}$  oder  $u_i \in \mathbb{B}$  sein kann.

Die Gleichungen in (3.41) können nun kompakt dargestellt werden als

$$\hat{y} = \mathcal{T}(\hat{\Theta}^T \cdot \underline{u} + b) \quad (3.42)$$

Es gelten die Anmerkungen zu Gleichung (3.3) in gleicher Weise.

### 3.10.3 Transferfunktionen

Besondere Bedeutung im Perzeptron hat die Transferfunktion  $\mathcal{T}(s)$ . Sie bildet den Summenausgang  $s$  nichtlinear auf den Ausgang  $\hat{y}$  ab. Hierfür werden am häufigsten sigmoide (S-förmige) Transferfunktionen verwendet. Diese Funktionen ermöglichen es dem Perzeptron, sowohl auf Signale mit kleiner Amplitude, als auch auf Signale mit großer Amplitude zu reagieren, und werden deshalb als *squashing* Funktionen bezeichnet. Weiterhin sind sigmoide Funktionen stetig und überall differenzierbar und werden aus diesem Grund weiche Transferfunktionen genannt. Im Gegensatz hierzu sind binäre Schrittfunctionen (Entscheidungsfunktionen) harte Transferfunktionen. Zu den weichen Transferfunktionen gehören die

häufig verwendete logistische Transferfunktion und die  $\tanh(s)$  Funktion. Harte Transferfunktionen sind die Signumfunktion und die Stufenfunktion. Typische Transferfunktionen sind in Tab. 3.1 abgebildet.

### 3.10.4 Mehrschichtiges MLP-Netz

Die in der Praxis verwendeten MLP-Netze haben meist mehrere Schichten versteckter Neuronen (technisches Perzeptron aus Abbildung 3.47) zwischen den Eingabeneuronen (input units) und den Ausgabeneuronen. Ein solches MLP-Netz ist in Abb. 3.48 dargestellt. Die Nomenklatur orientiert sich hauptsächlich an den Arbeiten [81, 44].

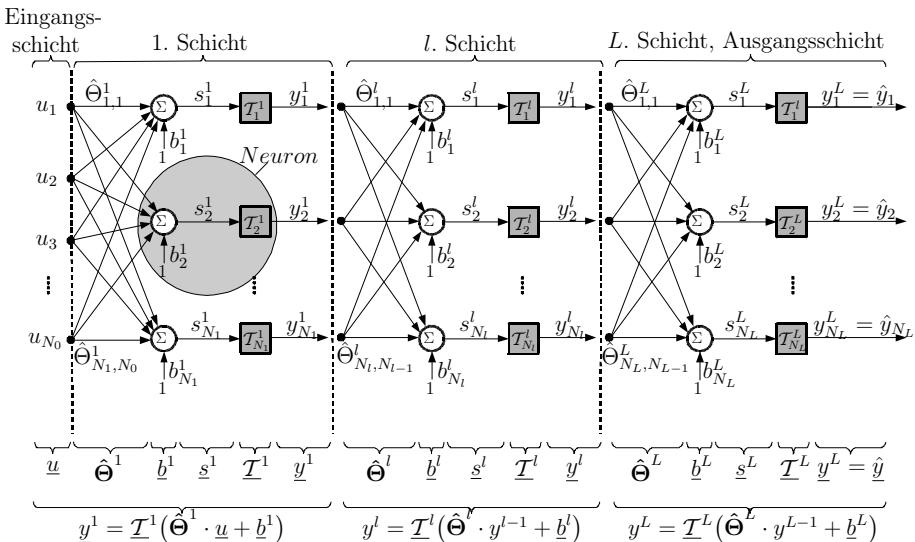


Abb. 3.48: Ein allgemeines feedforward Netzwerk mit  $L$  trainierbaren Schichten

Es handelt sich hier um ein  $L$ -stufiges Netz, d.h um ein Netzwerk mit  $L$  Schichten trainierbarer Verbindungen<sup>1)</sup>. Die  $N_0$  Zellen der Eingabeschicht (input layer) leiten die Eingabe in das Netz weiter, dienen also nur zur Auffächerung der Eingangssignale und haben keine Verarbeitungsfunktion. Die einzelnen Neuronen — auch Knoten des Netzwerkes genannt — in den  $L - 1$  versteckten Schichten (hidden layers) und die der Ausgabeschicht (output layer) dienen zur Informationsverarbeitung und entsprechen im einzelnen Abb. 3.47.

<sup>1)</sup> Die Anzahl der Stufen neuronaler Netze wird leider in der Literatur nicht eindeutig verwendet. Hier wird ein Netz als  $L$ -stufig bezeichnet, wenn es  $L$ -Schichten trainierbarer Verbindungen hat. Derartige Netze haben dann  $L + 1$  Schichten von Neuronen, davon  $L - 1$  verdeckte Schichten.

	$\mathcal{T}(s)$	$\mathcal{T}'(s) = \partial\mathcal{T}(s)/\partial s$	$\mathcal{T}(s)$
weiche Transferfunktionen	$\tanh(s) = \frac{1 - e^{-2s}}{1 + e^{2s}}$ tanh Funktion	$1 - \mathcal{T}^2(s)$	
	$\frac{1}{1 + e^{-s}}$ Logistikfunktion	$\mathcal{T}(s) \cdot (1 - \mathcal{T}(s))$	
	$\frac{O - U}{1 + e^{-cs}} + U$ verallg. Logistikfunktion	$(\mathcal{T}(s) - U) \cdot (O - \mathcal{T}(s)) \cdot \frac{c}{O - U}$	
harte Transferfunktionen	$c \cdot s$ lineare Funktion	$c$	
	$\text{sign}(s)$ Signumfunktion		
	$= \begin{cases} +1, & s \geq 0 \\ 0, & \text{sonst} \end{cases}$ Stufenfunktion		

Tabelle 3.1: Typische Transferfunktionen  $\mathcal{T}(s)$

Für das feedforward Netz aus Abb. 3.48 berechnet sich der Netzausgang zu:

$$\underline{\hat{y}} = \underline{\mathcal{T}}^L(\hat{\Theta}^L \cdot \underline{\mathcal{T}}^{L-1}(\hat{\Theta}^{L-1} \cdot \underline{\mathcal{T}}^{L-2}(\cdots \underline{\mathcal{T}}^2(\hat{\Theta}^2 \cdot \underline{\mathcal{T}}^1(\hat{\Theta}^1 \cdot \underline{u} + \underline{b}^1) + \underline{b}^2 \cdots \cdots) + \underline{b}^{L-2}) + \underline{b}^{L-1}) + \underline{b}^L) \quad (3.43)$$

mit

$$\begin{aligned} \underline{u} &= \begin{bmatrix} u_1 \\ \vdots \\ u_{N_0} \end{bmatrix} & \underline{\hat{y}} &= \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_{N_L} \end{bmatrix} & \underline{b}^l &= \begin{bmatrix} b_1^l \\ \vdots \\ b_{N_l}^l \end{bmatrix} \\ \underline{\mathcal{T}}^l &= \begin{bmatrix} \mathcal{T}_1^l \\ \vdots \\ \mathcal{T}_{N_l}^l \end{bmatrix} & \hat{\Theta}^l &= \begin{bmatrix} \hat{\Theta}_{1,1}^l & \hat{\Theta}_{1,2}^l & \cdots & \hat{\Theta}_{1,N_{l-1}}^l \\ \hat{\Theta}_{2,1}^l & \hat{\Theta}_{2,2}^l & \cdots & \hat{\Theta}_{2,N_{l-1}}^l \\ \vdots & \vdots & \vdots & \vdots \\ \hat{\Theta}_{N_l,1}^l & \hat{\Theta}_{N_l,2}^l & \cdots & \hat{\Theta}_{N_l,N_{l-1}}^l \end{bmatrix} \end{aligned}$$

Hierbei ist  $\underline{u}$  der Eingangsvektor,  $\underline{\hat{y}}$  der Ausgangsvektor,  $\underline{b}^l$  der bias-Vektor, die Elemente in  $\underline{\mathcal{T}}^l$  sind die einzelnen Transferfunktionen der Neuronen in der  $l$ -ten Schicht. Die Matrix  $\hat{\Theta}^l$  enthält die Verbindungsgewichte der  $l$ -ten trainierbaren Schicht. Für die einzelnen Gewichte  $\hat{\Theta}_{i,j}^l$ <sup>2)</sup> gilt folgende Konvention:

- $\hat{\Theta}_{i,j}^l = 0$  gibt an, dass keine Verbindung zwischen Neuron  $j$  in der  $(l-1)$ -ten Schicht und seinem Nachfolger  $i$  in der  $l$ -ten Schicht existiert.
- $\hat{\Theta}_{i,j}^l < 0$  gibt an, dass Neuron  $j$  in der  $(l-1)$ -ten Schicht seinen Nachfolger  $i$  in der  $l$ -ten Schicht durch ein Gewicht mit der Stärke  $|\hat{\Theta}_{i,j}^l|$  hemmt.
- $\hat{\Theta}_{i,j}^l > 0$  gibt an, dass Neuron  $j$  in der  $(l-1)$ -ten Schicht seinen Nachfolger  $i$  in der  $l$ -ten Schicht durch ein Gewicht mit der Stärke  $|\hat{\Theta}_{i,j}^l|$  stärkt.

### 3.10.5 Auslegung von feedforward Netzen

Zur Auslegung von feedforward Netzen gibt es bis heute keine allgemeingültige Regel. Es gilt jedoch folgender Satz aus der Approximationstheorie:

**Satz:** Zwei- bzw. mehrschichtige, vorwärtsgerichtete neuronale Netze mit monoton steigenden Transferfunktionen mit squashing Charakter erlauben es, jede kontinuierliche Funktion  $\mathcal{N}\mathcal{C}$  in  $C(\underline{u})$  beliebig genau zu approximieren, d.h.

$$||\widehat{\mathcal{N}\mathcal{C}}(\underline{u}) - \mathcal{N}\mathcal{C}(\underline{u})|| < \epsilon \quad \text{für } \epsilon \in \mathbb{R}^+$$

---

<sup>2)</sup> Anmerkung: der hochgestellte Index  $l$  steht hier für die  $l$ -te trainierbare Schicht. Es gilt  $1 \leq l \leq L$ .  $N_l$  steht für die Anzahl der Neuronen in der  $l$ -ten Schicht. Der tiefgestellte Index  $i, j$  bedeutet, dass das Gewicht das Neuron  $j$  aus der Schicht  $l-1$  verbindet mit dem Neuron  $i$  aus der Schicht  $l$ .

$C(\underline{u})$  ist die Menge aller stetigen Funktionsabbildungen der kompakten Wertemenge  $U \subset \mathbb{R}^n$  auf die reelle Zahlengerade  $\mathbb{R}$ , d.h.

$$\mathcal{NL} : \quad U \longrightarrow \mathbb{R}$$

Dieser fundamentale Satz geht zurück auf Weierstrass (1903) bzw. Cybenko/Hornik (1989).

Da dieser Satz nicht konstruktiv ist, muss die Struktur des neuronalen Netzes durch gezieltes Ausprobieren ermittelt werden.

Einige Faustregeln zur Festlegung der Netztopologie besagen:

- Zu wenig Schichten, bzw. Neuronen pro Schicht verringern die Repräsentationsfähigkeit der Funktion durch das neuronale Netz.
- Zu viele Schichten bzw. Neuronen (hoher Lernaufwand) verringern die Verallgemeinerungsfähigkeit des neuronalen Netzes
- Eine hohe Anzahl an Neuronen in einer Schicht kann eventuell durch Hinzufügen einer weiteren versteckten Schicht vermieden werden.

### 3.10.6 Beispiele

#### Approximation einer eindimensionalen statischen Nichtlinearität

Zunächst soll die eindimensionale Nichtlinearität  $\mathcal{NL}(u) = 3 \cdot \arctan(2 \cdot u)$  identifiziert werden. In einem ersten Versuch wurde ein 2-schichtiges MLP-Netz verwendet, wobei in der einzigen versteckten Schicht 2 Neuronen mit der  $\tanh(s)$  Transferfunktion und 1 Neuron mit der linearen Transferfunktion implementiert sind. Das Ausgangsneuron hat ebenfalls eine lineare Transferfunktion.

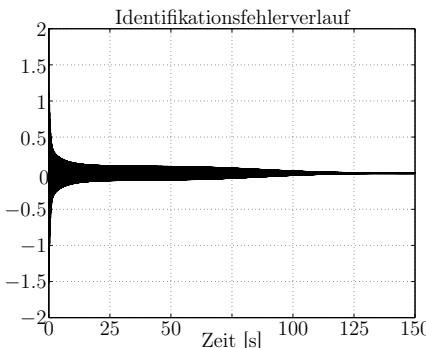


Abb. 3.49: Identifikationsfehlerverlauf  $e = y - \hat{y}$

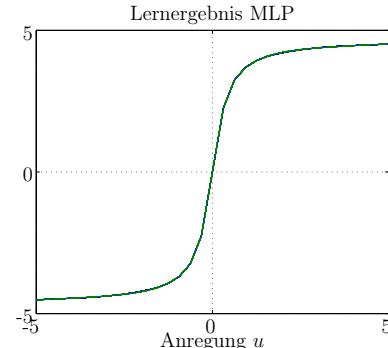


Abb. 3.50: Gelernte Kennlinie (..) und wahre Kennlinie (-)

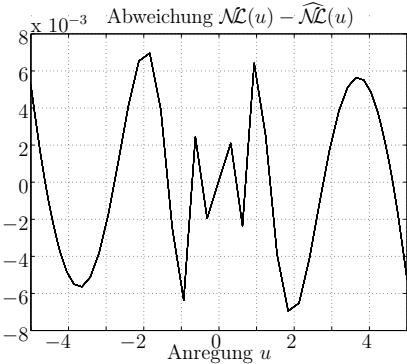
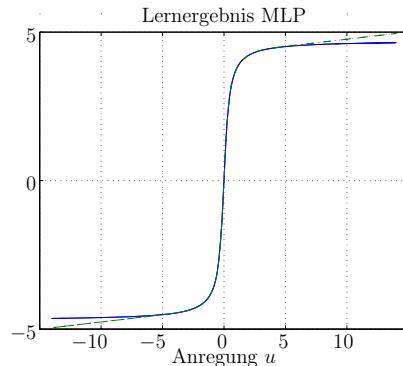
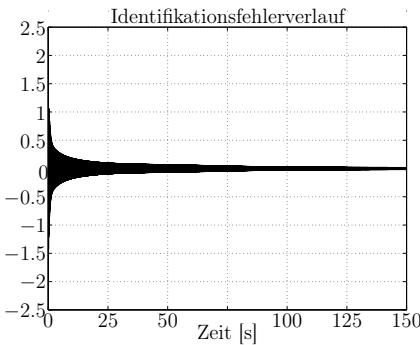
Abb. 3.51:  $NLL(u) - \widehat{NLL}(u)$ 

Abb. 3.52: Extrapolationsverhalten außerhalb des Eingangsbereichs (- Vorgabe .- Netzausgabe)

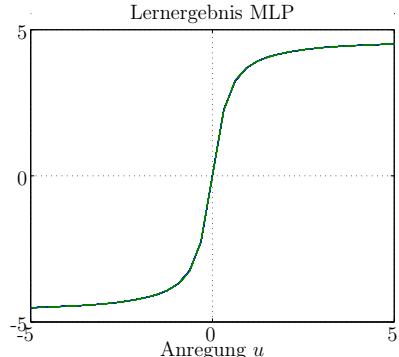
In Abb. 3.49 ist zunächst der Identifikationsfehlerverlauf dargestellt. Es ist gut zu sehen, dass der Fehler zunächst sehr schnell kleiner wird, ehe er dann zwischen 25s und ca. 70s ungefähr konstant bleibt, um dann wiederum stark abzunehmen. Dies bedeutet, dass auf der Fehlerfläche zunächst ein flaches Plateau überwunden werden musste. In Abb. 3.50 ist das Identifikationsergebnis dargestellt. Der Unterschied zwischen der gelernten Kennlinie und der Vorgabe ist hierbei zu vernachlässigen. Dies zeigt auch Abb. 3.51. Hier ist zu sehen dass die Abweichung  $NLL(u) - \widehat{NLL}(u)$  nach erfolgtem Lernvorgang im gesamten Eingangsbereich unter 0.008 bleibt, was ungefähr 0.26% Abweichung entspricht. Die vorgegebene Kennlinie ist also sehr gut approximiert worden. In Abb. 3.52 ist schließlich das Extrapolationsverhalten des MLP-Netzes dargestellt. In dem Bereich, in dem zuvor gelernt ( $-5 < u < 5$ ) worden ist, stimmen die Kennlinien nahezu überein (siehe auch Abb. 3.50). In den äußeren Bereichen nimmt allerdings die Abweichung deutlich zu. Es ist zu sehen, dass die approximierte Kennlinie linear ansteigt. Dies liegt an der Tatsache, dass einerseits die verwendeten sigmoiden Transferfunktionen ( $\tanh$ ) in diesem Bereich in Sättigung gehen und damit andererseits der Einfluss der Neuronen mit linearem Verhalten überwiegt.

In einem zweiten Versuch wird die Neuronenzahl erhöht. Das 2-schichtiges MLP Netz hat nun 5  $\tanh(s)$  Transferfunktionen und 3 lineare Transferfunktionen in der verdeckten Schicht und eine lineare Transferfunktion in der Ausgangsschicht.

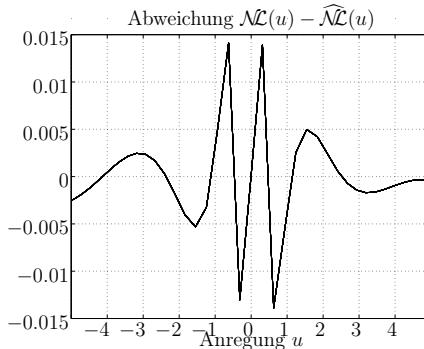
In Abb. 3.53 ist wiederum der Identifikationsfehlerverlauf dargestellt. Auch hier gibt es zunächst ein schnelles Abklingen des Fehlersignals, aber ab ca. 25s verlangsamt sich die Abnahme deutlich. In Abb. 3.55 ist zu sehen, dass die Abweichung  $NLL(u) - \widehat{NLL}(u)$  nach erfolgtem Lernvorgang im gesamten Eingangsbereich deutlich höher ist als im vorherigen Beispiel. Ein überdimensioniertes MLP Netz führt erwartungsgemäß zu einer deutlich längeren Konvergenzzeit; es wurde aber



**Abb. 3.53:** Identifikationsfehlerverlauf  $e = y - \hat{y}$



**Abb. 3.54:** Gelernte Kennlinie (,-) und wahre Kennlinie (-)



**Abb. 3.55:** Differenz zwischen vorgegebener nichtlinearen Funktion und Identifikationsergebnis über dem Eingangsbereich

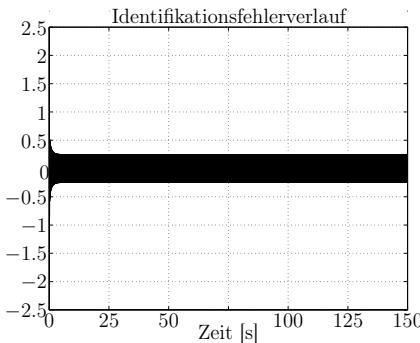
hier — wie vorher — eine Lernzeit von 150s angesetzt, so dass damit die Lernzeit zu kurz ist.

In einer weiteren Simulation wird versucht, die Nichtlinearität nur durch ein Neuron in der versteckten Schicht zu identifizieren. Die Ergebnisse sind in Abb. 3.56 und Abb. 3.57 dargestellt.

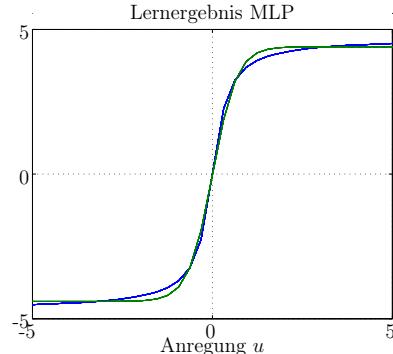
In Abb. 3.56 ist zu sehen, dass der Fehler schnell kleiner wird, aber dann während des gesamten Lernvorgangs konstant bleibt. In Abb. 3.57 ist die Abweichung zwischen der gelernten Nichtlinearität und der vorgegebenen Nichtlinearität deutlich zu sehen. Dieses Netz ist also falsch dimensioniert.

### Approximation einer zweidimensionalen statischen Nichtlinearität

Gegenüber anderen statischen Funktionsapproximatoren eignet sich das MLP-Netz besonders zur Approximation von mehrdimensionalen nichtlinearen Funk-



**Abb. 3.56:** Identifikationsfehlerverlauf  $e = y - \hat{y}$



**Abb. 3.57:** Gelernte Kennlinie (--) und wahre Kennlinie (-)

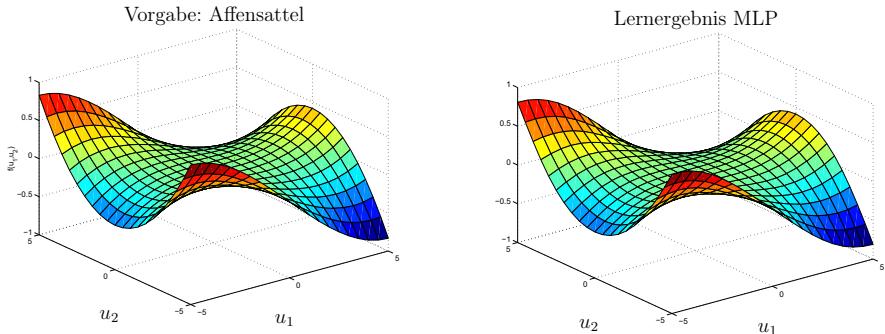
tionen, da die Anzahl der zu trainierenden Parameter nur gering mit der Eingangsdimension steigt. In diesem Abschnitt soll der sogenannte ‘Affensattel’ identifiziert werden.

$$\mathcal{N}\mathcal{L}(u_1, u_2) = (u_1^3 - 3 \cdot u_1 \cdot u_2^2)/300 \quad (3.44)$$

Hierfür wird ein MLP-Netz mit einer versteckten Schicht verwendet, wobei das Ausgangsneuron eine lineare Transferfunktion besitzt. Die versteckte Schicht enthält 15 Neuronen mit  $\tanh(s)$  als Transferfunktion, sowie 15 Neuronen mit linearer Transferfunktion. Zwischen den beiden Eingangsneuronen und den 30 Neuronen in der versteckten Schicht gibt es  $2 \cdot 30 = 60$  Verbindungsgewichte. Die 30 Neuronen der versteckten Schicht sind wiederum mit dem Ausgangsneuron über  $30 \cdot 1 = 30$  Gewichte verbunden. Insgesamt sind somit  $30 + 60 = 90$  Verbindungsgewichte vorhanden. Zu den Verbindungsgewichten kommen die Biasgewichte. Jedes Neuron der versteckten und der letzten Schicht hat ein eigenes Biasgewicht. Deshalb verfügt das MLP in diesem Beispiel über 31 Biasgewichte. Insgesamt gibt es demnach 121 Gewichte, 90 Verbindungsgewichte und 31 Biasgewichte. Die vorgegebene nichtlineare Funktion ist in Abb. 3.58 dargestellt, in Abb. 3.59 das Identifikationsergebnis, sowie in Abb. 3.60 die Abweichung zwischen Identifikationsergebnis und vorgegebener statischer Nichtlinearität aus Gleichung (3.44).

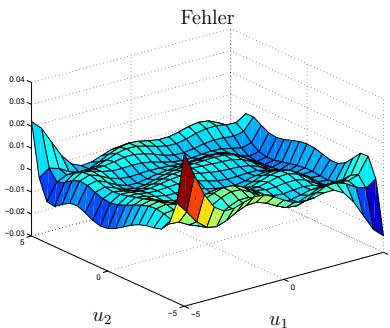
In Abb. 3.60 ist deutlich zu sehen, dass die vorgegebene nichtlineare Funktion sehr gut approximiert worden ist. Nur in den Randbereichen ist die Abweichung größer, da hier nicht ausreichend angeregt wurde.

In einer zweiten Simulation wird nun ein 3-schichtiges MLP-Netz mit der in Tabelle 3.2 angegebenen Struktur verwendet. In der ersten und zweiten versteckten Schicht sind jeweils  $5 + 2 = 7$  Neuronen enthalten. Zwischen den beiden Eingangsneuronen und den 7 Neuronen in der ersten versteckten Schicht gibt es  $2 \cdot 7 = 14$  Verbindungsgewichte. Zwischen der ersten und zweiten versteckten



**Abb. 3.58:** vorgegebene statische Nichtlinearität

**Abb. 3.59:** Identifikationsergebnis



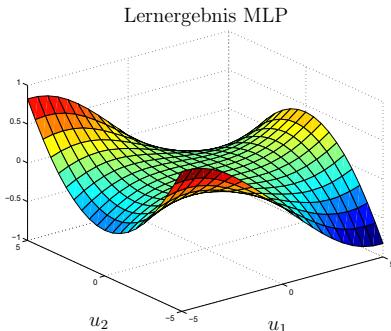
**Abb. 3.60:**  $\mathcal{NL}(u_1, u_2) - \widehat{\mathcal{NL}}(u_1, u_2)$

Schicht befinden sich  $7 \cdot 7 = 49$  Verbindungsgewichte, während die zweite verdeckte Schicht mit dem Ausgang über  $7 \cdot 1 = 7$  Verbindungsgewichte verknüpft ist. Insgesamt sind somit  $7 + 14 + 49 = 70$  Verbindungsgewichte vorhanden. Zu den Verbindungsgewichten kommen wieder die Biasgewichte. Jedes Neuron der verdeckten und der letzten Schicht hat ein eigenes Biasgewicht. Deshalb verfügt das MLP in diesem Beispiel über  $7 + 7 + 1 = 14$  Biasgewichte. Insgesamt gibt es demnach  $70 + 14 = 84$  Gewichte.

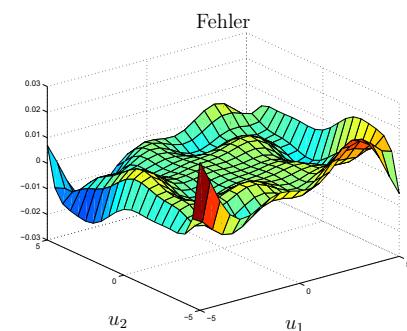
k	Anzahl $T(s) = \tanh(s)$	Anzahl $T(s) = c \cdot s$
1	5	2
2	5	2
3	0	1

**Tabelle 3.2:** verwendetes 3-schichtiges MLP-Netz

In Abb. 3.61 ist das Identifikationsergebnis, sowie in Abb. 3.62 die Abweichung zwischen der vorgegebenen nichtlinearen Funktion und dem Identifikationsergebnis, dargestellt.



**Abb. 3.61:** Identifikationsergebnis



**Abb. 3.62:**  $\mathcal{N}(u) - \widehat{\mathcal{N}}(u)$

Aus dem Vergleich zwischen Abb. 3.60 und Abb. 3.62 ist zu sehen, dass ähnlich gute Identifikationsergebnisse erzielt worden sind. Allerdings führt die Verwendung einer weiteren verdeckten Schicht zu einer deutlichen Parameterreduzierung.

## 3.11 Bewertung und Vergleich der Funktionsapproximatoren

### 3.11.1 Bewertung der Eigenschaften des GRNN und RBF–Netzwerks

RBF und GRNN sind stützstellenbasierte neuronalen Netze. Die lokale Wirkung der Stützstellen erlaubt eine anschauliche Interpretation der Netzwerkparameter. Zudem ist das Einbringen von Vorwissen durch Vorbelegung der Stützwerte möglich. Der Vorteil bei der Dimensionierung des GRNN gegenüber eines MLP Netzwerks liegt in der Tatsache, dass grundsätzlich nur zwei Schichten vorhanden sind. Die freien Parameter sind die Stützwertezahl  $p$ , der Glättungsfaktor  $\sigma$  und der Eingangsbereich innerhalb dessen die Stützwerte verteilt werden. Diese Größen können an die jeweilige Problemstellung leicht angepasst werden. Die Dimensionierung eines GRNN Netzwerks gestaltet sich im Vergleich zu Multilayer Netzwerken sehr einfach.

Der Vorteil des GRNN gegenüber dem RBF–Netzwerk liegt für regelungs-technische Anwendungen in den folgenden Punkten:

- Ist die zu approximierende Funktion zwischen 2 Stützwerten monoton, so wird dies im Fall der Verwendung eines GRNN auch in der Approximation abgebildet. Das RBF dagegen neigt zu Welligkeit zwischen 2 Stützstellen.

- Aufgrund dieser Eigenschaft kann die Stützstellenanzahl beim GRNN gewöhnlich kleiner als die eines RBF–Netzwerks gewählt werden.
- Beim Verlassen des Eingangsbereichs behält der Ausgang des GRNN den Wert des Stützwertes am Rand des Eingangsbereichs. Diese Extrapolationseigenschaft ist in vielen Fällen sinnvoll, vor allem bei Funktionen mit Sättigungscharakter.

Der Rechenaufwand sowohl für das Netzwerktraining als auch zur Auswertung steigt mit der Eingangsdimension und der Stützstellenanzahl gemäß Gleichung (3.23) an. Aus diesem Grund liegt das Haupteinsatzgebiet des GRNN und RBF–Netzwerks in der Approximation nichtlinearer Zusammenhänge mit 1 oder 2 Eingangsdimensionen.

### 3.11.2 Bewertung HANN

Das harmonisch aktivierte neuronale Netz (HANN) verwendet wie das GRNN Basisfunktionen zur Approximation unbekannter Nichtlinearitäten. Das HANN ist speziell auf die Approximation periodischer (zeitlich und örtlich) Funktionen ausgelegt. Dies zeigt sich in der Verwendung periodischer Basisfunktionen. Die Netzwerkgewichte entsprechen nach abgeschlossenem Lernvorgang den Fourierkoeffizienten der reellen frequenzdiskreten Fourierreihe. Durch diesen Zusammenhang lässt sich das HANN auch für Diagnosezwecke in mechatronischen Systemen einsetzen. Die Berücksichtigung weiterer Eingangsdimensionen (periodische oder nicht periodische Signale) ist zur mehrdimensionalen Funktionsapproximation möglich. Praktische Anwendungsbeispiele zur Schwingungsdämpfung in Kraftfahrzeugen finden sich in [17].

### 3.11.3 Bewertung LOLIMOT

Aus den speziellen Eigenschaften von LOLIMOT ergibt sich auch der bevorzugte Einsatzbereich. Durch die schnelle Parameterberechnung und den relativ geringen Anstieg der Parameteranzahl mit der Eingangsdimension des Netzes, ist LOLIMOT besonders für mehrdimensionale Identifikationsaufgaben ( $N \geq 2$ ) geeignet. Diese Eingangsdimension ist vor allem bei der Identifikation nichtlinearer dynamischer Systeme sehr schnell erreicht. Für ein sprungfähiges nichtlineares dynamisches System 2. Ordnung hat man z.B. schon  $N = 5$  Eingangsgrößen. Die Probleme, die bei der Identifikation nichtlinearer dynamischer Systeme auftreten, werden im Kapitel 8 genauer erläutert.

### 3.11.4 Bewertung MLP Netz

Mit einem MLP–Netz lassen sich nichtlineare statische Funktionen hoher Dimension sehr gut approximieren. Aus Regelungstechnischer Sicht ist das MLP–Netz

jedoch nur bedingt geeignet, da sich eine Änderung der Stützwerte nicht lokal sondern global auswirkt [193]. Zudem ist die Gestaltung einer optimalen Netzwerkstopologie schwierig.

### 3.11.5 Einsatzbereich der Netztypen

Netztyp	Interpolation	Extrapolation	Lernverfahren	Parameteranzahl
RBF	Gauß	fällt auf 0 ab	LS oder Gradientenverfahren	$M^N$
GRNN	normierter Gauß	konstant	LS oder Gradientenverfahren	$M^N$
MLP	abhängig von $\mathcal{T}$	nichtlinear	Gradientenverfahren	pro Schicht $(N_{l-1} + 1) \cdot N_l$
LOLIMOT	normierter Gauß	linear	LS oder Gradientenverfahren	$M \cdot (N + 1)$

**Tabelle 3.3:** Vergleich der einzelnen Netztypen

Netztyp	Einsatz
RBF	Identifikation von 1 oder 2 dimensionalen Fkt.
GRNN	Identifikation von 1 oder 2 dimensionalen Fkt.
MLP	Identifikation von höher dimensionalen Fkt.
LOLIMOT	Identifikation von höher dimensionalen Fkt.

**Tabelle 3.4:** Einsatzbereich der Netztypen

## 4 Lernen bei statischer Funktionsapproximation: Grundlagen

Die herausragende Eigenschaft Neuronaler Netze ist es, anhand von Trainingsdaten unbekannte Zusammenhänge zu approximieren und sie nach abgeschlossener Lernphase zu reproduzieren, sowie das gelernte Wissen auf ungelernte Eingangsdaten zu extrapolieren bzw. zu interpolieren. Hierfür besitzen Neuronale Netze eine begrenzte Anzahl an Parametern, die während der Lernphase eingestellt werden.

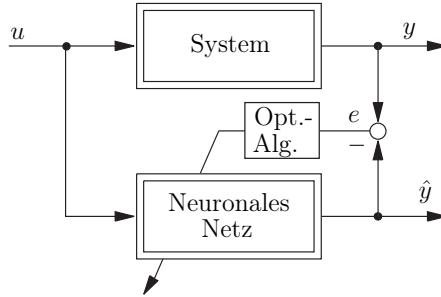
Unter Lernen in Neuronalen Netzen wird im allgemeinen die Modifikation dieser Parameter verstanden, um eine bessere Übereinstimmung zwischen erwarteter und tatsächlicher Ausgabe des Neuronalen Netzes zu erhalten.

Es wird zwischen drei Arten des Lernens unterschieden [248]

- Überwachtes Lernen
- Bestärkendes Lernen
- Unüberwachtes Lernen

Beim *überwachten* Lernen gibt ein externer „Lehrer“ (reales System) zu jedem Eingangssignal der Trainingsmenge das korrekte bzw. das beste Ausgangssignal dazu an. Aufgabe des Lernverfahrens ist es, die Parameter des Modells so zu verändern, dass das Modell nach Abschluss der Lernphase die Assoziation zwischen Ein- und Ausgang selbstständig auch für unbekannte, ähnliche Eingangssignale (Generalisierung) vornehmen kann. Diese Art des Lernens ist üblicherweise die schnellste Methode, ein Neuronales Netz für eine Aufgabe zu trainieren. Außerdem ist dieses Verfahren das einzige der drei oben benannten Verfahren, welches sich für eine Online-Identifikation eignet. Aus diesem Grund wird in diesem Beitrag einzig und alleine das überwachte Lernen verwendet, wobei in diesem Fall der externe „Lehrer“ das zu approximierende System ist. Die allgemeine Form des überwachten Lernens wird durch Abbildung 4.1 veranschaulicht. Diese Darstellung wird in der Literatur auch als Ausgangsfehleranordnung bezeichnet.

Beim *bestärkenden* Lernen gibt der Lehrer nicht die erwünschte Ausgabe an, sondern nur, ob die geschätzte Ausgabe richtig oder falsch ist. Eventuell wird zusätzlich noch der Grad der Richtigkeit mit angegeben. Jedoch sind keine Zielwerte für den Ausgang des Neuronalen Netzes vorhanden. Das Lernverfahren



**Abb. 4.1:** Prinzip des überwachten Lernens

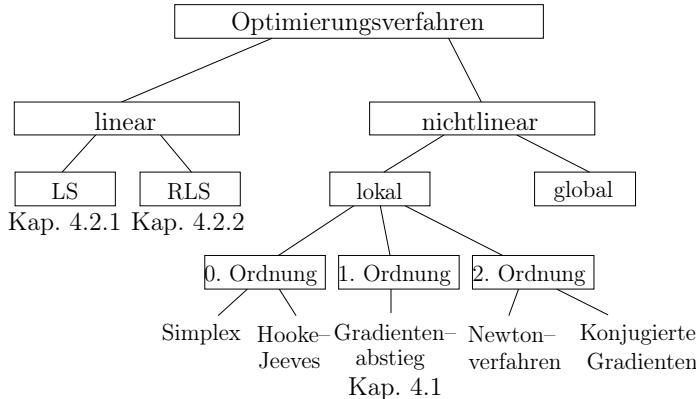
muss selbst die richtige Ausgabe finden. Diese Art des Lernverfahrens ist neurobiologisch plausibler, weil man einfache Rückkopplungsmechanismen der Umwelt (Bestrafung bei falscher Entscheidung, Belohnung bei richtiger) bei niederen und höheren Lebewesen beobachten kann. Dagegen haben bestärkende Lernverfahren den Nachteil, dass sie für Aufgaben, bei denen die erwünschte Ausgabe bekannt ist, viel länger zum Lernen brauchen als überwachte Lernverfahren, da weniger Informationen zur korrekten Modifikation der Parameter verwendet werden.

Beim *unüberwachten* Lernen gibt es überhaupt keinen externen Lehrer. Lernen geschieht durch Selbstorganisation. Dem Netz werden nur die Eingangsssequenzen präsentiert. Das bekannteste Beispiel unüberwachten Lernens sind die selbstorganisierten Karten bzw. die Kohonen-Netze [126].

Das überwachte Lernen, welches im Rahmen dieses Beitrags ausschließlich behandelt wird, kann nach [168] in drei Klassen eingeteilt werden: lineare, nichtlineare lokale und nichtlineare globale Optimierungsverfahren. In Abbildung 4.2 sind die verschiedenen Optimierungsverfahren im Überblick dargestellt.

Lineare Optimierungsverfahren können immer dann angewendet werden, wenn der Modellausgang linear in den Parametern ist. Als Gütfunktional<sup>1)</sup> wird die Summe des quadratischen Fehlers minimiert. Lineare Optimierungsverfahren führen immer zum globalen Minimum des Gütfunktionalen [13]. Bei den linearen Verfahren unterscheidet man nichtrekursive und rekursive Algorithmen. So stellt der Abschnitt 4.2.1 den bekannten Least-Squares-Algorithmus (LS) vor. Mit der für praktische Anwendungen interessanteren Variante des LS-Algorithmus, dem sogenannten rekursiven Least-Squares-Algorithmus (RLS) beschäftigt sich der Abschnitt 4.2.2. Diese Verfahren können bei den Gleichungsfehlermodellen — beispielsweise dem ARX-Modell (Kapitel 7.2) — angewendet werden, da bei Gleichungsfehlermodellen der Modellausgang linear in den Parametern eingeht. Nichtlineare Optimierungsverfahren müssen in den Fällen angewendet werden, in denen die unbekannten Parameter nichtlinear in den Modellausgang eingehen. Wenn der Modellausgang nichtlinear in den Parametern — wie beispielsweise

<sup>1)</sup> Es wird auch von der Verlustfunktion bzw. von der Kostenfunktion gesprochen.



**Abb. 4.2:** Optimierungsverfahren im Überblick

beim OE-Modell im Kapitel 7.2 — ist, kann beispielsweise das Gradientenabstiegsverfahren verwendet werden (siehe auch Abb. 4.2). Lokale Lernverfahren versuchen, ein Minimum des Gütfunktional, ausgehend von einem Startpunkt zu finden. Diese Verfahren führen zu einem Minimum in der Umgebung des Startpunktes, was häufig ein lokales und nicht ein globales Minimum ist. Die Methoden 0. Ordnung, wie z.B. die Simplex–Search–Methode oder der Hooke–Jeeves–Algorithmus, basieren ausschließlich auf der Auswertung des Gütfunktional [185]. Die partiellen Ableitungen des Gütfunktional nach den Parametern werden nicht benötigt, da ein globales Minimum des Gütfunktional entlang vordefinierter Richtungen gesucht wird. Die Verfahren 0. Ordnung sind einfach zu implementieren, jedoch haben sie eine langsame Konvergenz. Können die partiellen Ableitungen bestimmt werden, ist es sinnvoller ein gradientenbasiertes Optimierungsverfahren zu verwenden. Zu den Verfahren 1. Ordnung zählt der Gradientenabstieg mit seinen vielen Varianten. Zu den Optimierungsverfahren 2. Ordnung gehören die Newton–Verfahren und die Verfahren mit Konjugierten Gradienten. Die Verfahren 2. Ordnung verwenden neben der 1. Ableitung auch die 2. Ableitung der Kostenfunktion. Durch diese zusätzliche Information arbeiten die Verfahren 2. Ordnung erheblich effizienter als der einfache Gradientenabstieg [44]. Bei den Newton–Verfahren setzen das Gauß–Newton–Verfahren und das Levenberg–Marquardt–Verfahren ein quadratisches Gütfunktional voraus, was zu einer einfachen Approximation für die sonst aufwendig zu berechnende 2. Ableitung führt. Nichtlineare globale Verfahren beinhalten in der Regel stochastische Elemente um dem Algorithmus die Möglichkeit zu geben, aus lokalen Minima zu entkommen und ein globales Minimum zu finden. Die Fähigkeit aus lokalen Minima entkommen zu können wird mit einem hohen Rechenaufwand und einer langsamen Konvergenz bezahlt. Typische nichtlineare globale Optimierungsverfahren sind das Simulated Annealing sowie evolutionäre und genetische

Algorithmen. Da die Kostenfunktion bei der nichtlinearen Optimierung von sehr starker Symmetrie geprägt ist, treten sehr viele gleich gute globale Minima auf. Deshalb ist es meist nicht sinnvoll, für derartige Optimierungsprobleme ein globales Verfahren anzuwenden.

Einen tieferen Einblick in die Parameteroptimierungsverfahren geben [44, 111, 112, 141, 168, 173, 216] und Kapitel 10, in dem alle Verfahren von Abb. 4.2 detailliert dargestellt werden. Generell sei darauf hingewiesen, daß bei der Identifikation das zu identifizierende System möglichst **ungestört** ist.

## 4.1 Gradientenabstiegsverfahren

Wie schon angemerkt, muss das Gradientenabstiegsverfahren eingesetzt werden, wenn der Modellausgang nichtlinear in den Parametern ist — beispielsweise beim OE-Modell (Kapitel 7.2). Die Adaption der Gewichte der statischen Neuronalen Netze erfolgt mit Hilfe eines Lernalgorithmus, dem sogenannten Lerngesetz. Das wohl bekannteste Lernverfahren stellt das Gradientenverfahren dar.

Durch das Lernverfahren sollen die Parameter bzw. die Gewichte des Neuronalen Netzes so angepasst werden, dass die Abweichung zwischen dem Ausgang  $y$  des zu identifizierenden Systems und dem Ausgang  $\hat{y}$  des Neuronalen Netzes minimiert wird.

Diese Abweichung zwischen wahren und geschätztem Wert wird als Ausgangsfehler

$$e(\hat{\Theta}) = (y - \hat{y}(\hat{\Theta})) \quad (4.1)$$

bezeichnet.

Ausgangspunkt für die folgenden Überlegungen ist das quadratische Fehlermaß  $E(\hat{\Theta})$ :

$$E(\hat{\Theta}) = \frac{1}{2} e^2(\hat{\Theta}) = \frac{1}{2} (y - \hat{y}(\hat{\Theta}))^2 \quad (4.2)$$

Die Einführung des Faktors  $\frac{1}{2}$  ist für die Adaption der Gewichte unerheblich, führt jedoch zu einem übersichtlicheren Lerngesetz.

Das Ziel des Lernverfahrens ist, Gleichung (4.2) bezüglich der Parameter  $\hat{\Theta}$  zu minimieren. Da im allgemeinen  $E(\hat{\Theta})$  nicht analytisch vorliegt, ist man auf eine iterative Lösung angewiesen [176, 44].

Die grundsätzliche algorithmische Struktur besteht aus nachfolgenden Schritten und wird in Abbildung 4.3 für den zweidimensionalen Fall illustriert.

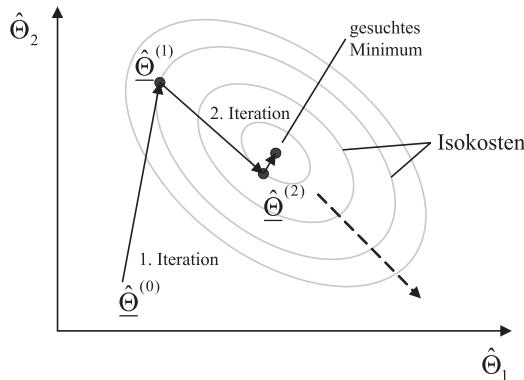
- (1) Wahl eines Startpunktes  $\hat{\Theta}^{(0)}$  und Festlegung des Iterationsindexes zu  $l = 0$ .
- (2) Bestimmung einer Suchrichtung  $s^{(l)}$
- (3) Bestimmung einer skalaren Schrittweite  $\eta^{(l)} > 0$  durch Lösung des folgenden eindimensionalen Minimierungsproblems

$$\min_{\eta>0} E \left( \hat{\underline{\Theta}}^{(l)} + \eta^{(l)} \underline{s}^{(l)} \right) \quad (4.3)$$

anschließend ergibt sich der  $(l+1)$ -te Parametervektor aus

$$\hat{\underline{\Theta}}^{(l+1)} = \hat{\underline{\Theta}}^{(l)} + \eta^{(l)} \underline{s}^{(l)} \quad (4.4)$$

- (4) Ist ein geeignetes Abbruchkriterium erfüllt, dann **stop**. Ansonsten:
- (5) Beginne mit einer neuen Iteration  $l := l + 1$  und Rücksprung nach (2).



**Abb. 4.3:** Iterative Suche eines lokalen Minimums

Das in [176] beschriebene Gradientenabstiegsverfahren verwendet für die Suchrichtung am jeweiligen Iterationspunkt die Richtung des steilsten Abstiegs, also die negative Gradientenrichtung  $-\underline{g}(\hat{\underline{\Theta}})$

$$\underline{s}^{(l)} = -\frac{\partial E(\hat{\underline{\Theta}}^{(l)})}{\partial \hat{\underline{\Theta}}} = -\underline{g}(\hat{\underline{\Theta}}) \quad (4.5)$$

Es sei an dieser Stelle ausdrücklich darauf hingewiesen, dass die Suchrichtung nicht immer der negative Gradient sein muss. Beispielsweise verwenden die leistungsfähigen Optimierungsverfahren 2. Ordnung im Kapitel 10 nur beim ersten Optimierungsschritt die Richtung des negativen Gradienten. Alle anderen Suchrichtungen weichen vom negativen Gradienten ab. In der in diesem Kapitel verwendeten Version des Gradientenabstiegsverfahrens wird auf die Bestimmung der skalaren Schrittweite  $\eta$  für jeden Iterationsschritt verzichtet. Es wird eine geeignet gewählte Schrittweite  $\eta$  als konstant angesetzt. Diese Schrittweite wird oft auch als Lernschrittweite oder auch Lernfaktor bezeichnet.

Für die Änderung der Gewichte des Neuronalen Netzes ergibt sich somit in zeitdiskreter Schreibweise folgendes Lerngesetz

$$\hat{\underline{\Theta}}[k+1] = \hat{\underline{\Theta}}[k] - \eta \frac{\partial E(\hat{\underline{\Theta}}[k])}{\partial \hat{\underline{\Theta}}} \quad (4.6)$$

In zeitkontinuierlicher Form lautet das Lerngesetz

$$\frac{d\hat{\underline{\Theta}}}{dt} = -\eta' \frac{\partial E(\hat{\underline{\Theta}})}{\partial \hat{\underline{\Theta}}} \quad (4.7)$$

wobei die Lernschrittweite  $\eta'$  der mit der Abtastzeit gewichteten Lernschrittweite für den zeitdiskreten Fall entspricht.

Für die statischen Neuronalen Netze unterscheiden sich die Lerngesetze nur in der Berechnung der Suchrichtung.

#### 4.1.1 Lerngesetz für das RBF und GRNN-Netz

Ausgehend von den Gleichungen (3.8) bzw. (3.16) für den Schätzwert  $\hat{y}$  am Ausgang eines RBF-Netzes

$$\hat{y}(\underline{u}) = \sum_{i=1}^p \hat{\Theta}_i \mathcal{A}_i(\underline{u}) \quad (4.8)$$

wird zur Adaption der Gewichte der bereits oben eingeführte Lernfehler  $e$  bzw. der daraus abgeleitete quadratische Fehler  $E$  herangezogen.

$$e(\underline{u}) = \sum_{i=1}^p \hat{\Theta}_i \mathcal{A}_i(\underline{u}) - y(\underline{u}) \quad (4.9)$$

$$E(\underline{u}) := \frac{1}{2} e^2 = \frac{1}{2} \left( \sum_{i=1}^p \hat{\Theta}_i \mathcal{A}_i(\underline{u}) - y(\underline{u}) \right)^2 \quad (4.10)$$

Die notwendige Änderung jedes Gewichts  $\hat{\Theta}_i$  wird durch ein Gradientenabstiegsverfahren (auch Delta-Lernregel genannt [197]) festgelegt. Dazu wird der quadratische Fehler nach Gleichung (4.10) nach dem jeweiligen Gewicht abgeleitet.

$$\frac{dE(\underline{u})}{d\hat{\Theta}_i} = \left( \sum_{k=1}^p \hat{\Theta}_k \mathcal{A}_k(\underline{u}) - y(\underline{u}) \right) \mathcal{A}_i(\underline{u}) = e(\underline{u}) \mathcal{A}_i(\underline{u}) \quad (4.11)$$

Somit bestimmen sich die notwendigen Änderungen der Gewichte zueinander wie es dem Beitrag jedes Gewichts zum Schätzwert  $\hat{y}$  und damit zum Lernfehler  $e$  entspricht. Eine zusätzliche Skalierung mit einem Lernfaktor  $\eta$  dient der Einstellung einer gewünschten Lerngeschwindigkeit bzw. Glättungswirkung bei

der Adaption. Das negative Vorzeichen stellt eine Anpassung der Gewichte in Richtung kleinerer Fehler sicher. Das vollständige Lerngesetz für jedes Gewicht lautet damit nach Gleichung (4.7)

$$\frac{d}{dt} \hat{\Theta}_i = -\eta e \mathcal{A}_i \quad (4.12)$$

Dadurch wird der quadratische Fehler minimiert. Für RBF-Netze, die mit dem Gradientenlerngesetz aus Gleichung (4.12) adaptiert werden, erfolgt in Abschnitt 4.1.1.2 ein Stabilitätsnachweis, sowie die Definition geeigneter Anregungssignale, die Parameterkonvergenz von geschätzter und realer Nichtlinearität garantieren.

#### 4.1.1.1 Gradientenverfahren mit Momentum Term

Bei flachen Plateaus in der Fehlerfläche  $e(\underline{u})$  (vergleiche dazu Abschnitt 4.1.4) kann es beim Gradientenverfahren nach Gleichung (4.12) zu sehr langen Konvergenzzeiten kommen, da der Gradient nur sehr kleine Werte annimmt. Dieser Effekt kann durch Einführung eines Momentum Terms im Lerngesetz verringert werden. Zu diesem Zweck wird das Lerngesetz zunächst zeitdiskretisiert, wobei  $k$  den jeweils aktuellen Abtastschritt kennzeichnet.

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] - \eta e[k] \mathcal{A}[k] = \hat{\Theta}[k] + \Delta \hat{\Theta}[k] \quad (4.13)$$

Der Momentum Term berücksichtigt nun zusätzlich Gewichtsänderungen aus den zurückliegenden Abtastschritten. Mit dem Skalierungsfaktor  $\alpha$  ( $0 < \alpha < 1$ ) lautet das modifizierte Lerngesetz:

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] \underbrace{- \eta e[k] \mathcal{A}[k] + \alpha \Delta \hat{\Theta}[k-1]}_{\Delta \hat{\Theta}[k]} \quad (4.14)$$

Durch Verwendung des Momentum Terms werden Gradienten aus vergangenen Abtastperioden so lange aufsummiert (Integration im Zeitbereich) und zum aktuellen Gradienten addiert, bis sich eine Vorzeichenumkehr des Gradienten ergibt. Zur Stützwerteadaption wird somit die Summe zurückliegender Gradienten verwendet, anstelle nur des aktuellen. Dies bewirkt an flachen Stellen der Fehlerfunktion eine Beschleunigung des Lernvorgangs.

Die folgenden Stabilitäts- und Konvergenzbetrachtungen beziehen sich wieder auf das einfache Gradientenverfahren ohne Momentum Term, da dies im Zeitkontinuierlichen das am häufigsten verwendete Verfahren darstellt.

#### 4.1.1.2 Stabilität nach Lyapunov

Die Stabilität eines Systems, das durch die nichtlineare Differentialgleichung [158]

$$\frac{d}{dt} \underline{x} = \underline{f}(\underline{x}, t) \quad (4.15)$$

beschrieben wird, ist nach Lyapunov wie folgt definiert:

**Definition:** Der Gleichgewichtszustand  $\underline{x}_0 = 0$  des Systems in Gl. (4.15) wird als *stabil* bezeichnet, wenn für jedes  $\varepsilon > 0$  und  $t_0 \geq 0$  ein  $\delta$  existiert, so daß aus  $\|\underline{x}(t_0)\| < \delta$  folgt  $\|\underline{x}(t)\| < \varepsilon$  für alle  $t \geq t_0$ .

Anschaulich gesprochen folgt aus einer kleinen Störung stets eine kleine Abweichung vom Gleichgewichtszustand, bzw. die Funktion  $\underline{x}$  bleibt nahe am Ursprung  $0$ , wenn ihr Anfangswert nur mit genügend kleinem Abstand zum Ursprung gewählt wird.

Unter Verwendung des Parameterfehlers  $\underline{\Phi} = \hat{\underline{\Theta}} - \underline{\Theta}$  aus Gl. (3.6) lässt sich die Stabilität des oben hergeleiteten Lerngesetzes nach Lyapunov beweisen. Für die Ableitung des Parameterfehlers gilt mit den Gleichungen (4.12), (3.5) und (3.6):

$$\frac{d}{dt} \underline{\Phi} = \frac{d}{dt} \hat{\underline{\Theta}} = -\eta \underline{\mathcal{A}} \underbrace{\left( \hat{\underline{\Theta}}^T - \underline{\Theta}^T \right)}_{-e(\hat{\underline{\Theta}}), \text{ siehe Gl. (4.1)}} \underline{\mathcal{A}} = -\eta \underline{\mathcal{A}} \underline{\Phi}^T \underline{\mathcal{A}} \quad (4.16)$$

Mit der Beziehung aus Gleichung (3.7) lässt sich diese Gleichung umformen zu

$$\frac{d}{dt} \underline{\Phi} = -\eta \underline{\mathcal{A}} \underline{\Phi}^T \underline{\mathcal{A}} = -\eta \underline{\mathcal{A}} \underline{\mathcal{A}}^T \underline{\Phi} \quad (4.17)$$

Als Lyapunov-Funktion  $V$  wird die positiv definite Funktion

$$V(\underline{\Phi}) = \frac{1}{2} \underline{\Phi}^T \underline{\Phi} \quad (4.18)$$

gewählt. Ihre zeitliche Ableitung entlang der durch Gl. (4.16) festgelegten Trajektorien bestimmt sich mit positivem Lernfaktor  $\eta$  zu

$$\begin{aligned} \frac{d}{dt} V(\underline{\Phi}) &= \frac{1}{2} 2 \frac{d}{dt} (\underline{\Phi}^T) \underline{\Phi} = (-\eta \underline{\mathcal{A}} \underline{\mathcal{A}}^T \underline{\Phi})^T \underline{\Phi} = -\eta (\underline{\mathcal{A}} \underline{\mathcal{A}}^T \underline{\Phi})^T \underline{\Phi} \quad (4.19) \\ &= -\eta \underbrace{\underline{\mathcal{A}}^T \underline{\Phi}}_e \underbrace{\underline{\mathcal{A}}^T \underline{\Phi}}_e = -\eta e^2 \leq 0 \end{aligned}$$

Damit ist  $dV/dt$  negativ semidefinit, wodurch die Beschränktheit des Parameterfehlers  $\underline{\Phi}$  und die Stabilität des obigen Systems nach Lyapunov gezeigt ist.

#### 4.1.1.3 Parameterkonvergenz

Neben der Stabilität nach Lyapunov ist auch die *asymptotische* Stabilität des betrachteten Lernvorgangs von Interesse, da erst diese die Parameterkonvergenz und damit eine sinnvolle Interpretierbarkeit der adaptierten Gewichte ermöglicht.

**Definition:** Der Gleichgewichtszustand  $\underline{x}_0 = 0$  des Systems in Gl. (4.15) wird als *asymptotisch stabil* bezeichnet, wenn er stabil ist und ein  $\delta > 0$  gewählt werden kann, so dass aus  $\|\underline{x}(t_0)\| < \delta$  folgt  $\lim_{t \rightarrow \infty} \underline{x}(t) = 0$ .

Für diesen Nachweis ist eine ausreichende Anregung (sogenannte *Persistent Excitation*) notwendig. Eine solche Anregung liegt vor, wenn für alle Einheitsvektoren  $v_i$ ,  $1 \leq i \leq p$ , die den  $p$ -dimensionalen Raum  $\mathbb{R}^p$  aufspannen, und für jedes positive  $\varepsilon_0$  und  $t_0$  ein endliches Zeitintervall  $T$  gefunden werden kann, so dass gilt

$$\frac{1}{T} \int_t^{t+T} |\underline{\mathcal{A}}^T(u) v_i| d\tau \geq \varepsilon_0 \quad \text{für alle } t \geq t_0 \quad (4.20)$$

Anschaulich gesprochen heißt dies, dass die Aktivierung für jedes Neuron nie dauerhaft zu Null werden darf, so dass sich jeder Parameterfehler stets über den Lernfehler  $e$  auswirkt und  $dV/dt$  somit negativ definit ist, solange keine vollständige Parameterkonvergenz erreicht ist.

Damit strebt die gewählte Lyapunov-Funktion  $V$  asymptotisch zu Null und damit auch der Parameterfehler.

$$\lim_{t \rightarrow \infty} V(t) = 0 \quad (4.21)$$

$$\lim_{t \rightarrow \infty} \underline{\Phi}(t) = \underline{\Theta} \quad (4.22)$$

Dadurch wird bei ausreichender Anregung die Konvergenz der Parameter erreicht:

$$\lim_{t \rightarrow \infty} \hat{\Theta}(t) = \underline{\Theta} \quad (4.23)$$

#### 4.1.1.4 Fehlermodell 1 für das Gradientenabstiegsverfahren

Ausgehend von dem oben hergeleiteten Lerngesetz werden in der Literatur verschiedene Fehlermodelle unterschieden, deren Einsatz im wesentlichen dadurch bestimmt ist, ob der messbare Lernfehler direkt oder nur indirekt bzw. verzögert vorliegt.

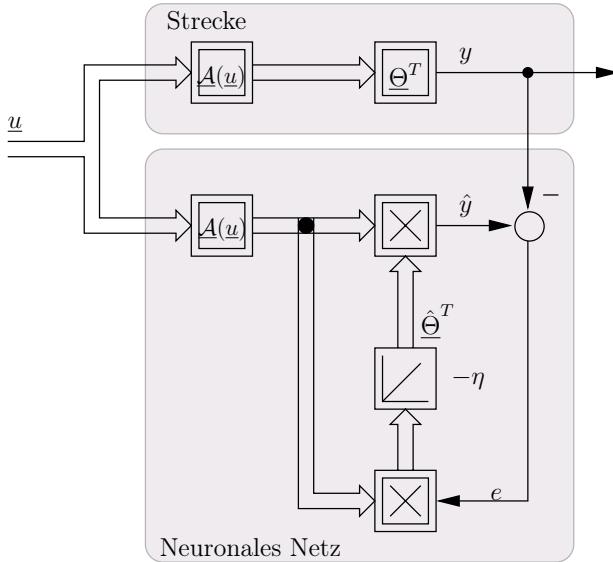
Bei direktem Vorliegen des Lernfehlers  $e$  kann nach Gleichung (4.12) die Struktur und die Parametrierung des sogenannten Fehlermodells 1 realisiert werden, wie in Abb. 4.4 für die Identifikation gezeigt<sup>2)</sup>. Die Nichtlinearität der Strecke wird dabei aus Gründen der Anschaulichkeit als Skalarprodukt nach Gl. (3.3) dargestellt.

#### 4.1.2 Lerngesetz für das HANN

Da die Funktionsapproximation mit harmonischen Basisfunktionen analog zu der mit radialen Basisfunktionen bei RBF-Netzen darstellbar ist, gilt die Herleitung des Lerngesetzes nach Abschnitt 4.1.1 sinngemäß auch für das HANN und wird daher nicht nochmals ausgeführt. Mit dem oben definierten Lernfehler  $e$

---

<sup>2)</sup> Die weiteren Fehlermodelle 2–4 werden verwendet, wenn der Ausgang der Nichtlinearität nicht direkt messbar ist. Diese weiteren Fehlermodelle werden in Kapitel 5 ausführlich behandelt.



**Abb. 4.4:** Identifikationsstruktur nach Fehlermodell 1

$$e(\varphi) = \hat{\Theta}_A^T \underline{A}(\varphi) + \hat{\Theta}_B^T \underline{A}_B(\varphi) - y(\varphi) \quad (4.24)$$

ergibt sich das Lerngesetz mit dem positiven Lernfaktor  $\eta$  zu

$$\begin{aligned} \frac{d}{dt} \hat{\Theta}_A &= -\eta e \underline{A}(\varphi) \\ \frac{d}{dt} \hat{\Theta}_B &= -\eta e \underline{A}_B(\varphi) \end{aligned} \quad (4.25)$$

Im Unterschied zu RBF-Netzen kann die Aktivierung beim HANN auch negative Werte annehmen. Die Stabilitätsbetrachtung kann jedoch wie beim RBF-Netz durchgeführt werden.

Die Lernstruktur und die verwendeten Fehlermodelle beim HANN sind aufgrund des ähnlichen Lerngesetzes von allgemeinen RBF-Netzen abgeleitet, wie sie in Abschnitt 4.1.1.4 dargestellt sind. Alle Ausführungen aus Abschnitt 4.1.1.4 sind auch für das HANN gültig.

### Stabilität und Parameterkonvergenz beim HANN

Für das HANN kann analog zu Kapitel 4.1.1.2 die Stabilität des Lernens nach Lyapunov nachgewiesen werden, wenn für die Stabilitätsbetrachtungen die entsprechenden Vektoren für gerade und ungerade Anteile wie oben jeweils zu einem einzigen Vektor  $\underline{\Phi}$  und  $\underline{A}$ , zusammengefasst werden.

Für den Fall eines ausreichend angeregten (*persistently excited*) HANN kann aus der Stabilität auch auf die Parameterkonvergenz geschlossen werden. Dies soll

für einen herausgegriffenen Parameterfehler  $\Phi_{Ai}$  gezeigt werden. Das zugehörige Lerngesetz ergibt sich nach Einsetzen der Fehlergleichung (3.30) in Gl. (4.25). Der besseren Übersichtlichkeit wegen wird der Gleichanteil bei der Herleitung nicht betrachtet.

$$\begin{aligned} \frac{d}{dt} \Phi_{Ai} &= -\eta e \mathcal{A}_{Ai}(\varphi) \\ &= -\eta \left( \sum_{k=1}^K \Phi_{Ak} \mathcal{A}_{Ak}(\varphi) + \Phi_{Bk} \mathcal{A}_{Bk}(\varphi) \right) \mathcal{A}_{Ai}(\varphi) \quad (4.26) \end{aligned}$$

Durch Integration von Gl. (4.26) erhält man den Parameterfehler  $\Phi_{Ai}$ . Aufgrund der Orthogonalität aller Elemente der Aktivierungsvektoren gilt für alle  $i \in \mathbb{N}_0$  und  $k \in \mathbb{N}$  bei Integration über eine oder mehrere Perioden der Länge  $2\pi$ :

$$\begin{aligned} \int_0^{2\pi} \mathcal{A}_{Ai}(\varphi) \mathcal{A}_{Ak}(\varphi) d\varphi &= 0 \quad \text{für } i \neq k \\ \int_0^{2\pi} \mathcal{A}_{Bi}(\varphi) \mathcal{A}_{Bk}(\varphi) d\varphi &= 0 \quad \text{für } i \neq k \\ \int_0^{2\pi} \mathcal{A}_{Ai}(\varphi) \mathcal{A}_{Bk}(\varphi) d\varphi &= 0 \quad \text{für beliebige } i, k \end{aligned} \quad (4.27)$$

Wird nun Gl. (4.26) ebenfalls über ein ganzzahliges Vielfaches der Periodendauer des Signals  $y(\varphi)$  ausgewertet, d.h. mit der zulässigen Näherung  $d\varphi/dt = const$  integriert, fallen alle Produkte ungleicher Elemente der Aktivierungsvektoren heraus; daher kann Gl. (4.26) wie folgt vereinfacht werden

$$\frac{d}{dt} \Phi_{Ai} = -\eta \Phi_{Ai} \mathcal{A}_{Ai}(\varphi) \mathcal{A}_{Ai}(\varphi) = -\eta \Phi_{Ai} \mathcal{A}_{Ai}^2(\varphi)$$

Bei ausreichender Anregung nach [158], d.h. wenn eine vollständige Periode des zu identifizierenden Signals stets in einer begrenzten Zeit  $T$  durchlaufen wird, gilt

$$\int_0^T \mathcal{A}_{Ai}^2 dt > 0$$

Der Lernfaktor  $\eta$  ist auch entsprechend dieser maximalen Periodendauer und des Energiegehalts der bei der Anregung nicht berücksichtigten Signalanteile zu wählen. Damit streben alle Parameterfehler asymptotisch gegen Null

$$\lim_{t \rightarrow \infty} \Phi_{Ai}(t) = 0 \quad \text{und} \quad \lim_{t \rightarrow \infty} \Phi_{Bi}(t) = 0$$

und damit auch die Stützwerte gegen die Fourierkoeffizienten der zu identifizierenden periodischen Funktion. Damit sind die Stabilität der Identifikation sowie

die Parameterkonvergenz beim HANN gewährleistet. Aufgrund der reellen Darstellung ergibt sich dabei für den identifizierten Gleichanteil  $\hat{\Theta}_{A0}$  ein Umrechnungsfaktor gegenüber dem entsprechenden Fourierkoeffizienten  $a_0$ .

$$\lim_{t \rightarrow \infty} \hat{\Theta}_{Ai}(t) = \Theta_{Ai} = a_i \quad \text{für } i \in \mathbb{N}$$

$$\lim_{t \rightarrow \infty} \hat{\Theta}_{Bi}(t) = \Theta_{Bi} = b_i \quad \text{für } i \in \mathbb{N}$$

$$\lim_{t \rightarrow \infty} \hat{\Theta}_{A0}(t) = \Theta_{A0} = \frac{a_0}{2}$$

Das Identifikationsergebnis des HANN entspricht im vollständig gelernten Zustand somit den Koeffizienten der Fourierreihe. Dies bedeutet insbesondere, dass das HANN einen *minimalen Approximationsfehler* anstrebt und die identifizierten Parameter wegen der Orthogonalität der Spektralkomponenten auch *eindeutig* sind. Schließlich ist die Fourierreihe – und damit die Darstellung durch das HANN – die *kompakteste Repräsentation* eines periodischen und bandbegrenzten Signals  $y(\varphi)$ .

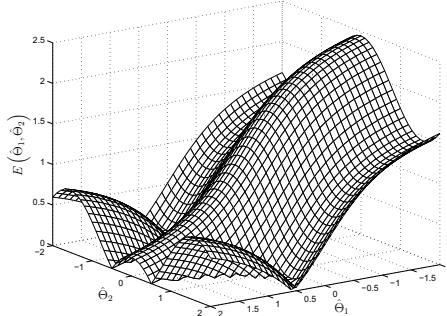
#### 4.1.3 Lerngesetz für mehrschichtige Netze

In diesem Abschnitt wird das zuvor behandelte Gradientenabstiegsverfahren so erweitert, dass auch mehrschichtige (in der Regel das MLP-Netz) Netze trainiert werden können.

Ausgangspunkt ist wie im Abschnitt 4.1 das quadratische Fehlermaß des Ausgangsfehlers. Dieses Fehlermaß kann als Funktion der Gewichte des Netzwerkes aufgefasst werden. Für mehr als ein Gewicht erhält man also eine Fehlerfläche. Für zwei Gewichte lässt sich diese Fehlerfläche — wie in Abbildung 4.5 gezeigt — noch graphisch anschaulich darstellen. Bei diesem Bild handelt es sich um eine typische Fehlerfläche, die entsteht, wenn bei mehrschichtigen Perzeptronennetzen zwei Gewichte (in Abbildung 4.5 mit  $\hat{\Theta}_1$  und  $\hat{\Theta}_2$  bezeichnet) in den versteckten Schichten verändert werden. Zu jeder Gewichtskombination wird die Kostenfunktion berechnet und der Wert als Fehlerflächenpunkt eingetragen. Dieses Fehlermaß, bzw. die Fehlerfunktion,  $E(\hat{\Theta}) = E(\hat{\Theta}^1, \underline{b}^1, \hat{\Theta}^2, \underline{b}^2, \dots, \hat{\Theta}^l, \underline{b}^l, \dots, \hat{\Theta}^L, \underline{b}^L)$  gibt den Fehler an, den das mehrschichtige neuronale Netz für die gegebenen Matrizen der Verbindungsgewichte  $\hat{\Theta}^1, \hat{\Theta}^2, \dots, \hat{\Theta}^l, \dots, \hat{\Theta}^L$  und die Vektoren der Biasgewichte  $\underline{b}^1, \underline{b}^2, \dots, \underline{b}^l, \dots, \underline{b}^L$  besitzt, wobei  $\hat{\Theta}^l$  und  $\underline{b}^l$  allgemein die Verbindungsgewichte und die Biasgewichte der  $l$ -ten Schicht enthalten. Die Verbindungsgewichte und Biasgewichte aller Schichten werden zu einem einzigen Gewichtsvektor  $\underline{\hat{\Theta}}$  zusammengefasst (für die Optimierung der Parameter ist die Position innerhalb des Netzes irrelevant).

##### 4.1.3.1 Herleitung der Backpropagation-Regel

Der Begriff Backpropagation steht in der Literatur häufig synonym für das Gradientenabstiegsverfahren. Streng genommen ist Backpropagation jedoch kein Lern-



**Abb. 4.5:** Fehlerfläche eines neuronalen Netzes als Funktion der Gewichte  $\hat{\Theta}_1$  und  $\hat{\Theta}_2$

verfahren, sondern eine Rechenvorschrift, mit der der Fehler am Netzausgang durch das MLP über die einzelnen Schichten zurückgerechnet werden kann, um den Gradienten  $\nabla E(\hat{\Theta}) = dE/d\hat{\Theta}$  der quadratischen Fehlerfunktion  $E(\hat{\Theta})$  zu bestimmen.

Im Folgenden wird nun der Backpropagation Algorithmus [81, 44] für das in Abschnitt 3.10 eingeführte MLP hergeleitet. Wie aus diesem Abschnitt bekannt ist, übergibt der Ausgang einer Schicht die Signale zum Eingang der nächsten Schicht. Zur Erinnerung sei nochmals die Simulationsgleichung (3.43) in allgemeiner Form angegeben:

$$\underline{y}^l = \underline{\mathcal{T}}^l(\hat{\Theta}^l \cdot \underline{y}^{l-1} + \underline{b}^l) = \underline{\mathcal{T}}^l(\underline{s}^l) \quad \text{für } l = 1, 2, \dots, L \quad (4.28)$$

Hierbei bezeichnet  $L$  die Gesamtzahl der Schichten des MLP. Die Neuronen der Eingangsschicht empfangen die externe Eingabe

$$\underline{y}^0 = \underline{u}, \quad (4.29)$$

welche den Startwert für Gleichung (4.28) liefert. Die Ausgänge der Neuronen in der letzten Schicht ergeben den Ausgangsvektor  $\hat{y}$  des MLP:

$$\hat{y} = \underline{y}^L \quad (4.30)$$

Das quadratische Fehlermaß  $E(\hat{\Theta})$  berechnet sich analog zu Gleichung (4.2) aus dem Abbildungsfehler  $\underline{e}$  zwischen dem Ausgang  $\underline{y}$  der Nichtlinearität und dem Netzausgang  $\hat{y}$ .

$$E(\hat{\Theta}) = \frac{1}{2} \underbrace{(\underline{y} - \hat{y}(\hat{\Theta}))^T}_{\underline{e}} \cdot \underbrace{(\underline{y} - \hat{y}(\hat{\Theta}))}_{\underline{e}} = \frac{1}{2} \underline{e}^T \cdot \underline{e} \quad (4.31)$$

Wie aus Gleichung (4.31) hervorgeht, ist bei einem mehrschichtigen Netz das Fehlermaß  $E(\hat{\Theta})$  keine explizite Funktion der Netzparameter. Zur Berechnung

der partiellen Ableitungen des Fehlermaßes nach den Netzparametern muss deshalb wiederholt die Kettenregel der Differentialrechnung zum Einsatz kommen. Die Herleitung des Backpropagation Algorithmus ist nicht schwierig aber aufwendig. Zum besseren Verständnis soll deshalb hier eine sehr ausführliche Form der Herleitung gebracht werden, welche den direkten Übergang in Matrix-Vektor Notation ermöglicht. Betrachten wir zunächst die noch näher zu bestimmende partielle Ableitung  $\frac{\partial E}{\partial \hat{\Theta}_{i,j}^l}$  von der Kostenfunktion zum Summenausgang  $s_i^l$  (siehe Abbildung 3.48) des  $i$ -ten Neurons in der Schicht  $l$  als Zwischenschritt für die gesuchte Ableitung. Dann kann nach der Kettenregel die Ableitung der Kostenfunktion nach einem Verbindungsgewicht  $\Theta_{i,j}^l$  bzw. nach einem Biasgewicht  $b_i^l$  (jeweils des  $i$ -ten Neurons in der allgemeinen Schicht  $l$ ) wie folgt dargestellt werden:

$$\begin{aligned}\frac{\partial E}{\partial \hat{\Theta}_{i,j}^l} &= \frac{\partial E}{\partial s_i^l} \cdot \frac{\partial s_i^l}{\partial \hat{\Theta}_{i,j}^l} \\ \frac{\partial E}{\partial b_i^l} &= \frac{\partial E}{\partial s_i^l} \cdot \frac{\partial s_i^l}{\partial b_i^l}\end{aligned}\quad (4.32)$$

Der zweite Term dieser Gleichungen ist leicht zu berechnen, da der Summenausgang  $s_i^l$  der Schicht  $l$  direkt von den Gewichten dieser Schicht abhängt:

$$s_i^l = \sum_{k=1}^{N_{l-1}} \hat{\Theta}_{i,k}^l y_k^{l-1} + b_i^l. \quad (4.33)$$

Somit gilt

$$\frac{\partial s_i^l}{\partial \hat{\Theta}_{i,j}^l} = y_j^{l-1}, \quad \frac{\partial s_i^l}{\partial b_i^l} = 1. \quad (4.34)$$

Durch die Definition der lokalen Gradienten

$$\delta_i^l = \frac{\partial E}{\partial s_i^l}, \quad (4.35)$$

können die Gleichungen (4.32) vereinfacht werden zu

$$\begin{aligned}\frac{\partial E}{\partial \hat{\Theta}_{i,j}^l} &= \delta_i^l \cdot y_j^{l-1} \\ \frac{\partial E}{\partial b_i^l} &= \delta_i^l.\end{aligned}\quad (4.36)$$

Diese Zusammenhänge ergeben in Vektorschreibweise:

$$\begin{aligned}\frac{\partial E}{\partial \hat{\Theta}^l} &= \underline{\delta}^l \cdot (\underline{y}^{l-1})^T \\ \frac{\partial E}{\partial \underline{b}^l} &= \underline{\delta}^l,\end{aligned}\quad (4.37)$$

mit dem lokalen Gradientenvektor

$$\underline{\delta}^l = \frac{\partial E}{\partial \underline{s}^l} = \begin{bmatrix} \frac{\partial E}{\partial s_1^l} \\ \frac{\partial E}{\partial s_2^l} \\ \vdots \\ \frac{\partial E}{\partial s_{N_l}^l} \end{bmatrix}. \quad (4.38)$$

Für die Berechnung der lokalen Gradienten in einem MLP ist wiederum die Kettenregel erforderlich. Die lokalen Gradienten werden ausgehend von der letzten Schicht durch Rekursion bis zur ersten Schicht zurückpropagiert ( $\underline{\delta}^L \rightarrow \underline{\delta}^{L-1} \rightarrow \dots \rightarrow \underline{\delta}^2 \rightarrow \underline{\delta}^1$ ). Das Zurückrechnen von bereits bekannten lokalen Gradienten auf noch unbekannte lokale Gradienten führte zur Namensgebung *Backpropagation*. Für die Herleitung dieses rekursiven Zusammenhangs definieren wir die folgende Jacobi Matrix:

$$\frac{\partial \underline{s}^{l+1}}{\partial \underline{s}^l} = \begin{bmatrix} \frac{\partial s_1^{l+1}}{\partial s_1^l} & \frac{\partial s_1^{l+1}}{\partial s_2^l} & \dots & \frac{\partial s_1^{l+1}}{\partial s_{N_l}^l} \\ \frac{\partial s_2^{l+1}}{\partial s_1^l} & \frac{\partial s_2^{l+1}}{\partial s_2^l} & \dots & \frac{\partial s_2^{l+1}}{\partial s_{N_l}^l} \\ \vdots & \vdots & & \vdots \\ \frac{\partial s_{N_l+1}^{l+1}}{\partial s_1^l} & \frac{\partial s_{N_l+1}^{l+1}}{\partial s_2^l} & \dots & \frac{\partial s_{N_l+1}^{l+1}}{\partial s_{N_l}^l} \end{bmatrix} \quad (4.39)$$

Diese Matrix soll die Verbindung zwischen den beiden Schichten  $l$  und  $l+1$  bei der Ableitungsberechnung herstellen. Als nächstes muss eine Berechnungsvorschrift für diese Jacobi Matrix gefunden werden. Dazu betrachtet man zunächst nur ein Element  $i, j$  der Jacobi Matrix. Mit den Gleichungen (4.33) und (4.28) und der Abbildung 3.48 folgt:

$$\begin{aligned} \frac{\partial s_i^{l+1}}{\partial s_j^l} &= \frac{\partial \left( \sum_{k=1}^{N_l} \hat{\Theta}_{i,k}^{l+1} y_k^l + b_i^{l+1} \right)}{\partial s_j^l} \\ &= \frac{\partial \left( \sum_{k=1}^{N_l} \hat{\Theta}_{i,k}^{l+1} y_k^l + b_i^{l+1} \right)}{\partial y_j^l} \cdot \frac{\partial y_j^l}{\partial s_j^l} \\ &= \hat{\Theta}_{i,j}^{l+1} \cdot \frac{\partial y_j^l}{\partial s_j^l} \\ &= \hat{\Theta}_{i,j}^{l+1} \cdot \dot{T}_j^l(s_j^l) \end{aligned} \quad (4.40)$$

Mit diesem Ergebnis lässt sich die Jacobi Matrix kompakt in Matrixschreibweise darstellen

$$\frac{\partial \underline{s}^{l+1}}{\partial \underline{s}^l} = \hat{\Theta}^{l+1} \cdot \dot{\mathbf{T}}^l(\underline{s}^l), \quad (4.41)$$

wobei wir die Matrix der abgeleiteten Transferfunktionen  $\dot{\mathbf{T}}^l$  als Transfermatrix bezeichnen und wie folgt definieren:

$$\dot{\mathbf{T}}^l(\underline{s}^l) = \begin{bmatrix} \dot{\tau}_1^l(s_1^l) & 0 & \cdots & 0 \\ 0 & \dot{\tau}_2^l(s_2^l) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \dot{\tau}_{N_l}^l(s_{N_l}^l) \end{bmatrix}. \quad (4.42)$$

Nun kann der rekursive Zusammenhang der lokalen Gradienten durch erneutes Anwenden der Kettenregel bestimmt werden:

$$\begin{aligned} \underline{\delta}^l &= \frac{\partial E}{\partial \underline{s}^l} = \left( \frac{\partial s^{l+1}}{\partial \underline{s}^l} \right)^T \cdot \frac{\partial E}{\partial \underline{s}^{l+1}} \\ &\stackrel{(4.41)}{=} \dot{\mathbf{T}}^l(\underline{s}^l) \cdot \left( \hat{\Theta}^{l+1} \right)^T \cdot \frac{\partial E}{\partial \underline{s}^{l+1}} \\ &\stackrel{(4.38)}{=} \dot{\mathbf{T}}^l(\underline{s}^l) \cdot \left( \hat{\Theta}^{l+1} \right)^T \cdot \underline{\delta}^{l+1} \end{aligned} \quad (4.43)$$

Jetzt fehlt nur noch die Berechnung des lokalen Gradienten  $\underline{\delta}^L$  der Ausgangsschicht, also die Bestimmung des Startwerts für die rekursive Berechnung des Backpropagation:

$$\underline{\delta}_i^L = \frac{\partial E}{\partial s_i^L} = \frac{\partial \frac{1}{2} (\underline{e}^T \underline{e})}{\partial s_i^L} = \frac{\partial \frac{1}{2} \sum_{k=1}^{N_L} e_k^2}{\partial s_i^L} = -e_i \frac{\partial \hat{y}_i}{\partial s_i^L} = -e_i \dot{\mathbf{T}}_i^L(s_i^L), \quad (4.44)$$

was auch wieder in Matrixform gebracht werden kann

$$\underline{\delta}^L = -\dot{\mathbf{T}}^L(\underline{s}^L)\underline{e}. \quad (4.45)$$

Falls die Transferfunktion in der Ausgangsschicht linear ist und die Steigung Eins hat, vereinfacht sich Gleichung 4.45 zu:

$$\underline{\delta}^L = -\underline{e}. \quad (4.46)$$

Bisher wurde der Backpropagation Algorithmus mit einfachem Fehlermaß beschrieben. Dies bedeutet, dass bei jeder Neuberechnung des Gradienten nur ein Trainingspaar  $\{\underline{u}_p, \underline{y}_p\}$  aus einer Menge von insgesamt  $P$  Trainingsdaten

$$\{\underline{u}_1, \underline{y}_1\}, \{\underline{u}_2, \underline{y}_2\}, \dots, \{\underline{u}_P, \underline{y}_P\}$$

an das MLP angelegt wurde. Die meisten Lernverfahren verwenden ein kumulierte Fehlermaß. Hierbei wird der quadratische Fehler über alle Trainingsdaten aufsummiert, um dann die Gradientenberechnung durchzuführen. Man spricht in diesem Fall auch vom Offline bzw. Batch Lernen. Es gilt dann:

$$E(\hat{\Theta}) = \frac{1}{P} \sum_{p=1}^P \frac{1}{2} (\underline{y}_p - \hat{\underline{y}}_p)^T (\underline{y}_p - \hat{\underline{y}}_p) \quad (4.47)$$

Der Gradient ergibt sich nun für das kumulative Fehlermaß in einfacher Weise ebenfalls durch Aufsummieren der einzelnen Gradienten  $\nabla E_p$ , die jeweils beim Anlegen der einzelnen Trainingspaare  $\{\underline{u}_p, \underline{y}_p\}$  erzeugt wurden:

$$\nabla E(\hat{\Theta}) = \frac{1}{P} \sum_{p=1}^P \nabla E_p(\hat{\Theta}) \quad (4.48)$$

#### 4.1.3.2 Zusammenfassung des BP-Algorithmus

An dieser Stelle lohnt es sich noch einmal alle Einzelschritte des **Backpropagation Algorithmus** zusammenzufassen:

1. Es werden alle  $P$  Eingangsdaten  $\underline{u}_p$  vorwärts durch das MLP propagiert und dabei die Netzsignale  $\underline{s}_p^l$  und  $\underline{y}_p^l$  für jede Schicht gespeichert:

$$\underline{y}_p^0 = \underline{u}_p \quad (4.49)$$

$$\underline{y}_p^l = \underline{\mathbf{T}}^l(\hat{\Theta}^l \cdot \underline{y}_p^{l-1} + \underline{b}^l) = \underline{\mathbf{T}}^l(\underline{s}_p^l) \quad \text{für } l = 1, 2, \dots, L \quad (4.50)$$

$$\hat{\underline{y}}_p = \underline{y}_p^L \quad (4.51)$$

2. Zurückpropagieren und Speichern der lokalen Gradienten:

$$\underline{\delta}_p^L = -\dot{\mathbf{T}}^L(\underline{s}_p^L) \underline{e}_p \quad (4.52)$$

$$\underline{\delta}_p^l = \dot{\mathbf{T}}^l(\underline{s}_p^l) \left( \hat{\Theta}^{l+1} \right)^T \underline{\delta}_p^{l+1} \quad (4.53)$$

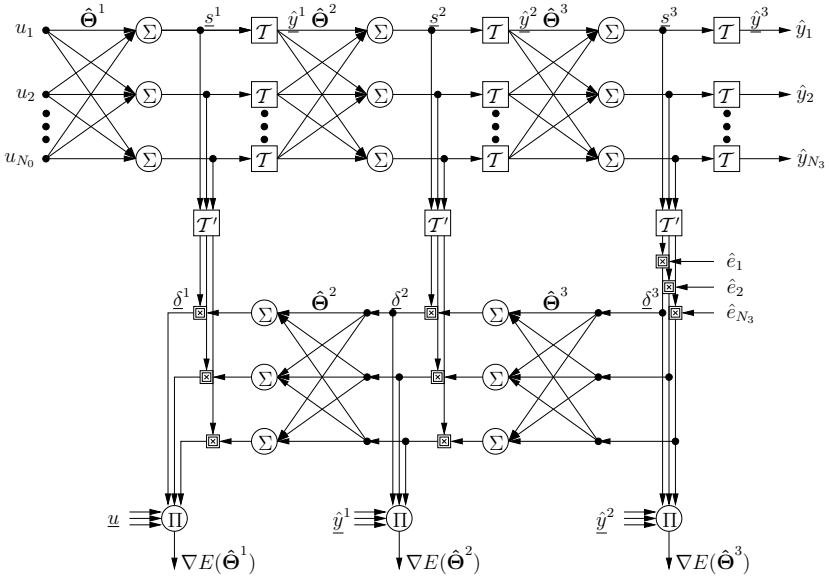
3. Berechnung des Gradienten aus den Netzsignalen und den lokalen Gradienten:

$$\frac{\partial E_p}{\partial \hat{\Theta}^l} = \underline{\delta}_p^l \cdot \left( \underline{y}_p^{l-1} \right)^T \quad (4.54)$$

$$\frac{\partial E_p}{\partial \underline{b}^l} = \underline{\delta}_p^l \quad (4.55)$$

$$\nabla E(\hat{\Theta}) = \frac{1}{P} \sum_{p=1}^P \nabla E_p(\hat{\Theta}) \quad (4.56)$$

Abbildung 4.6 zeigt die Gradientenberechnung für ein 3-schichtiges Netz mit  $N_0$  Eingängen und  $N_3$  Ausgängen. Zur besseren Übersicht wurde auf die Darstellung der Biasgewichte verzichtet.



**Abb. 4.6:** Graphische Veranschaulichung des Backpropagation Algorithmus mit  $\underline{\mathcal{T}}^1 = \underline{\mathcal{T}}^2 = \underline{\mathcal{T}}^3 = \underline{\mathcal{T}}$

#### 4.1.3.3 Gradientenverfahren bei mehrschichtigen Netzen

Analog zu den anderen statischen Funktionsapproximatoren soll in diesem Abschnitt das einfache Lerngesetz des Gradientenabstiegs für mehrschichtige Netze formuliert werden. Der Gradient berechnet sich — wie in Abschnitt 4.1.3.1 ausführlich hergeleitet und in Abschnitt 4.1.3.2 übersichtlich zusammengefasst — mit Hilfe der Backpropagation-Regel. Mit diesem Gradienten  $\nabla E(\hat{\Theta}[k])$  gilt das Lerngesetz:

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] - \eta \cdot \nabla E(\hat{\Theta}[k]) \quad (4.57)$$

Wie in Abschnitt 4.1.3.1 beschrieben, lässt sich der Gradient mit dem einfachen Fehlermaß und einem Trainingsdatenpaar berechnen oder mit Hilfe mehrerer Trainingsdatenpaare und dem kumulativen Fehlermaß. Das einfache Fehlermaß erzeugt einen unruhigen Abstiegspfad auf der Fehlerfläche, da sich zu jedem Lernschritt  $k$  auch das Trainingsmuster und somit das lokal zu optimierende Fehlermaß  $E(\hat{\Theta})$  ändert. Dies führt zwar zu einer verzögerten Minimierung des quadratischen Fehlers, verhindert aber durch die Unruhe ein verfrühtes Hängenbleiben in einem lokalen Minimum. Bei diesem Lernverfahren spricht man von online Gradientenabstieg oder auch von stochastischem Gradientenabstieg (erst die Summe aller Gradienten des Trainingsdatensatzes führt zum eigentlichen Gradienten des Optimierungsproblems). Das kumulative Fehlermaß führt zu einem ruhigeren Verlauf auf der Fehlerfläche. Dabei ist die Gefahr größer, in einem schlechten lokalen Minimum der Fehlerfläche hängen zu bleiben.

Die einfachen Beispiele in Abschnitt 4.1.3.5 verdeutlichen die durchzuführenden Rechenschritte beim Gradientenverfahren und helfen bei der praktischen Implementierung.

#### 4.1.3.4 Gradientenverfahren mit Momentum Term

Das Gradientenverfahren mit Momentum Term [191, 190] ist eine häufig benutzte Methode zur Vermeidung der Probleme, welche beim einfachen Gradientenabstieg auf flachen Plateaus und in steilen Schluchten der Fehlerfunktion auftreten (siehe auch Abschnitt 4.1.4). Hier hängt die Änderung der Gewichte nicht mehr ausschließlich von dem aktuellen Gradienten ab, sondern auch von der Gewichtsänderung des letzten Lernschrittes. Das Adoptionsgesetz lautet damit:

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + \Delta\hat{\Theta}[k] = \hat{\Theta}[k] - \eta \cdot \nabla E(\hat{\Theta}[k]) + \alpha \cdot \Delta\hat{\Theta}[k-1] \quad (4.58)$$

Dies bewirkt eine Beschleunigung in weiten Plateaus und ein Abbremsen in stark zerklüfteten Fehlerflächen. Die Variable  $\alpha$  hat üblicherweise Werte zwischen 0.2 und 0.99 [248].

#### 4.1.3.5 Beispiele

Die folgenden Ausführungen betrachten wieder die nichtlineare Kennlinie

$$\mathcal{NL}(u) = 3 \cdot \arctan(2 \cdot u), \quad (4.59)$$

welche bereits bei den Beispielen in den Abschnitten 3.7.3, 3.9.3 und 3.10.6 zum Einsatz kam.

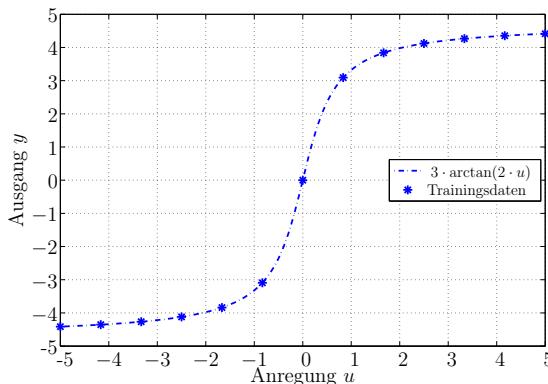
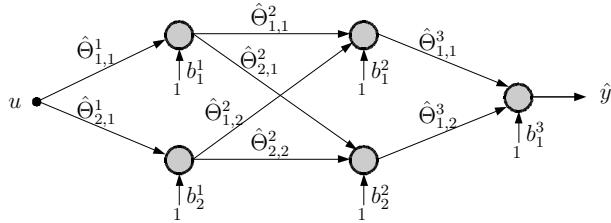


Abb. 4.7: Nichtlineare Kennlinie und Trainingsdaten für die Beispiele

Wie in Abbildung 4.7 und in Tabelle 4.1 dargestellt, umfassen die Trainingsdaten für die statische Funktionsapproximation insgesamt 13 Paare, äquidistant verteilt im Bereich  $u \in [-5 \dots 5]$ . Das verwendete MLP-Modell ist in Abbildung



**Abb. 4.8:** Beispiel-MLP zur Identifikation der arctan-Kennlinie von Abbildung 4.7

4.8 skizziert. Es besteht aus drei Schichten mit jeweils zwei tanh-Neuronen in den versteckten Schichten, das Ausgangsneuron enthält eine lineare Transferfunktion mit der Steigung 1. Wie gleich zu Beginn des Abschnitts 4.1.3 eingeführt, setzt sich der Gewichtsvektor  $\hat{\Theta}$  aus den Verbindungsgewichten und den Biasgewichten zusammen. Für das hier verwendete MLP von Abbildung 4.8 lautet der Gewichtsvektor:

$$\hat{\Theta} = [\hat{\Theta}_{1,1}^1 \ \hat{\Theta}_{2,1}^1 \ b_1^1 \ b_2^1 \ \hat{\Theta}_{1,1}^2 \ \hat{\Theta}_{2,1}^2 \ \hat{\Theta}_{1,2}^2 \ \hat{\Theta}_{2,2}^2 \ b_1^2 \ b_2^2 \ \hat{\Theta}_{1,1}^3 \ \hat{\Theta}_{1,2}^3 \ b_1^3]^T \quad (4.60)$$

Diesen Gewichtsvektor gilt es nun im Folgenden so anzupassen, damit das MLP-Modell die in Gleichung (4.59) beschriebene statische Nichtlinearität möglichst gut nachbilden kann. Zunächst soll im Beispiel 1 das zu lösende Optimierungsproblem mit Hilfe einer Fehlerfläche visualisiert werden, damit der Leser ein Gefühl dafür bekommt, welche Aufgabe bei der statischen Funktionsapproximation mit MLP-Netzen gelöst werden muss. Das anschließende Beispiel 2 zeigt ausführlich die Gradientenberechnung mit dem Backpropagation-Algorithmus für ein Trainingsdatum. Zuletzt greift das Beispiel 3 das einführende Fehlerflächenbeispiel wieder auf und zeigt den zurückgelegten Weg bei der Optimierung mit dem Gradientenabstieg.

### Beispiel 1: MLP-Fehlerfläche

Mit dem Gewichtsvektor

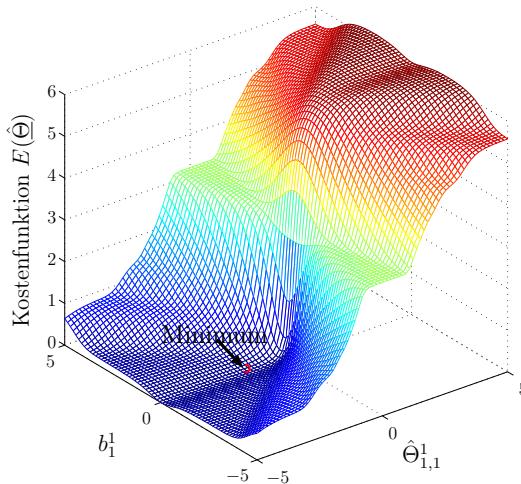
$$\hat{\Theta} = [-1.61 \ -0.15 \ 0 \ 0 \ 0.02 \ -0.78 \ 5.36 \ -1.26 \ 0 \ 0 \ -1.42 \ 3.25 \ 0]^T \quad (4.61)$$

approximiert das Neuronale Netz von Abbildung 4.8 sehr gut die in Gleichung (4.59) formulierte nichtlineare Kennlinie. Das Optimierungsproblem ist also be-

Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13
$u$	-5.00	-4.17	-3.33	-2.50	-1.67	-0.83	0	0.83	1.67	2.50	3.33	4.17	5.00
$y$	-4.41	-4.35	-4.27	-4.12	-3.84	-3.09	0	3.09	3.84	4.12	4.27	4.35	4.41

**Tabelle 4.1:** Trainingsdaten für die statische Funktionsapproximation

reits gelöst. Um nun ein Bild von der Fehlerfläche zu erhalten, werden im Folgenden zwei Gewichte verändert, während alle anderen Gewichte konstant bleiben. Die Fehlerfläche berechnet sich offline aus den in Tabelle 4.1 dargestellten 13 Trainingsdaten mit Gleichung (4.47). Abbildung 4.9 zeigt die resultierende Fehlerfläche bei Veränderung der beiden Gewichte  $\hat{\Theta}_{1,1}^1$  und  $b_1^1$  im Bereich  $[-5 \dots 5]^2$ . Die anderen 11 Gewichte behalten dabei die optimalen Werte von Gleichung



**Abb. 4.9:** Resultierende Fehlerfläche bei Veränderung der beiden Gewichte  $\hat{\Theta}_{1,1}^1$  und  $b_1^1$

(4.61). Das Minimum der Fehlerfläche tritt bei  $\hat{\Theta}_{1,1}^1 = -1.61$  und  $b_1^1 = 0$  auf. An dieser Stelle ist die Kostenfunktion mit  $E(\hat{\Theta}) = 1.3 \cdot 10^{-4}$  sehr klein. In dem Bereich um das Minimum ist die Fehlerfläche sehr flach, was bei der Optimierung mit dem Gradientenabstieg zu langsamer Parameterkonvergenz führt (siehe Beispiel 3).

### Beispiel 2: Gradientenberechnung mit Backpropagation

Dieses Beispiel berechnet den Gradienten für das erste Trainingsdatenpaar  $\{u, y\} = \{-5.00, -4.41\}$ . Die dabei durchgeführten Berechnungen laufen im Normalfall automatisiert im Rechner ab, was die Implementierung der in Abschnitt 4.1.3.2 zusammengefassten Rechenschritte erforderlich macht. Für das Beispiel wird der Gewichtsvektor von Gleichung (4.61) an zwei Stellen modifiziert:

$$\hat{\Theta}_{1,1}^1 = 1.00 \quad b_1^1 = 1.00 \quad (4.62)$$

Diese geringfügige Veränderung führt zu einem deutlich schlechteren Modell mit einem Fehlerwert von  $E(\hat{\Theta}) = 5.2$  (vergleiche dazu die Abbildung 4.9). Das nächste Beispiel 3 hat schließlich die Aufgabe, die beiden veränderten Gewichte

$\hat{\Theta}_{1,1}^1$  und  $b_1^1$  wieder zu optimieren. Für die Parameter im Modell gilt mit den Gleichungen (4.60), (4.61) und (4.62):

$$\hat{\Theta}^1 = \begin{bmatrix} \hat{\Theta}_{1,1}^1 \\ \hat{\Theta}_{2,1}^1 \\ 1.00 \\ -0.15 \end{bmatrix} \quad \hat{\Theta}^2 = \begin{bmatrix} \hat{\Theta}_{1,1}^2 & \hat{\Theta}_{1,2}^2 \\ \hat{\Theta}_{2,1}^2 & \hat{\Theta}_{2,2}^2 \\ 0.02 & 5.36 \\ -0.78 & -1.26 \end{bmatrix} \quad \hat{\Theta}^3 = \begin{bmatrix} \hat{\Theta}_{1,1}^3 & \hat{\Theta}_{1,2}^3 \\ -1.42 & 3.25 \end{bmatrix}$$

$$\underline{b}^1 = \begin{bmatrix} b_1^1 \\ b_2^1 \\ 1.00 \\ 0.00 \end{bmatrix} \quad \underline{b}^2 = \begin{bmatrix} b_1^2 \\ b_2^2 \\ 0.00 \\ 0.00 \end{bmatrix} \quad \underline{b}^3 = b_1^3 = 0.00$$

Mit diesen Gewichten folgen nun die drei Schritte Vorwärtspropagieren, Zurückpropagieren und Gradientenberechnung nach Abschnitt 4.1.3.2:

#### 1. Schritt: Vorwärtspropagieren des ersten Trainingsdatums

$$(4.49) : \quad \underline{y}_1^0 = -5$$

$$\begin{aligned} (4.50) : \quad l = 1 : \quad \underline{y}_1^1 &= \underline{\mathcal{T}}^1 \left( \hat{\Theta}^1 \cdot \underline{y}_1^0 + \underline{b}^1 \right) \\ &= \underline{\mathcal{T}}^1 \left( \begin{bmatrix} 1.00 \\ -0.15 \end{bmatrix} \cdot (-5) + \begin{bmatrix} 1.00 \\ 0.00 \end{bmatrix} \right) \\ &= \underline{\mathcal{T}}^1 \left( \begin{bmatrix} -4.00 \\ 0.75 \end{bmatrix} \right) \Rightarrow \underline{s}_1^1 = \begin{bmatrix} -4.00 \\ 0.75 \end{bmatrix} \\ &= \begin{bmatrix} \tanh(-4.00) \\ \tanh(0.75) \end{bmatrix} = \begin{bmatrix} -1.00 \\ 0.64 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} l = 2 : \quad \underline{y}_1^2 &= \underline{\mathcal{T}}^2 \left( \hat{\Theta}^2 \cdot \underline{y}_1^1 + \underline{b}^2 \right) \\ &= \underline{\mathcal{T}}^2 \left( \begin{bmatrix} 0.02 & 5.36 \\ -0.78 & -1.26 \end{bmatrix} \cdot \begin{bmatrix} -1.00 \\ 0.64 \end{bmatrix} + \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix} \right) \\ &= \underline{\mathcal{T}}^2 \left( \begin{bmatrix} 3.41 \\ -0.03 \end{bmatrix} \right) \Rightarrow \underline{s}_1^2 = \begin{bmatrix} 3.41 \\ -0.03 \end{bmatrix} \\ &= \begin{bmatrix} \tanh(3.41) \\ \tanh(-0.03) \end{bmatrix} = \begin{bmatrix} 1.00 \\ -0.03 \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
l = 3 : \quad \underline{y}_1^3 &= \underline{\mathcal{T}}^3 \left( \hat{\Theta}^3 \cdot \underline{y}_1^2 + \underline{b}^3 \right) \\
&= \underline{\mathcal{T}}^3 \left( \begin{bmatrix} -1.42 & 3.25 \end{bmatrix} \cdot \begin{bmatrix} 1.00 \\ -0.03 \end{bmatrix} + 0 \right) \\
&= 1 \cdot (-1.52) = -1.52 \quad \Rightarrow \quad \underline{s}_1^3 = -1.52
\end{aligned}$$

$$\begin{aligned}
(4.51) : \quad \hat{y}_1 &= \underline{y}_1^3 = -1.52 \\
\Rightarrow e_1 &= \underline{y}_1 - \hat{y}_1 = -4.41 + 1.52 = -2.89
\end{aligned}$$

1. Schritt: Zurückpropagieren

$$\begin{aligned}
\underline{\dot{\mathbf{T}}}^3(\underline{s}_1^3) &= 1 \quad (\text{Ausgangsneuron hat eine lineare Transferfunktion}) \\
\underline{\dot{\mathbf{T}}}^2(\underline{s}_1^2) &= \begin{bmatrix} 1 - \tanh^2(\underline{s}_1^2) & 0 \\ 0 & 1 - \tanh^2(\underline{s}_2^2) \end{bmatrix} \\
\underline{\dot{\mathbf{T}}}^1(\underline{s}_1^1) &= \begin{bmatrix} 1 - \tanh^2(\underline{s}_1^1) & 0 \\ 0 & 1 - \tanh^2(\underline{s}_2^1) \end{bmatrix}
\end{aligned}$$

$$(4.52) : l = 3 : \underline{\delta}_1^3 = -1 \cdot e_1 = -1 \cdot (-2.89) = 2.89$$

$$\begin{aligned}
(4.53) : l = 2 : \underline{\delta}_1^2 &= \underline{\dot{\mathbf{T}}}^2(\underline{s}_1^2) \left( \hat{\Theta}^3 \right)^T \underline{\delta}_1^3 \\
&= \begin{bmatrix} 1 - \tanh^2(3.41) & 0 \\ 0 & 1 - \tanh^2(-0.03) \end{bmatrix} \cdot \begin{bmatrix} 1.42 \\ 3.25 \end{bmatrix} \cdot 2.89 \\
&= \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} \cdot \begin{bmatrix} 1.42 \\ 3.25 \end{bmatrix} \cdot 2.89 \\
&= \begin{bmatrix} 0.00 \\ 3.25 \end{bmatrix} \cdot 2.89 = \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
l = 1 : \underline{\delta}_1^1 &= \underline{\dot{\mathbf{T}}}^1(\underline{s}_1^1) \left( \hat{\Theta}^2 \right)^T \underline{\delta}_1^2 \\
&= \begin{bmatrix} 1 - \tanh^2(-4.00) & 0 \\ 0 & 1 - \tanh^2(0.75) \end{bmatrix} \cdot \begin{bmatrix} 0.02 & -0.78 \\ 5.36 & -1.26 \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix} \\
&= \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.60 \end{bmatrix} \cdot \begin{bmatrix} 0.02 & -0.78 \\ 5.36 & -1.26 \end{bmatrix} \cdot \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix} \\
&= \begin{bmatrix} 0.00 & 0.00 \\ 3.22 & -0.76 \end{bmatrix} \cdot \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix} = \begin{bmatrix} 0.00 \\ -7.14 \end{bmatrix}
\end{aligned}$$

### 3. Schritt: Gradientenberechnung

$$(4.54) : \begin{aligned} l = 1 : \quad & \frac{\partial E_1}{\partial \hat{\Theta}^1} = \underline{\delta}_1^1 \cdot \left( \underline{y}_1^0 \right)^T = \begin{bmatrix} 0.00 \\ -7.14 \end{bmatrix} \cdot (-5.00) = \begin{bmatrix} 0.00 \\ 35.70 \end{bmatrix} \\ l = 2 : \quad & \frac{\partial E_1}{\partial \hat{\Theta}^2} = \underline{\delta}_1^2 \cdot \left( \underline{y}_1^1 \right)^T = \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix} \cdot [-1.00 \quad 0.64] = \begin{bmatrix} 0.00 & 0.00 \\ -9.39 & 6.01 \end{bmatrix} \\ l = 3 : \quad & \frac{\partial E_1}{\partial \hat{\Theta}^3} = \underline{\delta}_1^3 \cdot \left( \underline{y}_1^2 \right)^T = 2.89 \cdot [1.00 \quad -0.03] = [2.89 \quad -0.09] \end{aligned}$$

$$(4.55) : \begin{aligned} l = 1 : \quad & \frac{\partial E_1}{\partial \underline{b}^l} = \underline{\delta}_1^1 = \begin{bmatrix} 0.00 \\ -7.14 \end{bmatrix} \\ l = 2 : \quad & \frac{\partial E_1}{\partial \underline{b}^2} = \underline{\delta}_1^2 = \begin{bmatrix} 0.00 \\ 9.39 \end{bmatrix} \\ l = 3 : \quad & \frac{\partial E_1}{\partial \underline{b}^3} = \underline{\delta}_1^3 = 2.89 \end{aligned}$$

Mit diesen Ergebnissen lautet der Gradient für das erste Trainingsdatum:

$$\frac{\partial E_1}{\partial \hat{\Theta}} = \nabla E_1 =$$

$$[0.00 \ 35.70 \ 0.00 \ -7.14 \ 0.00 \ -9.39 \ 0.00 \ 6.01 \ 0.00 \ 9.39 \ 2.89 \ -0.09 \ 2.89]^T$$

Die Berechnungen beim 2. bis 13. Trainingsdatum laufen analog ab. Der Gradient für das gesamte Optimierungsproblem berücksichtigt alle Trainingsdaten und berechnet sich aus dem Mittel aller Teilgradienten, wie in Gleichung (4.56) beschrieben.

### Beispiel 3: Gradientenabstieg

In diesem Beispiel soll der Gradientenabstieg für die beiden Gewichte  $\hat{\Theta}_{1,1}^1$  und  $b_1^1$  nach Gleichung (4.62) durchgeführt werden. Die restlichen Gewichte behalten — wie bereits in den Beispielen 1 und 2 vorausgesetzt — die idealen Werte von Gleichung (4.61). Für die Berechnung des Gradienten müssen die Teilgradienten aller Trainingsdatenpaare vorliegen. Das Beispiel 2 bestimmte den Gradienten für das erste Trainingsdatenpaar. Analog dazu laufen die Berechnungen für die weiteren Trainingsdatenpaare ab. Die Ergebnisse fassen die beiden Tabellen 4.2 und 4.3 zusammen. Mit diesen Werten folgt nach Gleichung (4.56) ein Gesamtgradient (für das zweidimensionale Teilproblem) von:

$$\begin{bmatrix} \frac{\partial E}{\partial \hat{\Theta}_{1,1}^1} \\ \frac{\partial E}{\partial b_1^1} \end{bmatrix} = \begin{bmatrix} \frac{1}{13} \sum_{p=1}^{13} \frac{\partial E_p}{\partial \hat{\Theta}_{1,1}^1} \\ \frac{1}{13} \sum_{p=1}^{13} \frac{\partial E_p}{\partial b_1^1} \end{bmatrix} = \begin{bmatrix} \frac{1}{13} \cdot 17.14 \\ \frac{1}{13} \cdot (-7.81) \end{bmatrix} = \begin{bmatrix} 1.32 \\ -0.60 \end{bmatrix}$$

$\frac{\partial E_1}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_2}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_3}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_4}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_5}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_6}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_7}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_8}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_9}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_{10}}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_{11}}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_{12}}{\partial \Theta_{1,1}^1}$	$\frac{\partial E_{13}}{\partial \Theta_{1,1}^1}$
0.00	0.24	1.04	3.84	8.46	2.59	0.00	0.60	0.27	0.08	0.02	0.00	0.00

**Tabelle 4.2:** Ableitung der Kostenfunktion nach dem Gewicht  $\hat{\Theta}_{1,1}^1$  für alle 13 Trainingsdaten

$\frac{\partial E_1}{\partial b_1^1}$	$\frac{\partial E_2}{\partial b_1^1}$	$\frac{\partial E_3}{\partial b_1^1}$	$\frac{\partial E_4}{\partial b_1^1}$	$\frac{\partial E_5}{\partial b_1^1}$	$\frac{\partial E_6}{\partial b_1^1}$	$\frac{\partial E_7}{\partial b_1^1}$	$\frac{\partial E_8}{\partial b_1^1}$	$\frac{\partial E_9}{\partial b_1^1}$	$\frac{\partial E_{10}}{\partial b_1^1}$	$\frac{\partial E_{11}}{\partial b_1^1}$	$\frac{\partial E_{12}}{\partial b_1^1}$	$\frac{\partial E_{13}}{\partial b_1^1}$
0.00	-0.06	-0.31	-1.54	-5.07	-3.11	1.36	0.72	0.16	0.03	0.01	0.00	0.00

**Tabelle 4.3:** Ableitung der Kostenfunktion nach dem Gewicht  $b_1^1$  für alle 13 Trainingsdaten

Damit lautet der 1. Optimierungsschritt mit dem Gradientenabstieg nach Gleichung (4.57) und einer Lernschrittweite von  $\eta = 2$ :

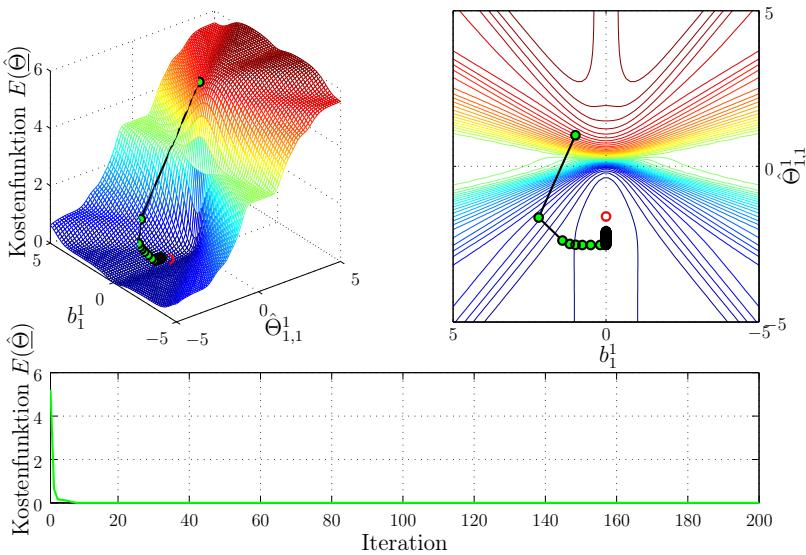
$$\begin{aligned}\hat{\Theta}[1] &= \hat{\Theta}[0] - \eta \cdot \nabla E(\hat{\Theta}[0]) \\ &= \begin{bmatrix} 1.00 \\ 1.00 \end{bmatrix} - 2 \cdot \begin{bmatrix} 1.32 \\ -0.60 \end{bmatrix} \\ &= \begin{bmatrix} -1.64 \\ 2.2 \end{bmatrix}\end{aligned}$$

Der Startpunkt  $\hat{\Theta}[0] = [1 \ 1]^T$  und das Ergebnis des ersten Optimierungsschrittes  $\hat{\Theta}[1] = [-1.64 \ 2.2]^T$  sind in der Fehlerfläche in Abbildung 4.10 als schwarze Kreise eingetragen. Die weiteren Schritte laufen analog ab. Abbildung 4.10 zeigt den Verlauf der Optimierung für die ersten 200 Schritte. Nach etwa 10 Optimierungsschritten erreicht der Gradientenabstieg einen sehr flachen Bereich der Fehlerfläche und die Optimierung konvergiert nur noch sehr langsam. Eine deutlich schnellere Konvergenz erreichen die Verfahren zweiter Ordnung, welche neben der Gradienteninformation auch noch die zweite Ableitung der Fehlerfläche ausnutzen. Näheres dazu folgt in Kapitel 10.

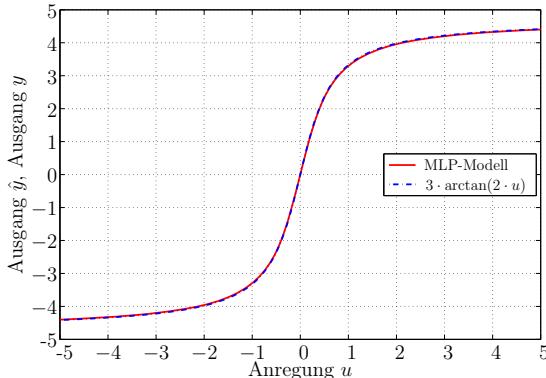
Abbildung 4.11 verdeutlicht, dass das dreischichtige MLP-Modell sehr gut in der Lage ist, die nichtlineare Kennlinie zu beschreiben. Die 13 Trainingsdaten von Tabelle 4.1 reichen für den Optimierungsprozess aus.

#### 4.1.4 Probleme beim einfachen Gradientenabstieg

Der einfache Gradientenabstieg besitzt eine Reihe von Problemen, die dadurch entstehen, dass es sich um ein lokales Verfahren handelt, welches keine Information über die Fehlerfläche  $E(\hat{\Theta})$  insgesamt hat, sondern nur aus der Kenntnis der lokalen Umgebung ein Minimum suchen muss. Der folgende Überblick orientiert sich an den Aufzeichnungen von Zell[248].



**Abb. 4.10:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren bei einer Lernschrittweite von  $\eta = 2$

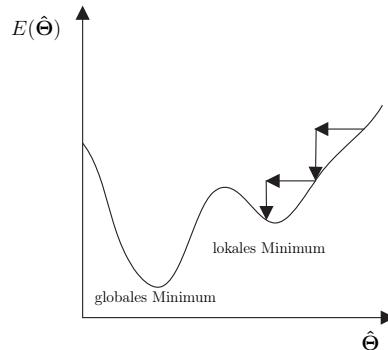


**Abb. 4.11:** Gelernte Kennlinie und wahre Kennlinie nach der Optimierung

#### 4.1.4.1 Lokale Minima der Fehlerfläche

Gradientenverfahren haben alle das Problem, dass sie in einem lokalen Minimum der Fehlerfläche hängen können. Dies ist in Abb. 4.12 graphisch dargestellt. Mit steigender Verbindungsanzahl wird die Fehlerfläche immer zerklüfteter und daher steigt die Wahrscheinlichkeit, in einem lokalen statt einem globalen Minimum zu landen [84]. Eine allgemeingültige Lösung gibt es für dieses Problem

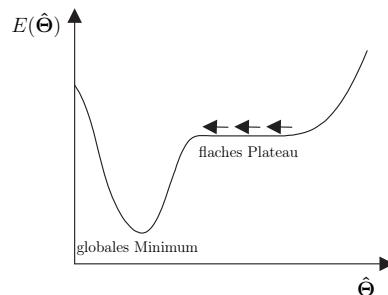
nicht, allerdings hat sich gezeigt, dass der Gradientenabstieg bei einer genügend kleinen Schrittweite  $\eta$  in sehr vielen Anwendungen ein lokales Minimum findet, das für die Anwendung akzeptabel ist.



**Abb. 4.12:** Lokales Minimum einer Fehlerfläche

#### 4.1.4.2 Flache Plateaus

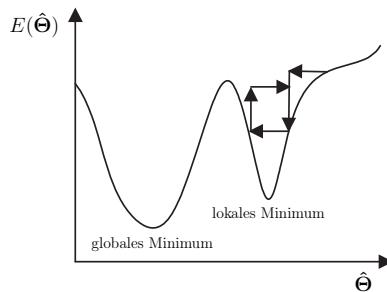
Da die Größe der Gewichtsänderung von dem Betrag des Gradienten abhängig ist, stagniert der Gradientenabstieg auf flachen Plateaus, d.h. das Lernverfahren benötigt viele Iterationsschritte (siehe Abb. 4.13). Bei einem vollständig ebenen Plateau wird keine Gewichtsänderung mehr durchgeführt, da der Gradient Null ist. Problematisch ist außerdem, dass man nicht erkennen kann, ob der Algorithmus auf einem flachen Plateau stagniert oder sich in einem lokalen oder globalen Minimum befindet (Gradient ist ebenfalls der Nullvektor). Für dieses Problem existieren einfache Verfahren (z.B. Momentum-Term aus Kapitel 4.1.1.1), die diese flachen Plateaus überwinden können.



**Abb. 4.13:** Fehlerfläche mit weitem Plateau

#### 4.1.4.3 Oszillationen in steilen Schluchten

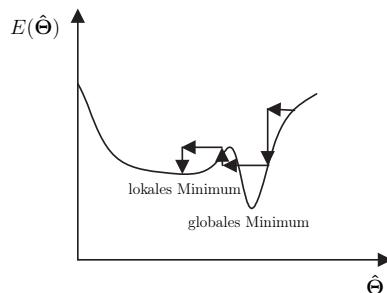
In steilen Schluchten der Fehlerfläche kann der Gradientenabstieg oszillieren. Dies geschieht, wenn der Gradient am Rande einer Schlucht so groß ist, dass durch die Gewichtsänderung ein Sprung auf die gegenüberliegende Seite der Schlucht erfolgt. Ist die Schlucht dort genauso steil, bewirkt dies einen Sprung zurück auf die erste Seite (Abb. 4.14). Auch hier kann eine Modifikation des Verfahrens (z.B. Momentum-Term) Abhilfe leisten.



**Abb. 4.14:** Oszillationen in steilen Schluchten

#### 4.1.4.4 Verlassen guter Minima

Es kann passieren, dass der Gradientenabstieg aus guten Minima heraußspringt. Bei sehr engen Tälern der Fehlerfläche kann der Betrag des Gradienten so groß sein, dass die Gewichtsänderung aus dem guten Minimum heraus in ein suboptimales Minimum führt. Dies ist in Abb. 4.15 dargestellt. In der Praxis passiert dies glücklicherweise sehr selten. Bei der Verwendung des Momentum-Terms oder der Erhöhung der Schrittweite, kann dieses Problem jedoch auftreten.



**Abb. 4.15:** Verlassen guter Minima

## 4.2 Lerngesetz: Least–Squares–Verfahren

Die Methode der kleinsten Quadrate, auch als Least–Squares–Algorithmus (LS) bezeichnet, oder deren rekursive Variante (RLS) spielen in der Signalverarbeitung und bei der Lösung überbestimmter linearer Gleichungssysteme eine wichtige Rolle. Wie schon in der Einführung des Kapitels 4 hingewiesen, sind die Least–Squares–Verfahren nur einsetzbar, wenn der Modellausgang linear in den Parametern ist, dies ist gegeben bei Gleichungsfehler–Modellen oder beim FIR–Modell (Kapitel 7.2).

Zunächst wird der Least–Squares–Algorithmus erklärt und anschließend die rekursive Form des Least–Squares–Algorithmus abgeleitet. Beide Methoden sollen abschließend zur optimalen Gewichtsberechnung eines RBF– und GRNN–Netzwerks besprochen werden.

### 4.2.1 Nichtrekursiver Least–Squares–Algorithmus (LS)

Die Berechnung der optimalen Parameter  $\hat{\underline{\Theta}}$  erfolgt beim Least–Squares–Algorithmus durch die Minimierung der Summe der quadrierten Gleichungsfehler. Ausgangspunkt ist dabei eine Gleichung<sup>3)</sup> für das geschätzte Ausgangssignal  $\hat{y}$  als Funktion der Parameterschätzwerte  $\hat{\theta}_i$ ,  $i = 1 \dots m + n$ .

$$\hat{y}[k] = u[k-1]\hat{\theta}_1 + \dots + u[k-m]\hat{\theta}_m + y[k-1]\hat{\theta}_{m+1} + \dots + y[k-n]\hat{\theta}_{m+n} \quad (4.63)$$

oder kurz,  $\hat{y}[k] = \underline{x}[k]^T \hat{\underline{\Theta}}$ , wobei

$$\hat{\underline{\Theta}} = [\underbrace{\hat{\theta}_1, \dots, \hat{\theta}_m}_{\text{Zählerterme}}, \underbrace{\hat{\theta}_{m+1}, \dots, \hat{\theta}_{m+n}}_{\text{Nennerterme}}]^T$$

Zu jeder Messung des Modellausgangs  $\hat{y}$  gehört also ein Datenvektor  $\underline{x}$  vergangener Ein– und Ausgangssignale des Systems. Für den Fall  $m = n = 3$  erhält man für jede Messung eine Zeile der folgenden Datenmatrix (Regressionsmatrix):

$$\mathbf{X} = \begin{bmatrix} \underline{x}[1]^T \\ \underline{x}[2]^T \\ \underline{x}[3]^T \\ \vdots \end{bmatrix} = \begin{bmatrix} u[0] & u[-1] & u[-2] & y[0] & y[-1] & y[-2] \\ u[1] & u[0] & u[-1] & y[1] & y[0] & y[-1] \\ u[2] & u[1] & u[0] & y[2] & y[1] & y[0] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (4.64)$$

Liegt ein Datensatz mit  $P$  Messungen vor, so ergibt sich

$$\begin{bmatrix} \hat{y}[1] \\ \hat{y}[2] \\ \vdots \\ \hat{y}[P] \end{bmatrix} = \hat{\underline{y}} = \mathbf{X} \cdot \hat{\underline{\Theta}} \quad (4.65)$$

<sup>3)</sup> Auf Gleichungen dieser Art wird in Kapitel 7.2 näher eingegangen

mit der  $[P \times n + m]$  Matrix  $\mathbf{X}$ . Die Matrix  $\mathbf{X}$  wird als Regressionsmatrix bezeichnet und enthält alle  $n+m$  Signale des Datenvektors zu den  $P$  verschiedenen Zeitpunkten. Das Gütefunktional zur Bestimmung der  $p$  unbekannten Parameter lautet:

$$\min_{\hat{\Theta}} E(\hat{\Theta}) = \min_{\hat{\Theta}} \frac{1}{2} \left\| \underline{y} - \mathbf{X} \cdot \hat{\Theta} \right\|^2 = \min_{\hat{\Theta}} \left[ \frac{1}{2} (\underline{y} - \mathbf{X} \cdot \hat{\Theta})^T (\underline{y} - \mathbf{X} \cdot \hat{\Theta}) \right] \quad (4.66)$$

mit  $\underline{y} = [y[1] \ y[2] \ \dots \ y[P]]^T$ .

Die Anzahl  $P$  der Trainingspaare muss mindestens gleich der Parameteranzahl  $p$  sein, damit das Gleichungssystem lösbar ist. Normalerweise wird die Zahl der Trainingspaare wesentlich größer als die Parameteranzahl sein ( $P > p$ ), so dass sich ein überbestimmtes Gleichungssystem ergibt (Redundanz). Durch zu Null setzen der Ableitung von Gl. (4.66) nach dem Parametervektor  $\hat{\Theta}$  ergibt sich der Parametervektor zu:

$$\frac{\partial E(\hat{\Theta})}{\partial \hat{\Theta}} = -\mathbf{X}^T \cdot (\underline{y} - \mathbf{X} \cdot \hat{\Theta}) \stackrel{!}{=} 0 \quad (4.67)$$

$$\Rightarrow \hat{\Theta} = \mathbf{X}^+ \cdot \underline{y} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \underline{y}$$

$\mathbf{X}^+$  ist die sogenannte **Pseudo–Links–Inverse** von  $\mathbf{X}$ , da sie ähnlich einer Inversen der nicht–quadratischen und somit auch nicht–invertierbaren Matrix  $\mathbf{X}$  benutzt wird. Für den Sonderfall  $P = p$  ergibt sich  $\mathbf{X}^+ = \mathbf{X}^{-1}$ .

In Gleichung (4.67) kommt der Matrix  $(\mathbf{X}^T \cdot \mathbf{X})$  – auch **Kovarianzmatrix** genannt – eine besondere Bedeutung zu, da solange diese Matrix invertierbar ist nur eine Lösung für das Gleichungssystem existiert. Für die Praxis folgt aus dieser Tatsache, dass das System genügend angeregt werden muss, um vollen Rang der Kovarianzmatrix zu garantieren.

Das vorgestellte Least–Squares–Verfahren ist in dieser Form nur als offline Verfahren einsetzbar, da alle Trainingspaare gleichzeitig vorliegen müssen. Häufig fallen jedoch Messwerte sukzessive während des Trainings an, so dass dies bei der Berechnung der optimalen Parameter berücksichtigt werden kann. Aufgrund der Tatsache, dass mit zunehmender Messwerteanzahl die Berechnung der optimalen Parameter  $\hat{\Theta}$  wegen der Zunahme der Zeilenanzahl in  $\mathbf{X}$  immer aufwändiger wird, soll im Folgenden die rekursive Form des Least–Squares–Verfahrens (RLS) vorgestellt werden, bei der der Rechenaufwand zu jedem Abtastschritt gleich bleibt.

#### 4.2.2 Rekursiver Least–Squares–Algorithmus (RLS)

Bei dem rekursiven Least–Squares–Verfahren bleibt der Rechenaufwand pro Zeitschritt konstant, obwohl immer neue Messvektoren hinzukommen. Zunächst wird von Gleichung (4.65) ausgehend für ein einzelnes Trainingspaar angesetzt:

$$\underline{y}[i] = \underline{x}^T[i] \cdot \hat{\Theta} \quad \text{mit} \quad \underline{x}^T[i] = \begin{bmatrix} u[i-1] & \dots & u[i-m] & y[i-1] & \dots & y[i-n] \end{bmatrix} \quad (4.68)$$

Für  $k > p$  Trainingspaare ergibt sich mit

$$\underline{y}[k] = \begin{bmatrix} y[1] & y[2] & \dots & y[k] \end{bmatrix}^T \quad \mathbf{X}[k] = \begin{bmatrix} \underline{x}^T[1] \\ \underline{x}^T[2] \\ \vdots \\ \underline{x}^T[k] \end{bmatrix} \quad (4.69)$$

das überbestimmte Gleichungssystem für  $k$  Messwertepaare zu

$$\underline{y}[k] = \mathbf{X}[k] \cdot \hat{\Theta}[k] \quad (4.70)$$

Der optimale Schätzwert bei  $k$  Gleichungen folgt gemäß Gleichung (4.67) zu

$$\hat{\Theta}[k] = (\mathbf{X}^T[k] \cdot \mathbf{X}[k])^{-1} \cdot \mathbf{X}^T[k] \cdot \underline{y}[k] \quad (4.71)$$

bzw. für  $k+1$  Gleichungen zu

$$\hat{\Theta}[k+1] = (\mathbf{X}^T[k+1] \cdot \mathbf{X}[k+1])^{-1} \cdot \mathbf{X}^T[k+1] \cdot \underline{y}[k+1] \quad (4.72)$$

Durch Aufteilen von Gleichung (4.72) in den neu hinzugekommenen Anteil und den bekannten Anteil, ergibt sich die Lösung für  $k+1$  Gleichungen zu

$$\begin{aligned} \hat{\Theta}[k+1] &= \\ &= \left( \begin{bmatrix} \mathbf{X}^T[k] & \underline{x}[k+1] \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}[k] \\ \underline{x}^T[k+1] \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} \mathbf{X}^T[k] & \underline{x}[k+1] \end{bmatrix} \cdot \begin{bmatrix} \underline{y}[k] \\ y[k+1] \end{bmatrix} \\ &= (\mathbf{X}^T[k] \cdot \mathbf{X}[k] + \underline{x}[k+1] \cdot \underline{x}^T[k+1])^{-1} \cdot (\mathbf{X}^T[k] \cdot \underline{y}[k] + \underline{x}[k+1] \cdot y[k+1]) \\ &= (\mathbf{X}^T[k] \cdot \mathbf{X}[k] + \underline{x}[k+1] \cdot \underline{x}^T[k+1])^{-1} \cdot \left( \mathbf{X}^T[k] \cdot \mathbf{X}[k] \cdot \hat{\Theta}[k] + \underline{x}[k+1] \cdot y[k+1] \right) \end{aligned} \quad (4.73)$$

Für die folgende Umformung ist das MATRIXINVERSIONSLEMMA [197] notwendig: Für eine reguläre Matrix  $\mathbf{A}$  und Spaltenvektoren  $\underline{b}$  und  $\underline{c}$  gilt:

$$(\mathbf{A} + \underline{b}\underline{c}^T)^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\underline{b})(\underline{c}^T\mathbf{A}^{-1})}{1 + \underline{c}^T\mathbf{A}^{-1}\underline{b}} \quad (4.74)$$

wobei  $(\mathbf{A} + \underline{b}\underline{c}^T)$  regulär ist.

Führt man die Abkürzung  $\mathbf{P}^{-1}[k] = \mathbf{X}^T[k] \cdot \mathbf{X}[k]$  ein und wendet das Lemma an, so gilt:

$$\hat{\Theta}[k+1] = \left( \mathbf{P}[k] - \frac{\mathbf{P}[k]\underline{x}[k+1]\underline{x}^T[k+1]\mathbf{P}[k]}{1 + \underline{x}^T[k+1]\mathbf{P}[k]\underline{x}[k+1]} \right) \left( \mathbf{P}^{-1}[k]\hat{\Theta}[k] + \underline{x}[k+1]y[k+1] \right) \quad (4.75)$$

Mit elementaren Umformungen kann schließlich der neue Schätzwert für den unbekannten Parametervektor durch folgende Gleichung bestimmt werden:

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + \underbrace{\frac{\mathbf{P}[k] \cdot \underline{x}[k+1]}{1 + \underline{x}^T[k+1] \cdot \mathbf{P}[k] \cdot \underline{x}[k+1]}}_{\text{Verstärkungsvektor } \underline{\gamma}[k]} \cdot \underbrace{\left( y[k+1] - \underline{x}^T[k+1] \cdot \hat{\Theta}[k] \right)}_{\text{Korrekturterm}} \quad (4.76)$$

Der Schätzwert für  $k+1$  Gleichungen wird basierend auf dem Schätzwert für  $k$  Gleichungen und der  $\mathbf{P}$ -Matrix durch Addition des alten Schätzwertes und eines, mit dem Verstärkungsvektor  $\underline{\gamma}[k]$  multiplizierten Korrekturterms, berechnet. Der Korrekturterm ist die Differenz aus tatsächlichem Messwert und der Prädiktion des Systemverhaltens auf Basis des letzten Schätzwertes der Parameter.

Mit Gleichung (4.76) kann nun für jedes neue Trainingspaar der Schätzwert der Parameter auf Basis des jeweils letzten Schätzwertes mit konstantem Rechenaufwand verbessert werden. Der Aufwand zur Berechnung der  $\mathbf{P}$ -Matrix nimmt jedoch mit jedem Messwertepaar zu. Durch Partitionierung analog zur  $\mathbf{X}$ -Matrix kann auch die Berechnung der  $\mathbf{P}$ -Matrix rekursiv gestaltet werden. Als Rekursionsformel ergibt sich:

$$\mathbf{P}[k+1] = \mathbf{P}[k] - \underline{\gamma}[k] \cdot \underline{x}^T[k+1] \cdot \mathbf{P}[k] \quad (4.77)$$

mit  $\underline{\gamma}[k]$  gemäß Gleichung (4.76). Gleichung (4.76) und (4.77) bilden zusammen ein einseitig verkoppeltes System von Rekursionsgleichungen.

Es bestehen nun zwei Möglichkeiten die Rekursion zu starten. Die erste Möglichkeit besteht in der Wahl beliebiger Startwerte  $\hat{\Theta}[0]$  und  $\mathbf{P}[0]$ . Die Rekursion beginnt dann bereits mit dem ersten Messwertepaar. Als Startwert für die Matrix  $\mathbf{P}[0]$  eignet sich eine obere Dreiecksmatrix, deren Werte zwischen 100 und 1000 liegen. Der Parametervektor  $\hat{\Theta}[0]$  kann mit Null initialisiert werden.

Eine zweite Möglichkeit besteht darin, die ersten  $p$  Trainingspaare zur Lösung des dann eindeutig bestimmten Gleichungssystems zu benutzen.

$$\hat{\Theta}[p] = \mathbf{X}^{-1}[p] \cdot \underline{y}[p] \quad \mathbf{P}[p] = (\mathbf{X}^T[p] \cdot \mathbf{X}[p])^{-1} \quad (4.78)$$

Die Rekursion startet dann mit dem  $(p+1)$ -ten Trainingspaar.

Der Vorteil der rekursiven Form des Least-Squares-Algorithmus ist, dass die bei der nichtrekursiven Methode nötige Matrixinversion der Kovarianzmatrix  $(\mathbf{X}^T \cdot \mathbf{X})^{-1}$  vermieden werden kann. Die Matrixinversion reduziert sich bei der rekursiven Methode in eine Division durch einen Skalar. Das rekursive Least-Squares-Verfahren (RLS) ist somit, wie auch das bereits vorgestellte Gradientenverfahren, bei konstantem Rechenaufwand zu jedem Abtastschritt online anwendbar.

### Einführung eines Vergessensfaktors

Für die Identifikation zeitvarianter Parameter ist es erforderlich den RLS-Algorithmus mit einem Vergessensfaktor  $\lambda$  zu versehen. Der Vergessensfaktor  $\lambda$

ermöglicht eine Gewichtung des Beitrags vergangener Trainingspaare zur Berechnung der aktuellen Parameter  $\hat{\Theta}$ . Durch die Einführung eines Vergessensfaktors  $\lambda \leq 1$  werden Trainingspaare, die  $j$  Zeitpunkte zurückliegen mit dem Faktor  $\lambda^j$  gewichtet, so dass ein sog. exponentielles Vergessen eintritt. Die Rekursionsgleichungen lauten dann:

$$\begin{aligned}\hat{\Theta}[k+1] &= \hat{\Theta}[k] + \underbrace{\frac{\mathbf{P}[k] \cdot \underline{x}[k+1]}{\lambda + \underline{x}^T[k+1] \cdot \mathbf{P}[k] \cdot \underline{x}[k+1]} \cdot \underbrace{\left( y[k+1] - \underline{x}^T[k+1] \cdot \hat{\Theta}[k] \right)}_{\text{Korrekturterm}}}_{\text{Verstärkungsvektor } \underline{\gamma}[k]} \\ \mathbf{P}[k+1] &= \frac{1}{\lambda} \left[ \mathbf{P}[k] - \underline{\gamma}[k] \cdot \underline{x}^T[k+1] \cdot \mathbf{P}[k] \right] \quad \lambda = 0.9 \dots 1\end{aligned}\tag{4.79}$$

Bei der Wahl des Vergessensfaktors muss stets ein Kompromiss gefunden werden zwischen einer besseren Elimination von Störeinflüssen ( $\lambda \rightarrow 1$ ) oder einem besseren Folgen der zeitveränderlichen Parameter ( $\lambda < 1$ ).

#### 4.2.3 Weighted-Least-Squares-Algorithmus (WLS)

Bei dem einfachen Least-Squares-Algorithmus wurden im Gütfunktional alle Gleichungsfehler gleich gewichtet. Versieht man diese Fehler mit unterschiedlichen Gewichten, dann erhält man eine allgemeinere Form des Least-Squares-Algorithmus. Die Verlustfunktion für den sog. gewichteten Least-Squares-Algorithmus lautet:

$$\min_{\hat{\Theta}} E(\hat{\Theta}) = \min_{\hat{\Theta}} \left[ \frac{1}{2} \left( \underline{y} - \mathbf{X} \cdot \hat{\Theta} \right)^T \cdot \mathbf{Q} \cdot \left( \underline{y} - \mathbf{X} \cdot \hat{\Theta} \right) \right]\tag{4.80}$$

Durch die Gewichtungsmatrix

$$\mathbf{Q}[k] = \begin{bmatrix} q[0] & 0 & \cdots & 0 \\ 0 & q[1] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q[k] \end{bmatrix} \quad \text{mit} \quad q[i] > 0\tag{4.81}$$

können die einzelnen Messgleichungen entsprechend ihrer Qualität mehr oder weniger stark gewichtet werden. Durch zu Null setzen der Ableitung von Gl. (4.80) nach dem Parametervektor  $\hat{\Theta}$  ergibt sich der optimale Parametervektor als Minimumsnormlösung zu:

$$\hat{\Theta} = (\mathbf{X}^T \cdot \mathbf{Q} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{Q} \cdot \underline{y}\tag{4.82}$$

Gleichung (4.82) beschreibt die nichtrekursive Form des Weighted-Least-Squares-Algorithmus (WLS). Die rekursive Form des Weighted-Least-Squares-Algorithmus lässt sich wie folgt angeben [111]:

$$\hat{\Theta}[k+1] = \hat{\Theta}[k] + \gamma[k] \cdot \left( y[k+1] - \underline{x}^T[k+1] \cdot \hat{\Theta}[k] \right) \quad (4.83)$$

$$\gamma[k] = \frac{\mathbf{P}_w[k] \cdot \underline{x}[k+1]}{1/q[k+1] + \underline{x}^T[k+1] \cdot \mathbf{P}_w[k] \cdot \underline{x}[k+1]} \quad (4.84)$$

$$\mathbf{P}_w[k+1] = \left[ \mathbf{P}_w[k] - \underline{\gamma}[k] \cdot \underline{x}^T[k+1] \cdot \mathbf{P}_w[k] \right] \quad (4.85)$$

mit der Abkürzung

$$\mathbf{P}_w[k] = \left[ \underline{x}^T[k] \cdot \underline{x}[k] \cdot w[k] \right]^{-1} \quad (4.86)$$

Der rekursive Least–Squares–Algorithmus mit Vergessensfaktor stellt einen Spezialfall des WLS–Algorithmus dar.

Im Laufe der letzten Jahrzehnte wurde eine Fülle verschiedener linearer Lernverfahren entwickelt, wobei aber die meisten davon auf der Idee des Least–Squares–Algorithmus beruhen. Sie haben zum Ziel, das Konvergenzverhalten und die Güte der Modelle bei stark verrauschten Signalen zu verbessern. An dieser Stelle wird auf die einschlägige Literatur verwiesen, wie z.B. „System Identification“ von Söderström T. und Stoica P. [216], „Identifikation dynamischer Systeme“ von Isermann R. [111, 112] oder „System Identification“ von Ljung L. [140], in denen die meisten linearen Lernverfahren in einheitlicher Form abgeleitet und diskutiert werden.

#### 4.2.4 Anwendung des Least–Squares–Algorithmus zur Parameteroptimierung bei RBF–Netzen

Im Folgenden soll die nichtrekursive und die rekursive Form des Least–Squares–Algorithmus zur Parameteroptimierung bei einem RBF–Netz bzw. bei einem GRNN angewendet werden. Dabei wird von einem Netz mit  $N$  Eingängen,  $p$  Stützstellen und einem Ausgang ausgegangen. Die Zentren  $\xi_i$  und der Glättungsfaktor  $\sigma$  seien bereits vorab festgelegt worden und sind deshalb konstant. Durch Anwendung des LS–Algorithmus ist eine analytische Bestimmung des optimalen Stützwertvektors  $\hat{\Theta}$  möglich. Der Unterschied zwischen RBF–Netz und GRNN liegt lediglich in den verschiedenen Aktivierungsfunktionen (siehe Gl. (3.9) und (3.12)). Deshalb sind die folgenden Erklärungen sowohl für das GRNN als auch das RBF–Netz gültig.

Der Netzwerkausgang  $\hat{y}$  berechnet sich wie bereits in Kapitel 3 beschrieben zu:

$$\hat{y}(u) = \mathcal{A}^T(u) \cdot \hat{\Theta} = \hat{\Theta}^T \cdot \mathcal{A}(u) \quad (4.87)$$

wobei  $\mathcal{A}(u)$  der Aktivierungsvektor beim aktuellen Eingangssignal  $u$  ist. Der tatsächliche Ausgang der zu schätzenden Nichtlinearität ist  $y$ , so dass das neuronale

Netz die Trainingspaare  $[\underline{u} \ y]$  möglichst genau wiedergeben muss. Liegen nun  $P$  solcher Trainingspaare vor, so ergibt sich ein lineares Gleichungssystem zur Bestimmung der  $p$  Gewichte.

$$\underline{y} = \mathbf{X} \cdot \hat{\Theta} \quad (4.88)$$

mit

$$\mathbf{X} = \begin{bmatrix} \mathcal{A}_1(\underline{u}[1]) & \mathcal{A}_2(\underline{u}[1]) & \dots & \mathcal{A}_p(\underline{u}[1]) \\ \mathcal{A}_1(\underline{u}[2]) & \mathcal{A}_2(\underline{u}[2]) & \dots & \mathcal{A}_p(\underline{u}[2]) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_1(\underline{u}[P]) & \mathcal{A}_2(\underline{u}[P]) & \dots & \mathcal{A}_p(\underline{u}[P]) \end{bmatrix} \quad (4.89)$$

$$\underline{y}^T = \begin{bmatrix} y[1] & y[2] & \dots & y[P] \end{bmatrix} \quad \underline{y} \in \mathbb{R}^P, \mathbf{X} \in \mathbb{R}^{P \times p}, \hat{\Theta} \in \mathbb{R}^p$$

Mit Gleichung (4.67) kann das überbestimmte Gleichungssystem gelöst werden und man erhält als Ergebnis den optimalen Stützwertvektor. Mit zunehmender Messwerteanzahl  $P$  wird jedoch die Berechnung des optimalen Stützwertvektors immer aufwändiger. Durch die Anwendung der rekursiven Form des Least–Squares–Algorithmus kann dies vermieden werden. In Gleichung (4.76) ist es erforderlich den aktuellen Regressionsvektor  $\underline{x}[k + 1]$  aufzustellen. Für das Beispiel eines RBF–Netzes bzw. GRNN gilt:

$$\underline{x}^T[k + 1] = \begin{bmatrix} \mathcal{A}_1(\underline{u}[k + 1]) & \mathcal{A}_2(\underline{u}[k + 1]) & \dots & \mathcal{A}_p(\underline{u}[k + 1]) \end{bmatrix} \quad (4.90)$$

#### 4.2.5 Bewertung des Least–Squares–Algorithmus

Die Wahl der Summe der quadratischen Fehler als Gütfunktional hat den Vorteil, dass das Gütfunktional als einziges Extremum ein eindeutiges globales Minimum [13] besitzt. Mit Hilfe des Least–Squares–Algorithmus kann die Lösung entweder offline oder mit Hilfe der rekursiven Form des Least–Squares–Algorithmus online eindeutig berechnet werden. In der Vergangenheit wurden viele numerisch stabile und schnelle Algorithmen entwickelt. Der rekursive Least–Squares–Algorithmus erreicht bereits nach  $p$  Zeitschritten eine sehr gute Lösung, was deutlich schneller ist als bei gradientenbasierten Lernverfahren. Der Einfluss von Störungen kann ohne Vergessensfaktor sehr gut eliminiert werden. Allerdings geht bei dem Beispiel des RBF–Netzes bzw. des GRNN die Lokalität des Lernvorganges weitestgehend verloren, da nicht wie bei dem Gradientenabstiegsverfahren der aktuelle Fehler, sondern die Summe über alle vergangenen Fehlerquadrate minimiert wird. Durch die Einführung eines Vergessensfaktors kann dieser Effekt reduziert werden. Abschließend sei noch erwähnt, dass auch für den rekursiven Least–Squares–Algorithmus Stabilität und Parameterkonvergenz des Lernens bewiesen werden kann. Auf den Beweis soll an dieser Stelle allerdings verzichtet werden.

## 5 Lernfähiger Beobachter

In den beiden vorangegangenen Kapiteln 3 und 4 wurden Verfahren zur statischen Funktionsapproximation mit Hilfe von neuronalen Netzen und deren Optimierung dargestellt. In diesem Kapitel wird die Identifikation statischer Nichtlinearitäten mit einer Systemidentifikation für eine spezielle Klasse von nichtlinearen dynamischen Systemen kombiniert. Zu diesem Zweck wird ein lernfähiger Beobachter [199] entworfen, der zwei Aufgaben erfüllen soll:

- Schätzung aller Systemzustände
- Identifikation einer statischen Nichtlinearität

Die Schätzwerte der Systemzustände können für Zustandsregelungen verwendet werden, wenn keine Meßwerte vorliegen. Die identifizierte Nichtlinearität, die das dynamische Verhalten beeinflußt, kann in einem nichtlinearem Regelungsverfahren (Kap. 12) schon beim Reglerentwurf berücksichtigt werden. Das Resultat der Identifikation soll ein interpretierbarer Zusammenhang  $\widehat{\mathcal{NL}}(\underline{x}, u)$  (geschätzte Nichtlinearität) sein, wobei ein stützwertbasiertes Approximationsverfahren bevorzugt wird. Dadurch läßt sich Vorwissen über die Form der Nichtlinearität einbringen.

Besonderes Augenmerk wird auf die garantierte Stabilität des Lernvorgangs gelegt, d.h. der Beobachterfehler  $e$  strebt gegen Null ( $\lim_{t \rightarrow \infty} e(t) = 0$ ) und  $\widehat{\mathcal{NL}} \rightarrow \mathcal{NL}$ .

Zunächst werden alle Ausführungen für den SISO–Fall betrachtet, eine Erweiterung auf den MISO–Fall ist aber problemlos möglich. Der SIMO– und MIMO–Fall wird schließlich in Abschnitt 5.5 betrachtet.

### 5.1 Strecken mit isolierter Nichtlinearität

Alle folgenden Betrachtungen beziehen sich auf Strecken mit „isolierter“ statischer Nichtlinearität. Der lernfähige Beobachter hat dann die Aufgabe, alle Streckenzustände zu schätzen, sowie die auftretende statische Nichtlinearität zu „lernen“ und dieses gelernte Wissen zu speichern. Dazu sind zunächst einige Definitionen notwendig:

**Definition 5.1 Strecke mit isolierter Nichtlinearität** Als Strecke mit isolierter Nichtlinearität wird eine Strecke bezeichnet, die in Zustandsdarstellung dargestellt werden kann als

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \underline{b}u + \underline{\mathcal{NL}}(\underline{x}, u) = \mathbf{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \underline{\mathcal{NL}}(\underline{x}, u) \quad \text{und} \quad y = \underline{c}^T \underline{x} + du, \quad (5.1)$$

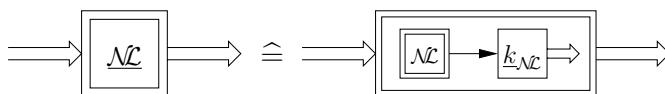
hierbei ist (siehe Abbildung 5.2)

- $u$  der skalare Systemeingang,
- $\underline{x} \in \mathbb{R}^n$  der Zustandsvektor mit  $n$  Zuständen,
- $\mathbf{A} \in \mathbb{R}^{n \times n}$  die Systemmatrix des linearen Streckenanteils,
- $\underline{b} \in \mathbb{R}^n$  der Einkopplungsvektor des Systemeingangs,
- $\underline{\mathcal{NL}}(\underline{x}, u)$  die statische Nichtlinearität,
- $\underline{k}_{\mathcal{NL}} \in \mathbb{R}^n$  der Einkoppelvektor der statischen Nichtlinearität,
- $\underline{c} \in \mathbb{R}^n$  der Auskopplungsvektor,
- $d$  der Durchgriff des Systemeingangs auf den Systemausgang und
- $y$  der skalare Systemausgang.

Die isolierte Nichtlinearität ist durch ein Produkt aus skalarer Nichtlinearität  $\underline{\mathcal{NL}}(\underline{x}, u)$  und Einkoppelvektor  $\underline{k}_{\mathcal{NL}}$  darstellbar.

Zum Beobachterentwurf wird in diesem Kapitel angenommen, dass die Systemmatrizen  $\mathbf{A}$ ,  $\underline{b}$ ,  $\underline{c}^T$ ,  $d$  und auch der Einkoppelvektor  $\underline{k}_{\mathcal{NL}}$  der Nichtlinearität konstant und bekannt sind.

Alle hier betrachteten Nichtlinearitäten sind statische Nichtlinearitäten, d.h. sie hängen nur von Zuständen und Eingängen, nicht aber explizit von der Zeit ab. Sie besitzen auch kein Gedächtnis und interne Speicher (eine Hysterese ist damit nicht darstellbar — wie Hysterese dargestellt und identifiziert werden kann, ist in Abschnitt 5.8 beschrieben). Zur Vereinfachung wird im Signalflußplan im Folgenden eine isolierte Nichtlinearität wie in Abbildung 5.1 dargestellt.



**Abb. 5.1:** Darstellung einer isolierten Nichtlinearität im Signalflußplan

Die allgemeine Darstellung einer isolierten Nichtlinearität im Signalflußplan ist in Bild 5.2 abgebildet. Darin sind alle einfacheren Anordnungen wie Rückkopplungen oder Serienschaltungen enthalten.

Für den Entwurf eines lernfähigen Beobachters stellt sich nun die Aufgabe, die isolierte Nichtlinearität durch ein neuronales Netz zu approximieren.

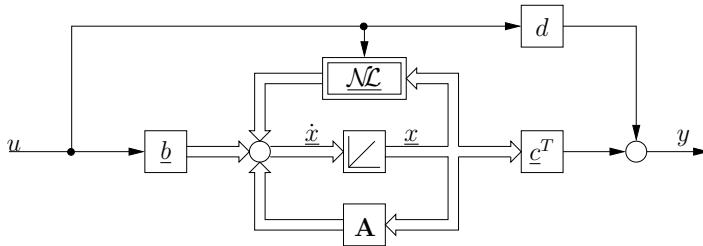


Abb. 5.2: Signalflussplan des nichtlinearen Systems mit isolierter Nichtlinearität

## 5.2 Beobachterentwurf bei messbarem Eingangsraum

Für die obige Klasse von Strecken mit isolierter Nichtlinearität wird nun ein Vorgehen zum Entwurf eines lernfähigen Beobachters vorgestellt. Das neuronale Netz wird in einen Zustandsbeobachter eingebettet und hat die Aufgabe, die isolierte Nichtlinearität  $\underline{N}(x, u)$  nachzubilden. Diese Approximation erfolgt mittels eines Lerngesetzes, in das auch der Fehler zwischen Beobachterausgang und Streckenausgang, also  $\hat{y} - y$ , eingeht.

### 5.2.1 Voraussetzungen

Die Nichtlinearität kann natürlich nur mit dem Streckenausgang identifiziert werden, wenn ihre Auswirkungen am Ausgang auch sichtbar werden.

**Definition 5.2 Sichtbarkeit der Nichtlinearität** *Die Sichtbarkeit einer Nichtlinearität am Ausgang einer SISO-Strecke ist genau dann gegeben, wenn die Übertragungsfunktion  $H_{\mathcal{N}}(s)$*

$$H_{\mathcal{N}}(s) = \underline{c}^T(s\mathbf{E} - \mathbf{A})^{-1}\underline{k}_{\mathcal{N}} \quad (5.2)$$

vom Angriffspunkt der Nichtlinearität zum Streckenausgang für alle  $s > 0$  ungleich Null ist.

Ein stabiles Lerngesetz kann dann abgeleitet werden, wenn die sogenannte Fehlerübertragungsfunktion (siehe Abschnitt 5.2.2)  $H(s) = \underline{c}^T(s\mathbf{E} - \mathbf{A} + l \underline{c}^T)^{-1}\underline{k}_{\mathcal{N}}$  asymptotisch stabil ist, d.h. alle Pole in der linken komplexen Halbebene liegen.

Ebenso muss die zeitinvariante Strecke im linearen Teil bekannt sein. Es muß Kenntnis über die Systemmatrizen, den Angriffspunkt sowie die Eingangsgrößen der Nichtlinearität vorliegen. Unter diesen Voraussetzungen kann der im Folgenden vorgestellte Beobachteransatz zur Identifikation der isolierten Nichtlinearität und zur Zustandsbeobachtung Verwendung finden.

### 5.2.2 Beobachterentwurf zur Identifikation der Nichtlinearität

Zunächst soll der Beobachterentwurf für eine Strecke mit statischer isolierter Nichtlinearität bei messbaren oder rückrechenbaren (z.B. durch Differenziation) Eingangsgrößen betrachtet werden. Das bedeutet, alle Eingänge von  $\mathcal{NL}(x_1 \dots x_k, u)$  sind bekannt (die Nichtlinearität muß aber nicht von allen Zuständen abhängen). Dies bedeutet, dass — und dies ist für das stabile und konvergente Lernen von außerordentlicher Bedeutung — das Eingangssignal der realen Nichtlinearität in der Strecke und das Eingangssignal des entsprechenden Funktionsapproximators identisch sind und somit immer die gleiche Aktivierungsfunktion bei lokal wirkenden Funktionsapproximatoren (RBF, GRNN) angeregt sind (vgl. Abbildung 5.3).

Der Eingangsraum der Nichtlinearität wird mit  $\underline{x}_E$  bezeichnet. Ziel ist die Identifikation der isolierten Nichtlinearität und die Beobachtung aller nicht messbaren Systemzustände.

#### Beobachteransatz

Der Beobachter für eine SISO-Strecke gemäß Gleichung (5.3) mit messbaren Eingangsgrößen der Nichtlinearität

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \mathcal{NL}(\underline{x}_E, u) \quad \text{und} \quad y = \underline{c}^T \underline{x} + d u \quad (5.3)$$

und bekannten und konstanten  $\mathbf{A}$ ,  $\underline{b}$ ,  $\underline{c}^T$ ,  $d$  und  $\underline{k}_{\mathcal{NL}}$  kann ähnlich zum Luenbergerbeobachter angesetzt werden als

$$\dot{\hat{x}} = (\mathbf{A} - \underline{l} \underline{c}^T) \hat{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\mathcal{NL}}(\underline{x}_E, u) + \underline{l} \underline{c}^T \underline{x} \quad \text{und} \quad \hat{y} = \underline{c}^T \hat{x} + d u \quad (5.4)$$

wobei der Beobachterfehler zu

$$e = \hat{y} - y \quad (5.5)$$

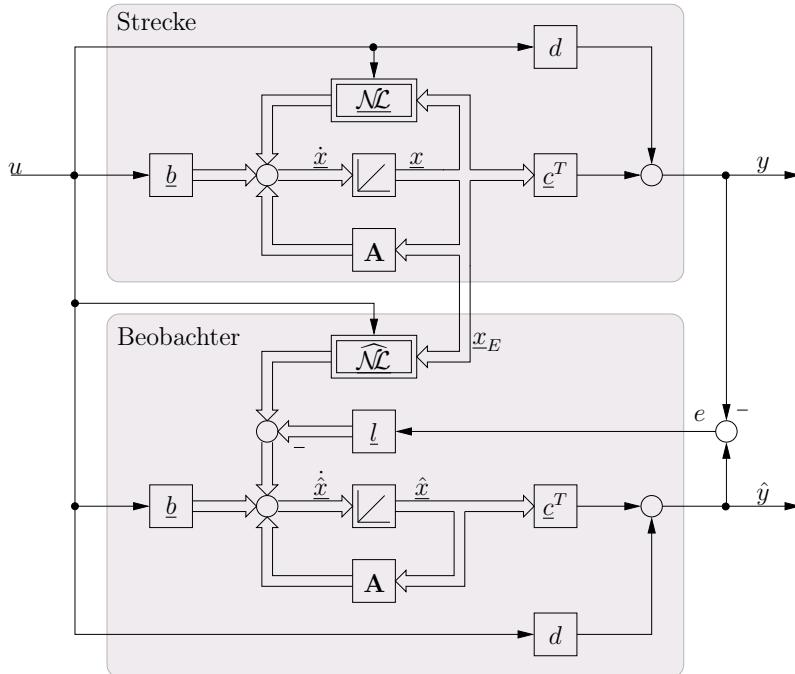
definiert wird. Bei den folgenden Betrachtungen wird nun das Ausgangssignal der isolierten Nichtlinearität als zusätzliches Eingangssignal einer ansonsten linearen Strecke betrachtet. Nur so können alle linearen Methoden wie Superpositionsprinzip und Darstellung als Übertragungsfunktion angewendet werden.

In Bild 5.3 ist die verwendete Struktur in Zustandsdarstellung gezeigt, wobei der Vektor  $\underline{x}_E$  den Eingangsraum der Nichtlinearität kennzeichnet.

Der Rückführvektor  $\underline{l}$  muß so dimensioniert werden, dass die Beobachterpole der Matrix  $(\mathbf{A} - \underline{l} \underline{c}^T)$  negativen Realteil besitzen und schneller als die Streckenpole sind, damit das Einschwingen des Beobachters garantiert ist. Die Dimensionierung kann wie im linearen Fall mittels LQ-Optimierung oder durch Polvorgabe erfolgen [142].

#### Bestimmung der Fehlerübertragungsfunktion $H(s)$

Wie aus Abbildung 5.3 zu erkennen ist, ist das Ausgangssignal der jeweils betrachteten Nichtlinearität nicht direkt messbar, sondern es ist nur das Streckenausgangssignal messbar; dies bedeutet, es ist jeweils eine zusätzliche Übertragungsfunktion  $H(s)$  zwischen Angriffspunkt (oder auch Einkoppelstelle) der



**Abb. 5.3:** Zustandsdarstellung von SISO-Strecke mit isolierter Nichtlinearität und lernfähigem Beobachter ([199], Patent)

Nichtlinearität und dem Streckenausgang wirksam. Diese Übertragungsfunktion  $H(s)$  wird Fehlerübertragungsfunktion genannt.

Zur Bestimmung der Fehlerübertragungsfunktion  $H(s)$  wird zunächst von der Zustandsdarstellung ausgegangen und dann eine gemischte Laplace–Zustandsnotation<sup>1)</sup> verwendet. Für die Strecke gilt:

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \underline{\mathcal{NL}}(\underline{x}_E, u) \quad \text{und} \quad y = \underline{c}^T \underline{x} + d u \quad (5.6)$$

Der Ansatz für den Beobachter lautet bei bekanntem  $\underline{A}$ ,  $\underline{b}$ ,  $\underline{c}$ ,  $\underline{k}_{\mathcal{NL}}$  und  $d$ :

$$\begin{aligned} \dot{\widehat{\underline{x}}} &= \widehat{\underline{A}}\widehat{\underline{x}} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\underline{\mathcal{NL}}}(\underline{x}_E, u) + \underline{l}\underline{c}^T(\underline{x} - \widehat{\underline{x}}) \quad \text{und} \quad \widehat{y} = \underline{c}^T \widehat{\underline{x}} + d u \\ \text{bzw. } \dot{\widehat{\underline{x}}} &= (\underline{A} - \underline{l}\underline{c}^T)\widehat{\underline{x}} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\underline{\mathcal{NL}}}(\underline{x}_E, u) + \underline{l}\underline{c}^T\underline{x} \quad \text{und} \quad \widehat{y} = \underline{c}^T \widehat{\underline{x}} + d u \end{aligned} \quad (5.7)$$

<sup>1)</sup> In der gemischten Laplace–Zustandsnotation werden Signale im Zeit- und im Frequenzbereich verwendet. Diese Schreibweise wird häufig in der englischsprachigen Literatur verwendet, und dient einer übersichtlicheren Schreibweise. Streng genommen ist diese Misch-Darstellung nicht korrekt, siehe Anmerkungen auf Seite 133.

Der Beobachterfehler in den Zuständen wird definiert zu

$$\underline{e}_Z = \hat{\underline{x}} - \underline{x} \quad (5.8)$$

und der Fehler im Ausgangssignal zu

$$e = \hat{y} - y \quad (5.9)$$

wobei  $e$  aus  $\underline{e}_Z$  durch folgende Beziehung hervorgeht:

$$e = \underline{c}^T \underline{e}_Z \quad (5.10)$$

Die Fehlerdifferentialgleichung des Zustandsfehlers kann somit dargestellt werden als:

$$\dot{\underline{e}}_Z = (\mathbf{A} - \underline{l} \underline{c}^T) \underline{e}_Z + \underline{k}_{\mathcal{N}\mathcal{L}} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \quad (5.11)$$

Das neuronale Netz wird im Lerngesetz jedoch mit dem Ausgangsfehler  $e$  adaptiert. Zur einfacheren Darstellung wird nun auf die Laplace-Notation übergegangen. Die Signale  $\mathcal{NL}$  und  $\widehat{\mathcal{NL}}$  werden dazu als normale Zeitsignale aufgefaßt, deren Laplace Transformation existiert. Die Fehlerdifferentialgleichung lautet dann folgendermaßen:

$$s\underline{e}_Z - \underline{e}_Z(t=0-) = (\mathbf{A} - \underline{l} \underline{c}^T) \underline{e}_Z + \underline{k}_{\mathcal{N}\mathcal{L}} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \quad (5.12)$$

Aufgrund der Annahme, die Systemmatrix  $\mathbf{A}$  sei bekannt und stabil sowie des Beobachteransatzes mit der Fehlerrückführung  $\underline{l}$  werden die unterschiedlichen Anfangszustände im realen System und im Beobachter und damit der Fehler  $\underline{e}_Z(t=0-)$  abgebaut. Damit verbleibt als Fehler  $(\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u))$ , dies ist beim Lernvorgang von großem Vorteil:

$$\begin{aligned} s\underline{e}_Z &= (\mathbf{A} - \underline{l} \underline{c}^T) \underline{e}_Z + \underline{k}_{\mathcal{N}\mathcal{L}} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \\ (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T) \underline{e}_Z &= \underline{k}_{\mathcal{N}\mathcal{L}} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \\ \underline{e}_Z &= (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}\mathcal{L}} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \end{aligned} \quad (5.13)$$

Der Ausgangsfehler  $e$  ergibt sich nun mit Gleichung (5.10) zu

$$e = \underline{c}^T \underline{e}_Z = \underbrace{\underline{c}^T (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}\mathcal{L}}}_{H(s)} \cdot (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \quad (5.14)$$

$$= H(s) (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) = \hat{y} - y \quad (5.15)$$

Die Fehlerübertragungsfunktion  $H(s)$  beschreibt somit das Übertragungsverhalten des Beobachters vom Angriffspunkt der Nichtlinearität, beschrieben durch  $\underline{k}_{\mathcal{N}\mathcal{L}}$ , auf das Fehlersignal  $e = \hat{y} - y$ . Sie hängt sowohl von den Streckenparametern als auch vom Rückführvektor  $\underline{l}$  ab, jedoch nicht vom Streckeneingang  $u$ .

Die Gleichung (5.15) und Abbildung 5.6 sind als Hammerstein-Modell zu interpretieren. Der interne Streckenzustand  $\underline{x}_E$  und der Systemeingang  $u$  bestimmen den Ausgang der Nichtlinearität  $\widehat{\mathcal{NL}}(\underline{x}_E, u)$ . Das Ausgangssignal ist also erst über die Fehlerübertragungsfunktion  $H(s)$  zu messen, d.h. am Eingang wirkt zuerst die Nichtlinearität, es folgt die Übertragungsfunktion  $H(s)$ . Die Gleichung (5.15) ist wie Gleichung (3.3) von rechts nach links zu lesen.

### Anmerkung

Für die mathematisch korrekte Darstellung des zuvor beschriebenen Sachverhaltes lautet Gleichung (5.12) mit  $\hat{y}_{\mathcal{N}\mathcal{L}}$  und  $y_{\mathcal{N}\mathcal{L}}$  als Ausgangssignal der geschätzten bzw. der zu approximierenden Nichtlinearität:

$$s\underline{e}_Z - \underline{e}_Z(t=0_-) = (\mathbf{A} - \underline{l} \underline{c}^T) \underline{e}_Z + \underline{k}_{\mathcal{N}\mathcal{L}} \cdot \underbrace{\mathcal{L}\{\{\hat{y}_{\mathcal{N}\mathcal{L}} - y_{\mathcal{N}\mathcal{L}}\}\}}_{e_{\mathcal{N}\mathcal{L}}(s)} \quad (5.16)$$

Mit  $\underline{e}_Z(t=0_-) = 0$  kann Gleichung (5.16) umgeformt werden zu:

$$\underline{e}_Z(s) = (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}\mathcal{L}} \cdot e_{\mathcal{N}\mathcal{L}}(s) \quad (5.17)$$

Der Ausgangsfehler  $e$  ergibt sich nun aus Gleichung (5.17) zu

$$e = \underline{c}^T \underline{e}_Z(s) = \underbrace{\underline{c}^T (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}\mathcal{L}}}_{H(s)} \cdot e_{\mathcal{N}\mathcal{L}}(s) \quad (5.18)$$

Diese Berechnung der Fehlerübertragungsfunktion  $H(s)$  führt auf das gleiche Ergebnis wie in Gleichung (5.14). Aus diesem Grund wird auf die explizite Laplace-Transformation der Ausgangssignale der Nichtlinearität verzichtet.

### Ableitung des stabilen Lerngesetzes

Nach Gleichung (3.3) kann eine statische Kennlinie beliebig genau durch ein GRNN dargestellt werden als

$$\mathcal{NL} = \underline{\Theta}^T \underline{\mathcal{A}}(\underline{x}) + d(\underline{x}_E, u) \quad (5.19)$$

mit dem mit steigender Stützwertezahl immer kleiner werdenden und damit vernachlässigbarem Approximationsfehler  $d(\underline{x}_E, u)$ . Die reale, zu identifizierende Nichtlinearität der Strecke ist deshalb repräsentiert durch ein bereits optimal trainiertes neuronales Netzwerk mit konstanten, jedoch unbekannten Gewichten  $\underline{\Theta}$ , gemäß Gleichung (5.20).

$$\mathcal{NL}(\underline{x}_E, u) = \underline{\Theta}^T \underline{\mathcal{A}}(\underline{x}_E, u) \quad (5.20)$$

Die zu identifizierende Nichtlinearität wird analog angesetzt zu

$$\widehat{\mathcal{NL}}(\underline{x}_E, u) = \widehat{\underline{\Theta}}^T \underline{\mathcal{A}}(\underline{x}_E, u) \quad (5.21)$$

Die Nichtlinearität wird somit durch einen zu adaptierenden Stützwertevektor  $\widehat{\underline{\Theta}}$  eines GRNN repräsentiert, dessen Aktivierung wiederum vom messbaren Eingangsraum  $\underline{x}_E$  der Nichtlinearität abhängt. Nach Gleichung (5.15) kann der Ausgangsfehler des Beobachters dargestellt werden als

$$e = \underline{c}^T e_Z = \underline{c}^T (s\mathbf{E} - \mathbf{A} + l \underline{c}^T)^{-1} \underline{k}_{\mathcal{NL}} (\widehat{\mathcal{NL}}(\underline{x}_E, u) - \mathcal{NL}(\underline{x}_E, u)) \quad (5.22)$$

$$e = H(s)(\widehat{\underline{\Theta}} - \underline{\Theta})^T \underline{\mathcal{A}}(\underline{x}_E, u) \quad (5.23)$$

Diese Art der vorteilhaften Darstellung ist durch die Messbarkeit der Eingangssignale der Nichtlinearität (vgl. Abbildung 5.1) gegeben.

Mit der Definition des sogenannten Parameterfehlervektors  $\underline{\Phi}$  zu

$$\underline{\Phi} = \widehat{\underline{\Theta}} - \underline{\Theta} \quad (5.24)$$

folgt die Fehlergleichung [158], die gemäß Abbildung 5.6 dargestellt werden kann.

$$e = H(s)\underline{\Phi}^T \underline{\mathcal{A}}(\underline{x}_E, u) \quad (5.25)$$

Erfüllt die Fehlerübertragungsfunktion  $H(s)$  die SPR-Bedingung, so kann die Adoption der Stützwerte  $\widehat{\underline{\Theta}}$  mittels des Lerngesetzes nach Fehlermodell 3 (Abb. 5.6) erfolgen; dieses führt im Vergleich zum allgemeinen Fall einer Nicht-SPR-Funktion zu einer erheblichen Ersparnis an Rechenzeit. Genauere Darstellungen zu dieser Problematik sind in Abschnitt 5.6 zu finden.

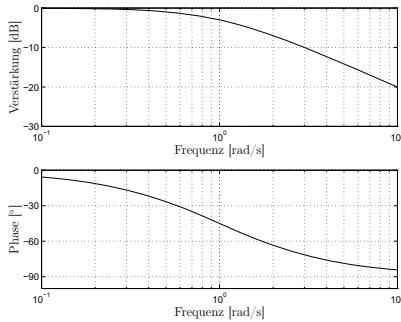
**Definition 5.3 SPR-Übertragungsfunktion** Eine Übertragungsfunktion ist eine streng positiv reelle (SPR) Übertragungsfunktion, wenn sie asymptotisch stabil ist, d.h. alle Pole in der linken Halbebene liegen und wenn der Realteil von  $H(s)$  längs der  $j\omega$ -Achse stets positiv ist.

$$\Re\{H(j\omega)\} > 0 \quad \text{für alle } \omega \geq 0 \quad (5.26)$$

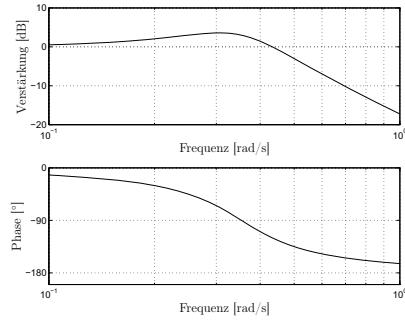
Anschaulich bedeutet die SPR-Bedingung, dass  $H(s)$  keine Phasendrehung von mehr als  $\pi/2$  hervorrufen darf. Als Beispiel sei hier ein  $PT_1$ -Glied genannt (Abb. 5.4). Für eine SPR-Übertragungsfunktion  $H(s)$  lautet die im Sinne von Ljapunov global stabile Adaptionsgleichung: (siehe auch Gleichung (4.12))

$$\dot{\underline{\Phi}} = \dot{\widehat{\underline{\Theta}}} = -\eta e \underline{\mathcal{A}}(\underline{x}_E, u) \quad \eta > 0 \quad (5.27)$$

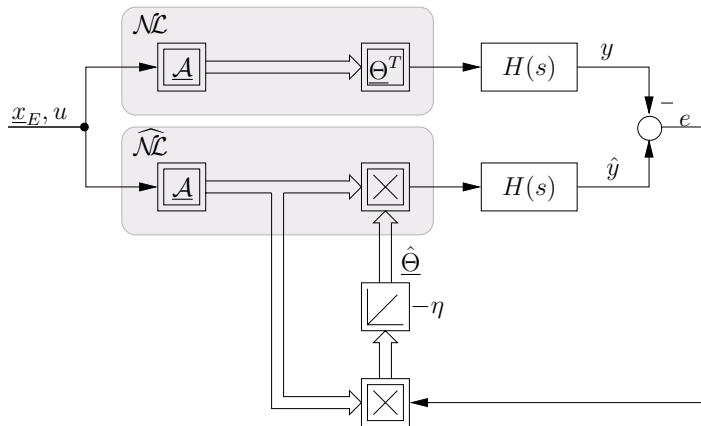
Der Parameter  $\eta$  bezeichnet die Lernschrittweite. Mit ihm kann die Geschwindigkeit des Adaptionsvorgangs beeinflusst werden.



**Abb. 5.4:** Bodediagramm der SPR–Funktion  $\frac{1}{1+s}$



**Abb. 5.5:** Bodediagramm der Nicht–SPR–Funktion  $\frac{1}{8s^2+2s+1}$



**Abb. 5.6:** Der Beobachterfehler wird als Differenz über  $H(s)$  zwischen der Nichtlinearität  $\underline{\mathcal{N}}(\underline{x}_E, u)$  und dem neuronalen Netz  $\widehat{\mathcal{N}}(\underline{x}_E, u)$  erzeugt (Fehlermodell 3)

Mit Gleichung (5.23) und Gleichung (5.27) kann der lernfähige Beobachter für eine SPR-Fehlerübertragungsfunktion als reine **Parallelstruktur** wie in Abbildung 5.6 dargestellt werden.

Erfüllt die Fehlerübertragungsfunktion  $H(s)$  die SPR-Bedingung nicht (Beispiel in Abb. 5.5), so wird ein stabiler Lernvorgang durch das Verfahren der **verzögerten Aktivierung** erreicht. Das stabile Lerngesetz lautet dann (Abbildung 5.7, siehe auch Abschnitt 5.6.4):

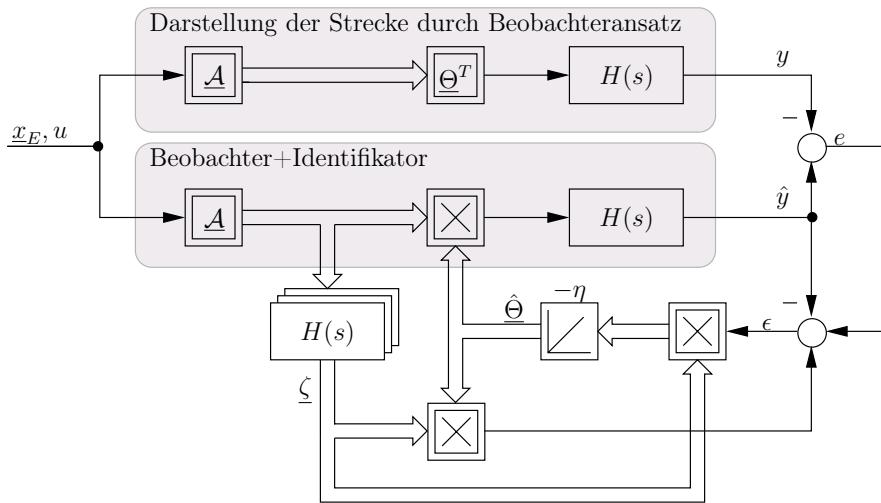
$$\dot{\underline{\Phi}} = \dot{\widehat{\Theta}} = -\eta \epsilon \underbrace{H(s)\underline{A}(\underline{x}_E, u)}_{\text{verzögerte Aktivierung}} \quad (5.28)$$

mit dem zur Adaption zu verwendenden erweiterten Fehler

$$\epsilon = e + \hat{\underline{\Theta}}^T H(s) \underline{\mathcal{A}}(\underline{x}_E, u) - H(s) \hat{\underline{\Theta}}^T \underline{\mathcal{A}}(\underline{x}_E, u) \quad (5.29)$$

Die verzögerte Aktivierung bewirkt, dass der Aktivierungsvektor  $\mathcal{A}(\underline{x}_E, u)$  so lange verzögert wird, bis die Auswirkung des Stützwertefehlers im Streckenausgang sichtbar wird. Dies stellt sicher, dass der verzögerte Aktivierungsvektor  $\zeta$  und das Fehlersignal  $e$  phasengleich sind. Der erweiterte Fehler  $\epsilon = \zeta^T \hat{\Theta} - y$  resultiert somit in einem Lern-Regelkreis, der stabil ist, da der Einfluß von  $H(s)$  durch die verzögerte Aktivierung außerhalb des Lernregelkreises angeordnet ist. Nach [158] wird dieses Lerngesetz zusammen mit dem Beobachteransatz und dem erweiterten Fehler als Fehlermodell 4 bezeichnet. Die Signalflußdarstellung des Prinzips des erweiterten Fehlers mit verzögter Aktivierung ist in Abb. 5.7 dargestellt. Der höhere Rechenaufwand entsteht durch die Verzögerung der Aktivierung durch die Fehlerübertragungsfunktion  $H(s)$ .  $H(s)$  muß so oft bereitgestellt werden, wie Stützwerte im neuronalen Netz vorhanden sind.

Die verschiedenen Fehlermodelle und die zugehörigen Beweise zur Stabilität der Parameteradaption werden in Abschnitt 5.6 näher beschrieben.



**Abb. 5.7:** Signalflußplan darstellung des Fehlermodells mit verzögerter Aktivierung (Fehlermodell 4, siehe Abschnitt 5.6.4)

Mit der Stabilitätstheorie nach Ljapunov ist in [158] bewiesen, dass für Fehlermodell 3 (SPR) **und** Fehlermodell 4 (verzögerte Aktivierung) für  $\eta > 0$  gilt

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (5.30)$$

und zusammen mit den Eigenschaften des neuronalen Netzes folgt nach [137] bei ausreichender Anregung (*persistent excitation*) durch  $\underline{\mathcal{A}}(\underline{x}_E, u)$  auch

$$\lim_{t \rightarrow \infty} \hat{\Theta} = \underline{\Theta} \quad (5.31)$$

Das bedeutet anschaulich, dass der Parameterfehlervektor  $\Phi$  zu Null geworden ist, wenn auch der Fehler  $e$  zu Null geworden ist, d. h. die Abbildung der Nichtlinearität ist konvergent.

### 5.2.3 Parameterkonvergenz

Das Problem der ausreichenden Anregung (*persistent excitation*) ist immer dann gegeben, wenn mit einem einzigen Fehlersignal mehrere Parameter adaptiert werden sollen. Die Fragestellung lautet also, ob bei einem bestimmten Eingangssignal  $\underline{x}_E$  und  $u$  welches dem Anregungssignal des GRNN entspricht, wirklich nur ein Parametersatz  $\hat{\Theta}$  das gewünschte Ergebnis erzeugt. Da diesem Anregungssignal eindeutig die Aktivierungsfunktionen des GRNN zugeordnet werden können, werden im Folgenden diese Aktivierungsfunktionen  $\underline{\mathcal{A}}(\underline{x}_E, u)$  betrachtet.

Es soll zunächst die mathematische Bedingung für *persistent excitation* angegeben werden, die für das GRNN bewiesen ist. Dabei soll die Definition nach [158] verwendet werden.

**Definition 5.4 Persistent Excitation** Wenn  $\underline{\mathcal{A}}(\underline{x}_E(t), u(t)) = \underline{\mathcal{A}}(t)$  stückweise stetig ist und für alle Einheitsvektoren  $\underline{v}_i$ ,  $1 \leq i \leq p$ , die den  $p$ -dimensionalen Raum  $\mathbb{R}^p$  aufspannen, und für jedes positive  $\varepsilon_0$  und  $t_0$  ein endliches Zeitintervall  $T$  gefunden werden kann, so dass gilt

$$\frac{1}{T} \int_t^{t+T} |\underline{\mathcal{A}}^T(\tau) \underline{v}_i| d\tau \geq \varepsilon_0 \quad \text{für alle } t \geq t_0 \quad (5.32)$$

dann ist das Signal  $\underline{\mathcal{A}}(t)$  persistently exciting.

Im Fall des GRNN gilt  $\underline{\mathcal{A}}(\underline{x}_E(t), u(t)) = \underline{\mathcal{A}}(t)$ . In [137] ist für ein GRNN mit Gaußscher Aktivierungsfunktion bewiesen, dass  $\underline{\mathcal{A}}(\underline{x}_E, u)$  immer persistently exciting ist. Der Beweis beruht darauf, dass beim GRNN jedes Neuron, wenn auch nur sehr gering, aktiviert ist. In praktischen Anwendungen bedeutet dies jedoch wegen der endlichen Rechengenauigkeit realer Rechner, dass Parameterkonvergenz

$$\lim_{t \rightarrow \infty} \hat{\Theta}_i = \Theta_i \quad (5.33)$$

nur für die Neuronen gewährleistet ist, deren Eingangsbereich auch bis zur Fehlerkonvergenz  $e = 0$  angeregt wurde.

Für das GRNN bedeutet das nun zusammenfassend: Für den bis zur Fehlerkonvergenz durchfahrenen Eingangsbereich des neuronalen Netzes sind die Stützwerte  $\hat{\Theta}_i$  gegen die wahren Werte  $\Theta_i$  (unter Vernachlässigung des Approximationfehlers) konvergiert.

Damit ist gezeigt, dass mit diesem Beobachteransatz eine isolierte Nichtlinearität  $\mathcal{NL}(\underline{x}_E, u)$  garantiert stabil und auch eindeutig durch das GRNN nachgebildet werden kann. Bisher wurde immer noch die Messbarkeit des Eingangsräumes der Nichtlinearität vorausgesetzt. Diese Bedingung wird im Folgenden nicht mehr benötigt, und man erhält dadurch ein wesentlich breiteres Einsatzspektrum in realen Anwendungen.

### 5.3 Beobachterentwurf bei nicht messbarem Eingangsraum

Im Gegensatz zu den vorherigen Ableitungen ist nun die Messbarkeit des gesamten Eingangsräumes der Nichtlinearität  $\mathcal{NL}(\underline{x}_E, u)$  nicht mehr gefordert, vielmehr genügt der Streckenausgang und -eingang.

#### 5.3.1 Zusätzliche Voraussetzung

Da jetzt eine Messung des Eingangsräumes der Nichtlinearität nicht mehr vorliegt, muß für den linearen Teil der Strecke zusätzlich zu allen Voraussetzungen in Abschnitt 5.2.1 noch die vollständige Zustandsbeobachtbarkeit gefordert werden.

**Definition 5.5 Beobachtbarkeit im linearen Teil** Eine SISO Strecke der Ordnung  $n$  mit isolierter Nichtlinearität

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \mathcal{NL}(\underline{x}, u) \quad \text{und} \quad y = \underline{c}^T \underline{x} + du \quad (5.34)$$

ist im linearen Teil genau dann vollständig zustandsbeobachtbar, wenn die Beobachtbarkeitsmatrix  $\mathbf{Q}_{obs}$

$$\mathbf{Q}_{obs} = [\underline{c} \quad \mathbf{A}^T \underline{c} \dots (\mathbf{A}^T)^{(n-1)} \underline{c}] \quad (5.35)$$

regulär ist, wenn also gilt

$$\det(\mathbf{Q}_{obs}) \neq 0 \quad (5.36)$$

Die Forderung nach Beobachtbarkeit im linearen Teil ist nötig, da die Streckenzustände, von denen die reale Nichtlinearität angeregt wird, nun beobachtet werden müssen. Mit diesen Voraussetzungen kann nun wieder ein lernfähiger Beobachter ähnlich zum vorhergehenden Abschnitt 5.2 entworfen werden.

#### 5.3.2 Beobachteransatz analog Luenberger

Wie beim Beobachter mit messbarem Eingangsraum der Nichtlinearitäten wird hier der Beobachter analog zum Luenberger-Beobachter angesetzt. Die jeweils zu identifizierende Nichtlinearität ist wieder eine Funktion des realen Eingangsräums  $\underline{x}_E$ . Dieser kann jedoch wegen der fehlenden Messung nicht bestimmt

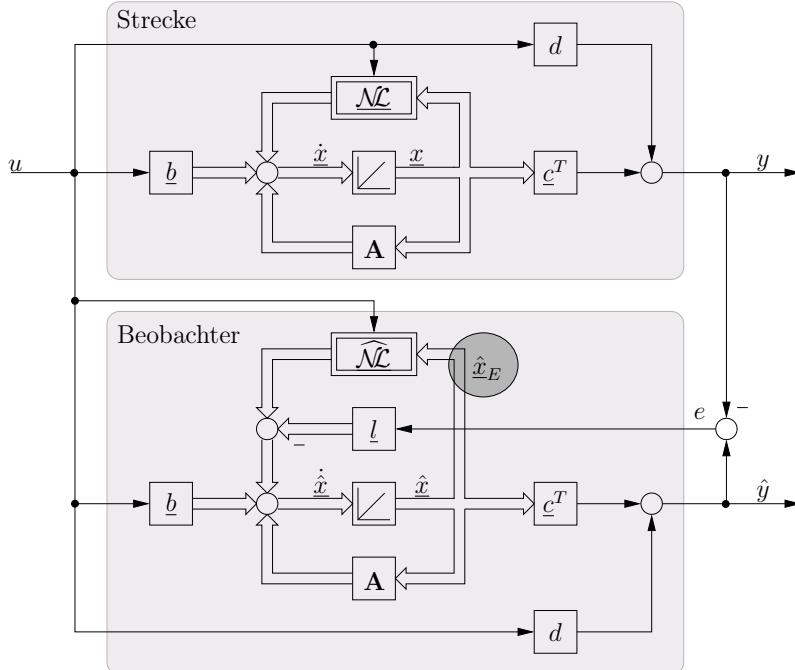
werden; deshalb wird nun im Beobachter die Nichtlinearität als Funktion vom beobachteten Eingangsraum  $\hat{x}_E$  abhängig angesetzt. Die Strecken-Darstellung lautet wieder

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\mathcal{NL}}(\hat{x}_E, u) \quad \text{und} \quad y = \underline{c}^T \underline{x} + du \quad (5.37)$$

Mit bekanntem  $\mathbf{A}$ ,  $\underline{b}$ ,  $\underline{c}$ ,  $\underline{k}_{\mathcal{NL}}$  und  $d$  lautet der Beobachteransatz nun

$$\dot{\hat{x}} = (\mathbf{A} - \underline{l} \underline{c}^T) \hat{x} + \underline{b}u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\mathcal{NL}}(\hat{x}_E, u) + \underline{l} \underline{c}^T \underline{x} \quad \text{und} \quad \hat{y} = \underline{c}^T \hat{x} + du \quad (5.38)$$

mit einstellbarem Rückführvektor  $\underline{l}$ . In Abbildung 5.8 ist die Signalflußdarstellung dieses Ansatzes dargestellt. Geht man davon aus, dass zu Beginn der



**Abb. 5.8:** Lernfähiger Zustandsbeobachter mit isolierter Nichtlinearität; Ziel: Beobachtung nichtmessbarer Zustände und Identifikation der Nichtlinearität ([199], Patent)

Zustandsschätzung  $\hat{x}(0) = \underline{x}(0)$  gilt, so folgt aus Gleichung (5.38) dass der Beobachterfehler nur auf die unbekannte Nichtlinearität zurückzuführen ist. Für den Fehler gilt nämlich

$$\dot{e}_z = (A - lc^T) e_z + \underline{k}_{\mathcal{NL}} [\widehat{\mathcal{NL}}(\hat{x}_E, u) - \mathcal{NL}(\underline{x}, u)] \quad (5.39)$$

Der Zustandsfehlervektor  $e_z$  reduziert sich am Ausgang des Systems zu  $e = \underline{c}^T e_z$ . Mit der Fehlerübertragungsfunktion

$$H(s) = \underline{c}^T (s\mathbf{E} - \mathbf{A} + l \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}} \quad (5.40)$$

ergibt sich die äquivalente Fehlergleichung im Laplacebereich:

$$e = H(s) [\widehat{\mathcal{N}}(\hat{x}_E, u) - \mathcal{N}(x_E, u)] = H(s) [\hat{\Theta}^T \underline{\mathcal{A}}(\hat{x}_E, u) - \underline{\Theta}^T \underline{\mathcal{A}}(x_E, u)] \quad (5.41)$$

## Einführung virtueller Stützwerte

Um die Fehlergleichung nun auch mit einem Parameterfehlervektor  $\underline{\Phi}$  darstellen zu können und somit die aus der Literatur bekannten Lerngesetze anwenden zu können, werden virtuelle Stützwerte eingeführt.

Die zu approximierende Nichtlinearität  $\widehat{\mathcal{N}}$  kann nun **nicht** durch  $\hat{\Theta}^T \underline{\mathcal{A}}(x_E, u)$  dargestellt werden, da der Eingangsraum der Nichtlinearität nicht mehr als Messgröße, sondern nur noch als beobachtete Größe vorliegt. Somit kann die Aktivierung  $\underline{\mathcal{A}}(x_E, u)$  nicht bestimmt werden. Es gilt also während des Lernvorgangs:

$$\underline{\mathcal{A}}(x_E, u) \neq \underline{\mathcal{A}}(\hat{x}_E, u) \quad (5.42)$$

Man kann jedoch ansetzen

$$\mathcal{N} = \underline{\Theta}^T \underline{\mathcal{A}}(x_E, u) = \underline{\Theta}^T \underline{\mathcal{A}} \hat{\equiv} \check{\underline{\Theta}}^T \hat{\underline{\mathcal{A}}} = \check{\underline{\Theta}}^T \underline{\mathcal{A}}(\hat{x}_E, u) \quad (5.43)$$

mit unbekannten aber durch die Gleichung

$$\check{\underline{\Theta}}^T \hat{\underline{\mathcal{A}}} \hat{\equiv} \underline{\Theta}^T \underline{\mathcal{A}} \quad (5.44)$$

bestimmten virtuellen Stützwerten  $\check{\underline{\Theta}}$ . Diese virtuellen Stützwerte  $\check{\underline{\Theta}}$  beschreiben somit die Bewegung der Nichtlinearität im Beobachterzustandsraum  $\hat{x}$  und sind zeitvariant, da sie sich ändern, wenn sich  $x$  und  $\hat{x}$  und somit auch die Aktivierungen  $\underline{\mathcal{A}}$  und  $\hat{\underline{\mathcal{A}}}$  ändern. Wegen der Lokalität der Stützwertewirkung gilt nach [137] jedoch, dass sich die virtuellen Stützwerte den realen angleichen, wenn sich die Beobachterzustände den Streckenzuständen angleichen, da dann

$$\hat{x} \rightarrow x \Rightarrow \hat{\underline{\mathcal{A}}} \rightarrow \underline{\mathcal{A}} \quad (5.45)$$

$$\hat{x} \rightarrow x \Rightarrow \check{\underline{\Theta}} \rightarrow \underline{\Theta} \quad (5.46)$$

gilt. Mit dem Einschwingen der Beobachterzustände konvergieren also auch die virtuellen Stützwerte gegen die realen. Es gilt nun zu zeigen, wann dies der Fall ist. Mit Gleichung (5.41) kann man ansetzen

$$\begin{aligned} e &= H(s) (\widehat{\mathcal{N}} - \mathcal{N}) \\ &= H(s) (\hat{\Theta}^T \underline{\mathcal{A}}(\hat{x}_E, u) - \underline{\Theta}^T \underline{\mathcal{A}}(x_E, u)) \\ &= H(s) (\hat{\Theta}^T \underline{\mathcal{A}}(\hat{x}_E, u) - \check{\underline{\Theta}}^T \underline{\mathcal{A}}(\hat{x}_E, u)) \\ &= H(s) (\hat{\Theta}^T - \check{\underline{\Theta}}^T) \underline{\mathcal{A}}(\hat{x}_E, u) \\ &= H(s) \underline{\Phi}^T \underline{\mathcal{A}}(\hat{x}_E, u) \end{aligned} \quad (5.47)$$

wobei der Parameterfehlervektor zu

$$\underline{\Phi} = \hat{\underline{\Theta}} - \check{\underline{\Theta}} \quad (5.48)$$

definiert wird. Im Falle eines nicht messbaren Eingangsraumes kann man demnach ansetzen

$$e = H(s) \underline{\Phi}^T \mathcal{A}(\hat{x}_E, u) \quad (5.49)$$

Für diese Fehlergleichung kann wieder die aus [158] bekannte global stabile Adaptionsgleichung angewandt werden.

$$\dot{\underline{\Phi}} = -\eta \epsilon H(s) \mathcal{A}(\hat{x}_E, u) \quad (5.50)$$

mit einem verallgemeinerten Fehler  $\epsilon$ , auf den in Kapitel (5.6.4) näher eingegangen wird.

$$\epsilon = e + \hat{\underline{\Theta}}^T H(s) \mathcal{A}(\hat{x}_E, u) - H(s) \hat{\underline{\Theta}}^T \mathcal{A}(\hat{x}_E, u) \quad (5.51)$$

Infolge der Zeitvarianz der virtuellen Stützwerte kann man nun aber **nicht** folgern, dass  $\dot{\underline{\Phi}} = \dot{\underline{\Theta}}$  ist. Die interessierenden Größen sind aber nach wie vor die Gewichte  $\hat{\Theta}_i$ . Im folgenden wird daher das Adaptionsgesetz

$$\dot{\hat{\underline{\Theta}}} = -\eta_{virt} \epsilon H(s) \mathcal{A}(\hat{x}_E, u) \quad (5.52)$$

verwendet, was für die Änderung des Parameterfehlervektors bedeutet

$$\dot{\underline{\Phi}} = \dot{\hat{\underline{\Theta}}} - \dot{\underline{\Theta}} = -\eta_{virt} \epsilon H(s) \mathcal{A}(\hat{x}_E, u) - \dot{\underline{\Theta}} = -\left(\eta_{virt} \epsilon H(s) \mathcal{A}(\hat{x}_E, u) + \dot{\underline{\Theta}}\right) \quad (5.53)$$

Aus Gleichung (5.53) und (5.50) 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  $\dot{\underline{\Theta}}$  gegenüber  $\dot{\underline{\Theta}}$  überwiegt. Da die Lernschrittweite  $\eta_{virt}$  positiv sein muß, ergibt sich eine Bedingung der Art

$$0 < \eta_{min} < \eta_{virt} < \eta_{max} \quad (5.54)$$

Zusammenfassend kann man feststellen, dass hiermit ein gedanklicher Ansatz besteht, mit dem, auch bei nicht messbaren Zuständen, die isolierte Nichtlinearität interpretierbar zu identifizieren ist und alle nichtmessbaren Systemzustände beobachtet werden können.

## 5.4 Zweimassensystem mit Reibung

In diesem Abschnitt wird anhand von Simulationen die Funktionsweise des lernfähigen Beobachters mit und ohne messbare Eingangssignale der Nichtlinearität untersucht. Als Beispiel dient ein elastisches, rotatorisch angeordnetes Zweimassensystem mit viskoser Reibung (keine Haftriebung!). Wie bereits mehrfach in

den vorherigen Abschnitten betont wurde, soll ein physikalisch interpretierbares Modell verwendet werden, d. h. es wird die Zustandsdarstellung gewählt, wobei der nichtlineare Einfluß durch einen Funktionsapproximator nachgebildet wird. Die Zustandsdarstellung ist durch folgende Gleichungen gegeben:

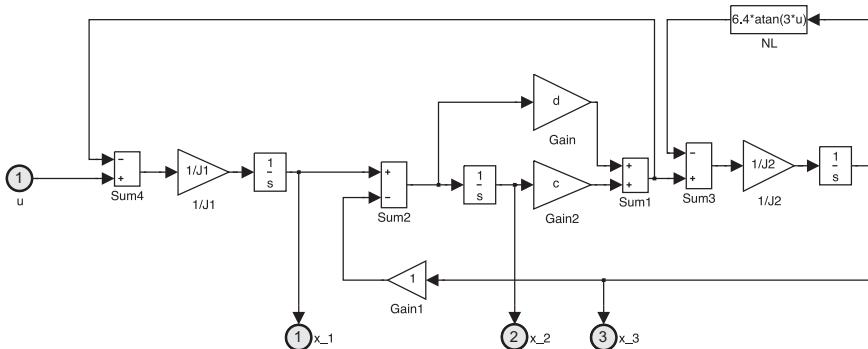
$$\dot{\underline{x}} = \underbrace{\begin{bmatrix} -\frac{d}{J_1} & -\frac{c}{J_1} & \frac{d}{J_1} \\ 1 & 0 & -1 \\ \frac{d}{J_2} & \frac{c}{J_2} & -\frac{d}{J_2} \end{bmatrix}}_{\mathbf{A}} \underline{x} + \underbrace{\begin{bmatrix} \frac{1}{J_1} \\ 0 \\ 0 \end{bmatrix}}_{\underline{b}} u + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J_2} \end{bmatrix}}_{k_{NL}} \cdot \underbrace{6.4 \arctan(3x_3)}_{\mathcal{NL}(x_3)} \quad (5.55)$$

$$y = \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}_{\underline{c}^T} \cdot \underline{x}$$

Der Zustandsvektor ist

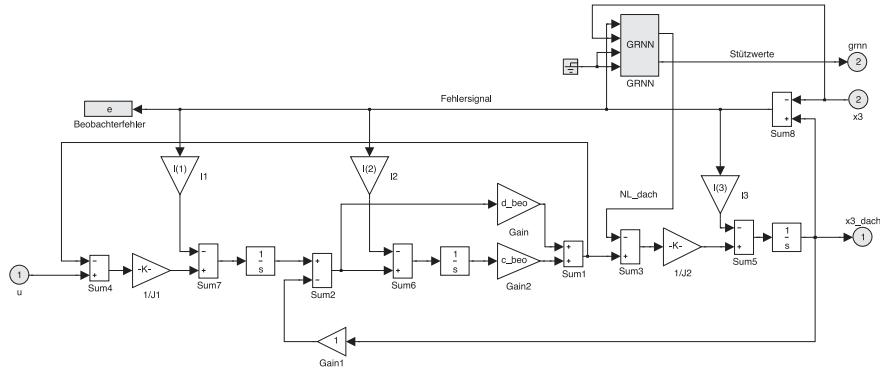
$$\underline{x} = \begin{bmatrix} N_1 \\ \Delta\alpha \\ N_2 \end{bmatrix} \quad (5.56)$$

wobei  $N_1$  die Motordrehzahl,  $N_2$  die Lastdrehzahl und  $\Delta\alpha$  die Wellenverdrehung der elastischen Verbindung ist. Der Eingang  $u$  ist das wirksame Antriebsmoment des Motors. Die isolierte Nichtlinearität stellt die viskose Reibung an der rotierenden Lastmasse dar. Der Signalflußplan des betrachteten Systems ist in Abb. 5.9 dargestellt. Zunächst wird angenommen, dass das Eingangssignal  $x_3$  der Nicht-



**Abb. 5.9:** Simulink-Modell des nichtlinearen Zweimassensystems

linearität messbar ist, d.h. der lernfähige Beobachter kann gemäß Abschnitt 5.2 ausgelegt werden. Die Struktur des lernfähigen Beobachters zeigt nachfolgende Abb. 5.10. Der Eingang des GRNN ist die gemessene Größe  $x_3$  ( $N_2$ ). Das Lern-



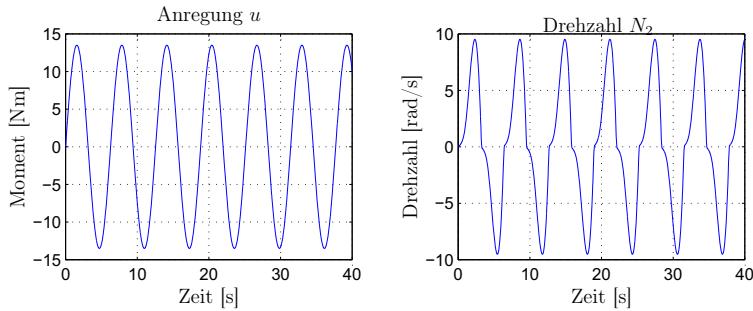
**Abb. 5.10:** Struktur des lernfähigen Beobachters für das nichtlineare Zweimassensystem (Hinweis: Das Erdungszeichen im Modell kennzeichnet, dass die entsprechenden Eingänge nicht belegt sind.)

gesetz und die verzögerte Aktivierung sind im Block *GRNN* gemäß den Gleichungen (5.28) und (5.29) realisiert. Die Pole des Identifikators sind gemäß dem Dämpfungsoptimum [204] bei einer Ersatzzeit von  $T_{ers} = 100$  ms eingestellt. Die Fehlerübertragungsfunktion  $H(s)$  ergibt sich aus der Zustandsdarstellung und den Rückführkoeffizienten  $\underline{l}$  zu

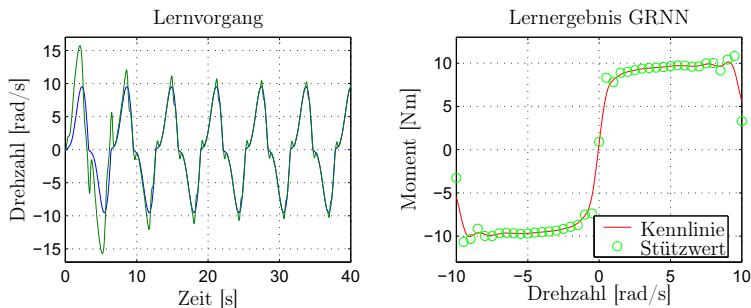
$$\begin{aligned}
 H(s) &= \underline{c}^T (s\mathbf{E} - \mathbf{A} + \underline{l} \underline{c}^T)^{-1} \underline{k}_{\mathcal{N}\mathcal{L}} = \\
 &= -(s^2 J_1 + sd + c)/(s^3 J_1 J_2 + s^2 (J_1 d + J_1 l_3 J_2 + J_2 d) + \\
 &\quad + s(J_1 c + J_1 d l_2 + d l_3 J_2 + c J_2 + d l_1 J_1) + c J_2 l_3 + c l_1 J_1) \quad (5.57)
 \end{aligned}$$

Im GRNN werden 41 Stützwerte über einen Bereich  $-10 \text{ [rad/s]} \leq N_2 \leq 10 \text{ [rad/s]}$  äquidistant verteilt. Der Glättungsfaktor  $\sigma$  ist gleich dem 0.75-fachen Stützwertabstand. Abb. 5.11 zeigt das Anregungssignal  $u$  und das Ausgangssignal  $N_2 = x_3$  der Strecke. Der Identifikationsvorgang, d.h. die Konvergenz des Beobachterausgangssignals gegen das Streckenausgangssignal ist aus Abb. 5.12 links ersichtlich. Das Identifikationsergebnis der Nichtlinearität nach einer Dauer von 40 s ist in Abb. 5.12 rechts dargestellt. Der Abfall der Kennlinie an den Randbereichen resultiert daraus, dass die Stützwerte noch nicht genügend oft aktiviert worden sind. Mit fortlaufender Lerndauer werden auch diese Stützwerte richtig gelernt.

Bei der Fehlerübertragungsfunktion  $H(s)$  handelt es sich um eine Nicht-SPR-Funktion, d.h. im Lerngesetz ist die verzögerte Aktivierung in Kombination mit dem erweiterten Fehler zu verwenden. Wenn nun trotzdem das einfache Lerngesetz gemäß Fehlermodell 3 verwendet wird, so ergibt sich der in Abb. 5.13 dargestellte Verlauf.

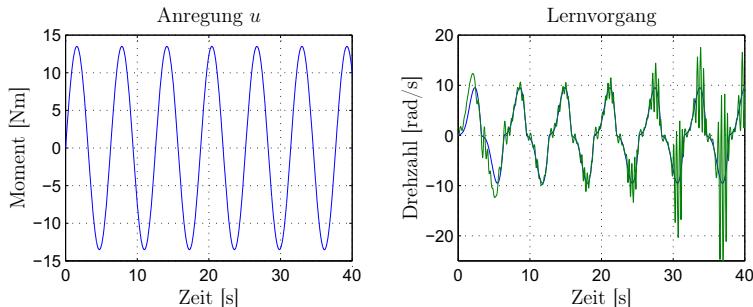


**Abb. 5.11:** Anregungssignal und Ausgangssignal des Zweimassensystems



**Abb. 5.12:** Lernvorgang und Identifikationsergebnis nach 40 s

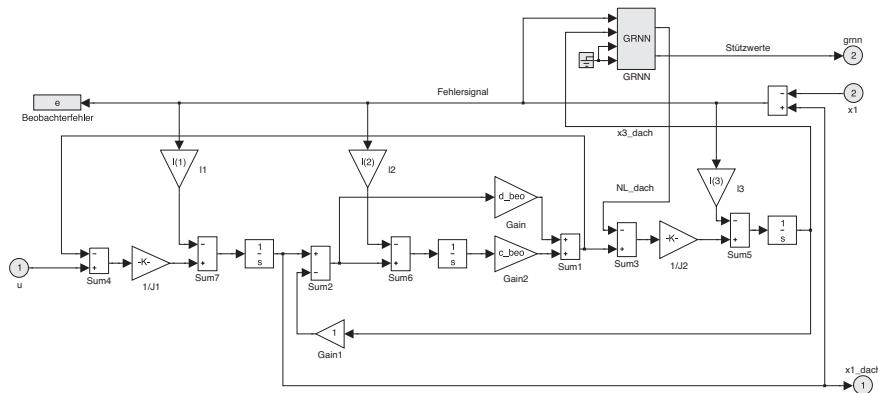
Da die Phasenverschiebung durch die Fehlerübertragungsfunktion von der Frequenz abhängig ist, ergibt sich folgendes Verhalten: Für niedere Frequenzen des Eingangssignals  $N_2$  des GRNN, für die die SPR–Bedingung noch erfüllt ist,



**Abb. 5.13:** Anregung und Lernvorgang bei Mißachtung der SPR–Bedingung im Lerngesetz

erfolgt zunächst korrektes Lernen. Der Beobachter konvergiert scheinbar gegen den Streckenausgang. Die Frequenzanteile, die eine größere Phasenverschiebung als  $\pi/2$  durch die Fehlerübertragungsfunktion erfahren, werden zwar einerseits stärker gedämpft als niedere Frequenzen, bewirken aber aufgrund der fehlenden Fehlerübertragungsfunktion im Lerngesetz einen instabilen Lernvorgang und wirken sich nach längerer Lerndauer aus. Der Identifikator wird schließlich instabil, da sich der Beitrag der höheren Frequenzen zum Lernergebnis immer weiter aufsummiert.

In einem weiteren Beispiel wird wieder von einer korrekten Implementierung des Lerngesetzes einschließlich verzögerter Aktivierung ausgegangen. Die Eingangsgröße des GRNN ist nun aber nicht mehr das gemessene Signal  $x_3$ , sondern das beobachtete Signal  $\hat{x}_3$ . Der Beobachterentwurf erfolgt nun gemäß Abschnitt 5.3, insbesondere ist die Ungleichungsbedingung nach Gleichung (5.54) zu beachten. Den Signalflussplan des Beobachters ohne messbare Eingangsgröße der Nichtlinearität zeigt Abb. 5.14.



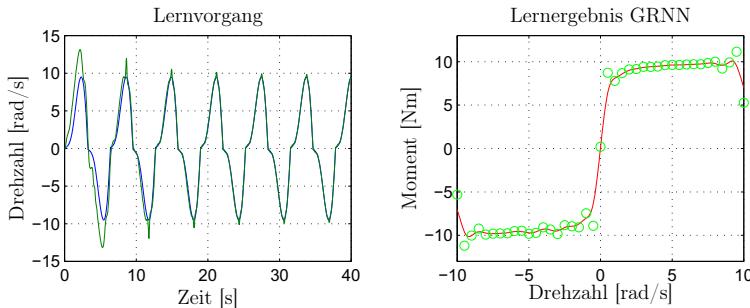
**Abb. 5.14:** Signalflussplan des Beobachters für das nichtlineare Zweimassensystem bei nicht messbarem Eingangssignal der Nichtlinearität (Hinweis: Das Erdungszeichen im Modell kennzeichnet, dass die entsprechenden Eingänge nicht belegt sind.)

Als Meßsignal und Fehlervergleichstelle liegt jetzt nicht mehr die Lastdrehzahl, sondern die Motordrehzahl  $x_1 = N_1$  vor. Die Ausgangsgleichung ändert sich dadurch zu

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{\underline{c}^T} \cdot x \quad (5.58)$$

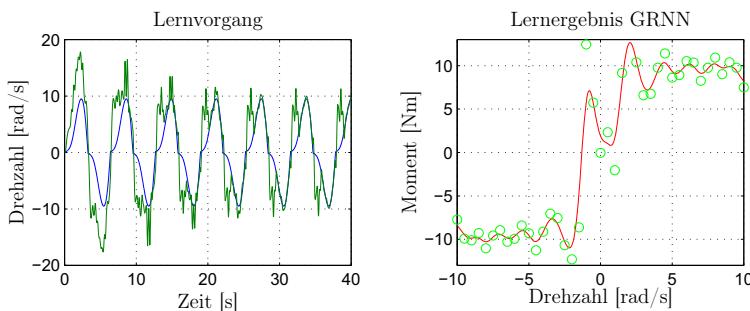
Die Beobachterpole werden wieder gemäß dem Dämpfungsoptimum mit einer Ersatzzeit von  $T_{ers} = 100$  ms festgelegt. Die Fehlerübertragungsfunktion kann in Analogie zu Gleichung (5.57) mit verändertem Ausgangsvektor  $\underline{c}$  berechnet werden. Die Dimensionierung des GRNN ist identisch mit dem vorherigen Fall bei

messbarem Eingangssignal der Nichtlinearität. Die Anregung der Strecke wurde ebenfalls identisch gewählt (siehe Abb. 5.11). Lediglich die Lernschrittweite  $\eta$  wird nun variiert, damit die Auswirkungen auf den Lernvorgang untersucht werden können. Bei einer Lernschrittweite von  $\eta = 40$  ergibt sich ein ähnlicher Verlauf wie im Falle mit messbarem Eingangsraum. Der Lernvorgang ist in Abb. 5.15 dargestellt.



**Abb. 5.15:** Stabiler Lernvorgang bei nicht messbarem Eingangssignal der Nichtlinearität

Wählt man die Lernschrittweite  $\eta$  zu klein, so kann der Fall eintreten, dass die Bewegung der virtuellen Stützwerte  $\dot{\Theta}$  gegenüber der gewünschten Lernrichtung überwiegt und dadurch ein instabiler Lernvorgang verursacht wird (Abschnitt 5.3). Für  $\eta$  muß eine untere Schranke  $\eta_{min}$  beachtet werden, deren Wert allerdings nicht ohne weiteres angegeben werden kann. Wählt man bei sonst unveränderten Daten eine Lernschrittweite von  $\eta = 10$ , so wurde  $\eta_{min}$  offensichtlich unterschritten und stabiles Lernen ist nicht mehr gewährleistet. Den zugehörigen Lernvorgang zeigt Abb. 5.16.



**Abb. 5.16:** Lernvorgang und -ergebnis nach 40 s bei Unterschreitung von  $\eta_{min}$

Auch nach längerer Lerndauer kann keine stationäre Lösung für  $\mathcal{NL}$  gefunden werden und der Lernvorgang ist nicht asymptotisch stabil.

Wenn das Eingangssignal der Nichtlinearität nicht messbar ist, so muß besonders auf die Wahl der Lernschrittweite und die Wahl der Beobachterpole geachtet werden (beide beeinflussen sich gegenseitig). Vor einer praktischen Implementierung sind unbedingt simulative Untersuchungen zur angesprochenen Parameterwahl nötig.

## 5.5 Identifikation mehrerer Nichtlinearitäten

Da komplexe Systeme häufig mehr als nur eine Nichtlinearität enthalten, ist eine Erweiterung auf mehrere isolierte Nichtlinearitäten erforderlich. Daher betrachten wir jetzt nichtlineare Systeme mit mehreren Ein- und Ausgängen mit folgender (MIMO)-Systembeschreibung,

$$\dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} + \underline{\mathcal{NL}}(\underline{x}_E) \quad (5.59)$$

$$\underline{y} = \mathbf{C}\underline{x} \quad (5.60)$$

$$\underline{\mathcal{NL}}(\underline{x}_E) = \sum_{i=1}^k e_{\mathcal{NL},i} \cdot \mathcal{NL}_i \quad (5.61)$$

wobei der Vektor  $\underline{\mathcal{NL}}(\underline{x}_E)$   $k$  unbekannte Komponenten besitzt, deren Anzahl den Rang der Ausgangsmatrix  $\mathbf{C}$  nicht übersteigen darf, d.h.

$$k \leq \text{Rang}(\mathbf{C}) \quad (5.62)$$

Den Beobachter setzen wir, analog zu unseren Überlegungen bei den SISO Systemen, zu

$$\begin{aligned} \dot{\hat{x}} &= \mathbf{A}\hat{x} + \mathbf{B}\underline{u} + \sum_{i=1}^k e_{\mathcal{NL},i} \widehat{\mathcal{NL}}_i(\underline{x}_E) - \mathbf{L}(\hat{y} - \underline{y}) \\ \hat{y} &= \mathbf{C}\hat{x} \end{aligned} \quad (5.63)$$

an. Beobachtbarkeit und Stabilität für den linearen Beobachterteil setzen wir wiederum voraus, so dass die gebräuchliche hybride Notation (Laplace Bereich + Nichtlinearität) verwendet werden kann. Damit ergibt sich für den  $i$ -ten Beobachterausgang der Zusammenhang

$$\hat{y}_i = c_i(s\mathbf{E} - \mathbf{A} + \mathbf{LC})^{-1} \left( \mathbf{B}\underline{u} + \mathbf{LC}\underline{x} + \sum_{i=1}^k e_{\mathcal{NL},i} \widehat{\mathcal{NL}}_i \right) \quad (5.64)$$

Analog erhält man für den  $i$ -ten Streckenausgang

$$y_i = c_i(s\mathbf{E} - \mathbf{A})^{-1} \left( \mathbf{B}\underline{u} + \sum_{i=1}^k e_{\mathcal{NL},i} \mathcal{NL}_i \right) \quad (5.65)$$

woraus sich der  $i$ -te Beobachterfehler gemäß Gleichung (5.14) zu

$$\underline{e}_i = \hat{y}_i - y_i = \underline{c}_i(s\mathbf{E} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1} \sum_{i=1}^n \underline{e}_{\mathcal{N}\mathcal{L},i} (\widehat{\mathcal{NL}}_i - \underline{\mathcal{NL}}_i) \quad (5.66)$$

bestimmt.  $c_i$  steht für den  $i$ -ten Zeilenvektoren der Ausgangsmatrix  $\mathbf{C}$ . Die unbekannten Komponenten von  $\underline{\mathcal{NL}}(\underline{x}_E)$  fassen wir zum Vektor  $\underline{\mathcal{NL}}(\underline{x}_E)^k$  der Länge  $k$  zusammen. Durch die Wahl ebensovieler Beobachterfehler  $\underline{e}^k$  erhalten wir letztlich ein System von  $k$  Fehlerdifferentialgleichungen der Form

$$\underline{e}^k = \mathbf{H}_F(s) \left( \widehat{\mathcal{NL}}(\underline{x}_E)^k - \underline{\mathcal{NL}}(\underline{x}_E)^k \right) \quad (5.67)$$

mit der Fehlerübertragungsmatrix  $\mathbf{H}_F(s)$ , deren Elemente sich aus Gleichung (5.66) ergeben. Die Fehlerübertragungsmatrix beschreibt den dynamischen Einfluß der einzelnen Schätzfehler der betreffenden Nichtlinearitäten auf den jeweiligen Beobachterfehler. Besitzt sie reine Diagonalgestalt, so liegt ein fehlerentkoppeltes System vor, was letztlich  $k$  Einzelsystemen mit je einer Nichtlinearität entspricht. In den überwiegenden Fällen wird jedoch kein entkoppeltes System vorliegen. Hier führen bisherige Ansätze [158] und [137] zur Identifikation des Vektors  $\underline{\mathcal{NL}}(\underline{x}_E)$  nicht zum Erfolg. Es wird im Folgenden eine Methode vorgestellt, das verkoppelte System in  $k$  entkoppelte Einzelsysteme überzuführen. Die hierfür entworfene „Fehlertransformation“ zeichnet sich durch ihren Aufbau aus Integratoren aus, was sich besonders in der Anwendung als großer Vorteil erweist.

Durch eine Linkstransformation des Fehlervektors mit der inversen Fehlermatrix  $\mathbf{H}_F(s)^{-1}$  erhalten wir den transformierten Fehlervektor  $\underline{e}_T$ .

$$\underline{e}_T = \mathbf{H}_F(s)^{-1} \underline{e}^k = \mathbf{H}_F(s)^{-1} \mathbf{H}_F(s) \left( \widehat{\mathcal{NL}}(\underline{x}_E)^k - \underline{\mathcal{NL}}(\underline{x}_E)^k \right) \quad (5.68)$$

Die Matrix  $\mathbf{E} = \mathbf{H}_F(s)^{-1} \mathbf{H}_F(s)$  besitzt selbstredend reine Diagonalgestalt. Damit ist gewährleistet, dass im Fehlervektor  $\underline{e}_T$  die Fehler entkoppelt vorliegen. Die einzelnen Komponenten der inversen Fehlerübertragungsmatrix  $\mathbf{H}_F(s)^{-1}$  werden in vielen Fällen nur durch differenzierende Elemente darstellbar sein. Dies stellt jedoch erhebliche Probleme in Bezug auf die praktische Realisierung an einer realen Strecke dar. Folglich ist es sinnvoll, ein Entkopplungsfilter zu entwerfen, welches sich nur durch Integratoren aufbauen lässt. Die Matrix  $\mathbf{H}_F(s)^{-1}$  stellt eine Filterung der  $k$  Fehlersignale dar. Durch Nachschalten eines weiteren Filternetzwerkes  $\mathbf{H}_W(s)$  in Diagonalgestalt

$$\mathbf{H}_W(s) = \begin{bmatrix} H_{11}(s) & 0 & \dots & 0 \\ 0 & H_{22}(s) & \dots & 0 \\ 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \ddots \\ 0 & \dots & 0 & H_{kk}(s) \end{bmatrix} \quad (5.69)$$

ergibt sich für das gesamte Entkopplungsfilter der Zusammenhang

$$\mathbf{H}_H(s) = \mathbf{H}_W(s)\mathbf{H}_F(s)^{-1} \quad (5.70)$$

Die  $k$  Elemente der Matrix  $\mathbf{H}_W(s)$  wählt man geschickterweise so, dass sich für die einzelnen Elemente des gesamten Entkopplungsnetzwerkes  $\mathbf{H}_H(s)$  Polynome von relativem Grad größer gleich null ergeben. Dadurch ist eine Realisierung des Entkopplungsnetzwerkes mit Integratoren gewährleistet. Wird nun das Entkopplungsfilter in Gleichung (5.67) eingesetzt, so ergibt sich diese zu

$$\underline{\epsilon}_H = \mathbf{H}_H(s)\underline{\epsilon}^k = \mathbf{H}_W(s)\mathbf{H}_F(s)^{-1}\mathbf{H}_F(s) \left( \widehat{\mathcal{N}}^k - \underline{\mathcal{N}}^k \right) = \mathbf{H}_W(s) \left( \widehat{\mathcal{N}}^k - \underline{\mathcal{N}}^k \right) \quad (5.71)$$

Die  $i$ -te Komponente des transformierten (gefilterten) Fehlervektors  $\underline{\epsilon}_H$  korreliert nun über die gewählten Übertragungsfunktionen  $H_{ii}(s)$  mit dem  $i$ -ten Approximationfehler von  $\widehat{\mathcal{N}}^k - \underline{\mathcal{N}}^k$ . Dadurch wurde das verkoppelte System in ein entkoppeltes System übergeführt, was sich durch Anwendung der aus der Literatur bekannten Fehlermodelle [158] darstellen lässt. Die prinzipiellen Adoptionsgesetze aus Abschnitt 5.2 sind demnach wieder anwendbar.

Das Entkopplungsnetzwerk  $\mathbf{H}_H(s)$  kann durch relativ freie Wahl der Matrix  $\mathbf{H}_W(s)$  entscheidend in seiner Dynamik beeinflusst werden. Es lassen sich die Frequenzgänge der einzelnen Übertragungsglieder der Matrix  $\mathbf{H}_H(s) = \mathbf{H}_W(s)\mathbf{H}_F(s)^{-1}$  in weiten Bereichen einstellen, was für eine praktische Realisierung von großem Vorteil ist. Durch die freie Wahl der einzelnen Elemente  $H_{ii}(s)$  von  $\mathbf{H}_W(s)$  lassen sich nicht nur, wie bereits beschrieben, die Frequenzgänge des Entkopplungsfilters beeinflussen, sondern auch die Lerndynamik einstellen. Da die Übertragungsglieder  $H_{ii}(s)$  im Lerngesetz in der sogenannten verzögerten Aktivierung auftauchen, bestimmen sie maßgeblich die Lerngeschwindigkeit des eingesetzten neuronalen Netzes. Aufgabe der Auslegung des Entkopplungsfilters ist es, sowohl akzeptable Frequenzgänge im Entkopplungsnetzwerk zu haben, als auch eine gute Lerngeschwindigkeit zu erreichen.

Aus den obigen Überlegungen bezüglich der Entkopplung der einzelnen Beobachterfehler erhalten wir für die einzelnen gefilterten Fehler  $k$  Fehlerdifferentialgleichungen der Form

$$e_{H,i} = H_{ii}(s) \left( \widehat{\mathcal{N}}_i^k - \underline{\mathcal{N}}_i^k \right) \quad (5.72)$$

Die  $i$ -te Komponente des Vektors  $\underline{\mathcal{N}}^k$  wird approximiert durch ein GRNN mit folgender Gleichung

$$\underline{\mathcal{N}}_i^k = \underline{\Theta}_i^T \cdot \mathcal{A}_i(\underline{x}_E) \quad (5.73)$$

Durch die bereits bekannte Bildung des erweiterten Fehlers

$$\epsilon_i = e_{H,i} - H_{ii}(s) \hat{\underline{\Theta}}_i^T \mathcal{A}_i + \hat{\underline{\Theta}}_i^T H_{ii}(s) \mathcal{A}_i \quad (5.74)$$

gelingt es, ein nach Ljapunov stabiles Lerngesetz zu formulieren [137, 158]. Mit dem Lerngesetz der verzögerten Aktivierung

$$\dot{\underline{\Theta}}_i = -\eta_i \epsilon_i H_{ii}(s) \mathcal{A}_i \quad (5.75)$$

ist mit  $\eta_i > 0$  sichergestellt, dass für die einzelnen Fehler

$$\lim_{t \rightarrow \infty} e_{H,i}(t) = 0 \quad (5.76)$$

folglich auch

$$\lim_{t \rightarrow \infty} \underline{e}(t) = 0 \quad (5.77)$$

und im Falle ausreichender Anregung gemäß [158] wiederum

$$\lim_{t \rightarrow \infty} \hat{\underline{\Theta}}_i = \underline{\Theta}_i \quad (5.78)$$

für jede einzelne geschätzte Nichtlinearität beim gewählten Beobachteransatz gilt und damit Parameterkonvergenz bewiesen ist. Durch Fehlerentkopplung ist es damit gelungen  $k$  isolierte Nichtlinearitäten mit  $k$  Meßsignalen separierbar zu identifizieren. Für diesen Identifikationsansatz ist zwingend notwendig, dass die Anzahl der Nichtlinearitäten die Anzahl der unabhängigen Meßgrößen nicht übersteigt.

## 5.6 Ergänzung: Fehlermodelle

Als Ergänzung zu den kurzen Ausführungen zu den verschiedenen Fehlermodellen in den Abschnitten 4.1.1.4 sowie 5.2 sollen an dieser Stelle die drei wichtigsten Fehlermodelle aus [158] mit den zugehörigen Lerngesetzen vorgestellt werden. Die Darstellungen orientieren sich am GRNN, obwohl sich ihre Anwendbarkeit auf alle Funktionsapproximatoren erstreckt, deren Ausgangssignal sich aus einem Produkt aus unbekanntem Gewichtsvektor  $\hat{\underline{\Theta}}$  und bekanntem Aktivierungsvektor  $\underline{\mathcal{A}}(u)$  berechnen lässt.

### 5.6.1 Fehlermodell 1

Beim *Fehlermodell 1* wird vorausgesetzt, dass der Ausgangsfehler direkt, d.h. ohne dynamische Beeinflussung, gemessen werden kann. Der Streckenausgang wird repräsentiert durch  $y = \underline{\Theta}^T \underline{\mathcal{A}}(u)$ , der Identifikatorausgang durch  $\hat{y} = \hat{\underline{\Theta}}^T \underline{\mathcal{A}}(u)$ , wobei nur der unbekannte Parametervektor  $\hat{\underline{\Theta}}$  zu adaptieren ist. Gemäß Abb. 5.17 findet keinerlei dynamische Beeinflussung zwischen Eingang  $u$  und Fehler  $e$  statt.

Mit der Definition des Parameterfehlervektors  $\underline{\Phi} = \hat{\underline{\Theta}} - \underline{\Theta}$  kann für das Fehlermodell 1 das folgende Adoptionsgesetz verwendet werden:

$$\dot{\underline{\Phi}} = \dot{\hat{\underline{\Theta}}} = -\eta e(t) \underline{\mathcal{A}}(t) \quad (5.79)$$

Dabei ist  $\eta$  die positive Lernschrittweite, mit der die Adoptionsgeschwindigkeit eingestellt werden kann. Gemäß Abb. 5.17 ergibt sich das Fehlersignal zu

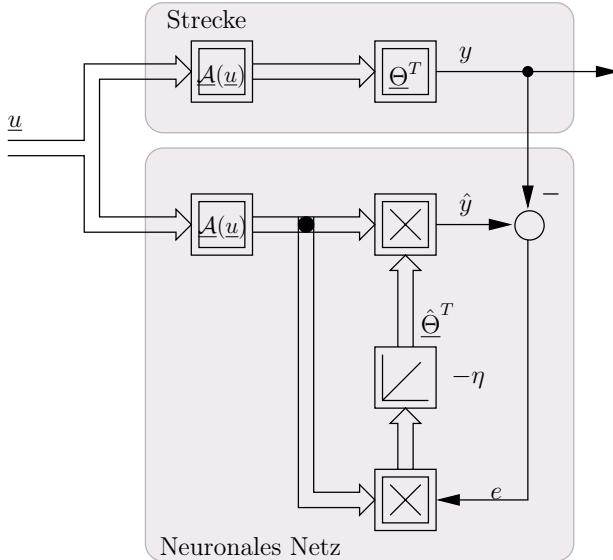


Abb. 5.17: Fehlermodell 1

$$e(t) = \hat{\underline{\Theta}}^T(t)\underline{\mathcal{A}}(t) - \underline{\Theta}^T\underline{\mathcal{A}}(t) = \underline{\Phi}^T(t)\underline{\mathcal{A}}(t) = \underline{\mathcal{A}}^T(t)\underline{\Phi}(t) \quad (5.80)$$

Das Adoptionsgesetz lautet damit

$$\dot{\underline{\Phi}}(t) = -\eta \underline{\mathcal{A}}(t)\underline{\mathcal{A}}^T(t)\underline{\Phi}(t) \quad (5.81)$$

Die Abbildung 5.17 zeigt die Struktur des Fehlermodells 1 mit dem Fehler-Regelkreis entsprechend Gleichung (5.79).

Die Stabilität des Fehlermodells soll anhand der folgenden, positiv definiten Ljapunov Funktion bewiesen werden.

$$V(\underline{\Phi}) = \frac{1}{2}\underline{\Phi}^T\underline{\Phi} \quad (5.82)$$

Die zeitliche Ableitung von  $V$  entlang der durch Gleichung (5.81) definierten Trajektorien ergibt sich damit zu

$$\dot{V}(t) = \frac{1}{2} \cdot 2 \cdot \underline{\Phi}^T(t)\dot{\underline{\Phi}}(t) = -\eta \underline{\Phi}^T(t)\underline{\mathcal{A}}(t)\underline{\mathcal{A}}^T(t)\underline{\Phi}(t) = -\eta e^2(t) \leq 0 \quad (5.83)$$

Da  $V(\underline{\Phi})$  positiv definit ist und die zeitliche Ableitung negativ semidefinit ist, ist  $V(\underline{\Phi})$  beschränkt und positiv, womit Stabilität nach Ljapunov bewiesen ist.

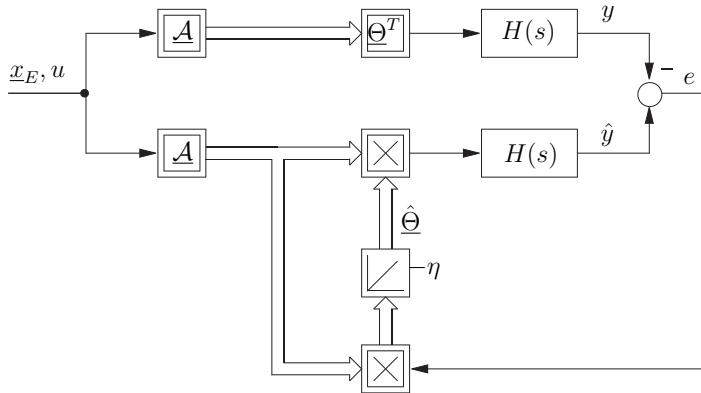
Eine zweite Problematik betrifft die Parameterkonvergenz. Um Parameterkonvergenz sicherzustellen, d.h.  $\hat{\underline{\Theta}}$  soll gegen  $\underline{\Theta}$  streben, muß der Aktivierungsvektor ausreichend anregend (*persistently exciting*) sein [158]. Wenn diese Bedingung erfüllt ist, dann streben die geschätzten Parameter gegen die wahren Werte. Damit ist, zumindest in groben Zügen, die Stabilität gemäß Ljapunov von Fehlermodell 1 bewiesen. Eine ausführliche Darstellung ist in [158] enthalten.

### 5.6.2 Fehlermodell 2

Fehlermodell 2 ist ein Spezialfall von Fehlermodell 3. Alle Zustandsgrößen der Fehlerübertragungsfunktion  $H(s)$  müssen messbar sein. Da dieser Sonderfall nur sehr selten zutrifft und Fehlermodell 2 im allgemeiner gefaßten Fehlermodell 4 eingeschlossen ist, soll auf die Darstellung von Fehlermodell 2 verzichtet werden. Genaueres kann wiederum [158] entnommen werden.

### 5.6.3 Fehlermodell 3

Fehlermodell 3 stellt eine Erweiterung von Fehlermodell 1 bezüglich des Ausgangssignals von Strecke und Approximation dar. Die Signale  $\underline{\Theta}^T \underline{\mathcal{A}}$  und  $\hat{\underline{\Theta}}^T \hat{\mathcal{A}}$  sind nicht mehr direkt für eine Fehlerbildung verfügbar. Beide Signale durchlaufen die lineare und bekannte Übertragungsfunktion  $H(s)$  gemäß Abb. 5.18.



**Abb. 5.18:** Fehlermodell 3

Weiterhin wird angenommen, dass  $H(s)$  streng positiv reell gemäß Definition 5.3 ist, d.h. dass sie keine Phasendrehung von mehr als  $\pi/2$  verursacht. Die Zustandsdarstellung von  $H(s)$  für das Streckenmodell und das Identifikationsmodell seien durch

$$\dot{x} = \mathbf{A}x + k_{\mathcal{N}} \underline{\Theta}^T \underline{\mathcal{A}}(u) \quad y = \underline{c}^T x \quad (5.84)$$

und

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + k_{\mathcal{N}} \hat{\underline{\Theta}}^T \hat{\mathcal{A}}(u) \quad \hat{y} = \underline{c}^T \hat{x} \quad (5.85)$$

gegeben.

$H(s)$  ergibt sich demnach zu  $H(s) = \underline{c}^T (s\mathbf{E} - \mathbf{A})^{-1} k_{\mathcal{N}}$ . Die Fehlereingleichung für  $e_Z = \hat{x} - x$  und die Fehlerausgangsgleichung für  $e$  lauten damit

$$\dot{\underline{e}}_Z = \mathbf{A}\underline{e}_Z + k_{\mathcal{N}} \underline{\Phi}^T \underline{\mathcal{A}}(u) \quad e = \underline{c}^T \underline{e}_Z \quad (5.86)$$

wobei  $\underline{\Phi}$  wieder der Parameterfehlervektor ist ( $\underline{\Phi} = \hat{\underline{\Theta}} - \underline{\Theta}$ ).

Als Adoptionsgesetz wird die bereits aus Abschnitt 5.6.1 bekannte Gleichung

$$\dot{\underline{\Phi}} = \dot{\underline{\Theta}} = -\eta e(t) \underline{\mathcal{A}}(t) \quad (5.87)$$

verwendet. Die Abbildung 5.18 zeigt die Struktur des Fehlermodells 3 mit dem Fehler-Regelkreis entsprechend Gleichung (5.87).

Solange  $H(s)$  streng positiv reell (SPR) ist, kann also das gleiche Adoptionsgesetz wie bei Fehlermodell 1 verwendet werden. Der Stabilitätsbeweis gestaltet sich aber etwas schwieriger:

Als positiv definite Ljapunov Funktion wird der Ausdruck

$$V(\underline{e}_Z, \underline{\Phi}) = \underline{e}_Z^T \mathbf{P} \underline{e}_Z + \underline{\Phi}^T \underline{\Phi} \quad (5.88)$$

verwendet. Die zeitliche Ableitung von  $V$  entlang der durch die Gleichungen (5.86) und (5.87) beschriebenen Trajektorien ergibt sich zu

$$\dot{V} = \dot{\underline{e}}_Z^T \mathbf{P} \underline{e}_Z + \underline{e}_Z^T \mathbf{P} \dot{\underline{e}}_Z - 2 \underline{\Phi}^T \underline{\mathcal{A}} e \quad (5.89)$$

Mit der Zwischenrechnung

$$\dot{\underline{e}}_Z = \mathbf{A}\underline{e}_Z + k_{\mathcal{N}} \underline{\mathcal{A}}^T \underline{\Phi} \quad (5.90)$$

lässt sich  $\dot{V}$  weiter berechnen zu

$$\begin{aligned} \dot{V} &= (\underline{e}_Z^T \mathbf{A}^T + \underline{\mathcal{A}}^T \underline{\Phi} k_{\mathcal{N}}) \mathbf{P} \underline{e}_Z + \underline{e}_Z^T \mathbf{P} (\mathbf{A}\underline{e}_Z + k_{\mathcal{N}} \underline{\Phi}^T \underline{\mathcal{A}}) - 2 \underline{\Phi}^T \underline{\mathcal{A}} e \\ &= \underline{e}_Z^T (\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}) \underline{e}_Z + \underline{\mathcal{A}}^T \underline{\Phi} k_{\mathcal{N}}^T \mathbf{P} \underline{e}_Z + \underline{e}_Z^T \mathbf{P} k_{\mathcal{N}} \underline{\Phi}^T \underline{\mathcal{A}} - 2 \underline{\Phi}^T \underline{\mathcal{A}} e \\ &= \underline{e}_Z^T (\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}) \underline{e}_Z + 2 \underline{e}_Z^T \mathbf{P} k_{\mathcal{N}} \underline{\Phi}^T \underline{\mathcal{A}} - 2 \underline{\Phi}^T \underline{\mathcal{A}} e \end{aligned} \quad (5.91)$$

dann gilt

$$2 \underline{e}_Z^T \mathbf{P} k_{\mathcal{N}} \underline{\Phi}^T \underline{\mathcal{A}} = 2 \underline{\Phi}^T \underline{\mathcal{A}} e \quad (5.92)$$

In [158] wird gezeigt, dass für eine streng positiv reelle Übertragungsfunktion  $H(s) = \underline{c}^T (s\mathbf{E} - \mathbf{A})^{-1} \underline{k}_{\mathcal{N}}$  positiv definite Matrizen  $\mathbf{P}$  und  $\mathbf{Q}$  existieren, für die gilt

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q} \quad (5.93)$$

$$\mathbf{P} k_{\mathcal{N}} = \underline{c} \quad (5.94)$$

Wählt man die Matrizen  $\mathbf{P}$  und  $\mathbf{Q}$  gemäß den Gleichungen (5.93) und (5.94), so folgt mit

$$\underline{e}_Z^T (\mathbf{P} k_{\mathcal{N}}) = \underline{e}_Z^T \underline{c} = e^T = e \quad (5.95)$$

schließlich

$$\dot{V} = -\underline{e}_Z^T \mathbf{Q} \underline{e}_Z \leq 0 \quad (5.96)$$

Da  $V$  eine positiv definite Funktion mit negativer Ableitung ist, muß  $V$  mit wachsender Zeit gegen Null streben. Dies ist wiederum nur dann möglich, wenn  $\hat{\Theta}$  gegen  $\underline{\Theta}$  und  $e_Z$  gegen null strebt. Damit ist die globale Stabilität von Fehlermodell 3 bewiesen. Als Einschränkung sei nochmals an die SPR-Bedingung der Fehlerübertragungsfunktion  $H(s)$  erinnert.

#### 5.6.4 Fehlermodell 4

Den allgemeinsten Fall der hier betrachteten Identifikationsstrukturen mit Adoptionsgesetzen stellt das *Fehlermodell 4* gemäß Abb. 5.19 dar.

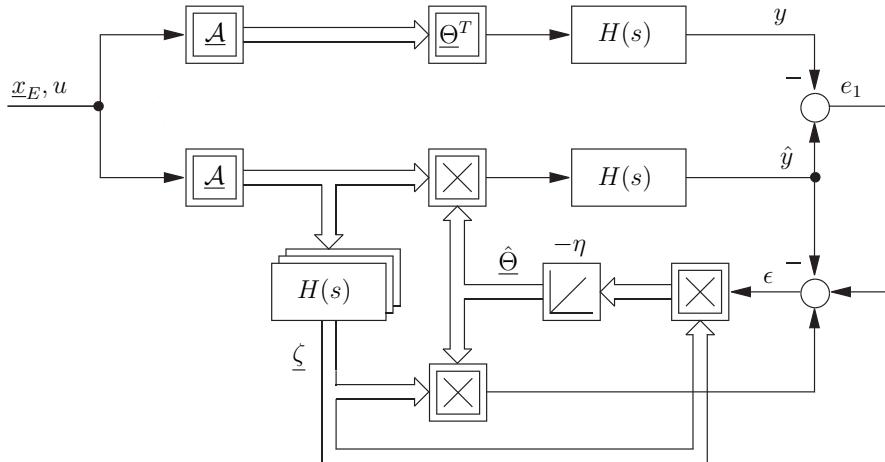


Abb. 5.19: Fehlermodell 4

Es handelt sich um eine Erweiterung von Fehlermodell 3. Die Fehlerübertragungsfunktion  $H(s)$  muß jetzt **nicht** mehr streng positiv reell sein. Um dennoch garantiert stabiles Adoptionsverhalten zu gewährleisten, sind Erweiterungen beim Adoptionsgesetz und bei der Fehlerbildung notwendig.

Der Ausgangsfehler  $e_1$  zwischen Strecke und Identifikator ergibt sich aus folgender Gleichung.

$$e_1 = H(s)\hat{\underline{\Theta}}^T \underline{\mathcal{A}} - H(s)\underline{\Theta}^T \underline{\mathcal{A}} = H(s) (\hat{\underline{\Theta}}^T - \underline{\Theta}^T) \underline{\mathcal{A}} = H(s) \underline{\Phi}^T \underline{\mathcal{A}} \quad (5.97)$$

Außerdem wird der sogenannte zusätzliche Fehler  $e_2$  gebildet, mit dem Ziel, den Stabilitätsbeweis von Fehlermodell 4 auf den Stabilitätsbeweis von Fehlermodell

1 in Abschnitt 5.6.1 zurückzuführen. Der zusätzliche Fehler  $e_2$  ist definiert durch

$$e_2 = \left( \hat{\underline{\Theta}}^T H(s) - H(s) \hat{\underline{\Theta}}^T \right) \underline{\mathcal{A}} \quad (5.98)$$

Nun soll der zusätzliche Fehler  $e_2$  durch den Parameterfehlervektor  $\underline{\Phi} = \hat{\underline{\Theta}} - \underline{\Theta}$  dargestellt werden. Setzt man in Gleichung (5.98) statt  $\hat{\underline{\Theta}}$  den Parameterfehlervektor  $\underline{\Phi}$  ein, so kann gezeigt werden, dass sich der zusätzliche Fehler  $e_2$  nicht verändert. Dazu muss beachtet werden, dass der Vektor  $\underline{\Theta}$  die Zielgröße der Identifikation ist und somit zeitinvariant und konstant ist.

$$\begin{aligned} e_2 &= [\underline{\Phi}^T H(s) - H(s) \underline{\Phi}^T] \underline{\mathcal{A}} \\ &= \left[ \left( \hat{\underline{\Theta}}^T - \underline{\Theta}^T \right) H(s) - H(s) \left( \hat{\underline{\Theta}}^T - \underline{\Theta}^T \right) \right] \underline{\mathcal{A}} = \\ &= \left[ \hat{\underline{\Theta}}^T H(s) - H(s) \hat{\underline{\Theta}}^T - \underline{\Theta}^T H(s) + H(s) \underline{\Theta}^T \right] \underline{\mathcal{A}} = \\ &= \left[ \hat{\underline{\Theta}}^T H(s) - H(s) \hat{\underline{\Theta}}^T \right] \underline{\mathcal{A}} + \underbrace{[H(s) \underline{\Theta}^T - \underline{\Theta}^T H(s)]}_{=0} \underline{\mathcal{A}} \quad (5.99) \end{aligned}$$

Der rechte Term in Gleichung (5.99) ist gleich null, wenn man berücksichtigt, dass es für den konstanten Vektor  $\underline{\Theta}$  irrelevant ist, ob er zuerst die Übertragungsfunktion  $H(s)$  durchläuft und dann das Skalarprodukt mit  $\underline{\mathcal{A}}$  gebildet wird oder umgekehrt. Die Reihenfolge der Multiplikationen in Gleichung (5.99) rechts kann damit vertauscht werden.

Da der zu jedem Zeitpunkt gültige geschätzte Parametervektor  $\hat{\underline{\Theta}}$  bekannt ist, kann der erweiterte Fehler  $\epsilon$  gemäß Abb. 5.19 gebildet werden und zur Adaption verwendet werden.

$$\epsilon = e_1 + e_2 \quad (5.100)$$

Der erweiterte Fehler  $\epsilon$  berechnet sich mit Gleichung (5.97) zu

$$\epsilon = e_1 + \underline{\Phi}^T H(s) \underline{\mathcal{A}} - H(s) \underline{\Phi}^T \underline{\mathcal{A}} = \underline{\Phi}^T \underline{\zeta} \quad (5.101)$$

mit dem Vektor der verzögerten Aktivierung  $\underline{\zeta}$ . Der erweiterte Fehler  $\epsilon$  wird gebildet, um die dynamischen Auswirkungen der Adaption der unbekannten Parameter im Fehler zu kompensieren.

$$\underline{\zeta} = H(s) \underline{\mathcal{A}} \quad (5.102)$$

Diese Fehlergleichung unter Verwendung des erweiterten Fehlers  $\epsilon$  hat die gleiche Form wie bei Fehlermodell 1 in Abschnitt 5.6.1. Das global stabile Lerngesetz lautet demnach

$$\dot{\underline{\Phi}} = \dot{\hat{\underline{\Theta}}} = -\eta \epsilon \underline{\zeta} \quad (5.103)$$

Eine weitere Stabilitätsuntersuchung mittels Ljapunov Funktionen ist nicht erforderlich, da die Gleichungen (5.101) und (5.103) exakt denen von Fehlermodell

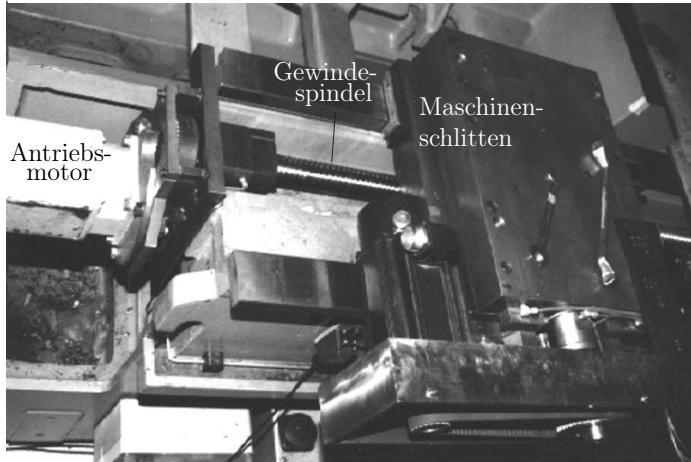
1 entsprechen, wobei der verzögerte Aktivierungsvektor  $\underline{\zeta}$  verwendet wird. Dieser geht aus  $\underline{\mathcal{A}}$  eindeutig hervor, wenn  $H(s)$  asymptotisch stabil ist.

Durch Verwendung des verzögerten Aktivierungsvektors und des erweiterten Fehlers läßt sich Fehlermodell 4 somit auf Fehlermodell 1 zurückführen. Alle in Abschnitt 5.6.1 gemachten Aussagen zu Stabilität und Konvergenz sind auch in diesem Fall gültig.

Durch die Einführung von Fehlermodell 4 ist es möglich, unbekannte Parametervektoren global stabil zu adaptieren, auch wenn der Ausgang des Identifikators eine beliebige, global stabile Übertragungsfunktion durchläuft. Diese Übertragungsfunktion muß **nicht** streng positiv reell sein.

## 5.7 Anwendung auf einen Vorschubantrieb

In diesem Abschnitt wird der lernfähige Beobachter auf einen Vorschubantrieb (siehe Abbildung 5.20) mit unbekannter Reibcharakteristik angewendet. Zunächst wird ein lineares Modell des Vorschubantriebes bestimmt, und darauf aufbauend wird der lernfähige Beobachter entwickelt, um die unbekannte Reibcharakteristik zu identifizieren. Im Anschluß daran wird die identifizierte Reibungskennlinie dazu verwendet, um eine Geschwindigkeits- bzw. Positionsregelung zu verbessern.



**Abb. 5.20:** Vorschubantrieb

### 5.7.1 Modellbildung

Abbildung 5.21 zeigt die Struktur des Vorschubantriebes schematisch mit den für eine Modellbildung interessierenden Größen und Zählpfeilen.

Prinzipiell kann ein derartiges mechanisches System durch eine Analyse mit finiten Elementen modelliert werden. Allerdings wird bei diesem Vorgehen die resultierende Ordnung des Modells sehr hoch werden. Zusätzlich sind die nichtlinearen Einflüsse zu beachten. Wenn dann anschließend dieses Modell mit sehr hoher Ordnung für einen Reglerentwurf genutzt werden soll, dann erfolgt im allgemeinen eine Ordnungsreduktion, wobei die Entscheidung, welche Ordnung das reduzierte Modell haben soll, eine Frage der Erfahrung ist. Außerdem ist zu beachten, dass eine Berücksichtigung der nichtlinearen Effekte bei der Ordnungsreduktion im allgemeinen nicht erfolgen kann, d. h. die nichtlinearen Effekte werden vernachlässigt. Weiterhin ist von außerordentlicher Bedeutung, dass das reduzierte Modell nicht mehr physikalisch interpretierbar ist. Um diese Nachteile zu vermeiden, soll der ingenieurtechnische Aspekt einer physikalischen Modellbildung als Ausgangsbasis genutzt werden.

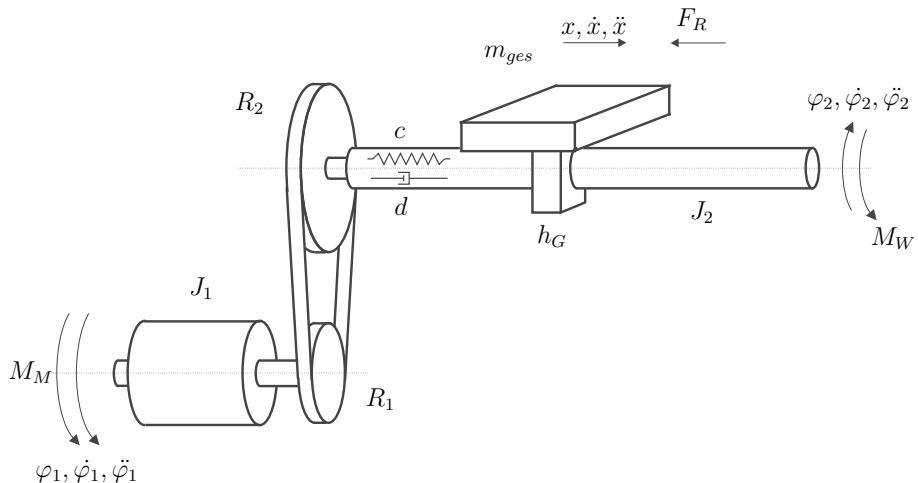


Abb. 5.21: Das Modell des Vorschubantriebes

Wendet man gemäß [21] das Freischneideprinzip von d'Alembert an so erhält man folgende Summen der Momente und Kräfte an den Teilsystemen:

$$\begin{aligned} \sum_{\text{1. Körper}} M_{\varphi_1} &= 0 : \quad J_1 \cdot \ddot{\varphi}_1 = M_M - c \cdot \frac{1}{\ddot{u}} (\frac{1}{\ddot{u}} \varphi_1 - \varphi_2) - d \cdot \frac{1}{\ddot{u}} (\frac{1}{\ddot{u}} \dot{\varphi}_1 - \dot{\varphi}_2) \\ \sum_{\text{2. Körper}} F_x &= 0 : \quad m_{ges} \cdot \ddot{x} = (c \cdot (\frac{1}{\ddot{u}} \varphi_1 - \varphi_2) + d \cdot (\frac{1}{\ddot{u}} \dot{\varphi}_1 - \dot{\varphi}_2)) / h_G - F_R \end{aligned} \quad (5.104)$$

mit  $\ddot{u} = R_2/R_1$ .

$J_1$  ist das Gesamtträgheitsmoment der Motorwelle und des Zahnriemengetriebes bezogen auf die Motorwelle.  $m_{ges}$  ist die Gesamtmasse bezogen auf die Translationsbewegung des Maschinenschlittens, die sich aus der rotierenden Masse der Spindel und der Tischmasse ergibt. Entsprechend ist  $J_2$  das Gesamtträgheitsmoment von Spindel und Maschinentisch bezogen auf die Spindel.

Die im System vorhandene Reibung wird als von der Schlittengeschwindigkeit bzw. von der Spindeldrehzahl abhängige Kennlinie betrachtet.

Die elektrischen Komponenten des Antriebsmotors müssen ebenfalls berücksichtigt werden. Ausgehend von dem Modell einer Gleichstrommaschine berechnet sich das antreibende Moment zu:

$$M_M = \underbrace{C_M \cdot \Psi}_{c_M} \cdot I_A \quad (5.105)$$

Mit dem konstanten Erregerfluß  $\Psi$  sowie der Maschinenkonstante  $C_M$ .

Der Ankerstrom  $I_A$  bildet sich gemäß der DGL:

$$\dot{I}_A = \frac{1}{L_A} (U_A - \varphi_1 \cdot \underbrace{C_E \cdot \Psi}_{c_E}) - I_A \cdot \frac{R_A}{L_A} \quad (5.106)$$

mit einer weiteren Maschinenkonstante  $C_E$ .

Für eine Regelungstechnische Interpretation werden die Gleichungen (5.104) und (5.106) in einer Zustandsdarstellung zusammengeführt. Mit

$$\underline{x} = \begin{bmatrix} \varphi_1 \\ \dot{\varphi}_1 \\ \varphi_2 \\ \dot{\varphi}_2 \\ I_A \end{bmatrix} \quad (5.107)$$

als Zustandsvektor lassen sich (5.104) und (5.106) wie folgt darstellen:

$$\begin{aligned} \dot{\underline{x}} &= \mathbf{A}\underline{x} + \underline{b}u + \underline{k}_{\mathcal{N}\mathcal{L}}\mathcal{N}\mathcal{L}(\underline{x}) \\ \underline{y} &= \mathbf{C}\underline{x} \end{aligned} \quad (5.108)$$

Dabei ergeben sich die Systemmatrizen und -vektoren  $\mathbf{A}$ ,  $\mathbf{C}$ ,  $\underline{b}$  und  $\underline{k}_{\mathcal{N}\mathcal{L}}$  zu

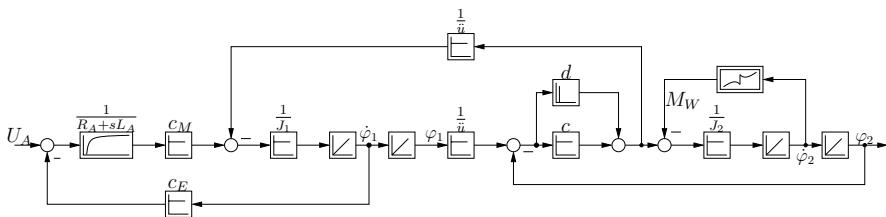
$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{c}{\ddot{u}^2 \cdot J_1} & -\frac{d}{\ddot{u}^2 \cdot J_1} & \frac{c}{\ddot{u} \cdot J_1} & \frac{d}{\ddot{u} \cdot J_1} & \frac{c_M}{J_1} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{c}{\ddot{u} \cdot J_2} & \frac{d}{\ddot{u} \cdot J_2} & -\frac{c}{J_2} & -\frac{d}{J_2} & 0 \\ 0 & -\frac{c_E}{L_A} & 0 & 0 & -\frac{R_A}{L_A} \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_A} \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & h_G & 0 & 0 \\ 0 & 0 & 0 & h_G & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \underline{k}_{\mathcal{N}\mathcal{L}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{1}{J_2} \\ 0 \end{bmatrix} \end{aligned} \quad (5.109)$$

und

$$u = U_A, \quad \mathcal{NL}(\underline{x}) = M_W = M_W(x_4) \quad (5.110)$$

Die Ausgangsmatrix  $C$  ist durch die Messsignale, die an dem Vorschubantrieb zur Verfügung stehen, gegeben. Diese sind für das mechanische Teilsystem die Motordrehzahl und die Schlittenposition. Die Schlittengeschwindigkeit wird durch einfache Differentiation des Positionssignals berechnet, da sie für den Identifikationsalgorithmus benötigt wird. Diese berechnete Größe wird hier als gemessene Größe behandelt, da es sich nicht um einen unabhängigen Zustand handelt. Der Ankerstrom des elektrischen Teilsystems kann ebenfalls gemessen werden. Sämtliche linearen Parameter sind bekannt.

Abbildung 5.22 zeigt den aus der Zustandsdarstellung abgeleiteten nichtlinearen Signalflußplan des Zweimassensystems.



**Abb. 5.22:** Signalflussplan des Vorschubantriebes

Um den Vorschubantrieb am Versuchsstand betreiben zu können, wird das oben dargestellte System mit Hilfe einer Kaskadenstruktur, bestehend aus einer PI-Stromregelung sowie einer PI-Drehzahlregelung, geregelt.

Um die Identifikationsergebnisse später bewerten zu können, wird zunächst von einem linearen Modell ausgegangen. Abbildung 5.23 zeigt den Vergleich zweier verschiedener Simulationen des linearen Zweimassenmodells mit Messdaten des Versuchsstandes. Die gemessenen Daten sind schwarz dargestellt, die simulierten Kurven sind grau gezeichnet. In der linken Spalte des Bildes 5.23 sieht man Ankerstrom, Motordrehzahl und Schlittenposition des Vorschubantriebs bei Anregung des Positionsregelkreises mit dreieckförmigem Sollwert. Während der Phasen konstanter Motordrehzahl zeigt die Messung des Ankerstroms, dass der Motor einem quasi konstanten Reibmoment entgegenwirken muss. Da die Reibung im Modell nicht berücksichtigt wurde, ist der simulierte Ankerstrom Null. Daraus resultiert auch im Verlauf der Motordrehzahl ein wesentlich höheres Überschwingen in der Simulation aufgrund der fehlenden Modellierung der dämpfenden Wirkung der Gleitreibung und daraus folgend eine deutliche Abweichung der Schlittenposition im Vergleich zur Messung.

Die rechte Spalte in Abb. 5.23 zeigt Ankerstrom, Motordrehzahl und Schlitzenposition des Vorschubantriebs bei Anregung des Positionsregelkreises mit si-

nusförmigem Sollwert. Im Verlauf des gemessenen Ankerstroms ist zu sehen, dass bei Vorzeichenumkehr der Motordrehzahl der Antrieb die Haftreibung des mechanischen Systems überwinden muss. Im weiteren Verlauf muss bei zunehmender Drehzahl ein der Drehzahl proportionales Reibmoment überwunden werden. Deutlich zu sehen in der Messung der Motordrehzahl ist das Hängenbleiben des Schlittens aufgrund der Haftreibung bei Vorzeichenumkehr der Drehzahl (Slipstick-Effekt). Die Simulation zeigt aufgrund der Vernachlässigung von Nichtlinearitäten keinen dieser Effekte.

### 5.7.2 Identifikation der Reibungskennlinie

Mit Hilfe von Abbildung 5.22 bzw. den Zustandsgleichungen (5.109) kann der lernfähige Beobachter entworfen werden.

Ausgangspunkt hierfür ist — wie in Abschnitt 5.2 beschrieben — das lineare System, in diesem Fall also die Gleichungen (5.108) und (5.109), wobei der Vektor  $\underline{\mathcal{N}}$  als zusätzlicher Eingang aufgefasst wird.

Für die Identifikation ist es sinnvoll nur ein Teilsystem, das konsequenterweise die Nichtlinearität enthält, zu verwenden. Bei dem Teilsystem aus dem Signalfußplan muß es sich um ein rückwirkungsfreies Teilsystem handeln, das möglichst geringe Ordnung aufweist. In dem Teilsystem müssen sowohl der Angriffspunkt bzw. das Einkoppelsignal in das System, als auch die Eingangsgröße der Nichtlinearität enthalten sein. In diesem Fall bietet sich das Teilsystem zwischen Motorstrom  $I_A$  und Schlittenposition  $\varphi_2$  an. Somit reduziert sich das System in (5.108) und (5.109) um einen Zustand.

Für den Beobachterentwurf wird nun folgendes lineares DGL-System verwendet.

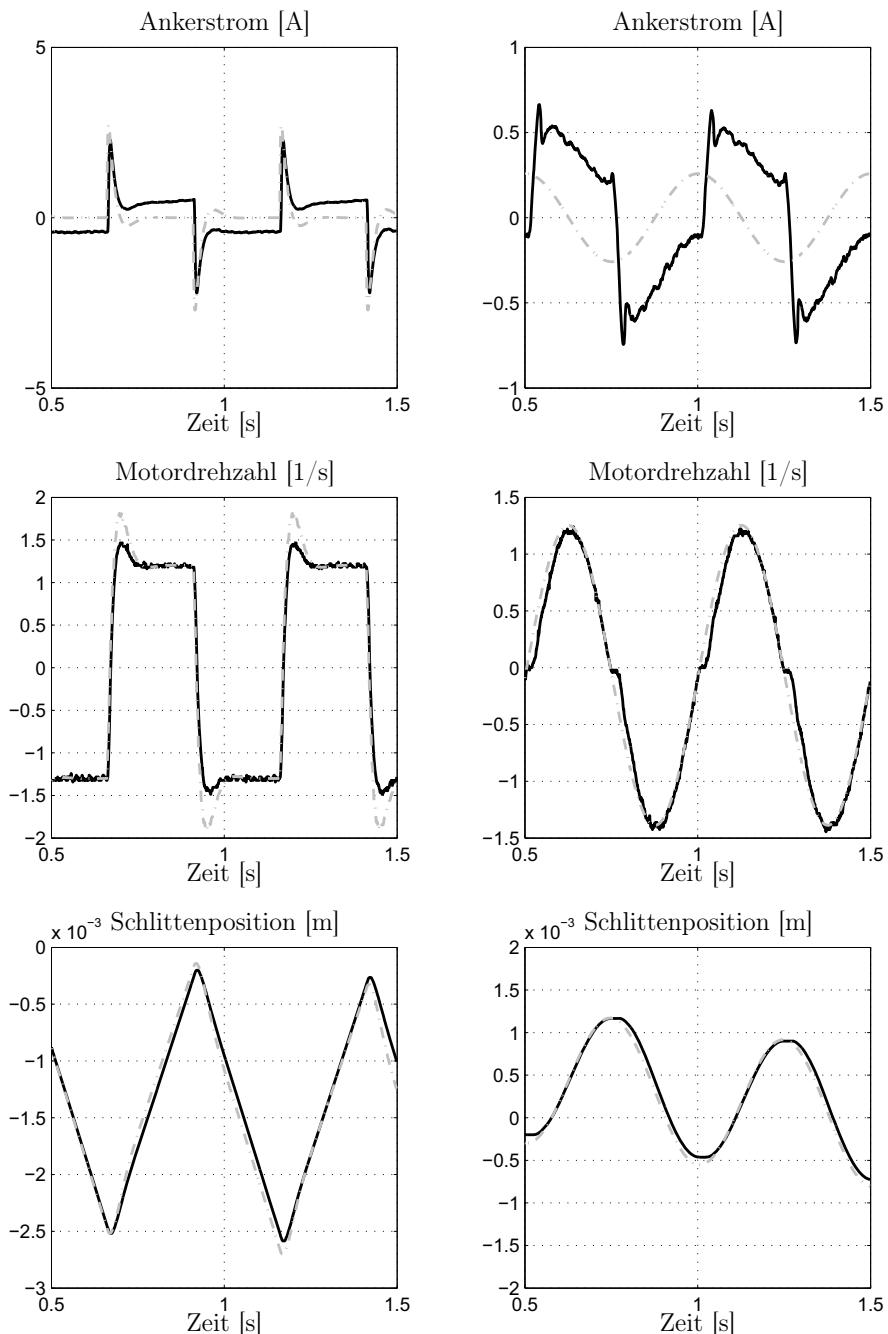
$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A}\underline{x} + \mathbf{B} \cdot \begin{bmatrix} u \\ \mathcal{N}(\underline{x}) \end{bmatrix} \\ \underline{y} &= \mathbf{C}\underline{x}\end{aligned}\quad (5.111)$$

Dabei ergeben sich die Systemmatrizen  $\mathbf{A}$ ,  $\mathbf{B}$  und  $\mathbf{C}$  zu:

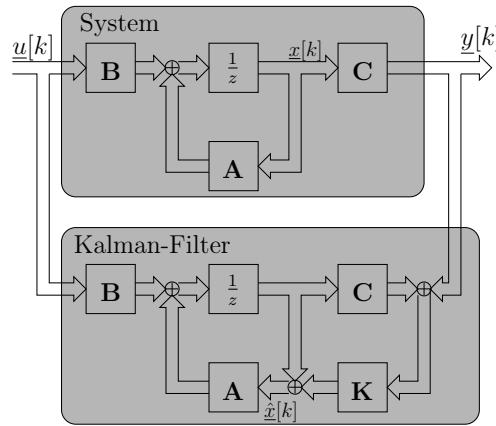
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{c}{\ddot{u}^2 \cdot J_1} & -\frac{d}{\ddot{u}^2 \cdot J_1} & \frac{c}{\dot{u} \cdot J_1} & \frac{d}{\dot{u} \cdot J_1} \\ 0 & 0 & 0 & 1 \\ \frac{c}{\ddot{u} \cdot J_2} & \frac{d}{\ddot{u} \cdot J_2} & -\frac{c}{J_2} & -\frac{d}{J_2} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{1}{J_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J_2} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & h_G & 0 \\ 0 & 0 & 0 & h_G \end{bmatrix}\quad (5.112)$$

Da die Identifikation auf einem DSP durchgeführt wird, müssen die Gleichungen (5.111) und (5.112) zeitdiskretisiert werden. Der Beobachterentwurf selbst erfolgt auf Basis eines zeitdiskreten Kalman-Filters [142] (siehe Abbildung 5.24).

Mit Hilfe von Standardtools (z. B. Matlab) kann die Rückführmatrix  $\mathbf{K}$  so bestimmt werden, dass die Fehlerübertragungsfunktion  $H(s)$  im interessierenden

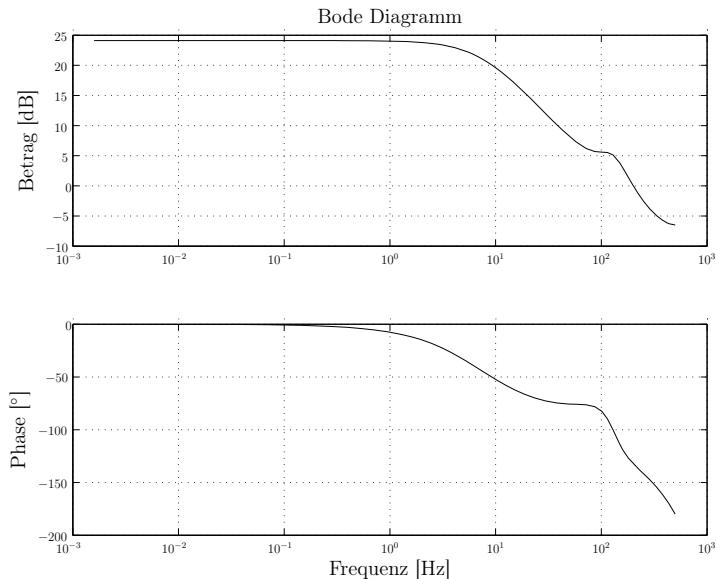


**Abb. 5.23:** Vergleich von simulierten linearem Modell und gemessenen Daten (schwarz: Gemessene Daten; grau: Simulations-Daten)



**Abb. 5.24:** System mit Kalman-Filter

Frequenzbereich die SPR-Bedingung erfüllt.<sup>2)</sup> Das Bodediagramm in Abbildung 5.25 verdeutlicht dies.



**Abb. 5.25:** Bode Diagramm von  $H(s)$

<sup>2)</sup> Diese Auslegung ist zweckmäßig, um die Rechenzeit auf dem DSP möglichst gering zu halten

Bei der Auslegung des Neuronalen Netzes ist noch zu beachten, dass das Netz aufgrund der Unstetigkeitsstelle bei  $\dot{\varphi}_2 = 0 \text{ [m/s]}$  der Reibungskennlinie zweigeteilt werden muss.

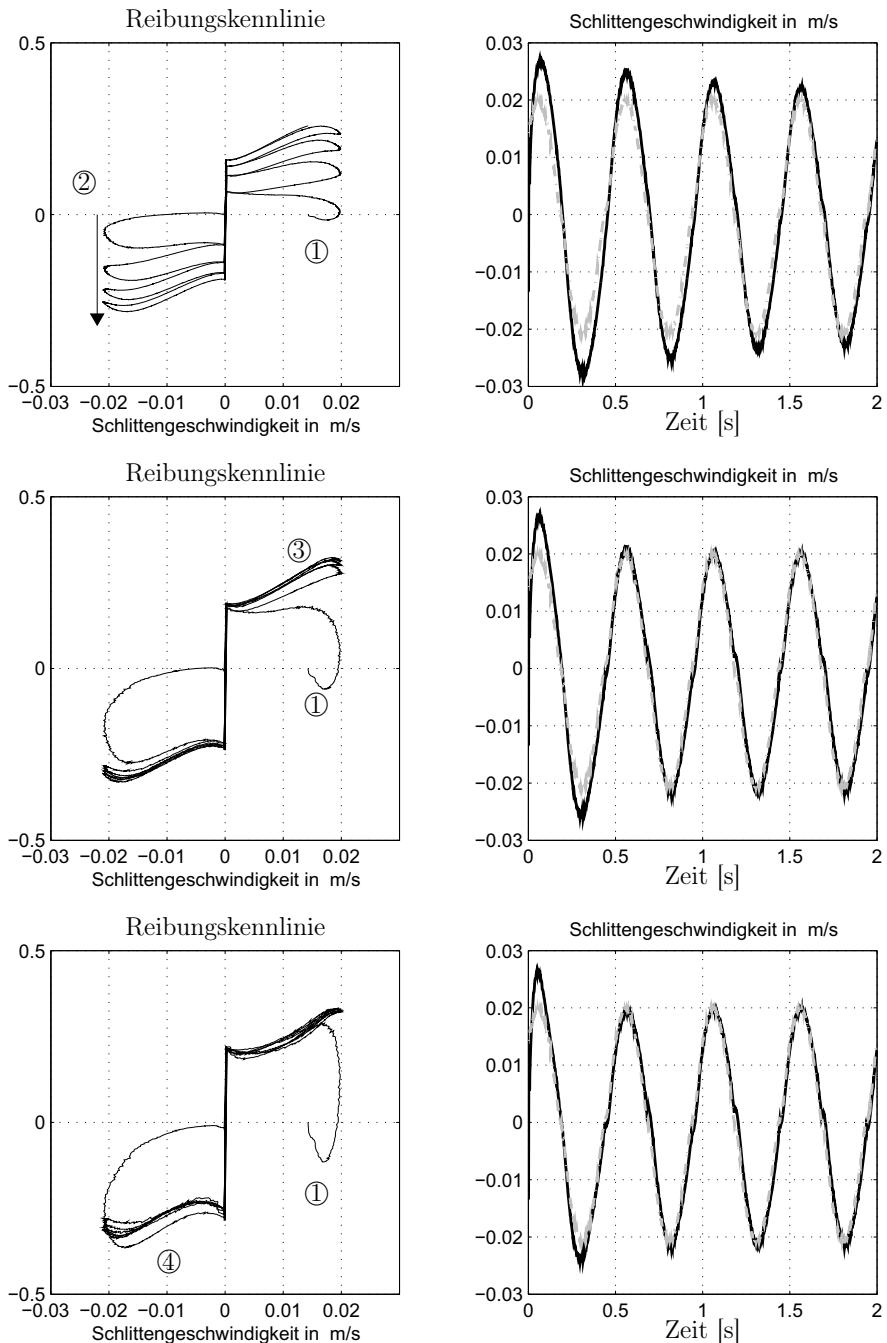
Abbildung 5.26 zeigt die Identifikation der Reibungskennlinie des Vorschubantriebs, aufgetragen über der Schlittengeschwindigkeit, für drei verschiedene Lernfaktoren. Rechts neben dem jeweiligen Identifikationsvorgang sind die beobachtete und die berechnete Schlittengeschwindigkeit über der Zeit aufgetragen, aus deren Differenz das Fehlersignal für die Adaption des Neuronalen Netzes gebildet wird.

Nach Abklingen des Einschwingvorgangs ① aufgrund unterschiedlicher Anfangswerte in der Strecke und im Beobachter lässt sich der Fortschritt ② beim Lernen des neuronalen Netzes beobachten. Der Lernfortschritt hängt im wesentlichen von dem Lernfaktor ab. Im ersten Bild ist ein Lernfaktor von 0.5 eingestellt. Nach dem Start des Lernvorgangs benötigt das Neuronale Netz bei der eingestellten Sollfrequenz drei Umdrehungen, bis eine nahezu stationäre Kennlinie ③ erreicht ist. Das entspricht einer Lerndauer von 1.5 Sekunden. Im zweiten Bild ist mit einem Lernfaktor von 1.0 eine wesentlich bessere Lerndynamik zu sehen, nach einer Umdrehung ist bereits fast das Endergebnis erreicht. Im letzten Bild wurde ein Lernfaktor von 2.0 eingestellt, damit bringt man das Neuronale Netz bereits zum Überschwingen ④, und die Adaptions-Dynamik kann nicht weiter verbessert werden.

Der Parametervektor des Neuronalen Netzes ist zu Beginn des Lernvorgangs gleich dem Nullvektor. Mit fortschreitendem Lernerfolg ist zu beobachten, wie der Unterschied zwischen beobachteter und berechneter Schlittengeschwindigkeit immer kleiner wird (Abbildung 5.26 rechts).

Geht man zurück zum linearen Modell und erweitert das lineare Zweimassenmodell um die gelernte nichtlineare Kennlinie, so lässt sich in etwa die Güte der gelernten Kennlinie zeigen. Abbildung 5.27 zeigt den Vergleich der Simulationen (vgl. Abb. 5.23) eines Modells, welches erstens als lineares Zweimassensystem modelliert wurde und zweitens eine drehzahlabhängige Reibungskennlinie enthält, mit Messdaten des Antriebsstrangs. In der linken Spalte der Abb. 5.27 sieht man Ankerstrom, Motordrehzahl und Schlittenposition des Vorschubantriebs bei Anregung des Positionsregelkreises mit dreieckförmigem Sollwert. Bei allen Größen zeigt sich eine wesentlich bessere Übereinstimmung zwischen Messung und Simulation als beim linearen Modell. Zu beachten ist allerdings die zu kurze Lerndauer und damit der Einfluss der Lerngeschwindigkeit auf den Verlauf der identifizierten Reibung. Dies zeigt sich insbesondere bei der Amplitude der Haftreibung, die etwas zu gering identifiziert wurde.

Die rechte Spalte in Abb. 5.27 zeigt Ankerstrom, Motordrehzahl und Schlittenposition des Vorschubantriebs bei Anregung des Positionsregelkreises mit sinusförmigem Sollwert. Im Verlauf des gemessenen Ankerstroms ist zu sehen, dass bei Vorzeichenumkehr der Motordrehzahl der Antrieb etwas weiter überschwingt als in der Messung. Dies ist darauf zurückzuführen, dass — wie schon oben angemerkt — aufgrund der zu kurzen Lerndauer eine etwas zu geringe Reibung



**Abb. 5.26:** Links: Identifikation der Reibungskennlinie mit verschiedenen  $\eta$ -Werten; Rechts: schwarz Beobachter; grau reales System

identifiziert wurde. Aufgrund der nichtlinearen Modellierung ist auch in der Simulation das Haften des Schlittens bei Nulldurchgang der Drehzahl zu sehen.

Aus diesen Ergebnissen ist eine nichtlineare Strecke identifiziert worden, d. h. es liegt ein Simulationsmodell vor, welches den ingenieurtechnischen Ansprüchen voll genügt und das physikalisch interpretierbar ist. Wesentlich ist dabei, dass eine Änderung der Nichtlinearität (z. B. durch eine Veränderung der Arbeitstemperatur) sofort erfasst werden kann, da der Beobachter im Online-Betrieb arbeitet. Zudem können neben der Geschwindigkeit auch andere Einflussfaktoren wie z. B. die Position des Schlittens, die Schmiermitteltemperatur und/oder das Werkstückgewicht berücksichtigt werden, was zu einer multidimensionalen Nichtlinearität führen würde.

Im nächsten Schritt soll nun dieses nichtlineare Beobachtermodell genutzt werden, um den unerwünschten nichtlinearen Einfluss zu unterdrücken.

### 5.7.3 Kompensation

Durch die Identifikation ist es möglich, eine unbekannte nichtlineare Kennlinie zu identifizieren. Nun stellt sich die Frage, in welcher Weise dieses Wissen am besten eingesetzt werden kann, um die Regelgüte des betrachteten Systems zu verbessern.

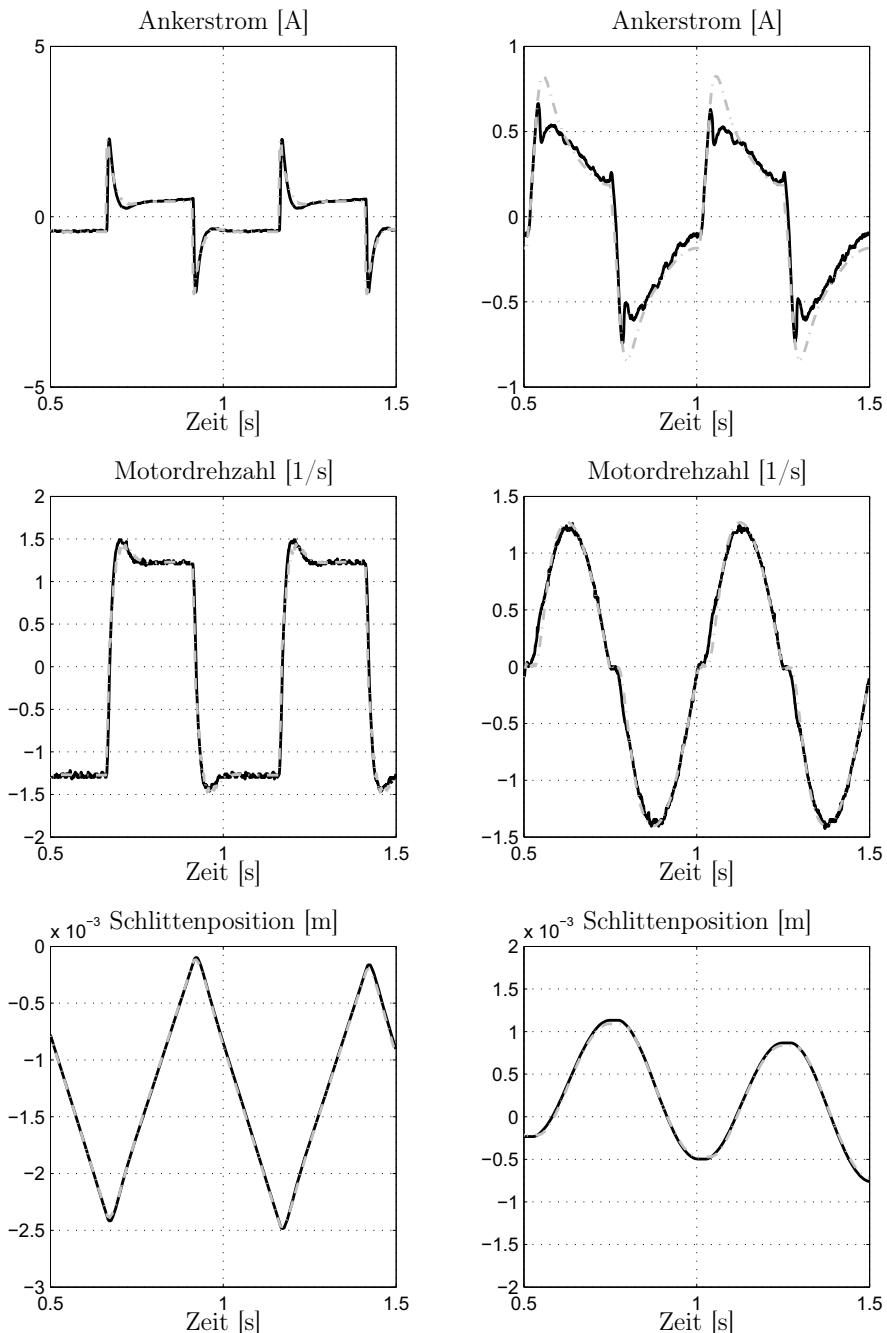
Eine Standardmethode der Regelungstechnik zur Kompensation eines nichtlinearen Einflusses ist z. B. die Störgrößenaufschaltung wie sie von Föllinger in [57] beschrieben wird. Hierbei wird versucht, durch Aufschalten von messbaren Störgrößen mit der inversen Übertragungsfunktion am Stelleingriff den Einfluss der Störgrößen zu kompensieren. Die Aufschaltung enthält also im Idealfall die invertierte Übertragungsfunktion  $G_K(s) = G_{S1}^{-1}(s)$  des Signalpfades zwischen Stellgrößeneingriff und dem Eingriffspunkt der Störgröße, vgl. Abbildung 5.28.

Das Problem der Störgrößenaufschaltung ist, dass die Übertragungsfunktion  $G_{S1}^{-1}(s)$  in den meisten Fällen nach der Invertierung mehrfach differenzierendes Verhalten aufweist, und deshalb technisch nicht sinnvoll realisierbar ist.

Eine näherungsweise Kompensation ist laut Föllinger [57] möglich, wenn die Übertragungsfunktion  $G_{S1}(s)$  durch ein  $PT_1$ -Glied approximiert werden kann. Die Übertragungsfunktion des Korrekturgliedes lautet dann

$$G_K(s) = \frac{1}{K_{S1}}(1 + T_{S1}s) \quad (5.113)$$

Diese Übertragungsfunktion stellt ein  $PD$ -Glied dar und kann nur realisiert werden, wenn eine zusätzliche Nennerzeitkonstante hinzugefügt wird. Ein Grund hierfür liegt zum einen in der Tatsache, dass durch ein  $PD$ -Glied die in jedem realen System vorhandene Störwelligkeit verstärkt wird. Durch die zusätzliche Nennerzeitkonstante kann der unerwünschten Verstärkung der Störwelligkeit zumindest teilweise entgegengewirkt werden. Der andere Grund liegt in den Beschränkungen der Stellglieder eines realen Systems, da diesen nicht unbegrenzt



**Abb. 5.27:** Vergleich von Simulation des nichtlinearen Zweimassenmodells mit gemessenen Daten (schwarz: Gemessene Daten; grau: Simulationsdaten)

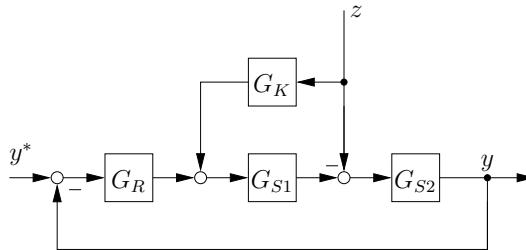


Abb. 5.28: Störgrößenaufschaltung

Energie zur Verfügung steht, d.h. es können keine unbegrenzten Signalamplituden erreicht werden. Daher ist es nicht sinnvoll, beliebig hohe Soll-Impulse zu erzeugen, da diese wegen der Stellgrößenbegrenzungen nicht auf die Strecke übertragen werden können.

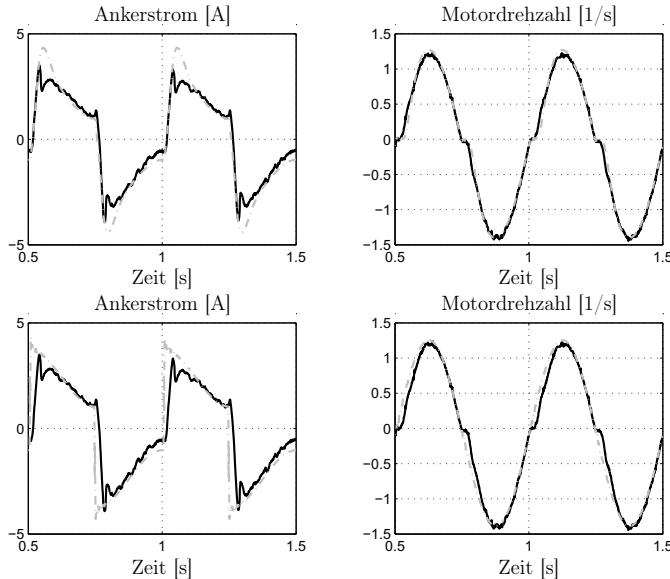
Die Übertragungsfunktion des Korrekturgliedes ergibt sich somit zu

$$G_K(s) = \frac{1}{K_{S1}} \frac{(1 + T_{S1}s)}{(1 + T_{Ns}s)} \quad (5.114)$$

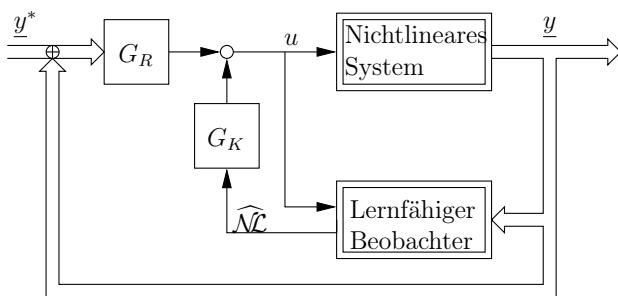
Inwiefern die Übertragungsfunktion  $G_{S1}^{-1}(s)$  durch ein  $PT_1$ -Glied approximiert werden kann, hängt von der jeweiligen Anwendung ab. Für eine mathematische Vorgehensweise können unter bestimmten Voraussetzungen die Näherungsmethoden nach Küpfmüller oder nach Samal bemüht werden. Die Approximation nach Küpfmüller ist eine Annäherung einer Übertragungsfunktion durch ein  $PT_1T_t$ -Glied. Hier muß die sich ergebende Totzeit so klein sein, dass sie vernachlässigbar ist und nurmehr ein  $PT_1$ -Glied als Approximation übrig bleibt. Die Approximation nach Samal ist eine Approximation durch ein  $PT_n$ -Glied. Hier kann unter Umständen, wenn das Verhältnis von großer zu kleiner Zeitkonstanten sehr groß ist, eine  $PT_2$ -Approximation durch eine  $PT_1$ -Approximation mit nur der großen Zeitkonstante ersetzt werden.

Zunächst wird in einer Simulation die Kompensationsaufschaltung an dem zuvor ermittelten nichtlinearen Regelkreismodell untersucht. Abbildung 5.29 zeigt in der oberen Reihe noch einmal den Vergleich von simuliertem und gemessenem Ankerstrom sowie von simulierter und berechneter Schlittengeschwindigkeit. Deutlich sind in beiden Fällen das Haften des Schlittens bei Nulldurchgang der Schlittengeschwindigkeit zu sehen. Nach Erweiterung des nichtlinearen Regelkreismodells um eine stationäre Störgrößenaufschaltung auf den Stromsollwert zeigt die Simulation (grau) in der unteren Reihe von Abbildung 5.29 eine deutliche Verbesserung des Regelkreisverhaltens gegenüber den gemessenen Daten ohne Kompensation der Reibung (schwarz).

Diese positiven Ergebnisse werden nun im Folgenden an einer Gildemeister GD 65 Drehmaschine experimentell verifiziert. Dazu wird eine online Identifikation und Kompensation an der Anlage implementiert. Die gesamte Struktur von Identifikation und Kompensation ist in Abbildung 5.30 dargestellt.



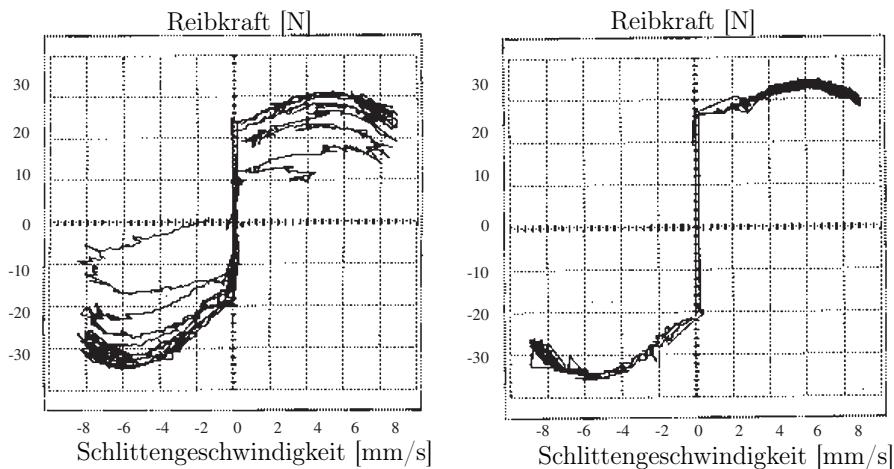
**Abb. 5.29:** Simulierte Kompensation des Haftriebungseinflusses (Obere Reihe: keine Kompensation; Untere Reihe: Kompensation)(grau: Simulationsmodell; schwarz System ohne Kompensation)



**Abb. 5.30:** Struktur der Onlineidentifikation und gleichzeitiger Kompensation

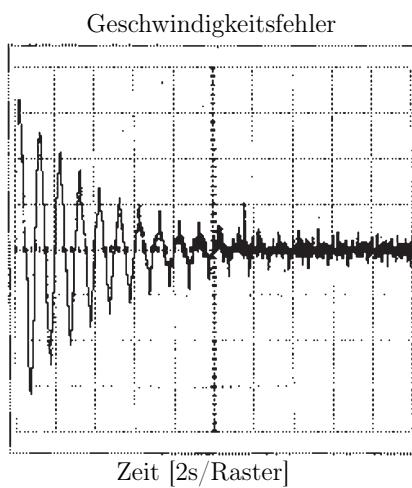
Der Lernvorgang der Reibungscharakteristik ist in Abbildung 5.31 abgebildet. Das Endergebnis dieser Identifikation ist in Abbildung 5.32 dargestellt. Aus Abbildung 5.32 ist zu erkennen, dass die Haftriebung in den beiden Bewegungsrichtungen unterschiedlich ist, eine Folge unterschiedlicher Oberflächenrauheit. Dieser Einfluß kann ortsabhängig sein und würde somit eine zusätzliche Eingangsdimension „Position“ erfordern. Der Lernfehler (Abweichung zwischen beob-

achteter und berechneter Schlittengeschwindigkeit) ist in Abbildung 5.33 veranschaulicht.



**Abb. 5.31:** Lernvorgang

**Abb. 5.32:** Gelernte Reibungscharakteristik

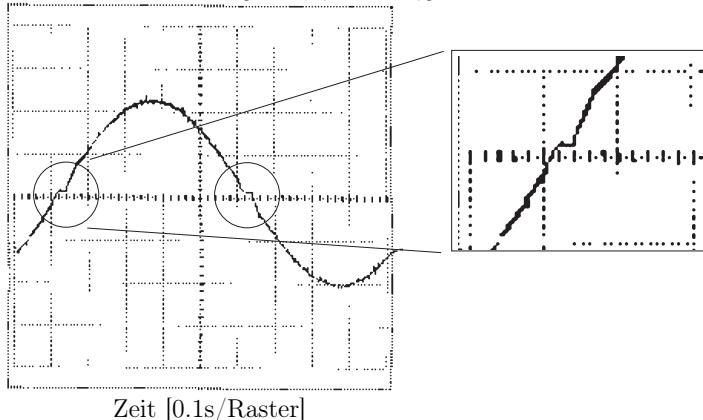


**Abb. 5.33:** Fehler zwischen berechneter und beobachteter Schlittengeschwindigkeit

Abbildung 5.34 und 5.35 zeigen die Schlittengeschwindigkeit bei Anregung des Drehzahlregelkreises mit sinusförmigem Sollwertsignal. In Abbildung 5.34 ist der

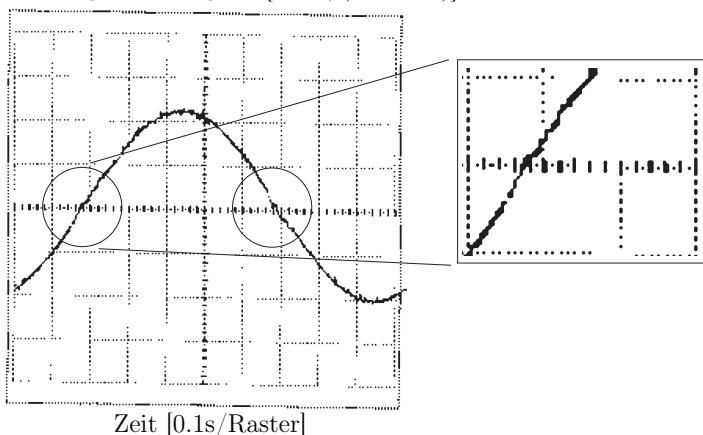
normale Betriebsfall ohne Kompensation mit dem deutlichen Effekt des Haftens bei Vorzeichenwechsel der Schlittengeschwindigkeit zu sehen. In Abbildung 5.35 ist der Einfluß der Kompensationsschaltung zu sehen, nachdem der Lernvorgang abgeschlossen ist, d. h. eine quasi stationäre Kennlinie gelernt wurde. Es ist jedoch weiterhin sinnvoll, weiterhin zu lernen, um im Fall sich ändernder Umgebungsvariablen, wie z. B. Temperatur oder Ölfilmverteilung, die neue Kennlinie sofort zu adaptieren.

Schlittengeschwindigkeit [8mm/(s Raster)]



**Abb. 5.34:** Berechnete Schlittengeschwindigkeit ohne Reibungskompensation

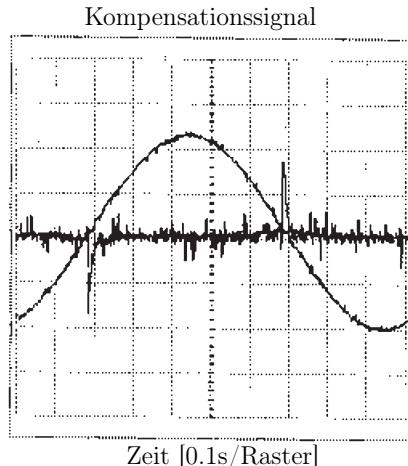
Schlittengeschwindigkeit [8mm/(s Raster)]



**Abb. 5.35:** Berechnete Schlittengeschwindigkeit mit Reibungskompensation

Deutlich ist eine wesentliche Verbesserung des Übertragungsverhaltens zu sehen, womit der Nachweis der praktischen Funktionalität der hier vorgestellten Identifikations- und Kompensationsmethode gezeigt wird.

Abbildung 5.36 zeigt schließlich invertiert das aufgeschaltete Kompensationssignal und die Schlittengeschwindigkeit. Deutlich ist der Peak zu sehen, der bei Vorzeichenwechsel der Drehzahl den Motor zusätzlich beschleunigt, um den Schlitten aus dem Haftbereich zu lösen.



**Abb. 5.36:** Kompensationssignal und Schlittengeschwindigkeit

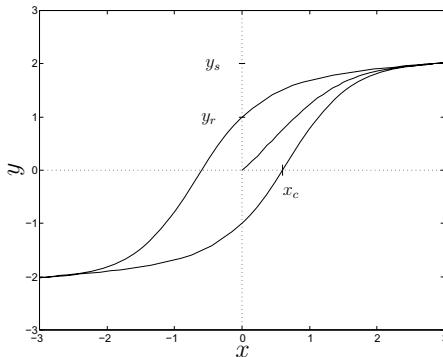
## 5.8 Identifikation von Hysterese

In diesem Abschnitt wird der lernfähige Beobachter verwendet, um eine Hysteresekennlinie zu identifizieren.

### 5.8.1 Modellierung der Hysterese

Wie in Abbildung 5.37 erkennbar, ist die Hysteresekennlinie nicht eindeutig und daher keine statische nichtlineare Funktion. Eine eindeutige Zuordnung eines  $y$ -Wertes zu einem gegebenen  $x$  ist nur mit Hilfe der Vorgeschichte möglich. Ein typisches Beispiel ist die Magnetisierung eines Eisenkerns: Ausgehend vom unmagnetisierten Zustand (Ursprung der Kennlinie) wird die Feldstärke  $H$  ( $x$ -Achse der Kennlinie) erhöht, bis der Eisenkern in Sättigung geht und sich  $B_{\text{satt}}$  ( $y_s$  auf der  $y$ -Achse der Kennlinie) einstellt. Wird die Feldstärke wieder zurückgenommen, sinkt die Flussdichte nicht wieder auf 0 sondern auf die Remanenzinduktion  $B_{\text{rem}}$  ( $y_r$ ) ab. Das heißt, der Arbeitspunkt hängt von seiner Vorgeschichte ab.

Bekannte Modelle wie in [67] verwenden innere Zustände, die der von Teilsystemen aufgenommenen und gespeicherten Energie entsprechen.



**Abb. 5.37:** Hysterese mit Neukurve

Da der aktuelle Ausgangswert  $y$  der Hysterese von der Vorgeschichte der Eingangsgröße  $x$  abhängt, liegt der Versuch nahe, Hysterese als zeitabhängige Funktion der aktuellen und vergangenen Eingangswerte zu beschreiben.

$$y(t) = f(x(t), x(t - \Delta t), x(t - 2\Delta t), \dots) \quad (5.115)$$

Zusätzlich lässt sich eine Hysterese im allgemeinen aufspalten in eine reversible und eine irreversible Komponente, wie anhand der Magnetisierung eines magnetisch weichen Eisenkerns nachvollzogen werden kann.

$$y(t) = y_{rev}(t) + y_{ir}(t) = f_{rev}(x(t)) + f_{ir}(x(t), x(t - \Delta t), x(t - 2\Delta t), \dots) \quad (5.116)$$

Während für den reversiblen Anteil die Vorgeschichte unerheblich ist, muß diese für den irreversiblen Anteil  $y_{ir}$  berücksichtigt werden. Dies ist möglich z.B. durch die Speicherung vergangener Werte von  $x$  als zusätzliche Eingangswerte für die Funktion  $f_{ir}$ . Alternativ könnte auch die Speicherung von Extremwerten innerhalb bestimmter Zeit- oder Eingangswertbereiche ausreichen.

Um jedoch die damit verbundene Problematik der Vieldimensionalität des Eingangsraumes bzw. des gezielten Rücksetzens der Extremwertspeicher zu umgehen, wird in diesem Skriptum eine grundsätzlich andere Vorgehensweise beschrieben.

### 5.8.2 Physikalisch motiviertes Modell der Hysterese

Hysteresen in physikalischen Systemen haben die Eigenschaft, dass in (für die Antriebstechnik relevanten) niedrigen Frequenzbereichen die Reihenfolge der angenommenen Werte der Eingangsgröße von Bedeutung ist, nicht jedoch, in welcher Zeit diese durchlaufen wurden.

Durch diesen physikalischen Hintergrund motiviert, wird von folgenden Annahmen ausgegangen, wobei lediglich die aufsummierte Wirkung physikalisch vorhandener Einzelprozesse betrachtet werden:

- Die irreversiblen Prozesse verhalten sich stochastisch, d.h. die Änderung des irreversiblen Anteils erfolgt mit einer Übergangswahrscheinlichkeit  $g_{irp}(x) \geq 0$  für  $dx/dt \geq 0$  bzw.  $g_{irn}(x) \geq 0$  für  $dx/dt < 0$  und proportional zur Änderung von  $x$ .
- Die irreversiblen Prozesse können erst nach einer Vorzeichenumkehr des Eingangssignals rückgängig gemacht werden, also  $g_{irp}(x) = 0$  für  $x < 0$  bzw.  $g_{irn}(x) = 0$  für  $x > 0$ . (Diese Festlegung ist nicht zwingend; genau genommen genügt die Bedingung  $\text{sign}(g_{irp}(x) - g_{irn}(x)) = \text{sign}(x)$  für ein stabiles Verhalten des Modells.)
- Es stellt sich in positiver wie negativer Richtung ein Sättigungswert  $y_{sirp}$  bzw.  $y_{sirn}$  ein.

Damit kann der irreversible Anteil der Hysterese mit zwei zeitunabhängigen Differentialgleichungen dargestellt werden.

$$\begin{aligned}\frac{dy_{ir}}{dx} &= (y_{sirp} - y_{ir}) \cdot g_{irp}(x) \quad \text{für} \quad \frac{dx}{dt} \geq 0 \\ \frac{dy_{ir}}{dx} &= (y_{ir} - y_{sirn}) \cdot g_{irn}(x) \quad \text{für} \quad \frac{dx}{dt} < 0\end{aligned}\quad (5.117)$$

Um eine geeignete Implementierung dieses Hysteresemodells zu ermöglichen, werden diese Differentialgleichungen um eine Zeitabhängigkeit wie folgt erweitert.

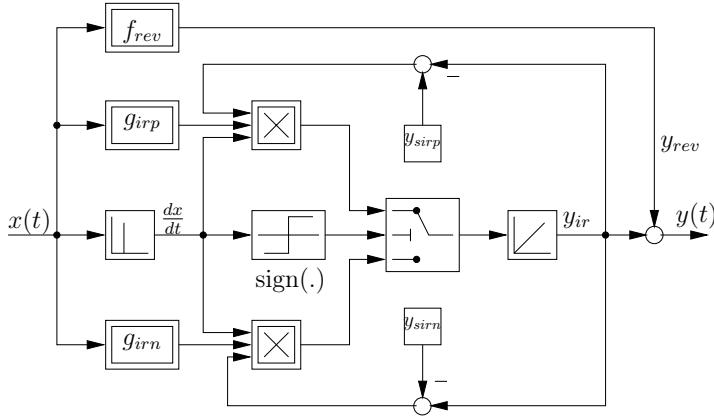
$$\begin{aligned}\frac{dy_{ir}}{dt} &= \frac{dx}{dt} \cdot (y_{sirp} - y_{ir}) \cdot g_{irp}(x) \quad \text{für} \quad \frac{dx}{dt} \geq 0 \\ \frac{dy_{ir}}{dt} &= \frac{dx}{dt} \cdot (y_{ir} - y_{sirn}) \cdot g_{irn}(x) \quad \text{für} \quad \frac{dx}{dt} < 0\end{aligned}\quad (5.118)$$

Eine implementierbare Struktur dieser Modellbeschreibung ist in Abb. 5.38 zu sehen; das Beispiel aus Abb. 5.37 wurde damit erzeugt.

### 5.8.3 Parametrierung

Da das vorgestellte Hysteresemodell nicht als Funktion sondern als Differentialgleichung vorliegt, ist für dessen Parametrierung die quantitative Kenntnis der beteiligten Prozesse von Vorteil. Wenn allerdings nur deren Wirkung bekannt ist, sollen im Folgenden einige Ansätze zur Parametrierung auf dieser Basis aufgezeigt werden. Dabei wird hier eine Hysterese vorausgesetzt, die sich punktsymmetrisch zum Ursprung verhält.

Der Sättigungswert  $y_s$  bestimmt sich für  $x \gg 0$  aus



**Abb. 5.38:** Implementierbare Struktur des Hysteresemodells

$$y_s = y_{sirp} + f_{rev}(x) \quad (5.119)$$

Aufgrund der Punktsymmetrie ist  $y_{rev}(0) = 0$  und somit die maximale Remanenz  $y_r$

$$y_r = y_{sirp} \quad (5.120)$$

Für die Bestimmung der koerzitiven Eingangsgröße  $x_c$ , die sich aus  $y(x_c) = 0$  bestimmen lässt, wird vereinfacht angenommen, dass die Übergangswahrscheinlichkeit mit der Funktion  $g_{irp} = x$  im Bereich  $x \geq 0$  gegeben ist. Mit dem Anfangswert  $y(0) = y_{sirn} = -y_{sirp}$  ergibt sich aus Gl. (5.117) die Differentialgleichung

$$\frac{dy_{ir}}{dx} = (y_{sirp} - y_{ir}) \cdot x \quad \text{für} \quad \frac{dx}{dt} \geq 0 \quad (5.121)$$

mit der Lösung

$$y_{ir} = y_{sirp} - 2y_{sirp} \exp\left(-\frac{x^2}{2}\right) \quad (5.122)$$

Da  $y(x_c) = y_{rev}(x_c) + y_{ir}(x_c) = 0$  sein soll, ergibt sich  $x_c$  als Lösung von

$$y_{sirp} - 2y_{sirp} \exp\left(-\frac{x_c^2}{2}\right) - f_{rev}(x_c) = 0 \quad (5.123)$$

Für den negativen Bereich gelten die Lösungen entsprechend; im Falle anderer Übergangswahrscheinlichkeiten  $g_{irp}, g_{irn}$  sind obige Gleichungen sinngemäß anzuwenden.

#### 5.8.4 Verallgemeinertes Hysteresemodell

Im oben beschriebenen Modell wird davon ausgegangen, dass zur Darstellung des irreversiblen Anteils der Hysterese die Kenntnis der Werte  $x(t), y(t)$  sowie der

zeitlichen Ableitung  $dx/dt$  ausreichen. Im folgenden wird nun hergeleitet, dass für den reversiblen Anteil  $y_{rev}$  dieselben Bedingungen genügen, und er damit als Sonderfall eines irreversiblen Anteils dargestellt werden kann. Aus Gl. (5.116) folgt

$$\begin{aligned} \frac{dy_{rev}}{dx} &= \frac{df_{rev}}{dx} \\ \text{und} \quad \frac{dy_{rev}}{dt} &= \frac{dx}{dt} \cdot \frac{df_{rev}}{dx} \end{aligned} \quad (5.124)$$

so dass die Hysterese in ihrer allgemeinen Form nach Gl. (5.117) erweitert werden kann zu:

$$\begin{aligned} \frac{dy}{dx} &= (y_{sirp} - y_{ir}) \cdot g_{irp}(x) + \frac{df_{rev}}{dx} \quad \text{für} \quad \frac{dx}{dt} \geq 0 \\ \frac{dy}{dx} &= (y_{ir} - y_{sirn}) \cdot g_{irn}(x) + \frac{df_{rev}}{dx} \quad \text{für} \quad \frac{dx}{dt} < 0 \end{aligned} \quad (5.125)$$

Unter der Bedingung der Stetigkeit von  $f_{rev}(x)$  ergibt sich somit die zeitbehaftete Darstellung

$$\begin{aligned} \frac{dy}{dt} &= \frac{dx}{dt} \left( (y_{sirp} - y_{ir}) \cdot g_{irp}(x) + \frac{df_{rev}}{dx} \right) \quad \text{für} \quad \frac{dx}{dt} \geq 0 \\ \frac{dy}{dt} &= \frac{dx}{dt} \left( (y_{ir} - y_{sirn}) \cdot g_{irn}(x) + \frac{df_{rev}}{dx} \right) \quad \text{für} \quad \frac{dx}{dt} < 0 \end{aligned} \quad (5.126)$$

Da  $y_{ir}$  nach Gl. (5.116) als Funktion von  $y$  und  $x$  darstellbar ist, lassen sich die Ausdrücke der rechten Seite in Gl. (5.125) zu einer Funktion  $f_p(x, y)$  bzw.  $f_n(x, y)$  zusammenfassen.

$$\begin{aligned} \frac{dy}{dx} &= f_p(x, y) \quad \text{bzw.} \quad \frac{dy}{dt} = \frac{dx}{dt} \cdot f_p(x, y) \quad \text{für} \quad \frac{dx}{dt} \geq 0 \\ \frac{dy}{dx} &= f_n(x, y) \quad \text{bzw.} \quad \frac{dy}{dt} = \frac{dx}{dt} \cdot f_n(x, y) \quad \text{für} \quad \frac{dx}{dt} < 0 \end{aligned} \quad (5.127)$$

Anschaulich bedeutet dies, dass sich von einem bestimmten Punkt  $(x(t), y(t))$  der Eingangs- und Ausgangsebene ausgehend stets das gleiche lokale Verhalten der Hysterese einstellt, wobei steigende und fallende Eingangsgröße  $x(t)$  zu unterscheiden sind, für die sich jeweils eine bestimmte differentielle Trajektorie einstellt.

Diese allgemeine Darstellung der Hysterese (siehe Abb. 5.39) enthält das physikalisch motivierte Modell als echte Teilmenge. Eine explizite Parametrierung wird zwar gegenüber dem physikalisch motivierten Modell erschwert, dafür aber eignet sich diese Struktur zur Identifikation einer Hysterese mit unbekannten Parametern und Eigenschaften.

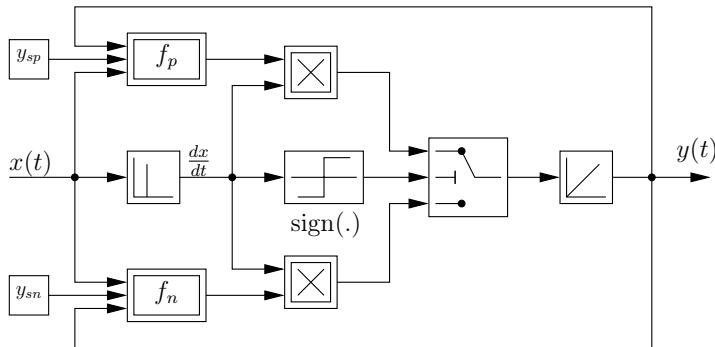


Abb. 5.39: Verallgemeinertes Hysteresemodell

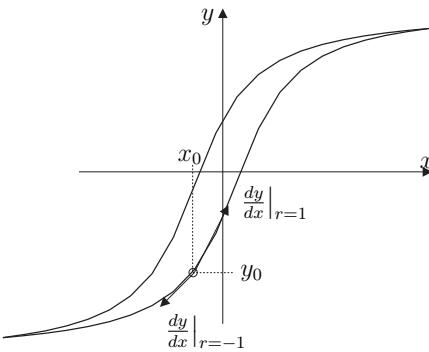


Abb. 5.40: Hysterese in „örtlicher“ Darstellung

### 5.8.5 Der allgemeingültige Signalflußplan

Ausgehend von den Überlegungen des vorherigen Abschnitts wird versucht, eine Beschreibungsform zu generieren, die es erlaubt, Hystereseffekte möglichst allgemein darzustellen.

Zu diesem Zweck betrachten wir zweidimensionale Hysteresen mit einem Eingangs- und einem Ausgangssignal. Eine mathematische Beschreibung gelingt in der folgenden Art und Weise. In Abhängigkeit der Änderungsrichtung

$$r = \text{sign} \left( \frac{dx}{dt} \right) \quad (5.128)$$

erfolgt eine differenzielle Änderung der Ausgangsgröße in funktionaler Darstellung von

$$\frac{dy}{dx} = f(x, y, r). \quad (5.129)$$

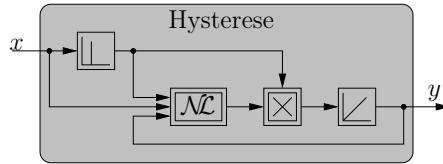


Abb. 5.41: Signalflußplan der allgemeinen Hysterese

Physikalische Hysteresevorgänge (magnetische Hysterese, Lose) lassen sich durch die obenstehende Differentialgleichung beschreiben, wobei es sich hierbei um eine „örtliche“ (d.h. die Zeit  $t$  kommt explizit nicht vor) Darstellungsweise handelt. Da jedoch i.a. das zeitliche Verhalten eines Systems von Interesse ist, muß die obige Darstellungsweise in eine zeitkontinuierliche Form übergeführt werden. Dies läßt sich durch Anwendung der aus der Mathematik bekannten Kettenregel der Differentialrechnung bewerkstelligen. Wird Gleichung (5.129) um die zeitliche Ableitung  $\frac{dx}{dt}$  zu

$$\frac{dy}{dx} \frac{dx}{dt} = f(x, y, r) \frac{dx}{dt} \quad (5.130)$$

erweitert, so ergibt sich diese mit der Kettenregel schließlich zu

$$\frac{dy}{dt} = f(x, y, r) \frac{dx}{dt}. \quad (5.131)$$

Das Verhalten der zeitlichen Ableitung des Ausgangssignals einer Hysterese-Kennlinie kann somit als statische Funktion des Eingangswertes, dessen zeitlicher Ableitung und des Ausgangswertes selbst dargestellt werden. Es ergibt sich damit eine Darstellung im Signalflußplan analog Abb. 5.41. Diese Darstellung von Hystereseverhalten wird im folgenden Abschnitt genutzt, ein geschlossenes Identifikationsverfahren zu entwerfen.

### 5.8.6 Identifikation von Hysterese

Nachdem diese allgemeine Modellbildung von Hystereseeffekten erarbeitet wurde, kann mit Hilfe der Signalflußdarstellung nach Abb. 5.41 ein lernfähiger Beobachter entworfen werden.

Mit folgendem Beobachteransatz

$$\dot{\hat{y}} = \widehat{\mathcal{N}}(y, x, \dot{x}) \cdot \dot{x} + \lambda(\hat{y} - y) \quad (5.132)$$

ergibt sich der Beobachterfehler zu

$$e = \frac{\dot{x}}{s - \lambda} \left( \widehat{\mathcal{N}}(y, x, \dot{x}) - \mathcal{N}(y, x, \dot{x}) \right) \quad (5.133)$$

Diese Fehlerdifferentialgleichung erlaubt mit dem Lerngesetz

$$\dot{\hat{\Theta}} = -\eta \cdot e \cdot \underline{\mathcal{A}} \quad (5.134)$$

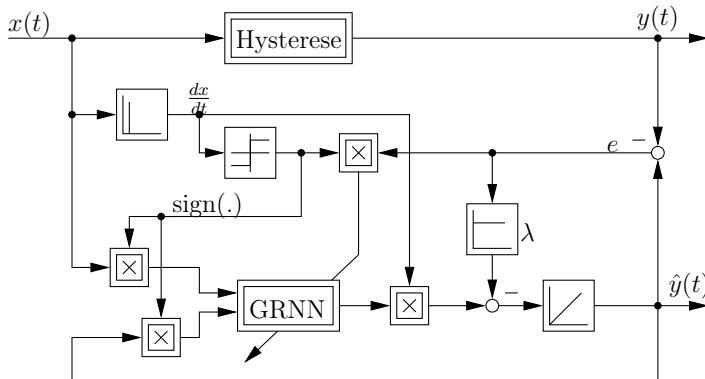
mit  $\eta > 0$  eine garantiert stabile Adaption der Stützwerte des neuronalen Netzes. Die zu approximierende Nichtlinearität  $\mathcal{N}$  ist eine statische Funktion seiner drei Abhängigkeiten  $x$ ,  $y$  und  $\dot{x}$ , wobei zusätzlich folgendes Vorwissen existiert: Die dritte Eingangsdimension  $\dot{x}$  geht nur mit Ihrem Vorzeichen  $r = \text{sign}(\dot{x})$  in die nichtlineare Funktion ein. Aus diesem Grund gelingt eine Approximation von  $\mathcal{N}$  mit einem richtungsselektiven, zweidimensionalen GRNN.

Dieser spezielle Netztyp eines GRNN zeichnet sich durch zwei getrennte Stützwertevektoren  $\hat{\Theta}_{\text{positiv}}$  für  $r = 1$  und  $\hat{\Theta}_{\text{negativ}}$  für  $r = -1$  aus. Die Umschaltung erfolgt automatisch in Abhängigkeit der Änderungsrichtung der Eingangskoordinate  $x$ .

Durch die Verwendung dieses speziellen Netztyps gelingt es, den vorhandenen dreidimensionalen Eingangsraum durch ein zweidimensionales GRNN mit zwei richtungsselektiven Stützwertevektoren abzudecken.

Der Lernaufwand kann weiter verringert werden, wenn von einer punktsymmetrischen Hysterese ausgegangen wird. In diesem Fall sind die beiden nichtlinearen Funktionen  $f_p$  und  $f_n$  identisch. Es ist jedoch zu berücksichtigen, dass  $f_n$  nur für ein negatives  $\frac{dx}{dt}$  gilt. Daher müssen die Eingangsgrößen sowie das Fehlerignal mit  $\text{sign}(\frac{dx}{dt})$  beaufschlagt werden. In diesem Fall kann die Hysterese mit Hilfe eines zweidimensionalen GRNN sowie einem Stützwertevektor identifiziert werden.

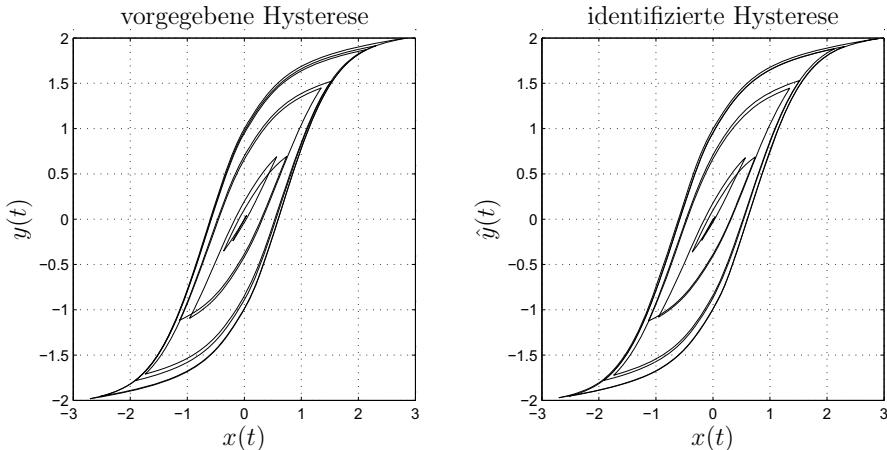
Die sich ergebende Lernstruktur ist in Abbildung 5.42 dargestellt.



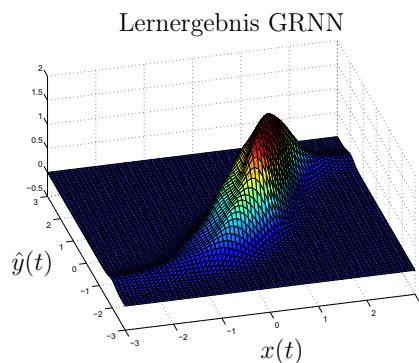
**Abb. 5.42:** Lernstruktur

Vorgegebene und gelernte Hysterese sind in Abbildung 5.43 dargestellt. Das identifizierte Steigungskennfeld des GRNN ist in Abbildung 5.44 abgebildet.

Das Identifikationsverfahren wurde an verschiedenen Hysteresemodellen getestet und liefert stets sehr zufriedenstellende Ergebnisse.



**Abb. 5.43:** Vorgegebene (links) und identifizierte (rechts) Hysterese



**Abb. 5.44:** Lernergebnis GRNN

## 5.9 Zusammenfassung und Bewertung

In diesem Kapitel wurde der lernfähige Beobachter [199] vorgestellt. Mit diesem Beobachter gelingt es unbekannte statische nichtlineare Charakteristiken in einer ansonsten bekannten Strecke zu identifizieren. Diese statischen Nichtlinearitäten werden mit Hilfe der in Kapitel 3.5 vorgestellten Funktionsapproximatoren identifiziert. Hierbei wurden zwei Fälle unterschieden, zum einen wenn der Eingangsraum der Nichtlinearität messbar ist, und zum anderen wenn der Eingangsraum nicht messbar ist. Für beide Fälle wurden stabile Lerngesetze vorgestellt. Anhand eines Simulationsbeispieles, sowie anhand eines realen Vorschubantriebes wurde das Potential dieses Verfahrens demonstriert. Ebenso wurde gezeigt, wie

mit Hilfe des lernfähigen Beobachters eine Hysterese identifiziert werden kann. Das Verfahren des Lernfähigen Beobachters [199] wurde inzwischen äußerst erfolgreich sowohl bei technischen Systemen wie dem Vorschubantrieb [88], bei mehrachsigen Robotern [132] und bei der Regelung eines  $i^2$ -CVTs [195] als auch in der Medizintechnik [6] eingesetzt.

Der Vorteil dieses Verfahrens ist, dass A-Priori-Wissen über das System mit eingebracht werden kann, die Identifikationsergebnisse physikalisch interpretierbar sind und dass nicht messbare Systemzustände beobachtet werden können. Nachteilig an diesem Verfahren ist, dass die genaue Kenntnis der linearen Parameter vorausgesetzt wird. Sind diese Parameter nicht exakt bekannt, so lassen sich mit Hilfe eines genügend schnell eingestellten Beobachters immer noch befriedigende Identifikationsergebnisse erzielen, allerdings werden hiermit die internen Systemzustände nicht mehr genau geschätzt und der Beobachter kann in Hinblick auf eine reale Anwendung das vorhandene Messrauschen nicht mehr optimal filtern. Diese erweiterte Aufgabenstellung „Identifikation der linearen Parameter der Strecke und der Nichtlinearitäten“ wird in Kapitel 6 dargestellt.

## 6 Identifikation nichtlinearer Systeme mit vorstrukturierten rekurrenten Netzen

Rekurrente Neuronale Netze eignen sich dazu, das Ein-/Ausgangsverhalten nichtlinearer dynamischer Systeme zu approximieren, ohne dass die Möglichkeit besteht, aus dem Identifikationsergebnis auf interne Zustände, lineare Parameter oder auch nichtlineare Charakteristiken schließen zu können. Um sowohl die linearen Parameter als auch die vorhandenen Nichtlinearitäten zu identifizieren, wird in diesem Kapitel ein rekurrentes Neuronales Netz vorgestellt, bei dem das Vorwissen über die Struktur des zu identifizierenden Systems mit berücksichtigt wird. Mit diesem vorstrukturierten bzw. strukturierten rekurrenten Netz wird zudem die physikalische Interpretierbarkeit des Identifikationsergebnisses erreicht.

In [25] wird zum ersten Mal ein strukturiertes rekurrentes Netz vorgestellt. Die Parameterkonvergenz ist jedoch abhängig von der Parameterinitialisierung sowie der Initialisierung der internen Systemzustände des rekurrenten Netzes. Dies führt häufig dazu, dass die internen Systemzustände den zulässigen bzw. physikalisch sinnvollen Wertebereich verlassen. Die Initialisierung erfolgt in [25], indem die Anfangszustände des rekurrenten Netzes als trainierbare Parameter des Netzes aufgefasst und getrennt von den eigentlich interessierenden Parametern gelernt werden. Durch diese Optimierung der Anfangszustände können jedoch unter Umständen Zustände entstehen, die im dynamischen System nicht auftreten.

Ein weiteres Problem tritt auf, wenn sich Nichtlinearitäten in den Rückkopplungen befinden. Die hierdurch auftretende alternierende Zustandsdivergenz wird unter der Verwendung der von T. BRYCHCY eingeführten Grenzwertheuristik vermieden. Hierfür muss jedoch das Netz so umkonfiguriert werden, dass der direkte Zusammenhang zwischen der Struktur des zu identifizierenden Systems und der Struktur des Netzes verloren geht. Die rechentechnische Lösung der zuvor beschriebenen Problematik ist aufgrund der iterativen Struktur nicht echtzeitfähig.

Im folgenden Kapitel wird ebenfalls von der Idee ausgegangen, den bekannten Signalflussplan als ein strukturiertes rekurrentes Netz aufzufassen. Dieses rekurrente Netz wird in einer Beobachterstruktur implementiert, wodurch das Problem der Anfangswertfindung der Systemzustände vermieden wird. Durch eine geeignete Wahl der Beobachterkoeffizienten kann außerdem eine Filterung der Messdaten erfolgen. Aufgrund der Erweiterung des rekurrenten Netzes zu einer Beobachterstruktur wird die Identifikation global integrierender Strecken

möglich, und somit das Problem der alternierenden Zustandsdivergenz vermieden. Auf eine spezielle Implementierung einer Grenzwertheuristik kann verzichtet werden.

## 6.1 Strukturierte rekurrente Netze

Ausgehend von dem Signalfussplan eines nichtlinearen Systems, welcher aus den elementaren Operatoren (Verstärker, Addierer, Integrierer, Differenzierer und Multiplikator) und den unbekannten Nichtlinearitäten besteht, wird das strukturierte rekurrente Netz aufgebaut. Hierbei werden die Summationspunkte des Signalfussplanes zu Neuronen und die linearen Parameter zu den Gewichten zwischen den Neuronen des rekurrenten Netzes. Die Integratoren werden mit Hilfe von Zeitverzögerungsgliedern gemäß der Integrationsregel

$$y_{int}[k+1] = h \cdot u_{int}[k] + y_{int}[k]$$

mit dem Integratoreingang  $u_{int}$ , dem Integratorausgang  $y_{int}$  und der Abtastzeit  $h$  nach L. EULER implementiert.<sup>1)</sup>

Während außer der anschaulichen Euler-Vorwärts-Regel auch andere Integrationsregeln verwendet werden können, ist für die numerische Differentiation nur die Form

$$y_{dif}[k] = \frac{1}{h} \cdot (u_{dif}[k] - u_{dif}[k-1])$$

möglich. Dabei sind  $u_{dif}$  der Eingang und  $y_{dif}$  der Ausgang des Differentiationsgliedes. Da die Differenz  $u_{dif}[k] - u_{dif}[k-1]$  bei kleinen Abtastzeiten  $h$  sehr klein wird, und diese Differenz mit dem Faktor  $1/h$  gewichtet wird, ist die numerische Differentiation im Vergleich zur numerischen Integration sehr empfindlich gegenüber Rechenun genauigkeiten. Zusätzlich muss beachtet werden, dass bei realen Systemen durch Messwerterfassung und Quantisierung verrauschte Signale vorliegen und die numerische Differentiation sehr empfindlich auf dieses Rauschen reagiert.

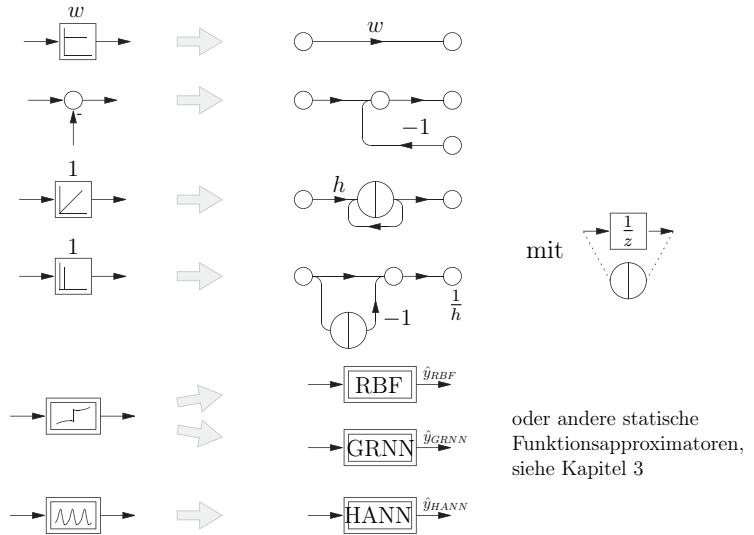
Statische Nichtlinearitäten werden mittels der in Abschnitt 3 beschriebenen statischen Neuronalen Netze als Subnetze im strukturierten rekurrenten Netz berücksichtigt.

Eine Zusammenfassung der elementaren Operatoren des Signalfussplans und ihre äquivalente Darstellung in einem rekurrenten Netz sind in Abbildung 6.1 aufgezeigt.

Durch die Transformation entsteht ein strukturiertes rekurrentes Neuronales Netz, welches eine Zuordnung der physikalischen Parameter des zu identifizierenden Systems und somit des Signalfussplanes zu den Gewichten des Netzes ermöglicht.

---

<sup>1)</sup> Diese Intergration wird auch oft als Euler-Vorwärts-Approximation oder auch als Rechteckapproximation bezeichnet [196].



**Abb. 6.1:** Elementare Operatoren eines Signalflussplanes und die äquivalenten Operatoren des strukturierten rekurrenten Netzes mit der Abtastzeit  $h$

### 6.1.1 Anwendung der Transformation

Zur besseren Veranschaulichung werden die beschriebenen Transformationen auf ein einfaches System, der Mechanik einer leerlaufenden elektrischen Maschine, angewandt.

Die Bewegungsdifferentialgleichung für dieses System lautet

$$\dot{\Omega} = \frac{1}{J} \cdot (M_L - M_W(\Omega))$$

mit

- der Winkelgeschwindigkeit  $\Omega$  in  $\frac{rad}{s}$ ,
- dem Massenträgheitsmoment  $J$  in  $kg\ m^2$ ,
- dem Luftspaltdrehmoment (antreibendes Drehmoment)  $M_L$  in  $N\ m$  und
- dem Reibungsdrehmoment<sup>2)</sup> (Widerstandsdrehmoment)  $M_W(\Omega)$  in  $N\ m$  abhängig von der Winkelgeschwindigkeit.

Abbildung 6.2 zeigt den Signalflussplan und die Transformation dieses Systems in ein rekurrentes Netz mit dem Parameter  $\hat{\Psi}_1 = 1/\hat{J}$ , der dem Kehrwert des Massenträgheitsmomentes entspricht und mit einem GRNN zur Approximation der Reibungskennlinie  $M_W(\Omega)$ , das die Parameter  $\hat{\Theta}_1$  bis  $\hat{\Theta}_r$  enthält.

<sup>2)</sup>  $M_W(\Omega)$  stellt eine statische Nichtlinearität, die Reibungskennlinie, dar.



**Abb. 6.2:** Transformation der Mechanik einer elektrischen Maschine in ein rekurrentes Netz

Für das rekurrente Netz ergibt sich entsprechend der Transformation aus Abbildung 6.2 die Differenzengleichung

$$\hat{\Omega}[k+1] = \hat{\Omega}[k] + h \cdot \hat{\Psi}_1 \cdot \left( M_L[k] - \hat{y}_{GRNN}(\hat{\Omega}[k], \hat{\Theta}) \right)$$

Neben der geschätzten Winkelgeschwindigkeit  $\hat{\Omega}[k]$  und dem geschätzten Reibungsdrehmoment  $\hat{y}_{GRNN}$  sind in dieser Gleichung auch der lineare Parameter  $\hat{\Psi}_1$  und die Stützwerte des GRNN  $\hat{\Theta} = [\hat{\Theta}_1 \dots \hat{\Theta}_r]^T$  enthalten, die zum Parametervektor

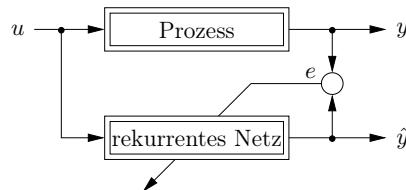
$$\hat{w} = [\hat{w}_1 \dots \hat{w}_p]^T = [\hat{\Psi}_1 \hat{\Theta}_1 \dots \hat{\Theta}_r]^T$$

des rekurrenten Netzes zusammengefasst werden. Im Parametervektor sind entsprechend alle zu identifizierenden Gewichte enthalten.

Im Allgemeinen setzt sich der Parametervektor aus  $p$  Elementen zusammen und beinhaltet die Gewichte  $\hat{\Psi}$ , die die linearen Systemanteile erfassen und die Gewichte  $\hat{\Theta}$ , welche die nichtlinearen Systemanteile über Funktionsapproximatoren berücksichtigen. Die Gewichte  $\hat{\Psi}$  werden daher im Folgenden als lineare und die Gewichte  $\hat{\Theta}$  als nichtlineare Parameter bezeichnet.

### 6.1.2 Parameteradaption

Für die Parameteradaption wird die in Kapitel 4 vorgestellte Ausgangsfehleranordnung verwendet. Dies ist in Abbildung 6.3 noch einmal dargestellt.



**Abb. 6.3:** Prinzip der Ausgangsfehleranordnung

Das Lerngesetz für das strukturierte rekurrente Netz wird analog zur Herleitung des Gradientenverfahrens für statische Neuronale Netze berechnet.

Ausgangspunkt ist der quadratische Fehler

$$E(\hat{w}) = \frac{1}{2} \cdot e^2(\hat{w}) = \frac{1}{2} (y - \hat{y}(\hat{w}))^2 \quad (6.1)$$

Analog zu Gleichung (4.6) ergibt sich folgende Gleichung für die Parameteradaption

$$\hat{w}[k+1] = \hat{w}[k] - \eta \cdot \left[ \begin{array}{c} \frac{\partial E[\hat{w}, k]}{\partial \hat{w}_1} \cdots \frac{\partial E[\hat{w}, k]}{\partial \hat{w}_p} \end{array} \right] \Big|_{\hat{w}[k]}$$

$\eta$  ist hierbei die an das Problem angepasste Lernschrittweite, wobei die Bedingung  $\eta > 0$  erfüllt sein muss.<sup>3)</sup>

Zur übersichtlicheren Schreibweise wird der Nabla-Operator eingeführt

$$\nabla = \left[ \begin{array}{c} \frac{\partial}{\partial \hat{w}_1} \cdots \frac{\partial}{\partial \hat{w}_p} \end{array} \right]^T$$

womit die Gleichung zur Parameteradaption in der Form

$$\hat{w}[k+1] = \hat{w}[k] - \eta \cdot \nabla E[\hat{w}, k] \Big|_{\hat{w}[k]}$$

angegeben werden kann. Kern dieser Gleichung ist der Gradient  $\nabla E[\hat{w}, k]$ , der entsprechend der Definition von  $E$  nach Gleichung (6.1) weiter zerlegt werden kann

$$\nabla E[\hat{w}, k] \Big|_{\hat{w}[k]} = \frac{1}{2} \cdot 2 \cdot \left( \underbrace{y[k] - \hat{y}[\hat{w}, k]}_{e[\hat{w}, k]} \right) \cdot \left( -\nabla \hat{y}[\hat{w}, k] \Big|_{\hat{w}[k]} \right) = -e[\hat{w}, k] \cdot \nabla \hat{y}[\hat{w}, k] \Big|_{\hat{w}[k]}$$

Wird dieser Gradient in die Gleichung zur Parameteradaption eingesetzt, ergibt sich das Lerngesetz nach dem Gradientenabstiegsverfahren zu

$$\hat{w}[k+1] = \hat{w}[k] + \eta \cdot e[k] \cdot \nabla \hat{y}[\hat{w}, k] \Big|_{\hat{w}[k]} \quad (6.2)$$

Dieses Lerngesetz wird dahingehend erweitert, dass die Lernschrittweite  $\eta$  nicht ein skalarer Wert ist, sondern für jedes Gewicht  $\hat{w}_i$  eine eigene Lernschrittweite  $\eta_i$  vorhanden ist. Damit erweitert sich  $\eta$  zu einem Vektor. Zusätzlich wird für jeden Parameter ein Momentumterm  $0 \leq \alpha_i < 1$  eingeführt, der in die Berechnung der aktuellen Gewichtsänderung auch die vergangenen Gewichtsänderungen einbezieht. Der Momentumterm hat den Vorteil, dass die Gewichtsanpassung unempfindlicher gegenüber Plateaus in der Fehlerebene  $E$  und Rauschanteilen im Gradienten  $\nabla E$  wird.

Diese Ergänzungen führen auf die bei strukturierten rekurrenten Netzen verwendete Form des Lerngesetzes:<sup>4)</sup>

<sup>3)</sup> Mit  $\eta < 0$  führt das Gradientenverfahren auf ein lokales Maximum in der Funktion  $E$ , womit eine konvergente Identifikation nicht möglich ist.

<sup>4)</sup>  $\text{diag}(\eta)$  ist eine Diagonalmatrix mit den Elementen des Vektors  $\eta$  in der Hauptdiagonalen.

$$\hat{w}[k+1] = \hat{w}[k] + \Delta\hat{w}[k] \quad (6.3)$$

mit

$$\Delta\hat{w}[k] = e[k] \cdot \text{diag}(\eta) \cdot \nabla \hat{y}[\hat{w}, k]|_{\hat{w}[k]} + \text{diag}(\alpha) \cdot \Delta\hat{w}[k-1]$$

Mit dem Lerngesetz kann der neue Parametervektor  $\hat{w}$  für den Abtastschritt  $k+1$  aus dem Parametervektor  $\hat{w}$ , dem Ausgangsfehler  $e[k]$  und den partiellen Ableitungen  $\nabla \hat{y}$  zum aktuellen Abtastschritt  $k$  und der vergangenen Gewichtsänderung  $\Delta\hat{w}[k-1]$  berechnet werden.

Da es sich bei dem Gradientenverfahren um eine iterative Methode handelt, sind für die erste Gewichtsanpassung die Startwerte  $\hat{w}[0]$  notwendig. Die Startwerte stellen Vorwissen dar, das bei der Identifikation zusätzlich zur Struktur des rekurrenten Netzes eingebracht werden muss. Wegen der Initialisierung des Gradientenverfahrens mit konstanten Startwerten  $\hat{w}[0]$  ergibt sich für die Gewichtsänderung  $\Delta\hat{w}[0] = 0$ . Nachteilig bei diesem Verfahren ist die große Abhängigkeit der Konvergenzgeschwindigkeit der Gewichte von den Lernparametern  $\eta$  und  $\alpha$ . Für die Wahl der Lernparameter, die manuell bestimmt werden müssen, gibt es keine mathematische Vorschrift. Grundsätzlich bewegen sich die Lernschrittweiten in derselben Größenordnung wie die zugehörigen Gewichte.

Zur Implementierung des Lerngesetzes aus Gleichung (6.3) sind die partiellen Ableitungen  $\nabla \hat{y}[\hat{w}, k]$  zum Abtastschritt  $k$  notwendig. Hierfür wird in Abschnitt 6.1.3 die Zustandsdarstellung für die strukturierten rekurrenten Netze eingeführt. Anhand dieser werden in Abschnitt 6.2.1 die partiellen Ableitungen  $\nabla \hat{y}[\hat{w}, k]$  allgemein berechnet. Die partiellen Ableitungen des RBF-Netzes, des GRNN und des HANN werden in Abschnitt 6.2.2 explizit bestimmt.

Zur Veranschaulichung werden diese Berechnungen im Anschluss am bereits eingeführten Beispiel der Mechanik einer elektrischen Maschine angewendet.

### 6.1.3 Zustandsdarstellung

Im Folgenden wird eine allgemeine Vorschrift zur Berechnung der Ableitungen  $\nabla \hat{y}$  entwickelt. Diese Vorschrift beruht auf der Zustandsbeschreibung eines nichtlinearen SISO-Systems. Diese kann für jeden Signalflussplan, der aus den in Abbildung 6.1 enthaltenen Elementen besteht, angegeben werden. Das Differentiationsglied stellt dabei einen Sonderfall dar, der in Kapitel 6.4 näher untersucht wird.

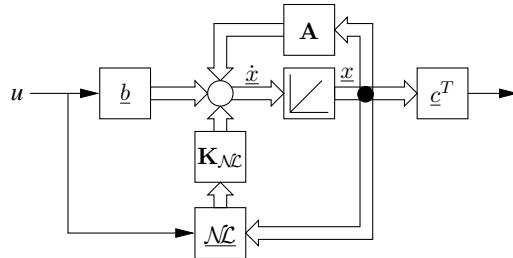
Signalflussplan und Zustandsbeschreibung stellen inherante Beschreibungen dar. Damit ist das Erstellen der Zustandsgleichung kein spezifisches Problem [175].

In der zeitkontinuierlichen Darstellung ist die Zustandsbeschreibung ein System aus nichtlinearen Differentialgleichungen erster Ordnung, das in der Form

$$\dot{x} = \mathbf{A} \cdot x + b \cdot u + \mathbf{K}_{\mathcal{N}} \cdot \underline{\mathcal{L}}(u, x) \quad \text{und} \quad y = c^T \cdot x \quad (6.4)$$

dargestellt werden kann. Der Anteil  $\mathbf{K}_{\mathcal{N}} \cdot \underline{\mathcal{L}}(u, x)$  repräsentiert die isoliert eingreifenden Nichtlinearitäten.  $\mathbf{A}$ ,  $b$  und  $c$  bilden den linearen Systemanteil. In

dieser Form wird durch die Zustandsbeschreibung ein nicht sprungfähiges System beschrieben (Durchgriff  $d = 0$ ). Dies ist bei realen Anwendungen praktisch immer gegeben [196]. Die Zustandsbeschreibung ist in Abbildung 6.4 graphisch dargestellt.



**Abb. 6.4:** Nichtlineare kontinuierliche Zustandsbeschreibung

Dabei sind

- $u$  der skalare Systemeingang,
- $\underline{x} \in \mathbb{R}^n$  der Zustandsvektor mit  $n$  Zuständen,
- $\mathbf{A} \in \mathbb{R}^{n \times n}$  die Systemmatrix,
- $\underline{b} \in \mathbb{R}^n$  der Einkopplungsvektor,
- $\mathbf{K}_{NL} \in \mathbb{R}^{n \times q}$  die Kopplungsmatrix der Nichtlinearitäten,
- $\underline{\mathcal{N}}(u, \underline{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^q$  der Vektor der statischen Nichtlinearitäten mit  $q$  skalaren Funktionen,
- $c \in \mathbb{R}^n$  der Auskopplungsvektor und
- $y$  der skalare Systemausgang.

In derselben Weise wie die kontinuierlichen Zustandsgleichungen aus dem Signalflussplan abgeleitet werden, können die diskreten Zustandsgleichungen aus dem rekurrenten Netz bestimmt werden. Dabei stellt jedes Verzögerungsneuron eine Differenzengleichung der Form  $\hat{x}_i[k+1] = \hat{f}(\hat{x}[k], u[k])$  dar. Durch Zusammenfassen dieser Gleichungen in Matrixschreibweise ergibt sich die diskrete Zustandsbeschreibung des rekurrenten Netzes.

Mit der eingeführten Euler-Vorwärts-Approximation kann aus der kontinuierlichen Systembeschreibung die diskrete Zustandsbeschreibung auch analytisch abgeleitet werden. Dazu wird von einer kontinuierlichen Abbildung der Strecke

$$\dot{\tilde{x}} = \tilde{\mathbf{A}} \cdot \tilde{x} + \tilde{\underline{b}} \cdot u + \tilde{\mathbf{K}}_{NL} \cdot \tilde{\underline{\mathcal{N}}}(u, \tilde{x}) \quad \text{und} \quad \tilde{y} = \tilde{c}^T \cdot \tilde{x}$$

ausgegangen. Die **Tilde** kennzeichnet, dass es sich um **geschätzte kontinuierliche** Größen handelt. In  $\tilde{\mathbf{A}}$ ,  $\tilde{b}$ ,  $\tilde{c}$ ,  $\tilde{\mathbf{K}}_{\mathcal{N}}$  und  $\tilde{\mathcal{L}}$  sind dieselben Elemente besetzt wie in  $\mathbf{A}$ ,  $b$ ,  $c$ ,  $\mathbf{K}_{\mathcal{N}}$  und  $\mathcal{L}$ . Die kontinuierliche Abbildung der Strecke hat somit dieselbe Struktur wie die zu identifizierende Strecke. Die Werte der Elemente können aber unterschiedlich sein und entsprechen den Werten des Parametervektors  $\hat{w}$ . Diese Beschreibung ist notwendig, da die Parameter der Strecke zunächst unbekannt sind und im weiteren Verlauf identifiziert werden sollen. Der Auskopp lungsvektor  $c$  stellt dabei eine Ausnahme dar, da die Berechnung der partiellen Ableitungen davon ausgeht, dass die Elemente des Auskopplungsvektors bekannt sind. Dies stellt bei realen Systemen praktisch keine Einschränkung dar, da der Systemausgang bei realen Anwendungen einem Zustand des Systems entspricht. Damit wird die Beziehung  $\tilde{c} = c$  für die Berechnung der partiellen Ableitungen vorausgesetzt.

Die Funktionen im Vektor  $\tilde{\mathcal{L}}$  sind durch statische Funktionsapproximatoren ersetzt.

Mit der Rechteckapproximation der Integratoren gilt für einen Zustand des rekurrenten Netzes  $\hat{x}_i[k]$  die Beziehung

$$\hat{x}_i[k+1] = h \cdot \hat{u}_{int}[k] + \hat{x}_i[k]$$

Wird diese Gleichung auf alle Zustände erweitert, ergibt sich mit den Bedingungen  $\hat{x}[k] = \tilde{x}[k]$  und  $\hat{u}_{int}[k] = \tilde{\mathbf{A}} \cdot \tilde{x}[k] + \tilde{b} \cdot u[k] + \tilde{\mathbf{K}}_{\mathcal{N}} \cdot \tilde{\mathcal{L}}(u[k], \tilde{x}[k])$ , die für kleine Abtastzeiten  $h$  gelten,<sup>5)</sup> die Gleichung

$$\hat{x}[k+1] = h \cdot \left( \tilde{\mathbf{A}} \cdot \hat{x}[k] + \tilde{b} \cdot u[k] + \tilde{\mathbf{K}}_{\mathcal{N}} \cdot \tilde{\mathcal{L}}(u[k], \hat{x}[k]) \right) + \hat{x}[k]$$

Durch Zusammenfassen kann diese Gleichung auf die Form

$$\hat{x}[k+1] = \underbrace{[h \cdot \tilde{\mathbf{A}} + \mathbf{E}]}_{\hat{\mathbf{A}}} \cdot \hat{x}[k] + \underbrace{h \cdot \tilde{b}}_{\hat{b}} \cdot u[k] + \underbrace{h \cdot \tilde{\mathbf{K}}_{\mathcal{N}} \cdot \tilde{\mathcal{L}}(u[k], \hat{x}[k])}_{\hat{\mathbf{K}}_{\mathcal{N}}}$$

gebracht werden. Dies ergibt die diskrete Zustandsbeschreibung

$$\hat{x}[k+1] = \hat{\mathbf{A}} \cdot \hat{x}[k] + \hat{b} \cdot u[k] + \hat{\mathbf{K}}_{\mathcal{N}} \cdot \hat{\mathcal{L}}(u[k], \hat{x}[k]) \quad \text{und} \quad \hat{y}[k] = \hat{c}^T \cdot \hat{x}[k] \quad (6.5)$$

mit

- $\hat{\mathbf{A}} = h \cdot \tilde{\mathbf{A}} + \mathbf{E}$
- $\hat{b} = h \cdot \tilde{b}$ ,
- $\hat{\mathbf{K}}_{\mathcal{N}} = h \cdot \tilde{\mathbf{K}}_{\mathcal{N}}$ ,
- $\hat{\mathcal{L}} = \tilde{\mathcal{L}}$  und

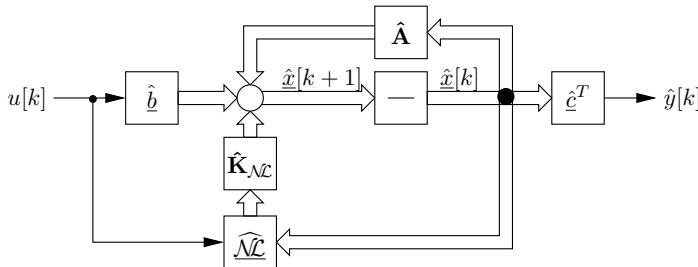
---

<sup>5)</sup> Mathematisch exakt gelten diese Bedingungen für den Grenzübergang  $h \rightarrow 0$ .

- $\hat{c} = \tilde{c} = c$ .

Diese Gleichungen gelten nur für die Euler-Vorwärts-Approximation der Integatoren und müssen für andere Methoden der numerischen Integration getrennt bestimmt werden [175].

Die Gleichungen (6.5) sind in Abbildung 6.5 graphisch dargestellt.



**Abb. 6.5:** Diskrete Zustandsbeschreibung

Das in Abbildung 6.1 enthaltene Differentiationsglied stellt bei dieser Betrachtung einen Sonderfall dar. Um diesen Sonderfall später berücksichtigen zu können, wird im Allgemeinen davon ausgegangen, dass die Anzahl  $n_d$  der diskreten Zustände  $\hat{x}[k]$  ungleich der Anzahl  $n$  der kontinuierlichen Zustände  $\underline{x}$  sein kann.

**Simulation:** Ohne zunächst auf die Bildung der partiellen Ableitungen explizit einzugehen, soll anhand einer Simulation auf ein wesentliches Problem aufmerksam gemacht werden.

In dieser Simulation wird nur die Massenträgheit der Maschine identifiziert. Das Reibungsdrehmoment der Strecke und des Identifikators sind in dieser ersten Anwendung zu Null gesetzt. Dies entspricht der üblichen Idealisierung, elektrische Maschinen reibungsfrei darzustellen. In diesem Fall besteht die Strecke aus einem einzelnen Integrator.

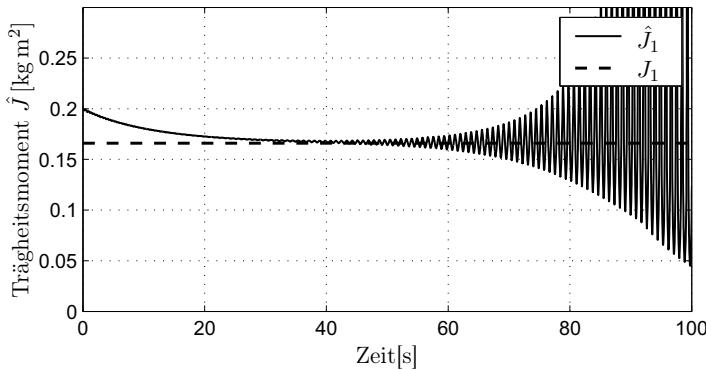
Das Massenträgheitsmoment in der Strecke wird zu  $0.166 \text{ kg m}^2$  gewählt, das Gewicht  $\hat{\Psi}_1[0]$  im rekurrenten Netz wird davon abweichend mit  $\frac{1}{0.2 \text{ kg m}^2} \approx 5 \frac{1}{\text{kg m}^2}$  initialisiert.

Die Anregung  $M_L$  wird mit einer einfachen Zwei-Punkt-Regelung realisiert. Überschreitet die Winkelgeschwindigkeit  $+20 \frac{\text{rad}}{\text{s}}$ , wird das Luftspaltmoment  $M_L$  mit  $-15 \text{ N m}$  vorgegeben. Werden  $-20 \frac{\text{rad}}{\text{s}}$  unterschritten, wird das Luftspaltmoment auf  $+15 \text{ N m}$  gesetzt. Damit erfolgt die Anregung der Strecke mit einem rechteckförmigen Drehmomentverlauf.

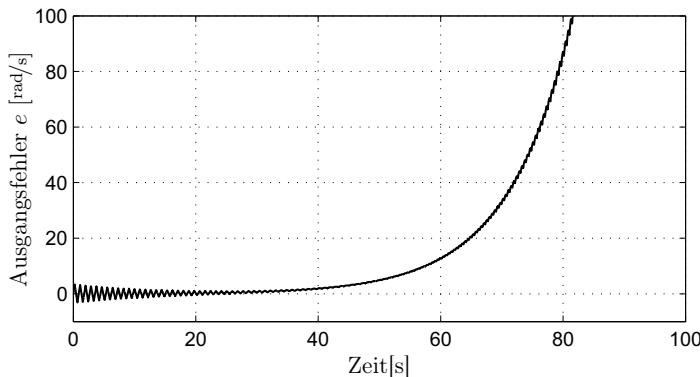
Für die Abtastzeit  $h$  wird in dieser Anwendung  $1 \text{ ms}$  gewählt. Die Lernparameter werden mit  $\eta_1 = 6 \cdot 10^{-6}$  und  $\alpha_1 = 0.95$  festgelegt.<sup>6)</sup>

<sup>6)</sup> Bei dieser Anwendung soll nur das Massenträgheitsmoment identifiziert werden, daher gilt für die Lernparameter  $\underline{\eta} = [\eta_1 \quad 0]^T$  und  $\underline{\alpha} = [\alpha_1 \quad 0]^T$ .

Der Identifikationsverlauf der Massenträgheit ist in Abbildung 6.6 dargestellt, der Verlauf des Ausgangsfehlers während der Identifikation ist in Abbildung 6.7 abgebildet.



**Abb. 6.6:** Identifikationsverlauf der Massenträgheit  $\hat{J}_1$



**Abb. 6.7:** Ausgangsfehlerverlauf

In Abbildung 6.6 ist zu erkennen, dass das rekurrente Netz nicht in der Lage ist das Trägheitsmoment richtig zu identifizieren, sondern es verlässt den physikalisch sinnvollen Bereich und wird instabil. Entsprechend strebt auch der Ausgangsfehler in Abbildung 6.7 gegen Unendlich. Der Grund für dieses Verhalten wird im Folgenden erläutert.

Das rekurrente Netz soll das Verhalten der Strecke nachbilden. Dies wird erreicht, wenn die Abweichung zwischen Strecke und rekurrentem Netz auch dynamisch zu Null wird. Nachdem die Modellierung mit Integratoren aufgebaut wird, ist dabei die Problematik unterschiedlicher Anfangswerte  $\Omega[0]$  und  $\hat{\Omega}[0]$  zu beachten. Bei der Simulation wird dieses Problem durch eine entsprechende

Initialisierung mit  $\Omega[0] = 0$  und  $\hat{\Omega}[0] = 0$  umgangen und für den Anfangsfehler wird  $e[0] = 0$  sichergestellt.

Durch die unterschiedliche Initialisierung der Massenträgheitsmomente zu Beginn der Identifikation ergibt sich jedoch schnell ein Ausgangsfehler. Entsprechend dem Lerngesetz wird über die Lernschrittweite und den Gradienten  $e \cdot \frac{\partial \hat{\Omega}}{\partial \hat{w}_1}$  das Gewicht  $\hat{w}_1$  so verändert, dass der Ausgangsfehler  $e$  minimiert wird. Bei ca. 40 Sekunden hat das Gewicht des rekurrenten Netzes den tatsächlichen Wert des Parameters erreicht. Bis zu diesem Zeitpunkt verringert sich auch die Amplitude des Ausgangsfehlers. Der Ausgangsfehler ist jedoch nicht zu Null geworden, weshalb das Gewicht  $\hat{w}_1$  weiter adaptiert wird.

Im Integrator des rekurrenten Netzes bleibt durch die vergangene Abweichung zwischen Strecke und rekurrentem Netz ein Fehler zurück. Da sich der Gradient aus dem Produkt von Ausgangsfehler und partieller Ableitung des Ausgangs nach dem Gewicht  $\hat{w}_1$  berechnet, kann der Gradient nur dann zu Null werden (und damit den Lernvorgang stoppen), wenn entweder der Ausgangsfehler oder der Term mit den partiellen Ableitungen zu Null werden. Tatsächlich entspricht die Berechnung der partiellen Ableitungen, wie später noch ersichtlich wird, der Berechnung des Systemausgangs. Damit ist  $\frac{\partial \hat{\Omega}}{\partial \hat{w}_1}$  solange von Null verschieden, solange die geschätzte Winkelgeschwindigkeit  $\hat{\Omega}$  nicht zu Null wird. Durch die externe Anregung des Systems wird jedoch erreicht, dass  $\hat{\Omega}$  nicht zu Null wird. Es verbleibt nur die Möglichkeit ein stabiles Lernergebnis zu erreichen, indem sichergestellt wird, dass der Ausgangsfehler zu Null wird, wenn die richtigen Parameter identifiziert sind.

Aus diesem Grund muss dafür gesorgt werden, dass der Ausgangsfehler tatsächlich zu Null werden kann.

Eine Variante zur Lösung dieser Problematik ist, die einzelnen Zustände des rekurrenten Netzes in regelmäßigen Intervallen so zu berechnen, dass der Ausgangsfehler Null wird. Im vorliegenden Beispiel bedeutet dies, die geschätzte Winkelgeschwindigkeit  $\hat{\Omega}$  auf den wahren Wert der Winkelgeschwindigkeit  $\Omega$  zu setzen. Tatsächlich sind reale Systeme aber wesentlich komplexer als dieses einfache Beispiel. Damit wird die Berechnung der einzelnen Zustände (bzw. Neuronen des rekurrenten Netzes) sehr aufwendig und stellt eine zusätzliche Fehlerquelle bei der Umsetzung der Identifikation dar.

In [25] wird dieses Problem gelöst, indem die Zustände des rekurrenten Netzes iterativ und von der eigentlichen Parameteridentifikation getrennt, bestimmt werden. Ein Online-Training ist bei dieser Vorgehensweise jedoch nicht möglich.

Eine wesentlich elegantere Lösung ist, das rekurrente Netz zu einem Luenberger-Beobachter zu erweitern. Dadurch wird zum einen die Anfangswertproblematik gelöst, und zum anderen werden die Zustände so nachgeführt, dass der Ausgangsfehler klein bleibt und stationär zu Null wird. Mit dieser Erweiterung ist es möglich, eine stabile Identifikation einer global integrierenden Strecke aufzubauen.

## 6.2 Erweiterung zum Luenberger-Beobachter

Analog zum klassischen Luenberger-Beobachter bzw. dem lernfähigen Beobachter aus Kapitel 5 wird der Ausgangsfehler auf die einzelnen Zustände des Rekurrenten Netzes zurückgeführt. Dies führt auf folgende kontinuierliche Beschreibung des Rekurrenten Netzes:

$$\dot{\tilde{x}} = \underbrace{\left( \tilde{\mathbf{A}} + \tilde{l} \cdot \tilde{c}^T \right)}_{\tilde{\mathbf{A}}_{\text{beo}}} \cdot \tilde{x} + \tilde{b} \cdot u - \tilde{l} \cdot y + \tilde{\mathbf{K}}_{\mathcal{NL}} \cdot \tilde{\mathcal{NL}}(u, \tilde{x}) \quad \text{und} \quad \tilde{y} = \tilde{c}^T \cdot \tilde{x} \quad (6.6)$$

Die Dynamik dieses Systems kann über die Pole der Matrix  $\tilde{\mathbf{A}}_{\text{beo}}$  und damit über die Beobachterkoeffizienten  $\tilde{l}$  vorgegeben werden. Dabei ist zu beachten, dass sich die Koeffizienten der Matrix  $\tilde{\mathbf{A}}$  während der Identifikation verändern, wodurch sich die Pole des Beobachters verschieben. Entsprechend muss bei der Beobachterdimensionierung beachtet werden, wie die Pole auf Änderungen der Koeffizienten reagieren. Eine Möglichkeit, um diese Problematik zu entschärfen, ist die Beobachterkoeffizienten während der Identifikation nachzuführen. Wie in Abschnitt 6.2.1 gezeigt wird, gehen die Beobachterkoeffizienten in die Berechnung der partiellen Ableitungen  $\nabla \tilde{y}$  ein. Entsprechend müssen auch die Gleichungen zur Veränderung der Beobachterkoeffizienten in die Berechnung der partiellen Ableitungen einfließen, was bei aufwendigen Beobachterdimensionierungen zu Rechenzeitproblemen führen kann.

Aus diesem Grund muss von Gebieten in der komplexen Ebene, in denen sich die Eigenwerte des Beobachters befinden, gesprochen werden. Um die Stabilität des Beobachters zu gewährleisten, muss sichergestellt sein, dass diese Gebiete immer im Bereich negativer Realteile liegen.

Zusätzlich ist zu beachten, dass es sich bei dem Gesamtsystem um ein nichtlineares System handelt und daher exakterweise nicht von Eigenwerten gesprochen werden kann.

Um die Beobachterdimensionierung einfach zu gestalten, wird von einem linearen Beobachter mit konstanten Koeffizienten, die den Startwerten der Identifikation entsprechen, ausgegangen. Dementsprechend erfolgt die Beobachterdimensionierung nach gängigen Verfahren wie der Polvorgabe.

Da bei realen Systemen immer eine verrauschte Umgebung vorliegt, muss bei der Beobachterdimensionierung ein Kompromiss zwischen schnellem Einschwingen und guter Filterwirkung getroffen werden. Im Fall der Dimensionierung nach dem Dämpfungsoptimum fließt dieser Kompromiss in die Wahl der Systemzeit ein.

Eine andere Möglichkeit, einen rauschoptimalen Beobachter zu entwerfen, liefert die *Kalman-Bucy-Filtertheorie* [119]. Das Ergebnis dieser Optimierung ist ein Luenberger-Beobachter mit einem Rückführvektor  $\tilde{l}$ , der über die positiv definite Lösung einer algebraischen Matrix-Riccati-Gleichung bestimmt wird.

Gemäß obigen Betrachtungen muss die Beobachterdimensionierung jetzt auf ihre Robustheit gegenüber Parameteränderungen der Matrix  $\tilde{\mathbf{A}}$  bzw. gegenüber

dem Einfluss der Nichtlinearitäten überprüft und gegebenenfalls korrigiert werden. Hierfür werden die Parameter wertmäßig beschränkt, d. h. die Parameter können sich nur innerhalb dieser definierten Grenzen verändern. Es ist nun zu überprüfen, ob das rekurrente Netz innerhalb dieser Schranken stabil arbeiten kann. Ist dies nicht der Fall, müssen entweder die Parameterschranken oder die Beobachterdimensionierung verändert werden. Die Parameterschranken werden sinnvollerweise in der anschließenden Identifikation beibehalten.

Auf Gleichung (6.6) können nun dieselben Umformungen wie in Kapitel 6.1.3 angewandt werden, womit sich

$$\hat{x}[k+1] = h \cdot \left( \left( \tilde{\mathbf{A}} + \tilde{l} \cdot \tilde{c}^T \right) \cdot \hat{x}[k] + \tilde{b} \cdot u[k] - \tilde{l} \cdot y[k] + \tilde{\mathbf{K}}_{\mathcal{N}\mathcal{L}} \cdot \tilde{\mathcal{N}\mathcal{L}}(u[k], \hat{x}[k]) \right) + \hat{x}[k]$$

ergibt. Durch einfache Zusammenfassungen und durch Berücksichtigung der Ausgangsgleichung ergibt sich aus dieser Gleichung die diskrete Zustandsbeschreibung des zum Luenberger-Beobachter erweiterten rekurrenten Netzes

$$\hat{x}[k+1] = \hat{\mathbf{A}}_{\text{rek}} \cdot \hat{x}[k] + \hat{b} \cdot u[k] - \hat{l} \cdot y[k] + \hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}} \cdot \hat{\mathcal{N}\mathcal{L}}(u[k], \hat{x}[k]) \quad \text{und} \quad \hat{y}[k] = \hat{c}^T \cdot \hat{x}[k] \quad (6.7)$$

mit

- $\hat{\mathbf{A}}_{\text{rek}} = h \cdot \tilde{\mathbf{A}}_{\text{beo}} + \mathbf{E} = h \cdot \left( \tilde{\mathbf{A}} + \tilde{l} \cdot \tilde{c}^T \right) + \mathbf{E}$ ,
- $\hat{b} = h \cdot \tilde{b}$ ,
- $\hat{l} = h \cdot \tilde{l}$ ,
- $\hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}} = h \cdot \tilde{\mathbf{K}}_{\mathcal{N}\mathcal{L}}$ ,
- $\hat{\mathcal{N}\mathcal{L}} = \tilde{\mathcal{N}\mathcal{L}}$  und
- $\hat{c} = \tilde{c}$ .

Diese Gleichungen gelten nur für die Euler-Vorwärts-Approximation der Integatoren und müssen für exaktere Integrationsmethoden separat bestimmt werden.

In Abbildung 6.8 ist die Kombination des realen Systems mit dem diskreten Beobachter über Digital-Analog- bzw. Analog-Digital-Wandler dargestellt.

### 6.2.1 Partielle Ableitungen

Aus der diskreten Zustandsbeschreibung können die partiellen Ableitungen  $\nabla \hat{y}$  zur Implementierung des Lerngesetzes aus Gleichung (6.3) berechnet werden.

Die partiellen Ableitungen  $\nabla \hat{y}$  ergeben sich nach Gleichung (6.7) zu:

$$\nabla \hat{y} = \frac{\partial \hat{c}^T}{\partial \hat{\mathbf{w}}} \cdot \hat{\mathbf{x}} + \hat{c}^T \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{w}}} \quad (6.8)$$

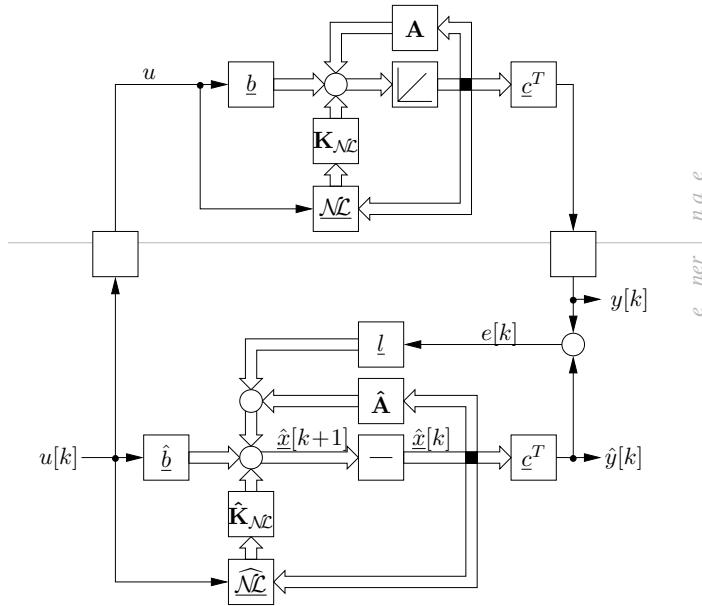


Abb. 6.8: Rekurrentes Netz als Luenberger-Beobachter

mit  $\hat{c} = \underline{c}$  vereinfacht sich (6.8) zu:

$$\nabla \hat{\mathbf{y}} = \underline{c}^T \cdot \frac{\partial \hat{\underline{x}}}{\partial \hat{w}} \quad (6.9)$$

Somit reduziert sich die Bestimmung von  $\nabla \hat{\mathbf{y}}$  auf die Berechnung von  $\partial \hat{\underline{x}} / \partial \hat{w}$ . Diese partiellen Ableitungen der Zustände nach den gesuchten Gewichten können mit Hilfe der Zustandsberechnung aus Gleichung (6.7) bestimmt werden. Hierfür ist es jedoch zunächst zweckmäßig neben dem bereits eingeführten Nabla-Operator die Jacobi-Matrix  $\hat{\mathbf{J}} \in \mathbb{R}^{n_d \times p}$  mit  $n_d$  diskreten Zuständen und  $p$  Parametern einzuführen.

$$\frac{\partial \hat{\underline{x}}[k]}{\partial \hat{w}} = \hat{\mathbf{J}}_{\hat{x}[k]} = \begin{bmatrix} (\nabla \hat{\mathbf{x}}_1[\mathbf{k}])^T \\ \vdots \\ (\nabla \hat{\mathbf{x}}_{n_d}[\mathbf{k}])^T \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{x}_1[k]}{\partial \hat{w}_1} & \dots & \frac{\partial \hat{x}_1[k]}{\partial \hat{w}_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{x}_{n_d}[k]}{\partial \hat{w}_1} & \dots & \frac{\partial \hat{x}_{n_d}[k]}{\partial \hat{w}_p} \end{bmatrix} \quad (6.10)$$

Die partielle Differentiation der Systembeschreibung (6.7) nach den einzelnen Gewichten führt auf

$$\begin{aligned}
\frac{\partial \hat{x}[k+1]}{\partial \hat{w}} &= \frac{\partial}{\partial \hat{w}} \left( \hat{\mathbf{A}}_{rek} \cdot \hat{x}[k] \right) + \\
&+ \frac{\partial}{\partial \hat{w}} \left( \hat{b} \cdot u[k] \right) - \\
&- \frac{\partial}{\partial \hat{w}} \left( \underline{L} \cdot y[k] \right) + \\
&+ \frac{\partial}{\partial \hat{w}} \left( \hat{\mathbf{K}}_{NL} \cdot \hat{\mathcal{N}}(u[k], \hat{x}[k]) \right)
\end{aligned}$$

Wegen der Unabhängigkeit von den wahren Größen  $u$  und  $y$  gilt  $\partial u[k]/\partial \hat{w} = 0$  und  $\partial y[k]/\partial \hat{w} = 0$  sowie  $\partial \underline{L}/\partial \hat{w} = 0$ . Mit diesen Bedingungen und der Produktregel kann die obige Gleichung weiter umgeformt werden

$$\begin{aligned}
\frac{\partial \hat{x}[k+1]}{\partial \hat{w}} &= \frac{\partial \hat{\mathbf{A}}_{rek}}{\partial \hat{w}} \cdot \hat{x}[k] + \hat{\mathbf{A}}_{rek} \cdot \frac{\partial \hat{x}[k]}{\partial \hat{w}} + \\
&+ \frac{\partial \hat{b}}{\partial \hat{w}} \cdot u[k] + \\
&+ \frac{\partial \hat{\mathbf{K}}_{NL}}{\partial \hat{w}} \cdot \hat{\mathcal{N}}(u[k], \hat{x}[k]) + \hat{\mathbf{K}}_{NL} \cdot \frac{\partial \hat{\mathcal{N}}(u[k], \hat{x}[k])}{\partial \hat{w}}
\end{aligned}$$

mit der eingeführten Jacobi-Matrix  $\hat{\mathbf{J}}_{\hat{x}[k]} = \partial \hat{x}[k]/\partial \hat{w}$  und elementaren Umformungen ergibt sich

$$\begin{aligned}
\hat{\mathbf{J}}_{\hat{x}[k+1]} &= \hat{\mathbf{A}}_{rek} \cdot \hat{\mathbf{J}}_{\hat{x}[k]} + \\
&+ \frac{\partial \hat{\mathbf{A}}_{rek}}{\partial \hat{w}} \cdot \hat{x}[k] + \\
&+ \frac{\partial \hat{b}}{\partial \hat{w}} \cdot u[k] + \\
&+ \frac{\partial \hat{\mathbf{K}}_{NL}}{\partial \hat{w}} \cdot \hat{\mathcal{N}}(u[k], \hat{x}[k]) + \hat{\mathbf{K}}_{NL} \cdot \frac{\partial \hat{\mathcal{N}}(u[k], \hat{x}[k])}{\partial \hat{w}}
\end{aligned}$$

Durch Zusammenfassen kann diese Gleichung in die Form

$$\hat{\mathbf{J}}_{\hat{x}[k+1]} = \hat{\mathbf{A}}_{rek} \cdot \hat{\mathbf{J}}_{\hat{x}[k]} + \hat{\mathbf{F}} \quad \text{mit} \quad \hat{\mathbf{F}} = \begin{bmatrix} \hat{f}_1 & \cdots & \hat{f}_i & \cdots & \hat{f}_p \end{bmatrix} \quad (6.11)$$

gebracht werden. Die Spalten der Matrix  $\hat{\mathbf{F}}$  ergeben sich dabei zu

$$\begin{aligned}
\hat{f}_i &= \frac{\partial \hat{\mathbf{A}}_{rek}}{\partial \hat{w}_i} \cdot \hat{x}[k] + \\
&+ \frac{\partial \hat{b}}{\partial \hat{w}_i} \cdot u[k] + \\
&+ \frac{\partial \hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}}}{\partial \hat{w}_i} \cdot \hat{\mathcal{N}\mathcal{L}}(u[k], \hat{x}[k]) + \hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}} \cdot \frac{\partial \hat{\mathcal{N}\mathcal{L}}(u[k], \hat{x}[k])}{\partial \hat{w}_i}
\end{aligned}$$

Entsprechend der Ausgangsgleichung  $\hat{y}[k] = \hat{c}^T \cdot \hat{x}[k]$  werden aus der Jacobi-Matrix  $\hat{\mathbf{J}}_{\hat{x}[k]}$  die gesuchten partiellen Ableitungen  $\nabla \hat{y}[\mathbf{k}]$  gewonnen

$$(\nabla \hat{y}[\mathbf{k}])^T = \left[ \begin{array}{ccc} \frac{\partial \hat{y}[k]}{\partial \hat{w}_1} & \dots & \frac{\partial \hat{y}[k]}{\partial \hat{w}_p} \end{array} \right] = \hat{c}^T \cdot \hat{\mathbf{J}}_{\hat{x}[k]} \quad (6.12)$$

Die Gleichungen (6.12) und (6.11) entsprechen der Systembeschreibung, wobei die Matrix  $\hat{\mathbf{F}}$  die partielle Differentiation der Systemmatrix  $\partial \hat{\mathbf{A}}_{rek}[k]/\partial \hat{w}_i$ , des Eingriffs  $\partial \hat{b} \cdot u[k-1]/\partial \hat{w}_i$  und der Nichtlinearitäten  $\frac{\partial}{\partial \hat{w}_i} (\hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}} \cdot \hat{\mathcal{N}\mathcal{L}}(u[k-1], \hat{x}[k-1]))$  zusammenfasst. Damit kann für die Berechnung der partiellen Ableitungen die Struktur des rekurrenten Netzes übernommen werden, wobei die durch die Matrix  $\hat{\mathbf{F}}$  beschriebenen Eingriffe ergänzt werden müssen.

Aus Gleichung (6.11) ist auch ersichtlich, dass die Jacobi-Matrix  $\hat{\mathbf{J}}_{\hat{x}[k+1]}$  aus der aktuellen Jacobi-Matrix  $\hat{\mathbf{J}}_{\hat{x}[k]}$  berechnet wird. Entsprechend ist zur Berechnung von  $\hat{\mathbf{J}}_{\hat{x}[1]}$  die Matrix  $\hat{\mathbf{J}}_{\hat{x}[0]}$  notwendig. Da für  $k \leq 0$  alle Zustände  $\hat{x}[k]$  und alle Gewichte  $\hat{w}$  konstant sind, gilt für die Jacobi-Matrix  $\hat{\mathbf{J}}_{\hat{x}[0]} = \mathbf{0}$ . Die Jacobimatrix wird also nicht durch explizites Differenzieren entsprechend Gleichung (6.10), sondern durch eine einfache Rekursion bestimmt.

### 6.2.2 Implementierung der statischen Neuronalen Netze

Wie bereits erwähnt, sind die statischen Neuronalen Netze im Allgemeinen nur Subsysteme des strukturierten rekurrenten Netzes. Das heißt, die Gewichte  $\hat{\Theta}$  eines statischen Neuronalen Netzes sind im Parametervektor  $\hat{w}$  neben den restlichen Gewichten des rekurrenten Netzes enthalten. Damit sind auch die Eingangsgrößen und die Ausgangsgröße eines Approximators abhängig von den Parametern  $\hat{w}$ . Entsprechend gilt für die partiellen Ableitungen der Eingangsgrößen nach den Gewichten

$$\frac{\partial \hat{u}}{\partial \hat{w}_i} \neq 0 \quad \text{mit} \quad \hat{u} \equiv \hat{x}_{\mathcal{N}\mathcal{L}}$$

Aus diesem Grund muss bei der Berechnung der partiellen Ableitungen  $\partial \hat{\mathcal{N}\mathcal{L}}/\partial \hat{w}_i$  unterschieden werden, ob nach einem Gewicht des Funktionsapproximators (es gilt  $\hat{w}_i = \hat{\Theta}_l$ )<sup>7)</sup> oder nach einem Gewicht des restlichen Netzes (es gilt  $\hat{w}_i \neq \hat{\Theta}_l$ ) differenziert wird. Im Folgenden gilt  $\hat{\mathcal{N}\mathcal{L}} = \hat{y}_{RBF}$  bzw.  $\hat{\mathcal{N}\mathcal{L}} = \hat{y}_{GRNN}$  etc.

<sup>7)</sup>  $\hat{\Theta}_l$  ist ein Gewicht des Funktionsapproximators mit  $1 \leq l \leq r$ , wobei  $r$  die Anzahl der Stützstellen des Funktionsapproximators darstellt.

Eine ausführliche Herleitung der partiellen Ableitungen würde den Rahmen dieser Vorlesung sprengen und kann in [88] nachgelesen werden. Die partiellen Ableitungen für das GRNN, HANN und RBF-Netz sind der Vollständigkeit halber in Tabelle 6.1 zusammengefasst.

Netz	partielle Ableitung
RBF	Für $\hat{w}_i \neq \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $1 \leq l \leq r$ $\frac{\partial \hat{y}_{RBF}}{\partial \hat{w}_i} = -\frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^r \hat{\Theta}_j \cdot \mathcal{A}_j(\hat{u}) \cdot \frac{\hat{u} - \xi_j}{\sigma_{norm}^2 \cdot \Delta \xi^2}$
	Für $\hat{w}_i = \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $1 \leq l \leq r$ $\frac{\partial \hat{y}_{RBF}}{\partial \hat{w}_i} = \mathcal{A}_l(\hat{u}) - \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^r \hat{\Theta}_j \cdot \mathcal{A}_j(\hat{u}) \cdot \frac{\hat{u} - \xi_j}{\sigma_{norm}^2 \cdot \Delta \xi^2}$
GRNN	Für $\hat{w}_i \neq \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $1 \leq l \leq r$ $\frac{\partial \hat{y}_{GRNN}}{\partial \hat{w}_i} = \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^r \mathcal{A}_j(\hat{u}) \cdot \frac{\hat{u} - \xi_j}{\sigma_{norm}^2 \cdot \Delta \xi^2} \cdot (\hat{y}_{GRNN} - \hat{\Theta}_j)$
	Für $\hat{w}_i = \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $1 \leq l \leq r$ $\frac{\partial \hat{y}_{GRNN}}{\partial \hat{w}_i} = \mathcal{A}_l(\hat{u}) + \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^r \mathcal{A}_j(\hat{u}) \cdot \frac{\hat{u} - \xi_j}{\sigma_{norm}^2 \cdot \Delta \xi^2} \cdot (\hat{y}_{GRNN} - \hat{\Theta}_j)$
HANN	Für $\hat{w}_i \neq \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $1 \neq l \neq r$ $\frac{\partial \hat{y}_{HANN}}{\partial \hat{w}_i} = \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^{\frac{r-1}{2}} j \cdot \left( \hat{\Theta}_{(\frac{r+1}{2}+j)} \cdot \cos(j \hat{u}) - \hat{\Theta}_{(j+1)} \cdot \sin(j \hat{u}) \right)$
	Für $\hat{w}_i = \hat{\Theta}_1$ mit $1 \leq i \leq p$ $\frac{\partial \hat{y}_{HANN}}{\partial \hat{w}_i} = 1 + \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^{\frac{r-1}{2}} j \cdot \left( \hat{\Theta}_{(\frac{r+1}{2}+j)} \cdot \cos(j \hat{u}) - \hat{\Theta}_{(j+1)} \cdot \sin(j \hat{u}) \right)$
	Für $\hat{w}_i = \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $2 \leq l \leq \frac{r+1}{2}$ $\frac{\partial \hat{y}_{HANN}}{\partial \hat{w}_i} = \cos(l \hat{u}) + \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^{\frac{r-1}{2}} j \cdot \left( \hat{\Theta}_{(\frac{r+1}{2}+j)} \cdot \cos(j \hat{u}) - \hat{\Theta}_{(j+1)} \cdot \sin(j \hat{u}) \right)$
	Für $\hat{w}_i = \hat{\Theta}_l$ mit $1 \leq i \leq p$ und $\frac{r+3}{2} \leq l \leq r$ $\frac{\partial \hat{y}_{HANN}}{\partial \hat{w}_i} = \sin(l \hat{u}) + \frac{\partial \hat{u}}{\partial \hat{w}_i} \cdot \sum_{j=1}^{\frac{r-1}{2}} j \cdot \left( \hat{\Theta}_{(\frac{r+1}{2}+j)} \cdot \cos(j \hat{u}) - \hat{\Theta}_{(j+1)} \cdot \sin(j \hat{u}) \right)$

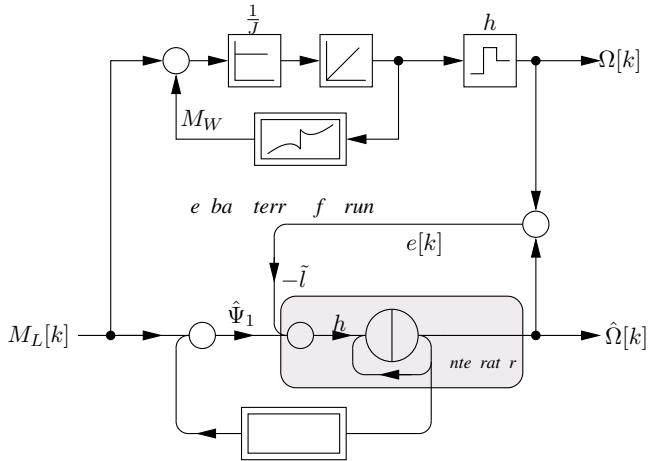
**Tabelle 6.1:** Zusammenstellung der partiellen Ableitungen der statischen Neuronalen Netze

### 6.2.3 Anwendung der Beobachterstruktur

Im Folgenden wird anhand des bereits eingeführten Beispiels der Mechanik einer elektrischen Maschine anschaulich erläutert, wie das rekurrente Netz zum Beobachter erweitert wird. Wie sich diese Erweiterung auf die Parameteradaption auswirkt, wird in Abschnitt 6.2.4 dargestellt.

In Abbildung 6.9 ist das rekurrente Netz aus Abbildung 6.2 inklusive der Strecke und der Beobachterrückführung dargestellt.

Der Eingriff der Beobachterrückführung erfolgt am Eingang der Integrator-Approximation, daher wird der Beobachterkoeffizient  $\tilde{l}$  des kontinuierlichen Beobachters eingesetzt. Bei dieser Darstellung der Beobachterrückführung kann die vorwärts Rechteckapproximation durch eine exaktere Methode ersetzt werden,



**Abb. 6.9:** Neuronaler Beobachter der Mechanik einer elektrischen Maschine

indem der Block für den Integrator im rekurrenten Netz entsprechend ersetzt wird.

Aus Abbildung 6.9 kann die Differenzengleichung

$$\hat{\Omega}[k+1] = \left(1 + h \cdot \tilde{l}\right) \cdot \hat{\Omega}[k] - h \cdot \tilde{l} \cdot \Omega[k] + h \cdot \hat{\Psi}_1 \cdot M_L[k] - h \cdot \hat{\Psi}_1 \cdot \hat{y}_{GRNN}(\hat{\Omega}[k])$$

abgelesen werden. Durch einen Vergleich der Differenzengleichung mit der diskreten Zustandsbeschreibung des neuronalen Beobachters nach Gleichung (6.7) ergeben sich die Elemente der Zustandsbeschreibung, wie in Tabelle 6.2 zusammengefasst.

dis. Zustandsbeschr.	$u$	$\hat{x}$	$\hat{\mathbf{A}}_{\text{rek}}$	$\hat{b}$	$\hat{l}$	$\hat{\mathbf{K}}_{\mathcal{N}}$	$\widehat{\mathcal{NL}}(u, \underline{x})$	$\hat{c}$	$\hat{y}$
Differenzengleichung	$M_L$	$\hat{\Omega}$	$1 + h \cdot \tilde{l}$	$h \cdot \hat{\Psi}_1$	$h \cdot \tilde{l}$	$-h \cdot \hat{\Psi}_1$	$\hat{y}_{GRNN}(\hat{\Omega})$	1	$\hat{\Omega}$

**Tabelle 6.2:** Elemente der diskreten Zustandsbeschreibung

#### 6.2.4 Durchführung der Identifikation

Mit den Elementen der diskreten Zustandsdarstellung aus Tabelle 6.2 und dem Parametervektor

$$\hat{\underline{w}} = \begin{bmatrix} \hat{\Psi}_1 & \hat{\Theta}_1 & \dots & \hat{\Theta}_r \end{bmatrix}^T$$

können die partiellen Ableitungen  $\nabla \hat{\Omega}$  unter Berücksichtigung der Beobachterrückführung nach Gleichung (6.11) und (6.12) bestimmt werden. Die Ergebnisse dieser Berechnungen sind

$$\hat{\mathbf{J}}_{\hat{x}[k+1]} = \hat{\mathbf{A}}_{rek} \cdot \hat{\mathbf{J}}_{\hat{x}[k]} + \hat{\mathbf{F}} = \hat{\mathbf{J}}_{\hat{\Omega}[k-1]} = \left(1 + h \cdot \tilde{l}\right) \cdot \hat{\mathbf{J}}_{\Omega[k]} + \hat{\mathbf{F}}$$

mit

$$\begin{aligned} \hat{f}_1 &= \frac{\partial \hat{\mathbf{A}}_{rek}}{\partial \hat{w}_1} \cdot \hat{x}[k] + \\ &+ \frac{\partial \hat{b}}{\partial \hat{w}_1} \cdot u[k] + \\ &+ \frac{\partial \hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}}}{\partial \hat{w}_1} \cdot \underline{\mathcal{NL}}(u[k], \hat{x}[k]) + \hat{\mathbf{K}}_{\mathcal{N}\mathcal{L}} \cdot \frac{\partial \underline{\mathcal{NL}}(u[k], \hat{x}[k])}{\partial \hat{w}_1} = \\ &= \frac{\partial \left(1 + h \cdot \tilde{l}\right)}{\partial \hat{w}_1} \cdot \hat{\Omega}[k] + \\ &+ \frac{\partial(h \cdot \hat{\Psi}_1)}{\partial \hat{w}_1} \cdot M_L[k] + \\ &+ \frac{\partial(-h \cdot \hat{\Psi}_1)}{\partial \hat{w}_1} \cdot \hat{y}_{GRNN}(\hat{\Omega}[k]) + (-h \cdot \hat{\Psi}_1) \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_1} = \\ &= 0 \cdot \hat{\Omega}[k] + \\ &+ h \cdot M_L[k] - \\ &- h \cdot \hat{y}_{GRNN}(\hat{\Omega}[k]) - h \cdot \hat{\Psi}_1 \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_1} = \\ &= h \cdot \left(M_L[k] - \hat{y}_{GRNN}(\hat{\Omega}[k])\right) - h \cdot \hat{\Psi}_1 \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_1} \end{aligned}$$

und für  $2 \leq i \leq p$

$$\begin{aligned}
\hat{f}_i &= \frac{\partial \hat{\mathbf{A}}_{rek}}{\partial \hat{w}_i} \cdot \hat{x}[k] + \\
&+ \frac{\partial \hat{b}}{\partial \hat{w}_i} \cdot u[k] + \\
&+ \frac{\partial \hat{\mathbf{K}}_{NL}}{\partial \hat{w}_i} \cdot \hat{\mathcal{NL}}(u[k], \hat{x}[k]) + \hat{\mathbf{K}}_{NL} \cdot \frac{\partial \hat{\mathcal{NL}}(u[k], \hat{x}[k])}{\partial \hat{w}_i} \frac{\partial (1 + h \cdot \hat{l})}{\partial \hat{w}_i} \cdot \hat{\Omega}[k] + \\
&+ \frac{\partial (h \cdot \hat{\Psi}_1)}{\partial \hat{w}_i} \cdot M_L[k] + \\
&+ \frac{\partial (-h \cdot \hat{\Psi}_1)}{\partial \hat{w}_i} \cdot \hat{y}_{GRNN}(\hat{\Omega}[k]) + (-h \cdot \hat{\Psi}_1) \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_i} = \\
&= 0 \cdot \hat{\Omega}[k] + \\
&+ 0 \cdot M_L[k] + \\
&+ 0 \cdot \hat{y}_{GRNN}(\hat{\Omega}[k]) - h \cdot \hat{\Psi}_1 \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_i} = \\
&= -h \cdot \hat{\Psi}_1 \cdot \frac{\partial \hat{y}_{GRNN}(\hat{\Omega}[k])}{\partial \hat{w}_i}
\end{aligned}$$

Das graphische Äquivalent dieser Berechnungen ist in Abbildung 6.10 enthalten.

Diese Abbildung zeigt, wie die Strecke, das rekurrente Netz (Abbildung 6.9), die Berechnung der partiellen Ableitungen und das Lerngesetz (6.3) kombiniert werden, um die Identifikation der Maschinenparameter (Massenträgheitsmoment und Reibungskennlinie) durchzuführen.

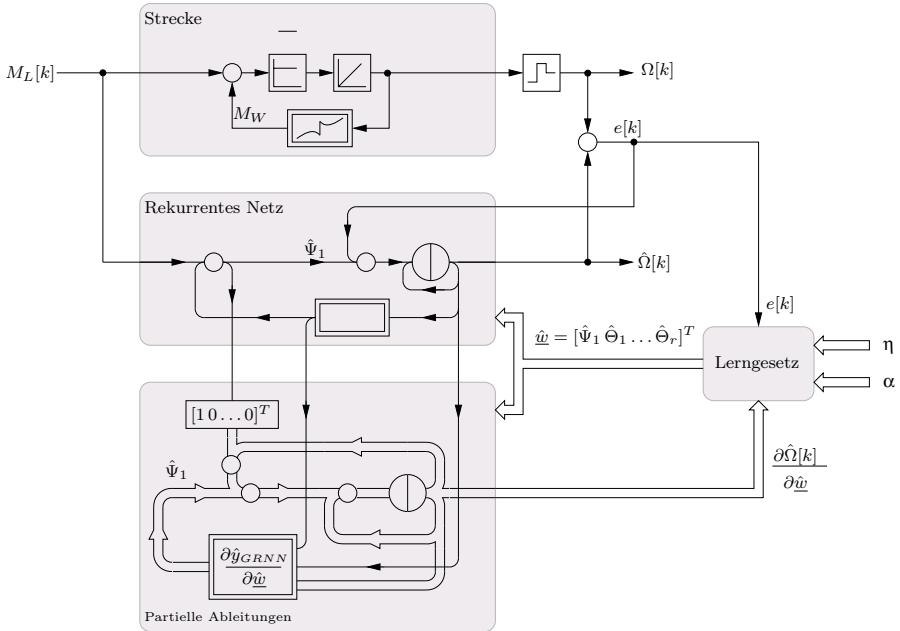
### Simulation

Anhand der in Abbildung 6.10 beschriebenen Gleichungen wird die simulative Parameteridentifikation durchgeführt.

Die Anregung der Strecke erfolgt wiederum mit der zuvor beschriebenen Zwei-Punkt-Regelung. Das Massenträgheitsmoment wird auf  $J_1 = 0.166 \text{ kg m}^2$  gesetzt. Im rekurrenten Netz wird der zugehörige Parameter mit  $\hat{\Psi}_1[0] = \frac{1}{0.2 \text{ kg m}^2} \approx 5 \frac{1}{\text{kg m}^2}$  initialisiert. Die Abtastzeit wird in dieser Anwendung mit  $h = 1 \text{ ms}$  festgelegt.

Abweichend gegenüber der zuvor durchgeföhrten Simulation wird hier die Maschine nicht idealisiert, d. h. es wird ebenfalls eine Reibungskennlinie vorgegeben. Damit wird bei dieser Anwendung neben dem Massenträgheitsmoment auch die Reibungskennlinie über die Approximation durch ein GRNN identifiziert. Für das GRNN werden zusätzlich der Eingangsbereich<sup>8)</sup>  $-20 \frac{\text{rad}}{\text{s}} \leq \hat{u}_{GRNN} \leq +20 \frac{\text{rad}}{\text{s}}$ , der Glättungsfaktor  $\sigma_{1,norm} = 1.6$  und die Anzahl der Stützstellen  $r_1 = 30$  festgelegt.

<sup>8)</sup> Die Grenzen des Eingangsbereiches sind gleichzeitig die Schaltschwellen der überlagerten Zwei-Punkt-Regelung.



**Abb. 6.10:** Rekurrentes Netz zur Identifikation der Mechanik einer elektrischen Maschine als Beobachter

Die Beobachterrückführung wird mit  $\tilde{l} = 15$  festgelegt. Der lineare Systemanteil hat damit einen reellen Eigenwert bei  $\tilde{\lambda} = -15$ , während der lineare Anteil der zu identifizierenden Strecke einen Eigenwert bei  $\lambda = 0$  und damit global integrierendes Verhalten aufweist.

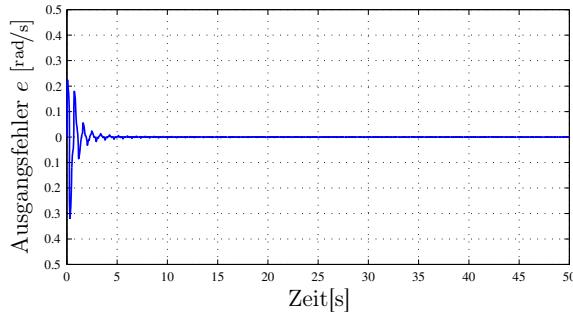
Durch die Berücksichtigung der Reibung und der Erweiterung zum Beobachter müssen die Lernparameter gegenüber der vorherigen Simulation neu festgelegt werden. Für den linearen Parameter wird  $\eta_1 = 1 \cdot 10^{-3}$  und  $\alpha_1 = 0.95$  festgelegt. Für die nichtlinearen Parameter, die Stützstellen der Reibungskennlinie, wird  $\eta_{2...31} = 5 \cdot 10^{-2}$  und  $\alpha_{2...31} = 0.95$  gewählt.

Abbildung 6.11 zeigt den Verlauf des Ausgangsfehlers während der Identifikation. Nach 3 Sekunden ist der Fehler gegenüber dem Maximalwert deutlich kleiner und nach 10 Sekunden praktisch zu Null geworden.

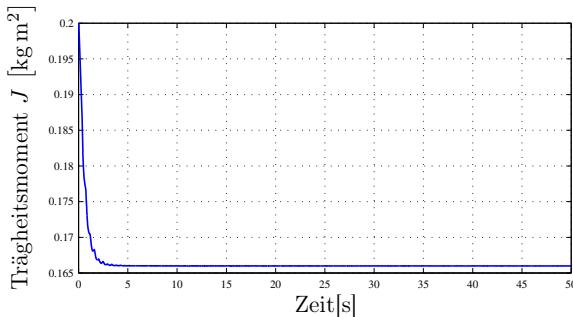
Der Verlauf des Massenträgheitsmoments während der Identifikation ist in Abbildung 6.12 dargestellt. Der Endwert  $J = 0.166 \text{ kg m}^2$  ist nach 10 Sekunden erreicht.

Der Verlauf der Gewichte des GRNN ist in Abbildung 6.13 abgebildet.

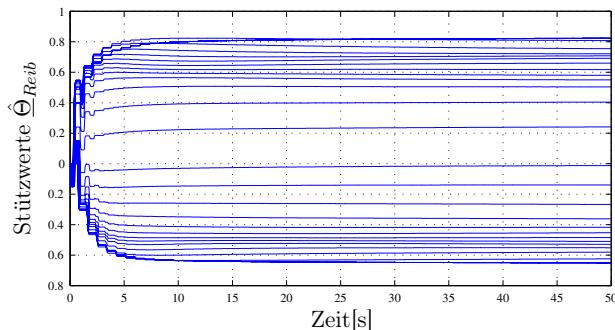
Die zu den Werten nach 50 Sekunden gehörende Reibungskennlinie ist in Abbildung 6.14 dargestellt.



**Abb. 6.11:** Ausgangsfehlerverlauf während der Identifikation



**Abb. 6.12:** Identifikationsverlauf der Massenträgheit  $\hat{J}$



**Abb. 6.13:** Identifikationsverlauf der Gewichte  $\hat{\Theta}_{Reib}$  des Funktionsapproximators

Im Bereich der Haftreibung weicht die identifizierte Reibungskennlinie leicht von der vorgegebenen Kennlinie ab. Da der Ausgangsfehler nach 30 Sekunden gegenüber dem Ausgangsfehler bei Identifikationsbeginn sehr klein ist, und die Gewichtsänderungen im rekurrenten Netz entsprechend dem Lerngesetz über den

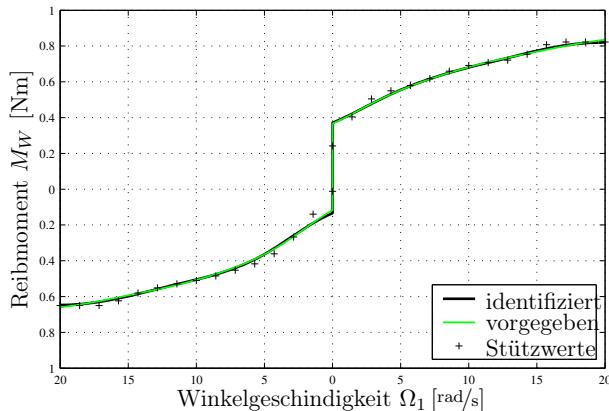


Abb. 6.14: Identifikationsergebnis der Reibungskennlinie  $M_R(\Omega)$

Gradienten  $e \cdot \nabla \hat{\Omega}$  vom Ausgangsfehler  $e$  abhängen, sinken mit dem Ausgangsfehler auch die Gewichtsänderungen. Damit wäre, um die Reibungskennlinie im Bereich der Haftreibungen weiter anzupassen, eine unverhältnismäßig große Simulationsdauer notwendig, die aber keine wesentliche Verringerung des Ausgangsfehlers mit sich bringt.

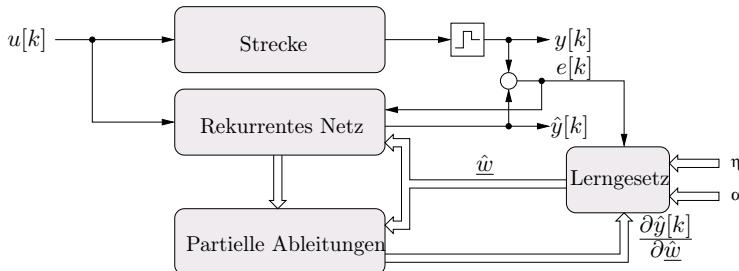
Vergleicht man die Identifikationsverläufe der Trägheitsmomente  $\hat{J}_I$  in den Abbildungen 6.6 und 6.12 sowie die Ausgangsfehler in den Abbildungen 6.7 und 6.11, so erkennt man, dass mit Hilfe der Erweiterung zur Beobachterstruktur ein stabiles und konvergentes Identifikationsergebnis erreicht wurde.

### 6.3 Beurteilung des Identifikationsverfahrens

Ausgehend von einer bekannten Systemstruktur wird entsprechend Abbildung 6.1, diese Struktur in ein rekurrentes Netz übertragen. Die statischen Nichtlinearitäten werden dabei durch statische Funktionsapproximatoren ersetzt. Die zu identifizierenden Parameter sind im rekurrenten Netz als Gewichte enthalten und werden im Parametervektor zusammengefasst.

Das rekurrente Netz wird zu einem Beobachter erweitert. Zur Auswertung des Lerngesetzes aus (6.3) werden aus der Zustandsbeschreibung unter Berücksichtigung der Beobachterrückführungen die partiellen Ableitungen  $\nabla \hat{y}$  berechnet. Strecke, rekurrentes Netz, Lerngesetz und die Berechnung der partiellen Ableitungen werden, wie in Abbildung 6.15 dargestellt, zur Adaption des Parametervektors kombiniert.

Vorteilhaft bei der vorgestellten Identifikationsmethode ist die physikalische Interpretierbarkeit der Identifikationsergebnisse. Die Ergebnisse können damit



**Abb. 6.15:** Kombination von Strecke, Beobachter, Lerngesetz und Berechnung der partiellen Ableitungen zur Parameteradaption

auf ihre Plausibilität überprüft werden. Ein weiterer Vorteil ist die Echtzeitfähigkeit des eingesetzten Gradientenverfahrens zur Parameteradaption.

Durch die Auskopplung der partiellen Ableitungen an den Ausgang des diskretisierten Integrators ist es möglich, ein beliebiges Integrationsverfahren mit konstanter Schrittweite zu verwenden.

Die Implementierung des rekurrenten Netzes in einer Beobachterstruktur bewirkt, zusätzlich zur Umgehung der Anfangswertproblematik und der Vermeidung von divergierenden Systemzuständen, eine Filterung der Messsignale. So mit wird eine eventuelle Phasenverschiebung der Signale, wie sie z. B. entsteht, wenn die Messsignale vor einer Identifikation gefiltert werden, verhindert.<sup>9)</sup>

Nachteilig wirkt sich die starke Abhängigkeit der Parameterkonvergenzgeschwindigkeit von den manuell zu bestimmenden Lernparametern  $\eta$  und  $\alpha$  aus. Weitere Nachteile sind, daß die Startwerte der Parameter nicht zu ungünstig gewählt werden dürfen (Vorwissen) und daß aufgrund der Änderung der linearen Parameter auch die Beobachter-Rückführungen  $l$  während der Identifikation angepaßt werden müssen.

Zusätzlich ist für eine erfolgreiche Parameteridentifikation ein genaues und eindeutiges Systemmodell in Form eines Signalflussplanes notwendig. Stimmt die Struktur des Systemmodells nicht mit dem realen System überein, werden die im Signalflussplan fehlenden Teile soweit wie möglich über die vorhandenen Gewichte ausgeglichen. Dies kann, wie in [5] erläutert, zu scheinbar physikalisch widersprüchlichen Ergebnissen führen, was jedoch eigentlich ein Zeichen für eine unzureichende Annahme der Systemstruktur ist.

<sup>9)</sup> Eine Vorwärts- mit anschließender Rückwärtsfilterung würde dieses Problem beseitigen, ist jedoch in einer Echzeitanwendung nicht durchführbar.

## 6.4 Anwendungsbeispiel

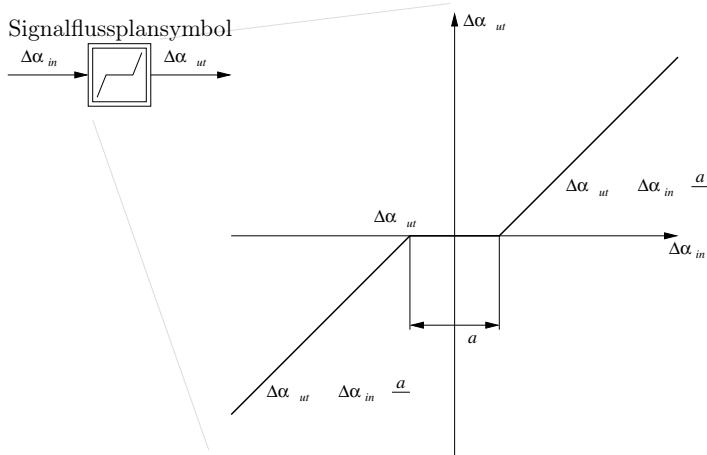
In diesem Abschnitt wird ein mit Lose und Reibung behaftetes Zweimassensystem identifiziert. Dieses System entspricht einen häufig vorkommenden Anwendungsfällen in der Antriebstechnik, und zwar immer dann, wenn eine Last über ein Getriebe von einem Antriebsmotor angetrieben wird. In diesen Fällen ist es oft auch so, dass nur die Drehzahl oder auch die Position des Antriebsmotors zur Verfügung stehen. Aus diesem Grund wird in dem folgenden Beispiel davon ausgegangen, dass lediglich Position und Drehzahl der ersten Masse zur Verfügung stehen.

Um die Lose zu approximieren, wird zunächst ein einfaches Losmodell vorgestellt, sowie die Berücksichtigung von Hafreibung dargestellt.

### 6.4.1 Losmodellierung und Approximation

Um die angenommene Lose identifizieren zu können, wird zunächst ein geeigneter Losapproximator entwickelt.

Die Lose wird in erster Näherung durch die in Abbildung 6.16 dargestellte Totzone beschrieben.



**Abb. 6.16:** Losekennlinie als Totzone und Signalflussplätsymbol

Bei dieser Modellierung kann die Lose durch einen einzigen Parameter, die Loseweite  $a$ , erfasst werden. Die mathematische Beschreibung dieser Kennlinie ergibt sich zu

$$\Delta\alpha_{out} = \begin{cases} \Delta\alpha_{in} - \frac{a}{2} & \text{wenn } \Delta\alpha_{in} > \frac{a}{2} \\ 0 & -\frac{a}{2} \leq \Delta\alpha_{in} \leq \frac{a}{2} \\ \Delta\alpha_{in} + \frac{a}{2} & \Delta\alpha_{in} < -\frac{a}{2} \end{cases} \quad (6.13)$$

Zur Vereinfachung wird der zu identifizierende Parameter  $\hat{\Theta}_L = \frac{\hat{a}}{2}$  festgelegt. Damit kann die beschriebene Totzone im rekurrenten Netz übernommen werden und es gilt

$$\Delta\hat{\alpha}_{out} = \begin{cases} \Delta\hat{\alpha}_{in} - \hat{\Theta}_L & \Delta\hat{\alpha}_{in} > \hat{\Theta}_L \\ 0 & -\hat{\Theta}_L \leq \Delta\hat{\alpha}_{in} \leq \hat{\Theta}_L \\ \Delta\hat{\alpha}_{in} + \hat{\Theta}_L & \Delta\hat{\alpha}_{in} < -\hat{\Theta}_L \end{cases} \quad (6.14)$$

Diese Losemodellierung ist stark vereinfacht, da der Stoßvorgang beim Eingreifen der Lose nicht berücksichtigt wird. Durch Materialverformungen beim Eingreifen der Lose entsteht ein Drehmoment, das dieser Bewegung entgegenwirkt. Dadurch wird die elastische Verbindung von Antriebs- und Arbeitsmaschine wesentlich stärker zum Schwingen angeregt.

Die Identifikation an der Versuchsanlage hat gezeigt, dass dieser Stoßvorgang nicht vernachlässigt werden kann. Durch den mechanischen Aufbau der Losekonstruktion entspricht das Eingreifen der Lose einem Stoß von zwei rotierenden Massen  $J_I$  und  $J_{II}$ . Bei idealisierter Betrachtung des Stoßvorgangs wird davon ausgegangen, dass während des Stoßvorgangs alle äußeren Einflüsse gegenüber dem Drehmomentimpuls vernachlässigbar klein sind, und dass der Stoßvorgang in unendlich kurzer Zeit abläuft.

Mit diesen Vereinfachungen ergibt sich eine sprungartige Änderung der Winkelgeschwindigkeiten.

Mit den Winkelgeschwindigkeiten  $\Omega_{I/II}^+$  nach dem Stoß und  $\Omega_{I/II}^-$  vor dem Stoß ergibt sich nach [15] der Zusammenhang

$$\begin{aligned} \hat{\Omega}_I^+ - \hat{\Omega}_I^- &= (\hat{\Omega}_I^- - \hat{\Omega}_{II}^-) \frac{1+\epsilon}{J_I + J_{II}} \cdot \hat{J}_{II} \\ \hat{\Omega}_{II}^+ - \hat{\Omega}_{II}^- &= (\hat{\Omega}_I^- - \hat{\Omega}_{II}^-) \frac{1+\epsilon}{J_I + J_{II}} \cdot \hat{J}_I \end{aligned} \quad (6.15)$$

Dabei ist  $\epsilon$  der Stoßkoeffizient, der für die aus Stahl gefertigte Losekonstruktion im Bereich von 0.6 bis 0.8 liegt.

Der Stoßvorgang wird im rekurrenten Netz durch einen Drehmomentimpuls auf die Maschinen nachgebildet. Der Drehmomentimpuls tritt immer dann auf, wenn die Lose eingreift. Da das rekurrente Netz ein zeitdiskretes System beschreibt, ist der Drehmomentimpuls einen Abtastschritt  $h$  lang. Zur Bestimmung des Drehmomentimpulses wird die Differenzengleichung der Maschinenmechanik einer Maschine auf die Form

$$\hat{M}[k] = \frac{\hat{J}}{h} \cdot (\hat{\Omega}[k+1] - \hat{\Omega}[k])$$

gebracht. Durch die Festlegung, dass der Drehmomentimpuls genau einen Abtastschritt lang sein soll, gilt  $\delta\hat{M} = \hat{M}[k]$ ,  $\hat{\Omega}^- = \hat{\Omega}[k]$  und  $\hat{\Omega}^+ = \hat{\Omega}[k+1]$ . Werden

diese Beziehungen in die Differenzengleichung eingesetzt, kann Gleichung (6.15) auf die Form

$$\begin{aligned}\delta\hat{M}_I &= \frac{\hat{J}_I \cdot \hat{J}_{II}}{h} \cdot (\hat{\Omega}_I^- - \hat{\Omega}_{II}^-) \frac{1+\epsilon}{\hat{J}_I + \hat{J}_{II}} \\ \delta\hat{M}_{II} &= \frac{\hat{J}_I \cdot \hat{J}_{II}}{h} \cdot (\hat{\Omega}_I^- - \hat{\Omega}_{II}^-) \frac{1+\epsilon}{\hat{J}_I + \hat{J}_{II}}\end{aligned}$$

gebracht werden. Durch Zusammenfassen ergibt sich

$$\begin{aligned}\delta\hat{M}_I &= \Delta\hat{\Omega}^- \cdot \hat{\Theta}_{S1} \quad \text{mit} \quad \hat{\Theta}_{S1} = \frac{\hat{J}_I \cdot \hat{J}_{II}}{h} \cdot \frac{1+\epsilon}{\hat{J}_I + \hat{J}_{II}} \\ \delta\hat{M}_{II} &= \Delta\hat{\Omega}^- \cdot \hat{\Theta}_{S2} \quad \text{mit} \quad \hat{\Theta}_{S2} = \frac{\hat{J}_I \cdot \hat{J}_{II}}{h} \cdot \frac{1+\epsilon}{\hat{J}_I + \hat{J}_{II}}\end{aligned}\quad (6.16)$$

$\hat{\Theta}_{S1}$  und  $\hat{\Theta}_{S2}$  werden als Stoßparameter bezeichnet und sind nur in dem Abtastschritt, in dem die Lose eingreift, wirksam.

Wie aus Gleichung (6.16) ersichtlich ist, sind bei dieser Betrachtung die Stoßparameter identisch. Tatsächlich sind am Stoßvorgang aber die Massen  $J_I$  und  $J_{II}$  nicht direkt beteiligt. Durch den mechanischen Aufbau der Versuchsanlage wird die Wirkung der Massen durch die elastische Welle gedämpft. Zusätzlich muss beachtet werden, dass die Versuchsanlage asymmetrisch aufgebaut ist. Dies bedeutet, dass der Einfluss des Massenträgheitsmomentes  $J_I$  durch die elastische Verbindung wesentlich stärker gedämpft wird als die Wirkung von  $J_{II}$ . Aus diesem Grund wird bei der Identifikation von zwei unterschiedlichen Stoßparametern ausgegangen.

Durch Zusammenfassen der halben Loseweite und der Stoßparameter ergibt sich der Parametervektor der Losemodellierung zu

$$\hat{\Theta}_{Lose} = [\hat{\Theta}_L \quad \hat{\Theta}_{S1} \quad \hat{\Theta}_{S2}]^T \quad (6.17)$$

Durch die Totzone und den idealisierten Stoßvorgang kann die Lose in der Struktur des rekurrenten Netzes übernommen werden. Die Lose wird damit nicht wie in [221] durch einen statischen Funktionsapproximator, ohne Möglichkeit die Stoßvorgänge zu berücksichtigen, sondern nur durch die Loseweite sowie die beiden Stoßparameter beschrieben.

### Partielle Ableitungen des Loseapproximators

Um den Loseapproximator im strukturierten rekurrenten Netz implementieren zu können, müssen ebenso wie für das GRNN bzw. RBF-Netz die partiellen Ableitungen nach den Gewichten berechnet werden, wobei die Unstetigkeitsstelle in der Kennlinie besonders berücksichtigt werden muss, was zu den folgenden Fallunterscheidungen führt.

Für den Fall  $\hat{w}_i = \hat{\Theta}_L$  ergibt sich

$$\frac{\partial \Delta\hat{\alpha}_{out}}{\partial \hat{w}_i} = \begin{cases} \frac{\partial \Delta\hat{\alpha}_{in}}{\partial \hat{w}_i} - 1 & \Delta\hat{\alpha}_{in} > \hat{\Theta}_L \\ 0 & \text{wenn } -\hat{\Theta}_L \leq \Delta\hat{\alpha}_{in} \leq \hat{\Theta}_L \\ \frac{\partial \Delta\hat{\alpha}_{in}}{\partial \hat{w}_i} + 1 & \Delta\hat{\alpha}_{in} < -\hat{\Theta}_L \end{cases}$$

$$\frac{\partial \delta \hat{M}_I}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_1 \quad \text{und} \quad \frac{\partial \delta \hat{M}_{II}}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_2$$

Für den Fall  $\hat{w}_i = \hat{\Theta}_1$  ergibt sich

$$\frac{\partial \Delta \hat{\alpha}_{out}}{\partial \hat{w}_i} = \begin{cases} \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} > \hat{\Theta}_L \\ 0 & -\hat{\Theta}_L \leq \Delta \hat{\alpha}_{in} \leq \hat{\Theta}_L \\ \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} < -\hat{\Theta}_L \end{cases}$$

$$\frac{\partial \delta \hat{M}_I}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_1 + \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \quad \text{und} \quad \frac{\partial \delta \hat{M}_{II}}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_2$$

Für den Fall  $\hat{w}_i = \hat{\Theta}_2$  ergibt sich

$$\frac{\partial \Delta \hat{\alpha}_{out}}{\partial \hat{w}_i} = \begin{cases} \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} > \hat{\Theta}_L \\ 0 & -\hat{\Theta}_L \leq \Delta \hat{\alpha}_{in} \leq \hat{\Theta}_L \\ \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} < -\hat{\Theta}_L \end{cases}$$

$$\frac{\partial \delta \hat{M}_I}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_1 \quad \text{und} \quad \frac{\partial \delta \hat{M}_{II}}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_2 + \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i}$$

Für den Fall  $\hat{w}_i \notin \hat{\Theta}_{Lose}$  ergibt sich

$$\frac{\partial \Delta \hat{\alpha}_{out}}{\partial \hat{w}_i} = \begin{cases} \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} > \hat{\Theta}_L \\ 0 & -\hat{\Theta}_L \leq \Delta \hat{\alpha}_{in} \leq \hat{\Theta}_L \\ \frac{\partial \Delta \hat{\alpha}_{in}}{\partial \hat{w}_i} & \Delta \hat{\alpha}_{in} < -\hat{\Theta}_L \end{cases}$$

$$\frac{\partial \delta \hat{M}_I}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_1 \quad \text{und} \quad \frac{\partial \delta \hat{M}_{II}}{\partial \hat{w}_i} = \frac{\partial \Delta \hat{\Omega}^-}{\partial \hat{w}_i} \cdot \hat{\Theta}_2$$

#### 6.4.2 Approximation der Reibungskennlinie

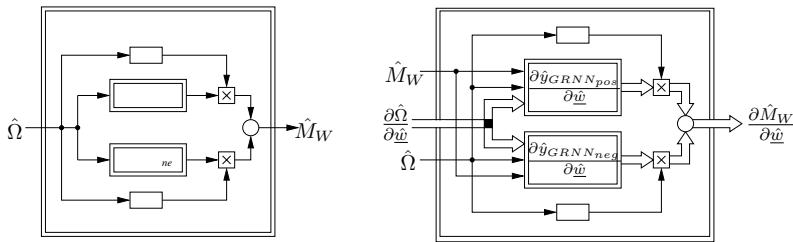
Aufgrund der Haftreibung ergibt sich in der Reibungskennlinie bei der Winkelgeschwindigkeit  $\Omega = 0$  eine Unstetigkeit. Mit dem in Abschnitt 3.7 beschriebenen GRNN ist es nicht möglich, eine Unstetigkeit zu approximieren. Daher werden bei der Identifikation der Reibungskennlinien zwei GRNN eingesetzt, wobei eines den positiven und das andere den negativen Ast der Reibungskennlinie approximiert. Die Zusammenführung der beiden Netze ist in Abbildung 6.17 dargestellt.

Der Gewichtsvektor des Funktionsapproximators zur Nachbildung der Reibungskennlinie  $\hat{\Theta}_{Reib}$  setzt sich entsprechend aus den Gewichten der GRNN zusammen. Es gilt

$$\hat{\Theta}_{Reib} = \begin{bmatrix} \hat{\Theta}_{GRNN, pos} \\ \hat{\Theta}_{GRNN, neg} \end{bmatrix} \quad (6.18)$$

Mit diesem Funktionsapproximator kann eine Funktion mit einem Sprung beim Eingangswert Null angenähert werden.

Bei der Berechnung der partiellen Ableitungen zur Auswertung des Lerngesetzes aus Gleichung (6.3) wird wie bei einem herkömmlichen GRNN gemäß den Gleichungen aus Tabelle 6.1 verfahren, wobei die Aktivierungen für den inaktiven Kennlinienast 0 sind. Zur Veranschaulichung ist die Berechnung der partiellen Ableitungen in Abbildung 6.17 dargestellt.



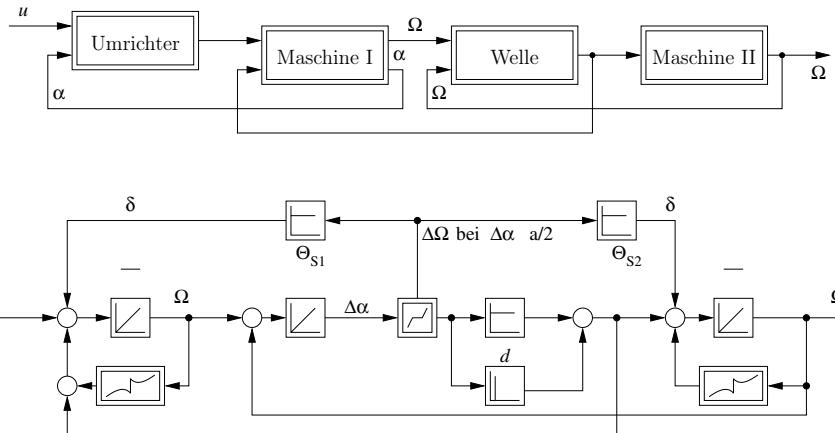
**Abb. 6.17:** Verbindung von zwei GRNN zur Approximation der unstetigen Reibungskennlinie und Berechnung der partiellen Ableitungen

Für die Implementierung muss zusätzlich das Verhalten bei  $\hat{\Omega} = 0$  definiert werden. Ist die Winkelgeschwindigkeit  $\hat{\Omega} = 0$ , bleibt der Zustand des Haftens so lange erhalten, bis das positive Haftdrehmoment überschritten, respektive das negative unterschritten wird. Erst wenn diese Bedingung erfüllt ist, kann sich wieder eine von 0 abweichende Winkelgeschwindigkeit ergeben und die Maschine geht in den gleitenden Zustand über. Umgekehrt kann die Maschine nur in den haftenden Zustand übergehen, wenn die Winkelgeschwindigkeit zu Null wird und das Drehmoment zu diesem Zeitpunkt innerhalb der Haftriebungsgrenzen liegt. Ansonsten bleibt der gleitende Zustand erhalten. Entsprechend gilt für den Zustand Haften, dass alle Aktivierungen und damit die partiellen Ableitungen 0 sind.

#### 6.4.3 Identifikation des losebehafteten Zweimassensystems

Die Struktur des Systems, sowie der kontinuierliche Signalflussplan ist in Abbildung 6.18 abgebildet.

Zur Beschreibung des Zweimassensystems sind die Lagedifferenz  $\Delta\alpha$  (Welle als Torsionsfeder  $M_C = \Delta\alpha \cdot c$ ) und die Winkelgeschwindigkeitsdifferenz  $\Delta\Omega$  (Dämpfung der Welle:  $M_D = \Delta\Omega \cdot d$ ) nach der Lose notwendig. Da der Ausgang der Losekennlinie die Lagedifferenz ist, muss die Winkelgeschwindigkeitsdifferenz durch Differentiation der Lagedifferenz gebildet werden. Aus diesem Grund muss



**Abb. 6.18:** Struktur des Systems (oben) und Signalflussplan des Zweimassensystems unter Vernachlässigung des Umrichters (unten)

im Signalflussplan des Zweimassensystems zur Beschreibung der elastischen Verbindung mit Lose ein Differentiationsglied enthalten sein.

In der kontinuierlichen Zustandsbeschreibung (vgl. Gleichung (6.19)) wird die Differentiation im Vektor der Nichtlinearitäten  $\underline{\mathcal{N}}$  berücksichtigt. Durch das Differentiationsglied selbst wird aber kein unabhängiger Zustand beschrieben.

Im rekurrenten Netz wird der Differentiationsblock durch die numerische Differentiation nach Euler ersetzt. Die numerische Differentiation beschreibt aber eine unabhängige Differenzengleichung erster Ordnung (Verzögerungsneuron). Daher ist in der diskreten Zustandsbeschreibung des rekurrenten Netzes eine zusätzliche Gleichung enthalten (vgl. Gleichung (6.20)).

Wird bei der Beobachterdimensionierung von einem kontinuierlichen Beobachter ausgegangen, werden nur die vier unabhängigen Zustände  $(\hat{\Omega}_1, \hat{\alpha}_1, \hat{\Omega}_2, \hat{\alpha}_2)$  mit je einer Zustandsrückführung beaufschlagt.

Die Stellgröße des Zweimassensystems ist das Motormoment der ersten Maschine  $M_I$ .

Damit können die Zustandsbeschreibungen der Strecke und des rekurrenten Netzes angegeben werden.

Die kontinuierliche Zustandsbeschreibung des Systems wird vom Signalflussplan aus Abbildung 6.18 abgeleitet. Dazu werden die Größen

$$\begin{aligned} u &= M_I \\ y &= \Omega_I \\ \underline{x} &= [x_1 \ x_2 \ x_3]^T = [\Omega_I \ \Delta\alpha \ \Omega_{II}]^T \end{aligned}$$

festgelegt, womit die Zustandsgleichungen

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underline{x} + \begin{bmatrix} \frac{1}{J_I} \\ 0 \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} -\frac{c}{J_I} & -\frac{c}{J_I} & -\frac{d}{J_I} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{c}{J_{II}} & \frac{d}{J_{II}} & -\frac{1}{J_{II}} \end{bmatrix} \cdot \begin{bmatrix} \mathcal{R}_I(x_1) \\ \mathcal{L}(x_2) \\ \dot{\mathcal{L}}(x_2) \\ \mathcal{R}_{II}(x_3) \end{bmatrix} \quad (6.19)$$

und die Ausgangsgleichung

$$y = [1 \ 0 \ 0] \cdot \underline{x}$$

des Systems angegeben werden können.

Hierbei gelten die in Tabelle 6.3 zusammengefassten Bezeichnungen. Zusätzlich bezeichnet  $\mathcal{L}$  die im System vorhandene Lose.

Beschreibung	Symbol	Einheit	Netzwerk Gewicht
Trägheitsmoment Maschine I	$J_I$	$\text{kg m}^2$	${}^1/\hat{\Psi}_1$
Dämpfungskonstante der Welle	$d$	$\text{Nm s/rad}$	$\hat{\Psi}_2$
Federkonstante der Welle	$c$	$\text{Nm/rad}$	$\hat{\Psi}_3$
Trägheitsmoment Maschine II	$J_{II}$	$\text{kg m}^2$	${}^1/\hat{\Psi}_4$
Reibungskennlinie I	$\mathcal{R}_I$	Nm	$\hat{\Theta}_{Reib,I}$
Reibungskennlinie II	$\mathcal{R}_{II}$	Nm	$\hat{\Theta}_{Reib,II}$
Losecharakteristik	$\mathcal{L}$	rad	$\hat{\Theta}_{Lose}$
Winkelgeschwindigkeit Maschine I	$\Omega_I$	$\text{rad/s}$	
Winkelgeschwindigkeit Maschine II	$\Omega_{II}$	$\text{rad/s}$	
Position Maschine I	$\alpha_I$	rad	
Differenzwinkel	$\Delta\alpha$	rad	
Antriebsmoment Maschine I	$M_I$	Nm	
Antriebsmoment Maschine II	$M_{II}$	Nm	
Sollmoment Umrichter	$M_I^*$	Nm	

**Tabelle 6.3:** Übersicht über die verwendeten Bezeichnungen

Die diskrete Zustandsbeschreibung des rekurrenten Netzes ergibt sich aus der kontinuierlichen Zustandsbeschreibung (6.19). Dazu werden die Größen

$$u[k] = M_I[k]$$

$$\hat{y}[k] = \hat{\Omega}_I[k]$$

$$\hat{x}[k] = [\hat{\Omega}_I[k] \ \Delta\hat{\alpha}[k] \ \hat{x}_3[k] \ \hat{\Omega}_{II}[k]]^T$$

festgelegt. Im rekurrenten Netz werden die Reibungen mit Hilfe von GRNN, sowie die Lose mit dem in Abschnitt 6.4.1 eingeführten Loseapproximator nachgebildet. Somit ergibt sich der Parametervektor des rekurrenten Netzes zu

$$\hat{w} = \begin{bmatrix} \hat{\Psi}_1 & \hat{\Psi}_2 & \hat{\Psi}_3 & \hat{\Psi}_4 & \hat{\Theta}_{Reib,I}^T & \hat{\Theta}_{Lose}^T & \hat{\Theta}_{Reib,II}^T \end{bmatrix}^T$$

mit den linearen Parametern  $\hat{\Psi}_1 = \frac{1}{\tilde{J}_I}$ ,  $\hat{\Psi}_2 = \hat{d}$ ,  $\hat{\Psi}_3 = \hat{c}$  und  $\hat{\Psi}_4 = \frac{1}{\tilde{J}_{II}}$  sowie den Stützwerten für die Reibung  $\hat{\Theta}_{Reib,I}$  und  $\hat{\Theta}_{Reib,II}$  sowie den Parametern für die Loseapproximation  $\hat{\Theta}_{Lose}$ .

Der Vektor mit den Beobachterkoeffizienten wird wie folgt eingeführt

$$\tilde{l} = [\tilde{l}_1 \ \tilde{l}_2 \ 0 \ \tilde{l}_3]^T$$

Wie bereits erwähnt, führt die Differentiation in der zeitdiskreten Darstellung auf eine zusätzliche Gleichung. Da diese zusätzliche Gleichung keinem unabhängigen Zustand in der zeitkontinuierlichen Darstellung entspricht, wird der interne Zustand des rekurrenten Netzes  $\hat{x}_3[k]$  nicht über einen Beobachterkoeffizient zurückgeführt. Aus diesem Grund ist der dritte Eintrag im Beobachtervektor  $\tilde{l}$  mit einer Null belegt.

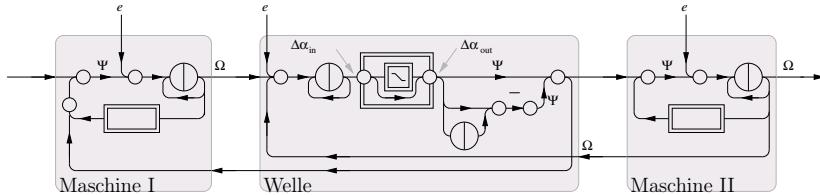
Da die numerische Differentiation einen diskreten Zustand beinhaltet, kann im Vektor der Nichtlinearitäten die Differentiation der Losenkenmlinie  $\dot{\mathcal{L}}$  entfallen. Mit diesen Festlegungen ergibt sich die diskrete Zustandsdarstellung sowie die Ausgangsgleichung des verwendeten rekurrenten Netzes zu

$$\begin{aligned} \hat{x}[k+1] = & \begin{bmatrix} h\tilde{l}_1 + 1 & -h\hat{\Psi}_1\hat{\Psi}_3 - \hat{\Psi}_1\hat{\Psi}_2 & \hat{\Psi}_1\hat{\Psi}_2 & 0 \\ h(\tilde{l}_2 + 1) & 1 & 0 & -h \\ 0 & 1 & 0 & 0 \\ h\tilde{l}_3 & h\hat{\Psi}_3\hat{\Psi}_4 + \hat{\Psi}_2\hat{\Psi}_4 & -\hat{\Psi}_2\hat{\Psi}_4 & 1 \end{bmatrix} \cdot \hat{x}[k] + \\ & + \begin{bmatrix} h\hat{\Psi}_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot u[k] - \begin{bmatrix} h\tilde{l}_1 \\ h\tilde{l}_2 \\ 0 \\ h\tilde{l}_3 \end{bmatrix} \cdot y[k] + \\ & + \begin{bmatrix} -h\hat{\Psi}_1 - h\hat{\Psi}_1\hat{\Psi}_3 + \hat{\Psi}_1\hat{\Psi}_2 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & h\hat{\Psi}_3\hat{\Psi}_4 - \hat{\Psi}_2\hat{\Psi}_4 & -h\hat{\Psi}_4 \end{bmatrix} \cdot \begin{bmatrix} \hat{y}_{Reib,I}(\hat{x}_1[k]) \\ \Delta\hat{\alpha}_{out}(\hat{x}_2[k]) \\ \hat{y}_{Reib,II}(\hat{x}_4[k]) \end{bmatrix} \quad (6.20) \end{aligned}$$

$$\hat{y}[k] = [1 \ 0 \ 0 \ 0] \cdot \hat{x}[k]$$

Mit Hilfe der diskreten Zustandsdarstellung (6.20) können die partiellen Ableitungen  $\nabla \hat{y}(\hat{w})$  gemäß Gleichungen (6.11) und (6.12) berechnet werden. Das Lerngesetz ergibt sich wiederum aus Gleichung (6.3).

Die Struktur des rekurrenten Netzes ist noch einmal in Abbildung 6.19 dargestellt.



**Abb. 6.19:** Strukturiertes rekurrentes Netz für das losebehaftete Zweimassensystem

#### 6.4.4 Identifikation

Die Identifikation wird in einer Simulationsumgebung unter idealisierten Bedingungen durchgeführt. Das heißt, die Struktur der Strecke und des rekurrenten Netzes stimmen exakt überein. Die Anregung des Systems erfolgt mit einer Zwei-Punkt-Regelung ( $\pm 10 \text{ N m}$ ). Die Abtastzeit wird mit  $h = 0.4 \text{ ms}$  festgelegt. Gelernt werden hierbei die Parameter und Nichtlinearitäten des Zwei-Massen-Systems.

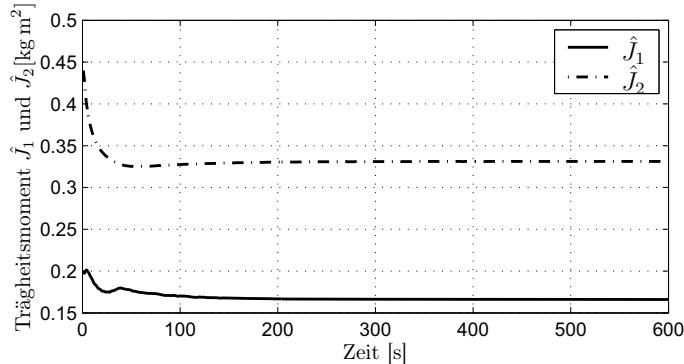
Alle für die Identifikation relevanten Parameter und Ergebnisse sind in Tabelle 6.4 zusammengefasst. In den Abbildungen 6.20 bis 6.27 sind die Zeitverläufe während der Identifikation und die identifizierten Nichtlinearitäten dargestellt.

Parameter	Strecke	Startwert	Ergebnis	$\eta$	$\alpha$	SFA
$J_I$ Reibung I	$0.166 \text{ kg m}^2$ Abb. 6.25	$0.2 \text{ kg m}^2$ 0	$0.166 \text{ kg m}^2$ Abb. 6.25	0.15 0.125	0.95 0.95	$\sigma_{\text{norm}} = 1.6$ $r_{\text{Reib},I} = 2 \cdot 15$
$d$	$0.6 \text{ N m s/rad}$	$1 \text{ N m s/rad}$	$0.6 \text{ N m s/rad}$	1.25	0.95	—
$c$	$1160 \text{ N m/rad}$	$1350 \text{ N m/rad}$	$1160 \text{ N m/rad}$	$5 \cdot 10^4$	0.95	—
$a$	$0.031 \text{ rad}$	$0 \text{ rad}$	$0.031 \text{ rad}$	$8 \cdot 10^{-6}$	0.95	—
$\Theta_{S1}$	$2 \text{ N m s/rad}$	$0 \text{ N m s/rad}$	$2 \text{ N m s/rad}$	5	0.95	—
$\Theta_{S2}$	$38 \text{ N m s/rad}$	$0 \text{ N m s/rad}$	$38 \text{ N m s/rad}$	10	0.95	—
$J_{II}$ Reibung II	$0.336 \text{ kg m}^2$ Abb. 6.25	$0.45 \text{ kg m}^2$ 0	$0.336 \text{ kg m}^2$ Abb. 6.25	$4 \cdot 10^{-3}$ 0.1	0.95 0.95	$\sigma_{\text{norm}} = 1.6$ $r_{\text{Reib},II} = 2 \cdot 15$
Beobachter	$\tilde{l} = [ 3.3 \quad 2.6 \quad -1197 \quad 98 \quad 227 ]$			—	—	—

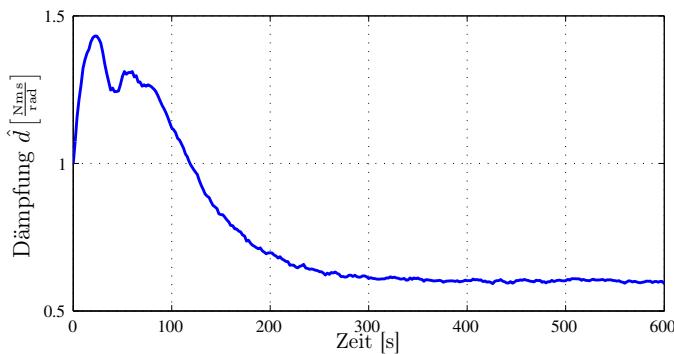
**Tabelle 6.4:** Parameter und Ergebnisse der simulativen Identifikation des Zweimassensystems, SFA: Statischer Funktionsapproximator

Wie Tabelle 6.4 und den Abbildungen 6.20 bis 6.24 zu entnehmen ist, werden sämtliche linearen Parameter sowie die Parameter des Loseapproximators exakt

identifiziert. Ebenso können die Reibungskennlinien 6.25 in der Simulationsumgebung exakt identifiziert werden.



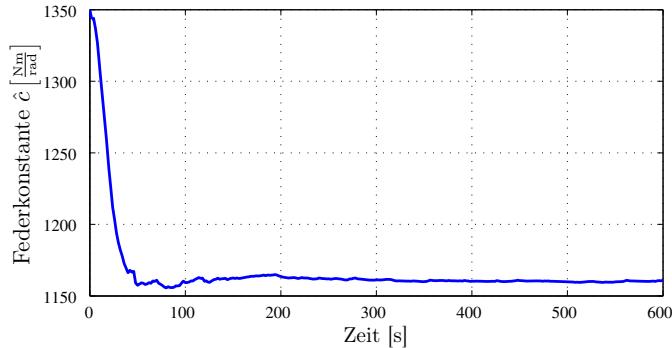
**Abb. 6.20:** Identifikationsverlauf der Maschinenparameter  $\hat{J}_I$  und  $\hat{J}_{II}$



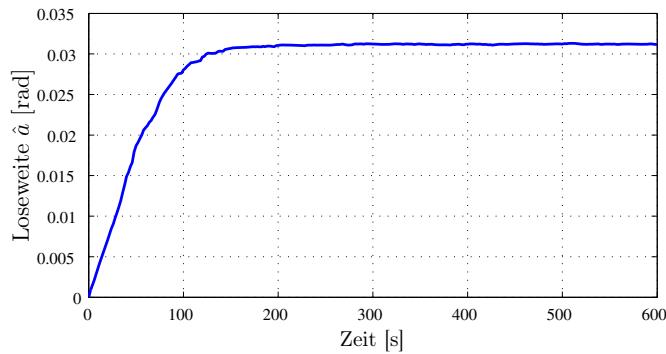
**Abb. 6.21:** Identifikationsverlauf des Dämpfungsmaßes  $\hat{d}$

In Abbildung ist exemplarisch die Konvergenz der Stützwerte  $\hat{\Theta}_{Reib,I}$  dargestellt. In Abbildung 6.27 ist zu erkennen, dass mit fortlaufender Identifikation der Ausgangsfehler  $e$  gegen Null strebt.

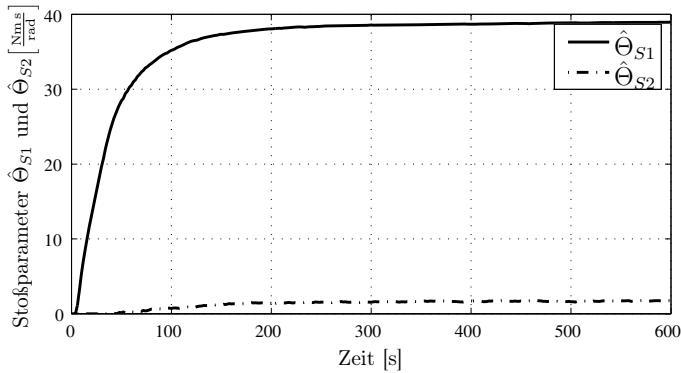
Die vorgestellte Identifikation eines losebehafteten Zweimassensystems wurde ebenfalls an einer Versuchsanlage des Lehrstuhls für elektrische Antriebssysteme durchgeführt, und lieferte sehr gute Identifikationsergebnisse [5, 90, 88].



**Abb. 6.22:** Identifikationsverlauf der Federsteifigkeit  $\hat{c}$



**Abb. 6.23:** Identifikationsverlauf der Loseweite  $\hat{a}$



**Abb. 6.24:** Identifikationsverlauf der Stoßparameter  $\hat{\Theta}_{S1}$  und  $\hat{\Theta}_{S2}$

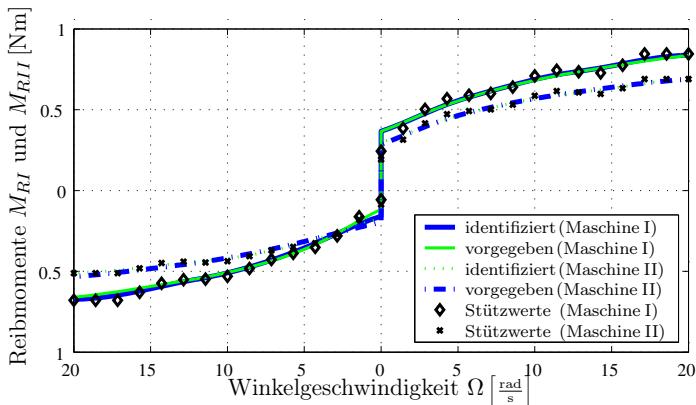


Abb. 6.25: Identifikationsergebnis der Reibungskennlinien

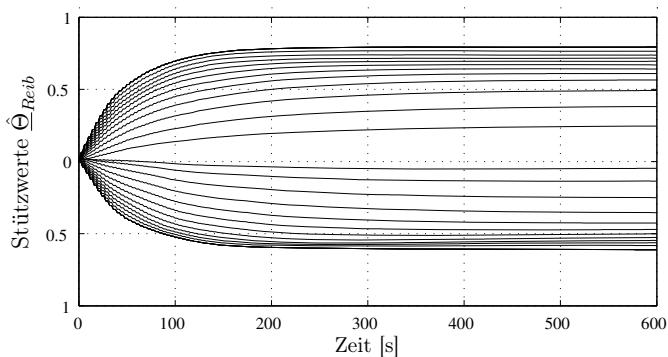


Abb. 6.26: Identifikationsverlauf der Stützwerte  $\hat{\Theta}_{Reib,I}$  der Reibungskennlinie

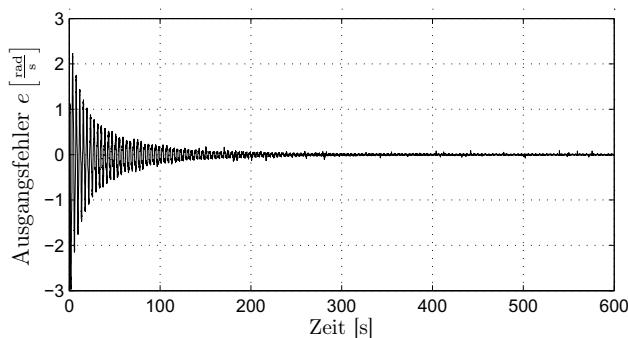


Abb. 6.27: Verlauf des Ausgangsfehlers  $e$  während der Identifikation

# 7 Identifikation linearer dynamischer Systeme

Die Identifikation linearer dynamischer Systeme ist ein umfassend erforschtes Gebiet. In diesem Kapitel sollen die wichtigsten linearen Modellstrukturen und Identifikationsverfahren vorgestellt werden. Es werden prinzipielle Unterschiede zwischen den Identifikationsverfahren erläutert und an Simulationsbeispielen veranschaulicht. Aufbauend auf dem Verständnis der Identifikation linearer dynamischer Systeme wird in Kapitel 8 die Identifikation nichtlinearer dynamischer System behandelt.

Detaillierte Ausführungen zur Identifikation linearer dynamischer Systeme können z.B. in [141, 168, 216, 230] gefunden werden.

Die Identifikation von linearen und insbesondere von nichtlinearen Systemen (Kapitel 6 und 8) ist eine wichtige Voraussetzung, um aussagekräftige Modelle des betrachteten Systems zu erhalten. Mit diesen aussagekräftigen Modellen können dann Simulationen durchgeführt werden, um eine gezielte Analyse der Aufgabenstellung im Vorfeld praktischer Untersuchungen durchzuführen und damit aufwendige und teure praktische Experimente möglichst zu minimieren oder sogar ganz zu vermeiden. Insofern ist eine physikalisch interpretierbare und konvergente Identifikation von größter Bedeutung.

## 7.1 Grundlagen der Identifikation

### 7.1.1 Parametrische und nichtparametrische Identifikationsverfahren

Die Verfahren zur Identifikation linearer dynamischer Systeme können in parametrische und nichtparametrische Ansätze unterteilt werden. Des Weiteren ist es hilfreich, zwischen dem Modell und dem Verfahren, das zur Ermittlung der Freiheitsgrade des Modells zum Einsatz kommt, zu unterscheiden. Parametrische und nichtparametrische Modelle können wie folgt voneinander abgegrenzt werden:

- Parametrische Modelle beschreiben das Systemverhalten exakt mit einer endlichen Anzahl an Parametern. Ein typisches Beispiel hierfür ist ein Modell, das auf einer Differentialgleichung bzw. einer Differenzengleichung basiert. Die Parameter haben in der Regel einen direkten Bezug zu physikalischen Größen wie z.B. einer Masse oder einem Trägheitsmoment.

- Nichtparametrische Modelle benötigen eine unendliche Anzahl an Parametern um das Systemverhalten exakt zu beschreiben. Ein typisches Beispiel ist ein Modell, das auf der Impulsantwort des Systems basiert.

Desweiteren können auch die Identifikationsverfahren in parametrische und nichtparametrische Methoden unterteilt werden:

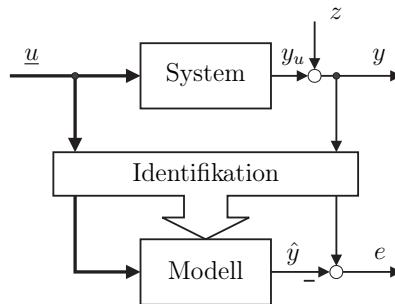
- Parametrische Methoden bestimmen eine endliche Anzahl von Parametern. Parametrische Methoden können auch dazu verwendet werden, um die Parameter eines nichtparametrischen Modells zu bestimmen, wenn die Parameter vorher auf eine endliche Anzahl reduziert wurden. Ein typisches Beispiel ist das FIR-Modell (finite impulse response), das die unendliche lange Impulsantwort eines Systems nachbildet.
- Nichtparametrische Methoden sind flexibler als parametrische Methoden. Sie werden meistens verwendet, wenn die Struktur des Systems nur unzureichend bekannt ist. Ein typisches Beispiel ist die Fourieranalyse, bei der eine endliche Parameteranzahl im allgemeinen nicht ausreicht. Nichtparametrische Methoden benötigen theoretisch unendlich viele Parameter. Bei einer Implementierung können aber nur endlich viele Parameter — das Systemverhalten kann in diesem Fall nicht mehr exakt beschrieben werden — berücksichtigt werden. Diese endliche Anzahl der Parameter ist aber wesentlich größer als bei parametrischen Methoden.

Neben der Unterscheidung von parametrischen und nichtparametrischen Methoden können die Identifikationsverfahren auch nach Zeitbereich und Frequenzbereich unterschieden werden. Im Folgenden sollen jedoch ausschließlich parametrische Identifikationsverfahren im Zeitbereich betrachtet werden, da ihnen die größte Bedeutung zukommt. Bei den vorgestellten Verfahren handelt es sich ausschließlich um Methoden im zeitdiskreten Bereich.

### 7.1.2 Identifikation

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  $u$ , 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 7.1 zeigt die grundsätzliche Struktur einer Identifikation.

Jede Identifikation setzt sich aus zwei grundlegenden Schritten zusammen:



**Abb. 7.1:** Prinzipielle Struktur einer Identifikation

1. Strukturauswahl bzw. Strukturbestimmung

2. Adaption der Parameter

Zunächst muss für das Modell eine Struktur festgelegt bzw. bestimmt [117, 127] 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.

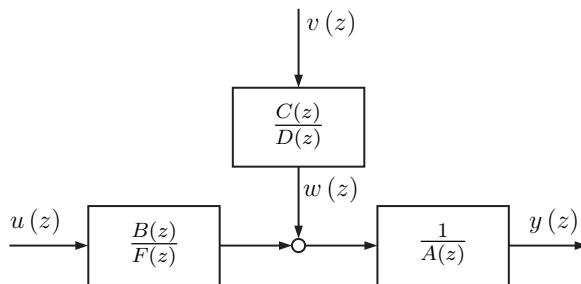
Im zweiten Schritt müssen die Parameter der gewählten Modellstruktur so angepasst werden, dass der Fehler  $e$  zwischen realem Prozess und Modell minimiert wird. Dieser Teil wird deshalb als Parameteradaption bezeichnet und wurde in Kapitel 4 schon genauer erläutert. Prinzipiell besteht bei der Parameteradaption von Neuronalen Netzen und linearen dynamischen Modellstrukturen kein Unterschied, da es sich um ein mathematisches Optimierungsproblem handelt, dessen Lösung ausschließlich von der Form des Gleichungssystems abhängt bzw. ob die unbekannten Parameter linear oder nichtlinear in den Ausgang eingehen.

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 [131]. 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. In der Literatur werden stochastische Signale wie z.B. das amplitudenmodulierte Pseudo-Rausch-Binärsignal [111, 112, 168] als besonders geeignet angesehen, da sie viele unterschiedliche Frequenzen und Amplituden beinhalten. Mit derartigen Eingangssignalen wird das Auffinden des globalen Minimums des Gütfunktions begünstigt. Auf die Wahl eines geeigneten Anregesignals wird in Kapitel 8 genauer eingegangen.

## 7.2 Lineare dynamische Modellstrukturen

In diesem Kapitel sollen, ausgehend von unterschiedlichen Modellstrukturen, verschiedene Ansätze zur Identifikation linearer dynamischer Systeme vorgestellt werden. Ziel ist es, ein grundlegendes Verständnis für die Problemstellungen zu schaffen, die bei der Identifikation dynamischer Systeme auftreten können.

Ausgangspunkt für die Identifikation von linearen dynamischen Systemen ist eine allgemeine modellhafte Vorstellung des linearen (zeitdiskreten) Systems, wie sie in Abb. 7.2 dargestellt ist.



**Abb. 7.2:** Allgemeine lineare Modellstruktur

Dieses Modell besteht aus einem deterministischen und einem stochastischen Anteil, wie in Gl. (7.1) beschrieben:

$$y(z) = \underbrace{\frac{B(z)}{F(z) \cdot A(z)}}_{\text{Eingangs-\\übertragungsfunktion}} \cdot u(z) + \underbrace{\frac{C(z)}{D(z) \cdot A(z)}}_{\text{Rausch-\\übertragungsfunktion}} \cdot v(z) \quad (7.1)$$

mit

$$A(z) = 1 + a_1 \cdot z^{-1} + \dots + a_{na} \cdot z^{-na} \quad (7.2)$$

$$B(z) = b_0 + b_1 \cdot z^{-1} + \dots + b_{nb} \cdot z^{-nb} \quad (7.3)$$

$$C(z) = 1 + c_1 \cdot z^{-1} + \dots + c_{nc} \cdot z^{-nc} \quad (7.4)$$

$$D(z) = 1 + d_1 \cdot z^{-1} + \dots + d_{nd} \cdot z^{-nd} \quad (7.5)$$

$$F(z) = 1 + f_1 \cdot z^{-1} + \dots + f_{nf} \cdot z^{-nf} \quad (7.6)$$

In Gl. (7.1) 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. 7.2 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 7.1 zeigt die wichtigsten Modellstrukturen.

Das MA–Modell (Moving Average) gehört ebenso wie das AR–Modell (Auto

Modellstruktur		Modellgleichung
MA	(Moving Average)	$y(z) = C(z) \cdot v(z)$
AR	(Auto Regressive)	$y(z) = \frac{1}{D(z)} \cdot v(z)$
ARMA	(Auto Regressive Moving Average)	$y(z) = \frac{C(z)}{D(z)} \cdot v(z)$
ARX	(Auto Regressive with eXogenous input)	$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z)$
ARMAX	(Auto Regressive Moving Average with eXogenous input)	$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{C(z)}{A(z)} \cdot v(z)$
OE	(Output Error)	$y(z) = \frac{B(z)}{F(z)} \cdot u(z) + v(z)$
FIR	(Finite Impulse Response)	$y(z) = B(z) \cdot u(z) + v(z)$

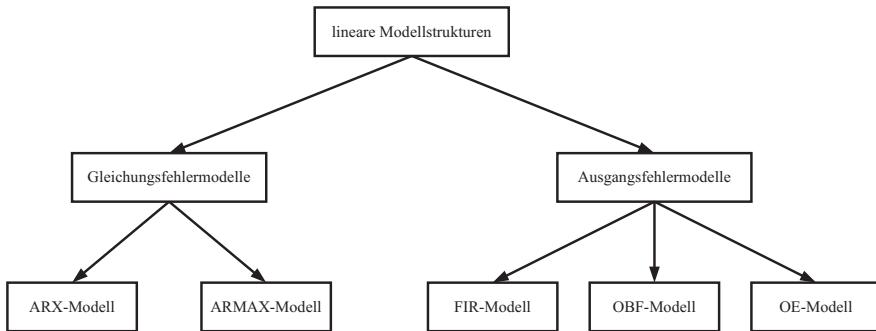
(abweichende Notation bei Goodwin, Narendra)

**Tabelle 7.1:** Lineare Modellstrukturen im Überblick

Regressive) und das ARMA–Modell (Auto Regressive Moving Average) zur Klasse der Zeitreihenmodelle. Ihnen ist gemeinsam, dass sie keinen deterministischen Anteil berücksichtigen. Sie kommen vor allem bei ökonomischen Problemstellungen zum Einsatz, wie z.B. der Vorhersage von Aktien– oder Wechselkursen, wo die deterministischen Einflussgrößen nur schwer zu bestimmen sind bzw. deren Anzahl sehr groß ist. Bei technischen Problemstellungen sind die deterministischen Einflussgrößen meistens sehr gut bekannt, so dass es sinnvoller ist ein Modell zu verwenden, wo diese auch berücksichtigt werden. Aus diesem Grund wird im Folgenden auf die Zeitreihenmodelle nicht weiter eingegangen.

Das ARX–Modell (Auto Regressive with eXogenous input) besitzt, ebenso wie das ARMAX–Modell (Auto Regressive 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. 7.3 ist diese Art der Klassifikation noch einmal verdeutlicht.



**Abb. 7.3:** Klassifikation der linearen Modellstrukturen

In Abb. 7.3 wurde zusätzlich das OBF–Modell (Orthonormal Basis Function) aufgeführt, das eine Erweiterung des FIR–Modells darstellt und in Kapitel 7.2.2.2 behandelt wird.

### 7.2.1 Modelle mit Ausgangsrückkopplung

In diesem Kapitel werden lineare Modelle mit Ausgangsrückkopplung, (z.B. ARX–, ARMAX– und OE–Modell) genauer betrachtet. Am Beispiel des ARX– und des OE–Modells soll der Unterschied zwischen einem Gleichungsfehler– und einem Ausgangsfehlermodell erläutert werden.

#### 7.2.1.1 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 Lernverfahren (vgl. Kapitel 4) eingesetzt werden können. In Abb. 7.4 ist die Gleichungsfehlerstruktur des ARX–Modells dargestellt.

Der ARX–Systemansatz ist definiert durch die Gleichung:

$$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z) \quad (7.7)$$

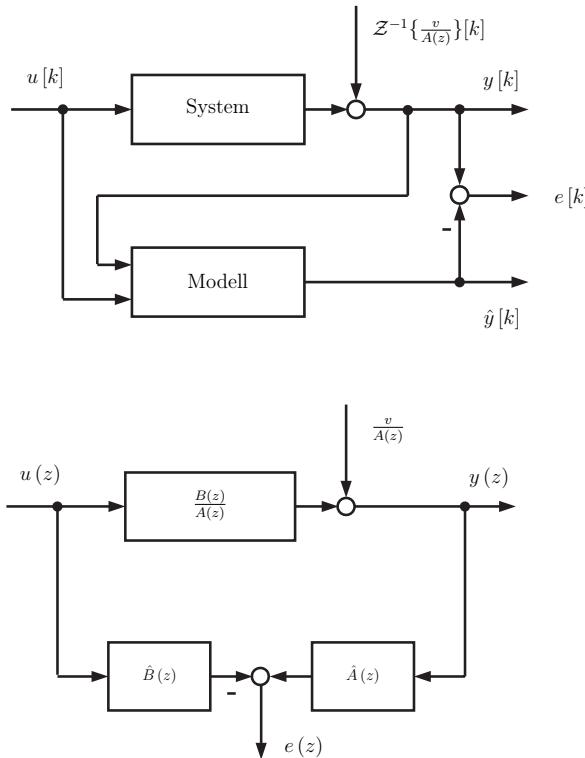


Abb. 7.4: Gleichungsfehlerstruktur des ARX-Modells

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

$$\begin{aligned} e(z) &= y(z) - \hat{y}(z) \\ &\stackrel{!}{=} v(z) \end{aligned} \quad (7.8)$$

Diese Forderung erscheint aufgrund der Filterung  $A(z)$  des weißen Rauschens  $v(z)$  nicht sinnvoll und soll daher überprüft werden:

$$e(z) = y(z) - \hat{y}(z) \quad (7.9)$$

$$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z) \quad (7.10)$$

Modellansatz (7.23)

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) + [1 - \hat{A}(z)] \cdot y(z) \quad (7.11)$$

$$e(z) = y(z) - \hat{B}(z) \cdot u(z) - y(z) + \hat{A}(z) \cdot y(z) \quad (7.12)$$

$$= -\hat{B}(z) \cdot u(z) + \hat{A}(z) \cdot y(z) \text{ Identifikationsgleichung} \quad (7.13)$$

jetzt  $y(z)$  eingesetzt

$$= -\hat{B}(z) \cdot u(z) + \hat{A}(z) \cdot \left[ \frac{B(z)}{A(z)} \cdot u(z) + \frac{1}{A(z)} \cdot v(z) \right] \quad (7.14)$$

$$A(z) = \hat{A}(z) \text{ und } B(z) = \hat{B}(z) \quad (7.15)$$

$$e(z) = v(z) \text{ q.e.d.} \quad (7.16)$$

$$(7.17)$$

Für das optimale Identifikationsergebnis  $e(z) = v(z)$  gilt  $\hat{A}(z) = A(z)$  und  $\hat{B}(z) = B(z)$ , womit für den Systemansatz

$$y(z) = \frac{\hat{B}(z)}{\hat{A}(z)} \cdot u(z) + \frac{1}{\hat{A}(z)} \cdot v(z) \quad (7.18)$$

geschrieben werden kann. Wird in diesen Ansatz die Bedingung für die optimale Identifikationsgleichung eingesetzt, ergibt sich

$$y(z) = \frac{\hat{B}(z)}{\hat{A}(z)} \cdot u(z) + \frac{1}{\hat{A}(z)} \cdot \underbrace{(y(z) - \hat{y}(z))}_{e(z) \stackrel{!}{=} v(z)} \quad (7.19)$$

Daraus kann für den Modellansatz (vgl. rechte Seite der Abbildung 7.4) die Gleichung

$$e(z) = -\hat{B}(z) \cdot u(z) + \hat{A}(z) \cdot y(z) \quad (7.20)$$

hergeleitet werden. Der Modellansatz kann auch in einen expliziten Ausdruck für den Modellausgang  $\hat{y}(z)$  umgeformt werden. Ausgehend von Gleichung (7.19) ergibt sich durch elementare Umformungen:

$$\hat{A}(z) \cdot y(z) = \hat{B}(z) \cdot u(z) + y(z) - \hat{y}(z) \quad (7.21)$$

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) + y(z) - \hat{A}(z) \cdot y(z) \quad (7.22)$$

Der Modellansatz lautet damit:

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) + [1 - \hat{A}(z)] \cdot y(z) \quad (7.23)$$

Wird Gl. (7.23) in eine Differenzengleichung umgewandelt, ergibt sich<sup>1)</sup>:

---

<sup>1)</sup> In Gl. (7.24) wird angenommen, dass der Prozess nicht sprungfähig ist und deswegen der Parameter  $b_0$  vernachlässigt werden kann.

$$\hat{y}[k] = b_1 \cdot u[k-1] + \dots + b_{nb} \cdot u[k-nb] - a_1 \cdot y[k-1] - \dots - a_{na} \cdot y[k-na] \quad (7.24)$$

Aus Gl. (7.24) wird deutlich, dass die unbekannten Gewichte  $b_1 \dots b_{nb}$  und  $a_1 \dots a_{na}$  linear in den Ausgang  $\hat{y}[k]$  eingehen. Außerdem ist die ARX-Identifikationsstruktur garantiert stabil, da die Identifikationsgleichung (7.20) keine Rückkopplungen enthält. Der Nachteil dieser Modellstruktur ist, dass mit der ARX-Identifikationsgleichung kein paralleles Modell, sondern ein seriell-paralleles Modell (Abbildung 7.4) bestimmt wird.

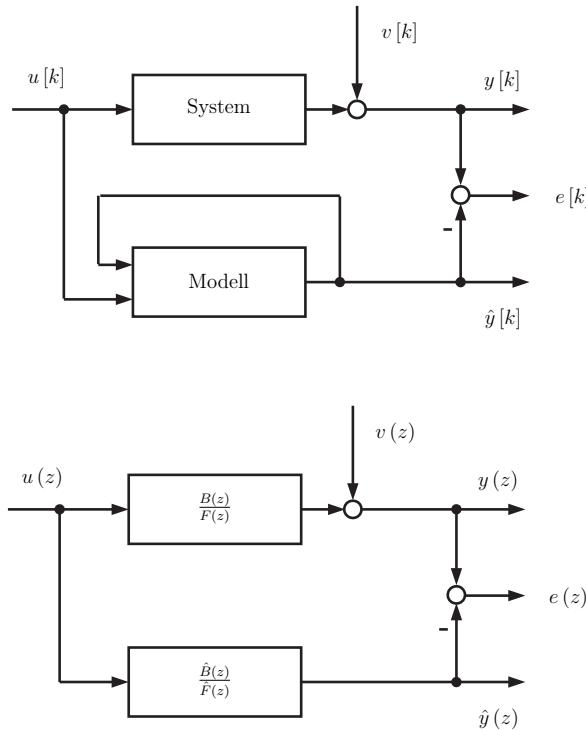
Beim ARX-Modell wird der sog. Gleichungsfehler (oder auch 1-Schritt-Prädiktionsfehler) minimiert. Die Bezeichnung Gleichungsfehler oder 1-Schritt-Prädiktionsfehler wird verwendet, weil der Identifikationsalgorithmus das aktuelle Ausgangssignal nicht eigenständig, sondern nur mit Hilfe der zuletzt gemessenen Ausgangssignale schätzt. Somit wird vom ARX-Modell nur eine 1-Schritt-Prädiktion zum neuen Ausgangssignal ausgeführt.

Wird der Gleichungsfehler minimiert, entsteht wie bereits erwähnt ein seriell-paralleles Modell. Bei genauer Betrachtung der Gleichungsfehlerstruktur nach Abb. 7.4 ist zu erkennen, dass System und Modell bezüglich des Eingangs  $u[k]$  parallel, aber bezüglich des Systemausgangs  $y[k]$  seriell verbunden sind.

In der Praxis wird jedoch oft ein paralleles Modell benötigt. Bei nicht oder wenig verrauchten Messsignalen stellt der Gleichungsfehler eine gute Näherung für den Ausgangsfehler (siehe Kapitel 7.2.1.2) dar [111], so dass das geschätzte Modell auch parallel betrieben werden kann. Dies gilt jedoch nicht mehr, wenn die Störungen auf das System zunehmen. Denn anders als bei der Ausgangsfehlerstruktur, bei der sich Messstörungen nur auf das gebildete Fehlersignal  $e$  auswirken, werden bei der Gleichungsfehlerstruktur zusätzlich die Eingangssignale des Modells durch Rauschen verfälscht. Die ARX-Modellstruktur ist somit nur bedingt zur Identifikation von parallelen Modellen geeignet, da mit zunehmenden Rauschen, das Identifikationsergebnis mit einem zunehmenden systematischen Fehler behaftet ist [111]. Aus diesem Grund wurden in der Vergangenheit aufwändige lineare Lernverfahren entwickelt, wie z.B. die Methode der verallgemeinerten kleinsten Quadrate, die Methode der Hilfsvariablen und die Methode der totalen kleinsten Quadrate, die alle zum Ziel haben, die Konvergenz der Schätzung bei gestörten Modelleingangssignalen zu verbessern. Die Praxis hat jedoch gezeigt, dass auch mit diesen aufwändigen Methoden keine vollständig fehlerfreien Schätzungen erzielt werden, da reales Messrauschen in der Regel nicht die restriktiven Bedingungen erfüllt, die zur Herleitung dieser Verfahren idealisiert angenommen werden.

### 7.2.1.2 Output Error Model

Das OE-Modell gehört 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 4) zu adaptieren. In Abb. 7.5 ist die Ausgangsfehlerstruktur des OE-Modells dargestellt.



**Abb. 7.5:** Ausgangsfehlerstruktur des OE-Modells

Der OE-Systemansatz aus Abb. 7.5 ist durch die folgende Gleichung definiert:

$$y(z) = \frac{B(z)}{F(z)} \cdot u(z) + v(z) \quad (7.25)$$

Als OE-Modellansatz bietet sich folgende Gleichung mit  $e(z) = v(z)$  an:

$$\hat{y}(z) = \frac{\hat{B}(z)}{\hat{F}(z)} \cdot u(z) \quad (7.26)$$

Im Gegensatz zu Kapitel 7.2.1.1 wird das optimale Identifikationsmodell nicht aus der Fehlergleichung, sondern durch Ansetzen eines echten Parallelmodells bestimmt. Die Umwandlung von Gl. (7.26) in eine Differenzengleichung verdeutlicht, warum das OE-Modell nichtlinear in den Parametern ist:

$$\hat{y}[k] = b_1 \cdot u[k-1] + \dots + b_{n_b} \cdot u[k-n_b] - f_1 \cdot \hat{y}[k-1] - \dots - f_{n_f} \cdot \hat{y}[k-n_f] \quad (7.27)$$

Im Vergleich zum ARX–Modell wurde in Gl. (7.27) der gemessene Ausgang  $y$  nun 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. Dies soll anhand des folgenden Beispiels kurz dargestellt werden:

$$\begin{aligned}\hat{y}[k] &= -a_1 \cdot \hat{y}[k-1] + b_1 \cdot u[k-1] \\ \hat{y}[k+1] &= -a_1 \cdot (-a_1 \cdot \hat{y}[k-1] + b_1 \cdot u[k-1]) + b_1 \cdot u[k] \\ &= a_1^2 \cdot \hat{y}[k-1] - a_1 \cdot b_1 \cdot u[k-1] + b_1 \cdot u[k]\end{aligned}\quad (7.28)$$

Die zurückliegenden Modellausgangssignale müssen wieder in die Differenzengleichung eingesetzt werden, was dazu führt, dass sogar bei einem linearen System erster Ordnung das Ausgangssignal nicht mehr linear in den Parametern ist.

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.

### 7.2.2 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 Differenzengleichung basieren. Daraus resultieren unterschiedliche Vor– und Nachteile für Modelle ohne Ausgangsrückkopplung.

Bei Ausgangsfehlermodellen ist das Ergebnis der Identifikation 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 garantiierte Stabilität, da Modelle ohne Ausgangsrückkopplung nur von Eingangssignalen abhängen. Dadurch hat das Rauschen am Prozessausgang keinen Einfluss auf die Eingangssignale des Identifikationsalgorithmus, so dass die Parameteradaption nur aufgrund des Fehlersignals — das aber Rauschen enthält — beeinträchtigt wird. Diesen Vorteilen steht wohl als Nachteil die hohe Anzahl an unbekannten 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.

### 7.2.2.1 Finite Impulse Response Model

Allgemein kann ein lineares dynamisches System zeitdiskret durch die Faltungssumme beschrieben werden [229]. Der Systemausgang berechnet sich entsprechend Gl. (7.29) aus der Faltung der Impulsantwort  $h$  mit dem Eingangssignal  $u^2)$ :

$$y[k] = \sum_{i=0}^k h[i] u[k-i] \quad (7.29)$$

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 gilt<sup>3)</sup>:

$$\lim_{i \rightarrow \infty} h[i] = 0 \quad (7.30)$$

Auf die Wahl der oberen Grenze  $n_b = m$ , 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 7.6 zeigt das FIR-Modell in seiner Struktur.

FIR-Modelle gehören grundsätzlich zur Klasse der Ausgangsfehlermodelle und sind linear in den Parametern, jedoch können prinzipiell nur Systeme mit abklingender Impulsantwort, d.h. stabile Systeme beschrieben werden. Der FIR-Systemansatz in Abb. 7.6 ist durch die folgende Gleichung definiert:

$$y(z) = B(z) \cdot u(z) + v(z) \quad (7.31)$$

Die optimale Identifikationsgleichung für ein FIR-Modell ergibt sich mit  $e(z) = v(z)$  zu:

$$\hat{y}(z) = \hat{B}(z) \cdot u(z) \quad (7.32)$$

Die Umwandlung von Gl. (7.32) führt zu:

$$\hat{y}[k] = b_1 \cdot u[k-1] + b_2 \cdot u[k-2] + \dots + b_m \cdot u[k-m] \quad (7.33)$$

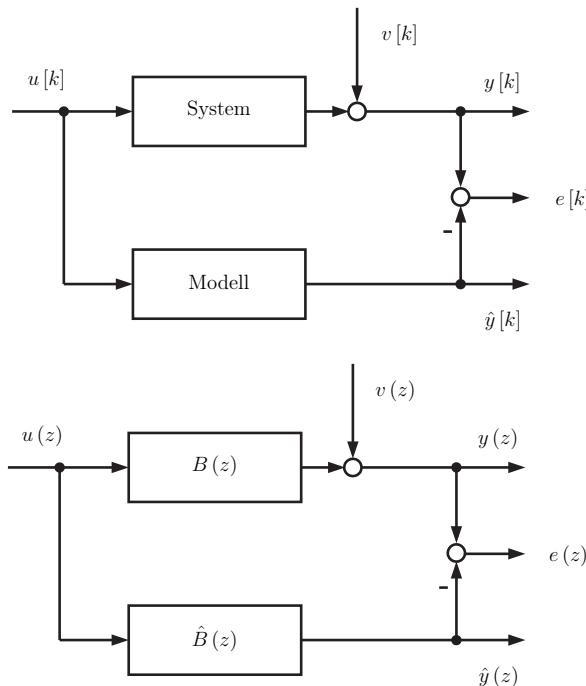
Gleichung (7.33) verdeutlicht noch einmal, dass das FIR-Modell aufgrund der fehlenden Ausgangsrückkopplung sowohl linear in den Parametern ist als auch zur Klasse der Ausgangsfehlermodelle gehört.

Den oben genannten Vorteilen steht der Nachteil gegenüber, dass die Anzahl der unbekannten Parameter sehr hoch ist. Die Anzahl der Parameter ist nach Gl. (7.33) identisch mit der Antwortlänge  $m$ , d.h.  $p_{FIR} = m$ . Die Antwortlänge hängt wiederum von der Systemdynamik und der Abtastzeit  $h$  ab.

---

<sup>2)</sup> Der Term  $h[0] \cdot u[k]$  kann vernachlässigt werden, wenn das System nicht sprungfähig ist.

<sup>3)</sup> Die Gleichung (7.30) gilt nicht für grenzstabile Systeme.



**Abb. 7.6:** Ausgangsfehlerstruktur des FIR–Modells

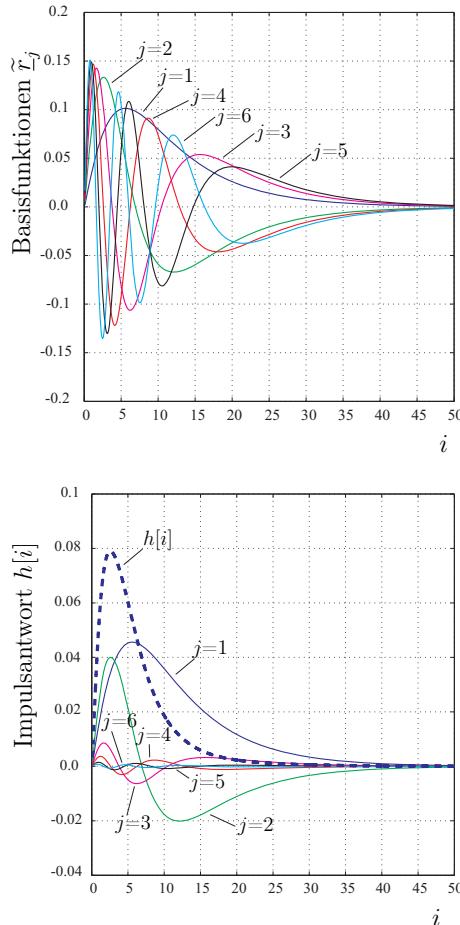
Als Faustformel wird in [131]  $m \simeq \frac{T_{99.9}}{h}$  angegeben<sup>4)</sup>. Im Vergleich dazu ist die Anzahl unbekannter Parameter bei Modellen mit Ausgangsrückkopplung durch  $p_{PARX/OE} = 2 \cdot n^5)$  gegeben. Ein Beispiel soll die Größenordnung der Anzahl unbekannter Parameter verdeutlichen. Betrachtet wird ein System zweiter Ordnung mit den auf die Abtastzeit  $h = 1\text{s}$  normierten Zeitkonstanten  $T_1 = 5$  und  $T_2 = 7$ . Für dieses System ergeben sich die Parameteranzahlen  $p_{PARX/OE} = 4$  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 sich das OBF–Modell ergibt.

<sup>4)</sup>  $T_{99.9}$  ist die Zeit, bis 99.9 % des Endwertes der Sprungantwort eines Systems erreicht sind.

<sup>5)</sup> Wenn angenommen wird, dass  $na = nb = n$  ist und das System nicht sprungfähig ist.

### 7.2.2.2 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 beschreiben<sup>6)</sup>. In Abb. 7.7 ist dieses Prinzip verdeutlicht. Mit der gewichteten Überlagerung orthonormaler Basisfunktionen



**Abb. 7.7:** Orthonormale Basisfunktionen (oben) und Überlagerung gewichteter Basisfunktionen zur Impulsantwort (unten)

kann die Impulsantwort (gestrichelte Kurve) eines Systems nachgebildet werden.

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

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:

$$y(z) = b_1 \cdot L_1(z) \cdot u(z) + b_2 \cdot L_2(z) \cdot u(z) + \dots + b_{m_r} \cdot L_{m_r}(z) \cdot u(z) + v(z) \quad (7.34)$$

$L_1(z) \dots L_{m_r}(z)$  kennzeichnet die Übertragungsfunktionen der orthonormalen Filter.  $m_r \in \mathbb{N}$  ist die Anzahl der orthonormalen Filter bzw. der orthonormalen Basisfunktionen, für die die Beziehung  $m_r \leq m$  gilt. Die Impulsantworten der orthonormalen Filter  $L_j(z)$  stellen die orthonormalen Basisfunktionen dar. Es gilt:

$$L_j(z) = \sum_{i=1}^{\infty} \tilde{r}_j[i] \cdot z^{-i} \quad \text{mit} \quad j = 1 \dots m_r \quad (7.35)$$

Das OBF–Modell nach Gl. (7.34) 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 [233] eignen sich für stark gedämpfte Systeme, da sie auf Vorwissen über einen reellen Pol des Systems beruhen. Umgekehrt eignen sich Kautz–Filter [234] für schwach gedämpfte, oszillierende Systeme, da sie Vorwissen über ein konjugiert komplexes Polpaar beinhalten. In [86, 168] 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 [123] werden als Basisfunktionen orthonormalisierte verzerrte Sinusfunktionen vorgeschlagen. Nach [131] 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:

$$r_j[i] = \frac{1}{\sqrt{\frac{m}{2}}} \cdot \sin \left[ j \cdot \pi \cdot \left( 1 - e^{-\frac{i-0.5}{\zeta}} \right) \right] \quad (7.36)$$

In Gl. (7.36) 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:

$$\mathbf{R} = \begin{bmatrix} \underline{r}_1^T \\ \underline{r}_2^T \\ \vdots \\ \underline{r}_{m_r}^T \end{bmatrix} \quad \text{mit} \quad \underline{r}_j^T = \left[ r_j[1], r_j[2], \dots, r_j[m] \right] \quad (7.37)$$

$\mathbf{R} \in \mathbb{R}^{m_r \times m}$  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  $\tilde{\mathbf{R}}$ :

$$\begin{aligned}\mathbf{R} &= \mathbf{C}^T \tilde{\mathbf{R}} & \mathbf{C} &\in \mathbb{R}^{m_r \times m} \\ \mathbf{R} \mathbf{R}^T &= \mathbf{C}^T \tilde{\mathbf{R}} \tilde{\mathbf{R}}^T \mathbf{C} = \mathbf{C}^T \mathbf{C} & (7.38) \\ \implies \tilde{\mathbf{R}} &= (\mathbf{C}^T)^{-1} \mathbf{R} & \tilde{\mathbf{R}} &\in \mathbb{R}^{m_r \times m}\end{aligned}$$

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

Die in Gl. (7.36) 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 [131] kann die Impulsantwort für den Fall  $n = 1$  besser rekonstruiert werden.

Die erweiterte Rekonstruktionsmatrix ergibt sich zu:

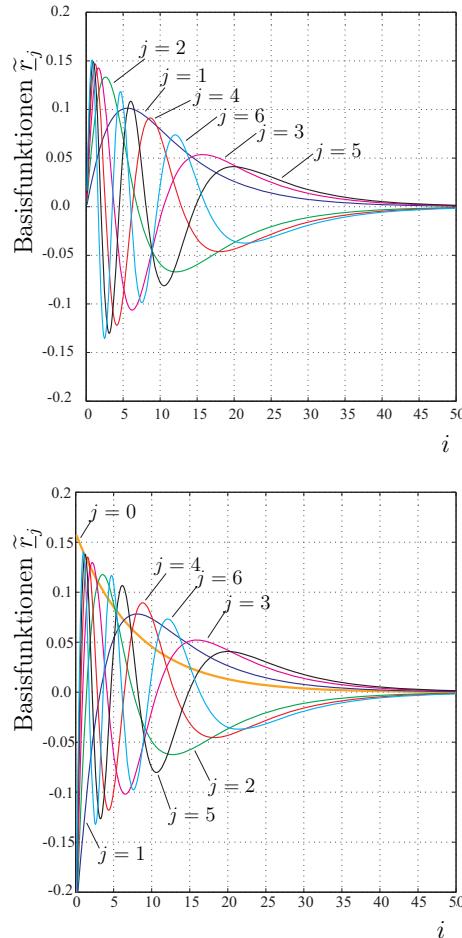
$$r_0[i] = \frac{1}{\sqrt{\frac{m}{2}}} \cdot e^{-\frac{j-i-0.5}{\zeta}} \quad \mathbf{R} = \begin{bmatrix} \underline{r}_0^T \\ \underline{r}_1^T \\ \vdots \\ \underline{r}_{m_r}^T \end{bmatrix} \quad \mathbf{R} \in \mathbb{R}^{(m_r+1) \times m} \quad (7.39)$$

Die orthonormierten Sinusfunktionen sind mit und ohne die abklingende Exponentialfunktion für die Einstellwerte  $m = 50$ ,  $\zeta = 9$ ,  $m_r = 5$  in Abb. 7.8 dargestellt. Für die richtige Wahl des Formfaktors  $\zeta$  sowie der Anzahl orthonormalisierter verzerrter Basisfunktionen  $m_r$  wurden in [131] die Faustformeln nach Tab. 7.2 festgelegt.

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

**Tabelle 7.2:** Faustformeln zur Wahl des Formfaktors  $\zeta$  und der Basisfunktionenanzahl  $m_r$  abhängig vom Dämpfungsgrad  $D$

Die Zeitkonstanten  $T_{63}$  und  $T_{95}$  bezeichnen die Zeit bis 63 % bzw. 95 % des End-



**Abb. 7.8:** Orthonormale verzerrte Sinusfunktionen ohne (oben) und mit (unten) abklingender Exponentialfunktion

wertes der Sprungantwort eines Systems erreicht sind. Diese Zeitkonstanten müssen noch auf die Abtastzeit  $h$  bezogen werden.

In Tab. 7.2 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:

$$\hat{y}[k] = b_1 \cdot \tilde{\underline{L}}_1^T \cdot \underline{u}[k] + b_2 \cdot \tilde{\underline{L}}_2^T \cdot \underline{u}[k] + \dots + b_{m_r} \cdot \tilde{\underline{L}}_{m_r}^T \cdot \underline{u}[k] \quad (7.40)$$

Die Vektoren  $\tilde{r}_j^T \in \mathbb{R}^{1 \times m}$  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:

$$\underline{u}^T[k] = [u[k-1], u[k-2], \dots, u[k-m]] \quad (7.41)$$

Werden die unbekannten Parameter  $b_i$  zu einem Parametervektor  $\underline{\Theta}$  zusammengefasst, vereinfacht sich Gl. (7.40) mit Hilfe der orthonormierten Basisfunktionenmatrix  $\tilde{\mathbf{R}}$  und das optimale OBF–Modell ergibt sich zu:

$$\hat{y}[k] = \underline{\Theta}^T \cdot \tilde{\mathbf{R}} \cdot \underline{u}[k] \quad (7.42)$$

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.

## 7.3 Identifikationsbeispiele

Die vorgestellten Modellstrukturen sollen in diesem Kapitel an einem Beispiel veranschaulicht werden. Betrachtet wird ein lineares System mit der Übertragungsfunktion

$$F(s) = \frac{1}{(1+sT_1)(1+sT_2)} = \frac{1}{s^2T_1T_2 + s(T_1 + T_2) + 1}$$

und den auf die Abtastzeit von  $h = 1\text{ s}$  normierten Zeitkonstanten

$$T_1 = 5 \quad T_2 = 7 \quad (\text{normiert}) \quad (7.43)$$

Im Folgenden sollen die vorgestellten Modellstrukturen mit und ohne Ausgangsrückkopplung an diesem Beispiel veranschaulicht werden.

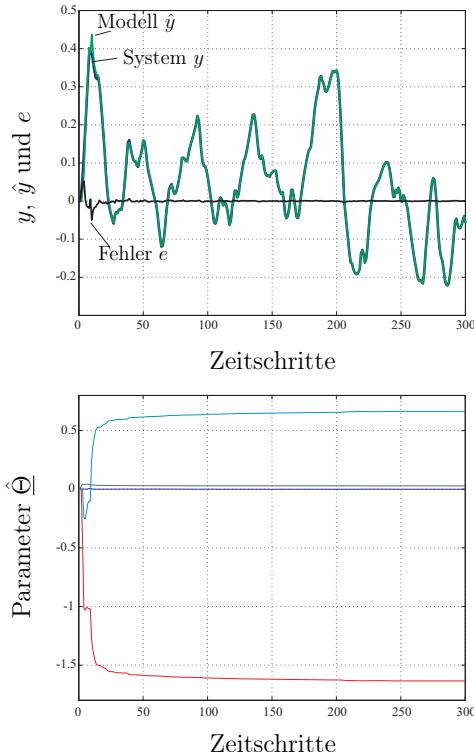
### 7.3.1 ARX–Modell

Der Ausgang des ARX–Modells berechnet sich entsprechend Gl. (7.24) zu:

$$\hat{y}[k] = \hat{\Theta}_1 \cdot u[k-1] + \hat{\Theta}_2 \cdot u[k-2] - \hat{\Theta}_3 \cdot y[k-1] - \hat{\Theta}_4 \cdot y[k-2] \quad (7.44)$$

Das ARX–Modell ist linear in den Parametern, so dass der RLS–Algorithmus zur Parameteradaption verwendet werden kann. Allerdings minimiert das ARX–Modell den Gleichungsfehler und nicht den Ausgangsfehler, so dass kein echtes Parallelmodell entsteht. Der Regressionsvektor setzt sich wie folgt zusammen:

$$\underline{x}^T[k] = [u[k-1], u[k-2], -y[k-1], -y[k-2]] \quad (7.45)$$



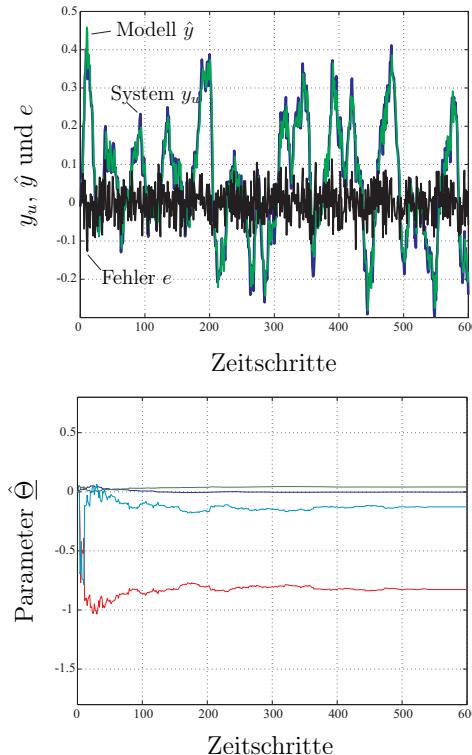
**Abb. 7.9:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim ARX-Modell

Es sind 4 unbekannte Parameter zu adaptieren. In Abb. 7.9 sind der Identifikationsverlauf und der Parameterverlauf dargestellt. Es ist zu erkennen, dass nach wenigen Zeitschritten der Fehler minimal wird und die Parameter konvergieren.

Nach 250 Zeitschritten werden die Parameter festgehalten, sodass das ARX-Modell als Parallelmodell fungiert. Wie in Abbildung 7.9 zu erkennen ist, steigt der Fehler zwischen Vorgabe und ARX-Modell in diesem Fall nicht an.

Wird dem Nutzsignal ein Rauschen mit ca. 10 % der Amplitude des Ausgangssignals überlagert, ergeben sich für den Identifikations- und Parameterverlauf die Ergebnisse nach Abb. 7.10.

Das starke Rauschen führt dazu, dass die Parameter zwar konvergieren, aber fehlerhaft bestimmt werden. Dies ist charakteristisch für das ARX-Modell, da nur im ungestörten Fall der Gleichungsfehler dem Ausgangsfehler gleichgesetzt werden kann. Die Konsequenz ist, dass der Fehler zwischen dem unverrauschten Ausgangssignal und dem Modell nicht klein wird. Im Parallelbetrieb ab 550 Zeit-



**Abb. 7.10:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim ARX-Modell mit Rauschen

schritten bleibt der Fehler groß. Das ARX–Modell liefert somit unbefriedigende Ergebnisse bei verrauschten Signalen.

### 7.3.2 OE–Modell

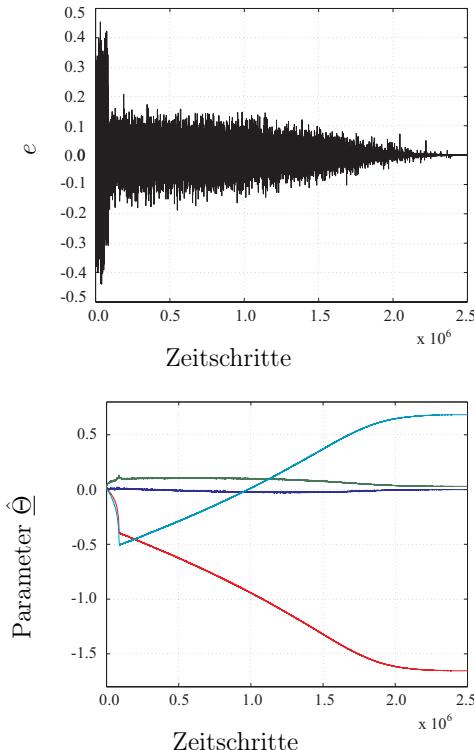
Der Ausgang des OE–Modells berechnet sich entsprechend Gl. (7.27) zu:

$$\hat{y}[k] = \hat{\Theta}_1 \cdot u[k-1] + \hat{\Theta}_2 \cdot u[k-2] - \hat{\Theta}_3 \cdot \hat{y}[k-1] - \hat{\Theta}_4 \cdot \hat{y}[k-2] \quad (7.46)$$

Das OE–Modell ist nichtlinear in den Parametern, so dass der RLS–Algorithmus zur Parameteradaption nicht verwendet werden kann. Stattdessen kommt das Gradientenabstiegsverfahren zum Einsatz. Die Lerngesetze für die 4 unbekannten Parameter lauten

$$\hat{\Theta}_i[k+1] = \hat{\Theta}_i[k] + \eta \cdot e[k] \cdot \frac{\partial \hat{y}[k]}{\partial \hat{\Theta}_i} \quad \text{mit} \quad i = 1 \dots 4 \quad (7.47)$$

mit dem Identifikationsfehler  $e[k] = y[k] - \hat{y}[k]$ . Beim OE–Modell wird der Ausgangsfehler minimiert, so dass ein echtes Parallelmodell entsteht. In Abb. 7.11 ist der Identifikationsverlauf und der Parameterverlauf dargestellt. Es ist zu er-

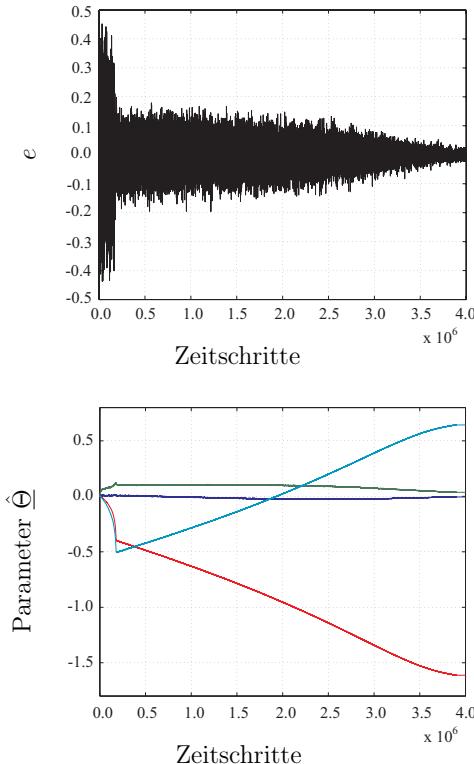


**Abb. 7.11:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim OE–Modell ohne Rauschen

kennen, dass aufgrund des nichtlinearen Lernverfahrens der Fehler sehr langsam gegen Null geht und die Parameter entsprechend langsam konvergieren. Nach 2.4 Mio.–Zeitschritten werden die Parameter festgehalten, so dass das OE–Modell als Parallelmodell arbeitet. Wieder zeigt sich, dass der Fehler im Parallelbetrieb nicht ansteigt.

In der Praxis sind jedoch solch lange Lernzeiten nicht sinnvoll, weswegen die Parameter in der Regel vorbelegt werden. Dazu wird zunächst ein ARX–Modell bestimmt, dessen Parameter dann zur Vorbelegung dienen.

Wird dem Nutzsignal ein Rauschen mit ca. 10 % der Amplitude des Ausgangssignals überlagert ergeben sich für den Identifikations– und Parameterverlauf die Ergebnisse nach Abb. 7.12.



**Abb. 7.12:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim OE-Modell mit Rauschen

Trotz des starken Rauschens wird der Fehler zwischen dem unverrauschten Ausgangssignal und dem Modell klein und die Parameter konvergieren. Die Parameter sind nicht wie beim ARX-Modell mit einem systematischen Fehler behaftet. Allerdings wird der Lernvorgang durch den Rauscheinfluss noch verlangsamt. Im Parallelbetrieb ab 3.9 Mio.-Zeitschritten steigt der Fehler nicht an.

### 7.3.3 FIR-Modell

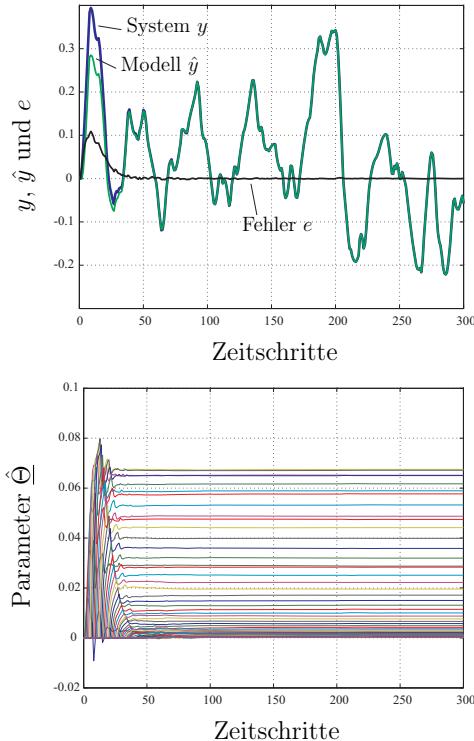
Der Ausgang des FIR-Modells berechnet sich entsprechend Gl. (7.33) zu:

$$\hat{y}[k] = \hat{\Theta}^T \cdot \underline{x}[k] = \hat{\Theta}_1 \cdot u[k-1] + \hat{\Theta}_2 \cdot u[k-2] + \dots + \hat{\Theta}_m \cdot u[k-m] \quad (7.48)$$

Das FIR-Modell ist linear in den Parameter, so dass der RLS-Algorithmus zur Parameteradaption verwendet werden kann. Der Regressionsvektor setzt sich wie folgt zusammen:

$$\underline{x}^T[k] = \left[ u[k-1], u[k-2], \dots, u[k-m] \right] \quad (7.49)$$

Die Impulsantwort wird bei der Antwortlänge  $m = 54$  abgeschnitten, d.h. es sind 54 unbekannte Parameter zu adaptieren. In Abb. 7.13 ist der Identifikationsverlauf und der Parameterverlauf dargestellt.



**Abb. 7.13:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim FIR-Modell

Es ist zu erkennen, dass nach ca. 60 Zeitschritten der Fehler gegen Null geht. Dies entspricht etwa der Anzahl der unbekannten Parameter. Die Parameter konvergieren sehr gut. Nach 250 Zeitschritten werden die Parameter festgehalten, so dass das FIR-Modell als Parallelmodell arbeitet. Der Fehler steigt im Parallelbetrieb nicht an. Die identifizierten Parameter ergeben die abgeschnittene Impulsantwort des System, wie in Abb. 7.14 deutlich wird.

Wird dem Nutzsignal ein Rauschen mit ca. 10 % der Amplitude des Ausgangssignals überlagert, ergeben sich für den Identifikations- und Parameterverlauf die Ergebnisse nach Abb. 7.15.

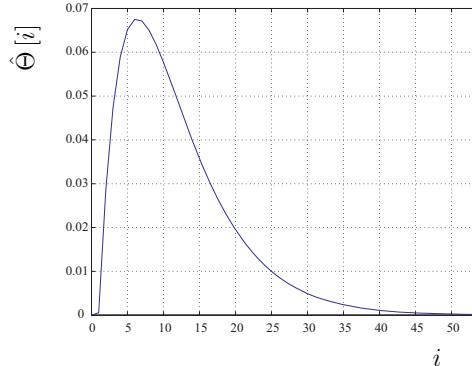


Abb. 7.14: Identifizierte Impulsantwort des FIR-Modells

Trotz des starken Rauschens wird der Fehler zwischen dem unverrauschten Ausgangssignal und dem Modell klein und die Parameter konvergieren. Im Parallelbetrieb ab 550 Zeitschritten steigt der Fehler nicht an. Die identifizierte Impulsantwort ist in Abb. 7.16 dargestellt.

Der Einfluss des Rauschens beeinträchtigt die Qualität der identifizierten Impulsantwort nur geringfügig. Der einzige gravierende Nachteil des FIR-Modells ist somit die hohe Anzahl an Parametern.

### 7.3.4 OBF-Modell

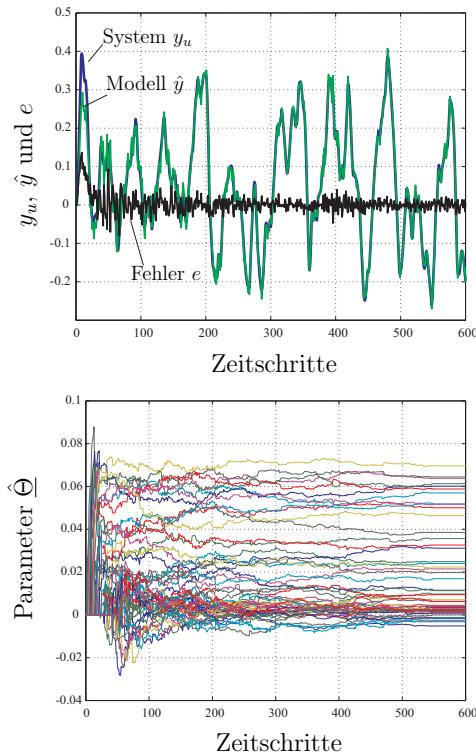
Mit dem auf Orthonormalen Basisfunktionen basierten Modell wird der Nachteil der hohen Parameteranzahl aufgehoben. Der Ausgang des OBF-Modells berechnet sich entsprechend Gl. (7.42) zu:

$$\hat{y}[k] = \hat{\Theta}^T \cdot \underbrace{\tilde{\mathbf{R}} \cdot \underline{u}[k]}_{\underline{x}[k]} \quad (7.50)$$

mit dem Vektor

$$\underline{u}^T[k] = [u[k-1], u[k-2], \dots, u[k-m]] \quad (7.51)$$

Das OBF-Modell ist linear in den Parameter, so dass der RLS-Algorithmus zur Parameteradaption verwendet werden kann. Der Regressionsvektor setzt sich nun aus dem Produkt der Basisfunktionenmatrix mit dem Vektor der vergangenen Eingangssignale zusammen. Die Anzahl der Parameter reduziert sich somit auf  $m_r$ . In diesem Beispiel wird  $m_r = 10$  und der Formfaktor der Basisfunktionen  $\zeta = 13.2$  gewählt. Als Antwortlänge ergibt sich wiederum  $m = 54$ . In Abb. 7.17 ist der Identifikationsverlauf und der Parameterverlauf dargestellt.



**Abb. 7.15:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim FIR-Modell mit Rauschen

Es ist zu erkennen, dass aufgrund der geringeren Parameteranzahl der Fehler schneller gegen Null geht als beim FIR-Modell. Die Parameter konvergieren sehr gut. Nach 250 Zeitschritten werden die Parameter festgehalten, so dass das OBF-Modell als Parallelmodell arbeitet. Der Fehler steigt im Parallelbetrieb nicht an. Die identifizierten Parameter ergeben zusammen mit der Basisfunktionenmatrix die abgeschnittene Impulsantwort des Systems, wie in Abb. 7.18 dargestellt.

Wird dem Nutzsignal wiederum ein Rauschen mit ca. 10 % der Amplitude des Ausgangssignals überlagert ergeben sich für den Identifikations- und Parameterverlauf die Ergebnisse nach Abb. 7.19.

Das starke Rauschen hat kaum einen Einfluss auf die Identifikation. Der Fehler zwischen dem unverrauschten Ausgangssignal und dem Modell wird klein und die Parameter konvergieren. Im Parallelbetrieb ab 550 Zeitschritten steigt der Fehler nicht an. Die identifizierte Impulsantwort ist in Abb. 7.20 dargestellt.

Der Einfluss des Rauschens ist in der Impulsantwort kaum zu sehen.

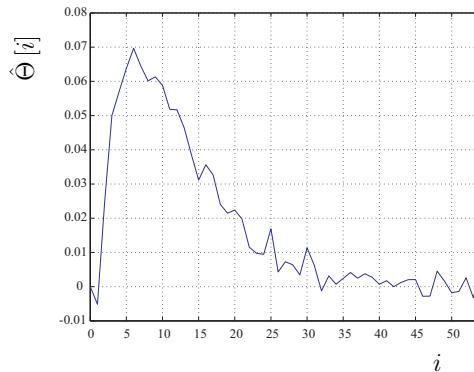
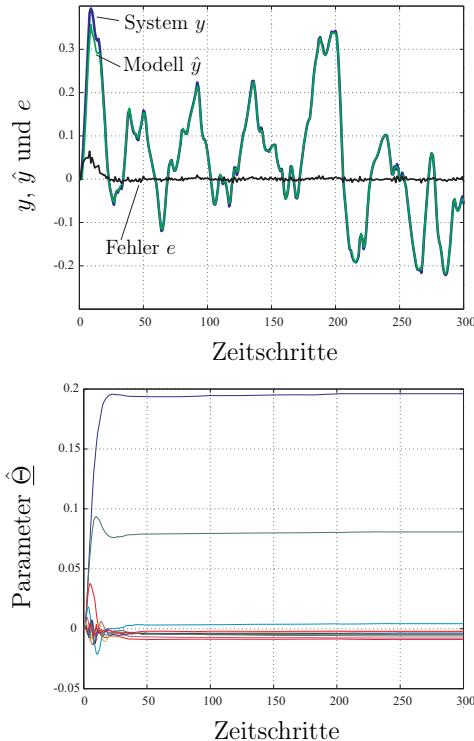


Abb. 7.16: Identifizierte Impulsantwort des FIR-Modells mit Rauschen

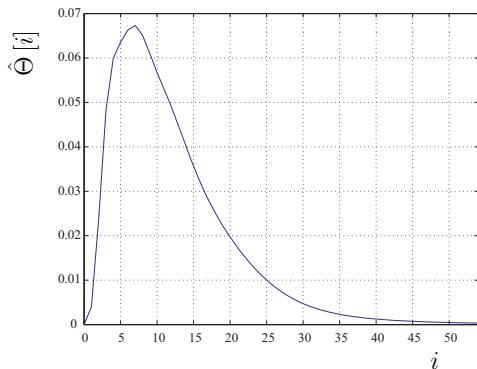
## 7.4 Zusammenfassung

In diesem Kapitel wurden die wichtigsten linearen Identifikationsverfahren im Überblick vorgestellt. Es wurde zwischen Modellstrukturen mit und ohne Ausgangsrückkopplung unterschieden und auf die verschiedenen Vorteile und Nachteile der einzelnen Identifikationsverfahren eingegangen. Wichtige Merkmale sind dabei, ob eine Minimierung des Gleichungsfehlers oder des Ausgangsfehlers erfolgt und ob die Modellstruktur linear oder nichtlinear in den Parametern ist. An einem Identifikationsbeispiel wurden das ARX- und das OE-Modell als zwei Vertreter für Modellstrukturen mit Ausgangsrückkopplung sowie das FIR- und das OBF-Modell als zwei Vertreter für Modellstrukturen ohne Ausgangsrückkopplung veranschaulicht.

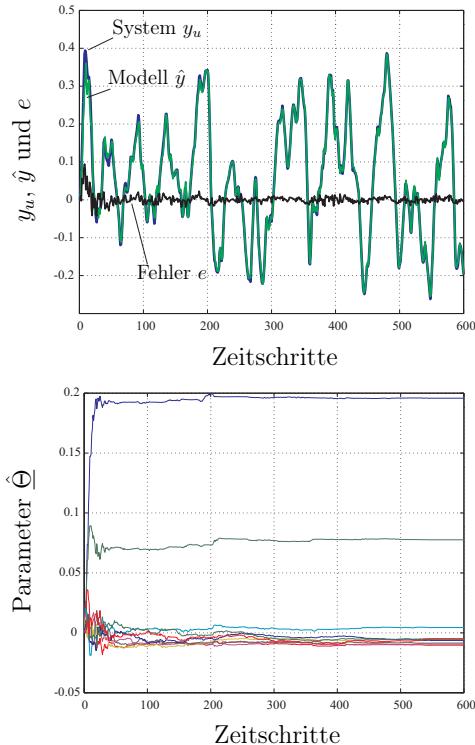
An dieser Stelle sei angemerkt, dass es in Software-Tools, wie z.B. *Matlab/Simulink* für lineare Identifikationsprobleme Standardlösungen gibt. In der *System Identification Toolbox* von *Matlab/Simulink* werden Identifikationsalgorithmen im Zeit- sowie Frequenzbereich bereitgestellt.



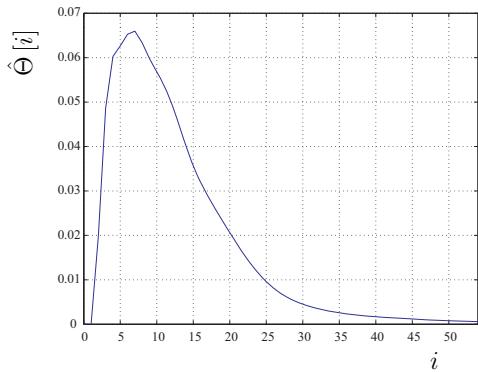
**Abb. 7.17:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim OBF-Modell



**Abb. 7.18:** Identifizierte Impulsantwort des OBF-Modells



**Abb. 7.19:** Identifikationsverlauf (oben) und Konvergenz der Parameter (unten) beim OBF-Modell mit Rauschen



**Abb. 7.20:** Identifizierte Impulsantwort des OBF-Modells mit Rauschen

## 8 Identifikation nichtlinearer dynamischer Systeme

Bei nichtlinearen dynamischen Systemen nimmt die Komplexität und Vielfalt, verglichen mit linearen dynamischen Systemen, deutlich zu. Bis heute gibt es keine einheitliche, in sich geschlossene, mathematische Theorie zur Beschreibung und Identifikation solcher Systeme. Im Rahmen von Forschungsarbeiten wurden deshalb immer Identifikations- und Regelverfahren für bestimmte *Klassen von nichtlinearen dynamischen Systemen* entwickelt. Der andauernde Forschungsaufwand kann dadurch gerechtfertigt werden, dass viele technische Prozesse ein deutlich nichtlineares Verhalten aufweisen, das sich nicht mehr um einen Arbeitspunkt linearisieren lässt, so dass nichtlineare Methoden angewendet werden müssen. In der Regel stellen die Identifikationsverfahren für nichtlineare dynamische Systeme eine Erweiterung linearer Verfahren dar. Im Gegensatz zu den linearen Systemen, wo in der Regel 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.

Nichtlineare dynamische Systeme unterscheiden sich sehr stark durch den Grad des strukturellen Vorwissens. Im Allgemeinen können folgende Ansätze zur Modellbildung, abhängig vom Vorwissen über das zu modellierende System unterschieden werden [11, 167]:

- **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**

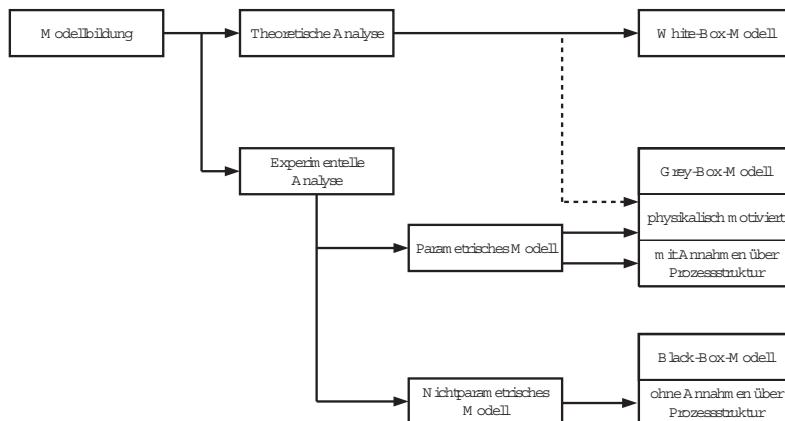
Soll eine aufwändige theoretische Analyse vermieden werden oder sind nur unzureichende Kenntnisse über das Systemverhalten vorhanden, kann eine

experimentelle Analyse durchgeführt 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. 8.1 sind die erläuterten Begriffe noch einmal grafisch gegeneinander abgegrenzt [11, 230].



**Abb. 8.1:** 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 5). Wird lediglich das Ein-/Ausgangsverhalten eines Modells zur Prädiktion, Simulation oder Diagnose benötigt, kann auf die experimentelle Analyse zurückgegriffen werden.

## 8.1 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.

### 8.1.1 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 [205]:

$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A} \cdot \underline{x} + \underline{b} \cdot u + \underline{k}_{\mathcal{N}\mathcal{L}} \cdot \mathcal{N}\mathcal{L}(\underline{x}, u) \\ y &= \underline{c}^T \cdot \underline{x} + d \cdot u\end{aligned}\tag{8.1}$$

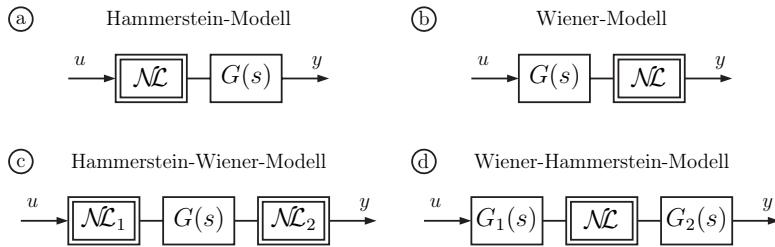
Die Nichtlinearität  $\mathcal{N}\mathcal{L}(\underline{x}, u)$  ist statisch und kann von der Eingangsgröße  $u$  sowie den Systemzuständen  $\underline{x}$  abhängen. Der Eingriff der statischen Nichtlinearität kann mit dem Vektor  $\underline{k}_{\mathcal{N}\mathcal{L}}$  beschrieben werden.

Die exakte Systemkenntnis erlaubt den Entwurf eines nichtlinearen Beobachters, der bereits im Kapitel 5 ausführlich beschrieben wurde.

### 8.1.2 Blockorientierte nichtlineare Modelle

Blockorientierte nichtlineare Modelle werden dann verwendet, wenn nur ein großes Strukturvorwissen des Systems vorhanden ist. In Abb. 8.2 sind einige blockorientierte nichtlineare Modelle dargestellt.

Sie alle entstehen durch die Kombination aus statischen Nichtlinearitäten  $\mathcal{N}\mathcal{L}_i$  mit dynamischen Übertragungsfunktionen  $G_i(s)$ . Diese nichtlinearen Modellstrukturen, besonders das Hammerstein- und das Wiener-Modell, stellen bekannte nichtlineare Prozesse dar. Das Hammerstein-Modell (vgl. Abb. 8.2 a) setzt sich aus einer statischen Nichtlinearität  $\mathcal{N}\mathcal{L}$ , gefolgt von einer dynamischen Übertragungsfunktion  $G(s)$ , zusammen. Im Gegensatz dazu wird beim Wiener-Modell (vgl. Abb. 8.2 b) die statische Nichtlinearität  $\mathcal{N}\mathcal{L}$  durch die vorgesetzte Übertragungsfunktion angeregt. Beide Modelle werden in Kapitel 8.2.2.2 noch

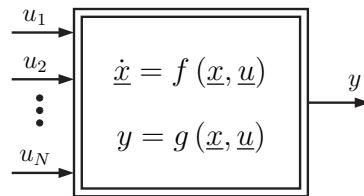


**Abb. 8.2:** Blockorientierte nichtlineare Modelle

genauer behandelt. Weiterhin sind auch Mischmodelle beider Ansätze denkbar, wie z.B. das in Abb. 8.2 c gezeigte Hammerstein–Wiener–Modell oder das in Abb. 8.2 d dargestellte Wiener–Hammerstein–Modell.

### 8.1.3 Allgemeine nichtlineare Systembeschreibung

In Abb. 8.3 ist die Beschreibung eines nichtlinearen dynamischen Systems in seiner allgemeinsten Form dargestellt.



**Abb. 8.3:** Allgemeine nichtlineare Systembeschreibung

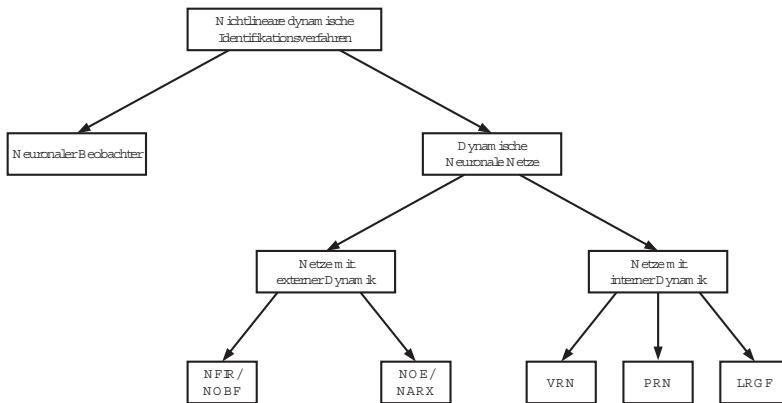
Das nichtlineare dynamische System lässt sich wie folgt beschreiben:

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad y = g(\underline{x}, \underline{u}) \quad (8.2)$$

In den Gleichungen (8.2) 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.

## 8.2 Verfahren zur Identifikation nichtlinearer dynamischer Systeme

In Abb. 8.4 sind verschiedene Verfahren zur Identifikation nichtlinearer dynamischer Systeme dargestellt.



**Abb. 8.4:** Übersicht dynamischer neuronaler Netze

Dynamische Neuronale Netze benötigen kein oder nur wenig strukturelles Vorwissen. Zur Identifikation *nichtlinearer dynamischer* Systeme, müssen die bekannten statischen Funktionsapproximatoren durch dynamische Elemente bzw. die bekannten linearen Identifikationsverfahren für nichtlineare Systeme erweitert werden. Eine Dynamikmodellierung ist sowohl außerhalb als auch innerhalb des Netzes möglich. Bei den Netzen mit externer Dynamik handelt es sich prinzipiell um statische Funktionsapproximatoren, die zur Dynamikmodellierung mit externen Filterketten erweitert werden. Man unterscheidet nichtlineare Modelle mit Ausgangsrückkopplung (NOE/NARX) und nichtlineare Modelle ohne Ausgangsrückkopplung (NFIR/NOBF). Das NFIR–Modell bzw. das NOBF–Modell kann jedoch auch als nichtlineare Erweiterung der Faltungssumme bzw. als Erweiterung des entsprechenden linearen Identifikationsverfahrens angesehen werden. Im Gegensatz dazu sind bei Netzen mit interner Dynamik die Verzögerungselemente und rekurrenten Verbindungen in die Netzstruktur selbst eingebunden. Man unterscheidet je nach Netzstruktur voll rekursive Netze (VRN), partiell rekursive Netze (PRN) und lokal rekursive Global Feedforward Netze (LRGF). Diese rekurrenten Netze wurden in Kapitel 6 besprochen.

Allgemein kann die externe Dynamik durch vorgeschaltete Filterketten auf die Netzeingänge realisiert werden [170]. Für die nichtlinearen Modelle mit Ausgangsrückführung ergeben sich prinzipiell zwei mögliche Strukturen. Diese sind in Abb. 8.5 links und in der Mitte dargestellt. Die dritte mögliche Struktur in

Abb. 8.5 rechts, ergibt sich für Ansätze, bei denen das Ausgangssignal allein durch verzögerte Eingangssignale bestimmt wird. Ausgangspunkt dafür ist die sog. Volterra–Funktionalpotenzreihe, die als nichtlineare Erweiterung der Faltungssumme betrachtet werden kann.

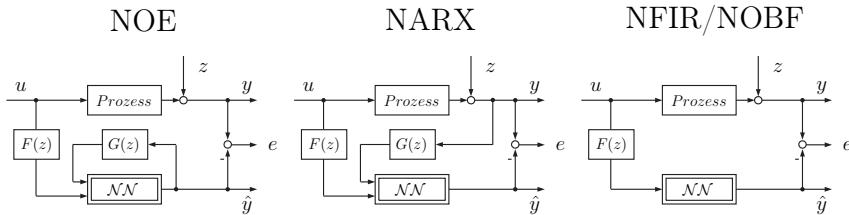


Abb. 8.5: Strukturen nichtlinearer Identifikationsmodelle

Die Abkürzungen bedeuten:

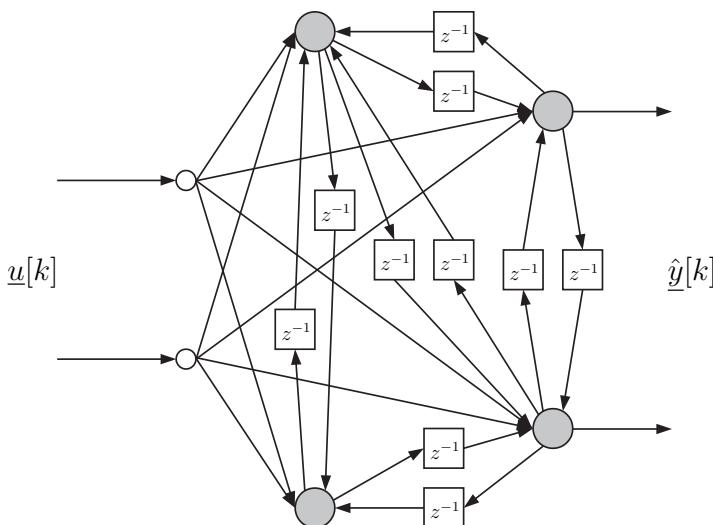
- |      |  |
|------|--|
| NOE  | Nonlinear Output Error                         |
| NARX | Nonlinear Auto Regressive with eXogenous input |
| NFIR | Nonlinear Finite Impuls Response               |
| NOBF | Nonlinear with Orthonormal Basis Functions     |

Beim NOE–Modell werden netzeigene Ausgangssignale  $\hat{y}$  verzögert auf die Netzeingänge zurückgeführt. Bei dieser rekurrenten Struktur wird der Ausgangsfehler minimiert und es kommt zu den bereits in Kapitel 7 diskutierten Nachteilen hinsichtlich Lernvorgang und Stabilität. Allerdings müssen Modelle, die für die Simulation eingesetzt werden, auf diese Weise trainiert werden, da nur bei der netzeigenen Rückführung ein echtes Parallelmodell entsteht. Beim NARX–Modell wird der gemessene Prozessausgang verzögert auf den Netzeingang zurückgeführt. Dadurch liegt die in Abb.8.5 in der Mitte gezeigte Anordnung vor, die nicht der Modellstruktur eines Parallelmodells entspricht. Bei dieser Anordnung wird der 1–Schritt–Prädiktions– oder Gleichungsfehler minimiert, was dazu führt, dass die Parameterschätzung bei Messrauschen im Allgemeinen mit einem systematischen Fehler behaftet ist. Die Nachteile der Gleichungsfehleranordnung wurden bereits in Kapitel 7 für lineare Systeme diskutiert. Dennoch wird in den meisten Fällen die NARX–Struktur gewählt, da sie die nicht unerheblichen Nachteile der NOE–Struktur vermeidet. Beim NFIR– bzw. NOBF–Modell wird der Ausgangsfehler minimiert. Auf die Vorteile von Ansätzen dieser Art wurde ebenfalls in Kapitel 7 ausführlich eingegangen. Die Volterra–Funktionalpotenzreihe [194, 232] in ihrer allgemeinen Form entspricht dem NFIR–Ansatz. Der Nachteil von NFIR–Modellen ist die große Parameteranzahl. Diese kann durch Einbringen von Vorwissen über die Prozessdynamik in Form von speziellen Filtern reduziert werden. Diese Filter zeichnen sich durch die Orthonormalität ihrer Impulsantworten aus [170]. Deshalb bezeichnet man Ansätze, die sich dieser Art der Komprimierung des Eingangsraumes bedienen als

NOBF-Ansätze. Dazu zählt z.B. der Ansatz der Volterra-Funktionalpotenzreihe unter Verwendung orthonormaler Basisfunktionen.

Bei Netzen mit interner Dynamik ist die explizite Verwendung von Vergangenheitswerten von Ein- und Ausgangssignalen am Netzeingang nicht erforderlich. Da diese Netze keine externe Rückkopplung besitzen, wird der Ausgangsfehler minimiert und ein echtes Parallelmodell identifiziert. In Abhängigkeit von der Struktur der eingefügten Speicherelemente können drei prinzipielle Architekturen unterschieden werden.

Die vollständig vernetzte Struktur (VRN) besteht – wie der Name schon andeutet – aus vollvernetzten Neuronen mit sigmoiden Transferfunktionen. Jedes Neuron stellt einen internen Zustandsspeicher dar. Abbildung 8.6 verdeutlicht die Struktur der voll rekurrenten Netze. Der Nachteil voll rekurrenter Netze ist das langsame Konvergenzverhalten und Stabilitätsprobleme beim Training, weshalb sich diese Struktur für Zwecke der Identifikation auch nicht durchgesetzt hat.



**Abb. 8.6:** Voll vernetzte Struktur

Die Architektur der partiell rekurrenten Netze (PRN) basiert auf mehrschichtigen Perzeptronen, die mit einer zusätzlichen Kontextschicht erweitert werden. Die Neuronen dieser Kontextschicht stellen die internen Speicher des Modells dar. Da die Rückführung eines Signals immer eine Zeiteinheit benötigt, repräsentieren die Kontextneuronen die internen Netzzustände. Bekannte Vertreter der partiell rekurrenten Netze sind das Jordan- und das Elman-Netz [81]. Beim Jordan-Netz koppelt der Netzausgang verzögert wieder in den Eingang über Kontextneuronen ein, der sonstige Aufbau entspricht dem MLP-Netzwerk von Kapitel 3.10. Für

jeden Ausgang ist ein eigenes Kontextneuron vorhanden. Wie in der dargestellten

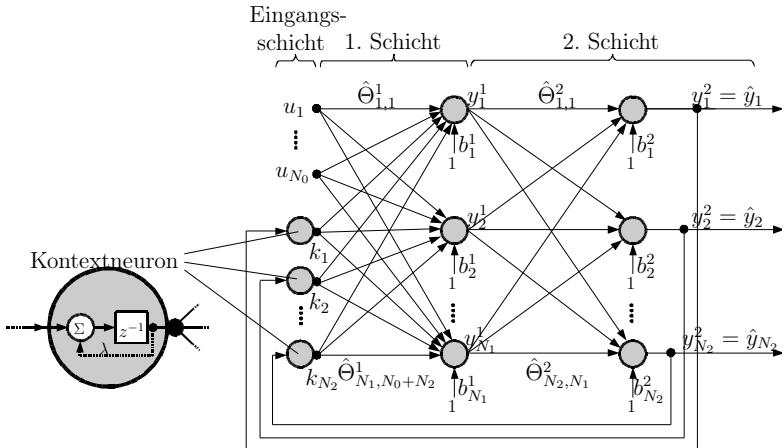


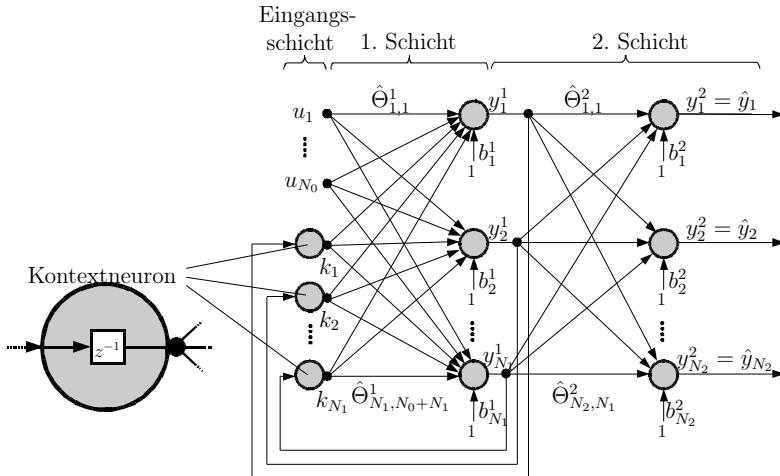
Abb. 8.7: Beispiel eines partiell rekurrenten Netzes: Jordan-Netzwerk

Netzarchitektur von Abbildung 8.7 zu sehen, verhalten sich die Kontextneuronen analog zu den anderen Neuronen der Eingangsschicht. Der weitere Signalweg verläuft nur vorwärtsgerichtet und trifft auf jedes Neuron der versteckten Schicht. Die Kontextneuronen können zusätzlich eigene Rückkopplungen enthalten, wie in Abbildung 8.7 gestrichelt eingezeichnet. Der Erinnerungsfaktor  $\lambda = 0 \dots 1$  bestimmt dabei, wie stark vergangene Signale in die Berechnungen einfließen. Der Erinnerungsfaktor  $\lambda$  und die Rückkopplung von den Netzausgängen zu den Kontextneuronen sind konstant und werden vom Optimierungsalgorithmus nicht angepasst.

Das Elman-Netz koppelt nicht — wie das Jordan-Netz — den Netzausgang zurück, sondern die Neuronenausgänge der ersten versteckten Schicht. Somit stimmt die Anzahl der Kontextneuronen mit der Neuronenzahl der versteckten Schicht überein. Gegenüber dem ähnlichen Jordan-Netz entfallen beim Elman-Netz außerdem die direkten Rückkopplungen bei den Kontextneuronen. Die resultierende Struktur des Elman-Netzes ist in Abbildung 8.8 dargestellt.

Die *Lokal Rekurrenten Global Feedforward* Netze (LRGF) beruhen auf statischen Feedforward Netzen, die durch zusätzliche lokale Rückkopplungen erweitert werden [11]. Feedback Verbindungen oder laterale Verbindungen sind nicht vorhanden. Die eingefügte rekurrente Struktur bleibt somit immer auf ein Neuron beschränkt und kann als lineares Filter interpretiert werden. In Abb. 8.9 sind verschiedene Strukturen abhängig von der Anordnung der linearen Filter im Netz dargestellt [170].

Bei der *Synapsendynamik* werden die konstanten Gewichte der Neuroneneingänge durch lineare Filter ersetzt. Die *Aktivierungsdynamik* stellt einen Spezialfall der Synapsendynamik dar. Für den Fall, dass die Übertragungsfunktionen in al-



**Abb. 8.8:** Beispiel eines partiell rekurrenten Netzes: Elman-Netzwerk

len Synapsen des Neurons gleich sind, können alle Übertragungsfunktionen durch ein lineares Filter hinter der Summationsstelle ersetzt werden. Bei der *Rückführungs dynamik* wird die Übertragungsfunktion im Rückführungszweig von Neuronausgang zu Neuroneingang plaziert.

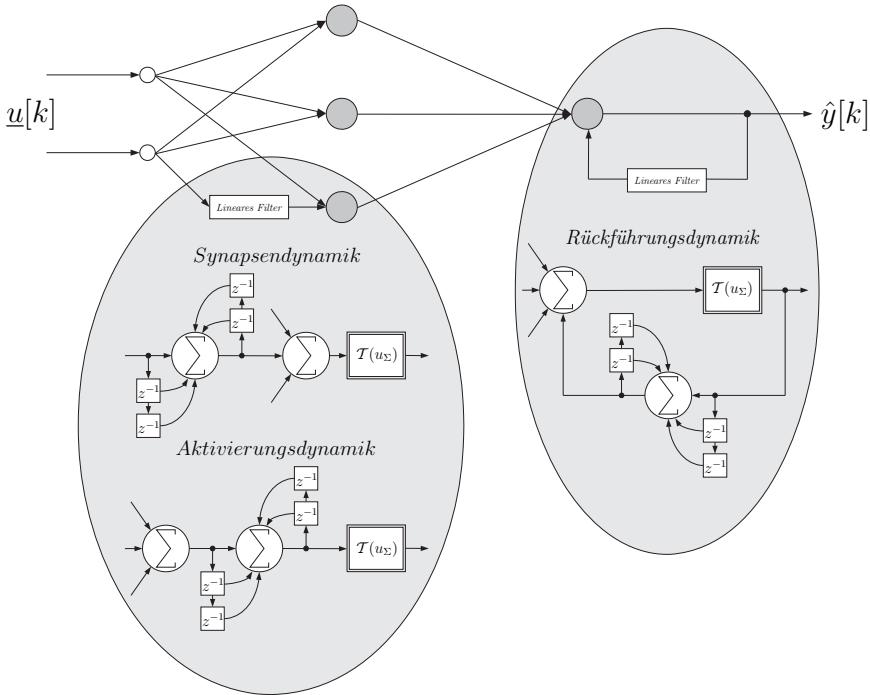
Im Gegensatz zu den dynamischen Neuronalen Netzen werden beim Neuronalen Beobachter [205] statische Neuronale Netze in einem Beobachter verwendet und somit statische Nichtlinearitäten innerhalb eines bekannten dynamischen Systems identifiziert. Dieser Ansatz setzt deutlich mehr Strukturwissen voraus als dynamische Neuronale Netze.

Der Neuronale Beobachter wurde bereits in Kapitel 5 ausführlich behandelt. Auch auf die Neuronalen Netze mit interner Dynamik soll in diesem Kapitel nicht weiter eingegangen werden, da diese in Kapitel 6 eigens behandelt werden. Im Folgenden sollen deshalb ausschließlich die Netze mit externer Dynamik genauer betrachtet werden.

Ein ausführlicherer Überblick zu diesem Forschungsgebiet kann beispielsweise in *Nonlinear System Identification* [168] von Nelles oder in *Simulation Neuronaler Netze* [248] von Zell nachgelesen werden.

### 8.2.1 Nichtlineare Modelle mit Ausgangsrückkopplung

Bei den nichtlinearen Modellen mit Ausgangsrückkopplung unterscheidet man, abhängig von der Art der Rückkopplung, das nichtlineare Ausgangsfehlermodell (NOE) und das nichtlineare Gleichungsfehlermodell (NARX). Analog zu den linearen Systemen sind auch Modelle mit komplexeren Störsignalmodellen möglich [168], wie z.B. das NARMAX-Modell (Nonlinear Auto Regressive Mo-



**Abb. 8.9:** Lokal Rekurrentes Global Feedforward Netz

ving Average model with eXogenous input). Bei den nichtlinearen Modellen mit Ausgangsrückkopplung wird der statische Funktionsapproximator mit Verzögerungsketten am Netzeingang erweitert. Diese sind zur Modellierung der Dynamik erforderlich. Die Abbildungen 8.10 und 8.11 zeigen nochmal die prinzipiellen Strukturen.

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:

$$F_i(z) = z^{-i} \quad \text{mit} \quad i = 1 \dots n \quad (8.3)$$

$$G_i(z) = z^{-i} \quad \text{mit} \quad i = 1 \dots n \quad (8.4)$$

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 3 vorgestellten Netztypen verwendet werden. Allerdings ist zu beachten, dass bei der Identifikation nichtlinearer dynamischer Systeme die Anzahl der Eingangssignale stark ansteigt. So

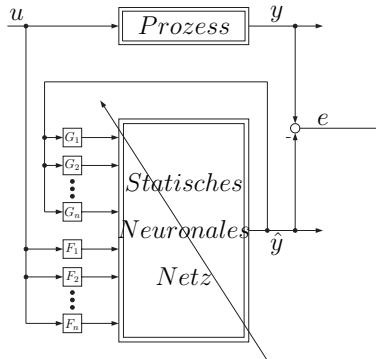


Abb. 8.10: NOE-Modell

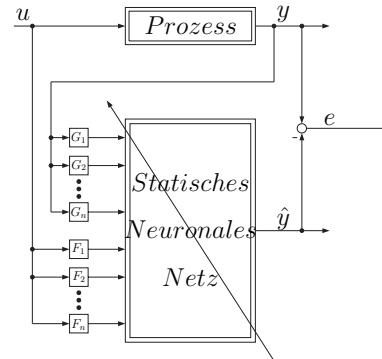


Abb. 8.11: NARX-Modell

hat der Eingangsraum bei der Identifikation eines nicht sprungfähigen Prozesses 2. Ordnung bereits die Dimension  $N = 4$ . Im Falle des RBF- und des GRNN-Netzes, bei denen die Parameteranzahl mit  $p = M^N$  ansteigt (vgl. Kapitel 3.11), würde dies bedeuten, dass bei  $M = 10$  Stützwerten je Dimension  $p = 10^4 = 10000$  Parameter zu adaptieren wären. Dies macht deutlich, warum das RBF-Netz bzw. das GRNN in der dargestellten Weise nur mit einer Strukturselektion [117] 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 [171, 248].

### 8.2.1.1 Time Delay Neural Network

Verwendet man als statischen Funktionsapproximator ein MLP-Netz und modelliert die Dynamik durch einfache zeitliche Schiebeoperatoren, wie in Gl. (8.4) dargestellt, erhält man das Time Delay Neural Network (TDNN). Abbildung 8.12 zeigt die Struktur des TDNN für die Identifikation eines nicht sprungfähigen Prozesses 2. Ordnung.

Das TDNN wird in der Regel in der Regel in der NOE-Struktur trainiert und betrieben. Das bedeutet:

$$\hat{y}[k] = f \left( u[k-1], u[k-2], \dots, u[k-n], \hat{y}[k-1], \hat{y}[k-2], \dots, \hat{y}[k-n] \right) \quad (8.5)$$

Da der Ausgang des MLP-Netzes nichtlinear von den Parametern abhängig ist, war bereits bei der Identifikation von statischen Nichtlinearitäten ein nichtlineares Optimierungsverfahren, wie z.B. das Gradientenabstiegsverfahren, erforderlich. Das bedeutet, dass bei der Identifikation dynamischer Systeme in

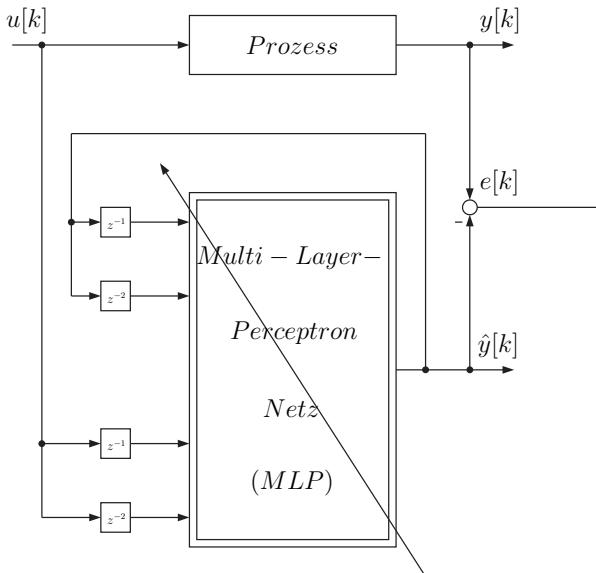


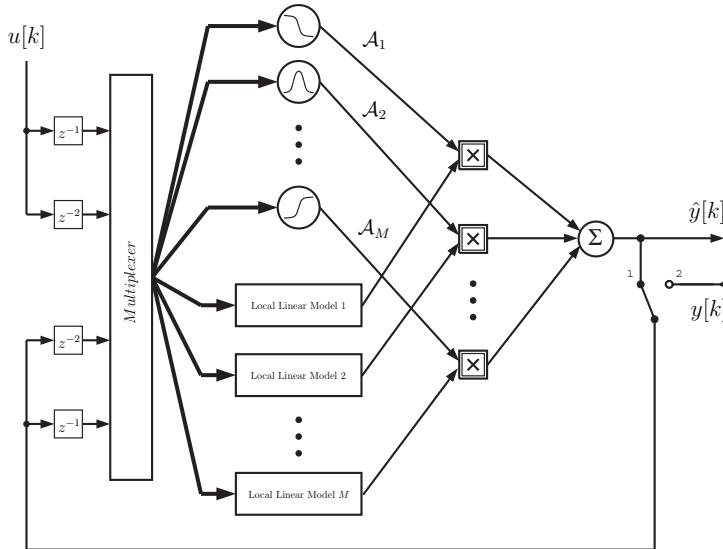
Abb. 8.12: Struktur des TDNN zur Identifikation eines Prozesses 2. Ordnung

der NOE-Struktur kein anderes Lernverfahren erforderlich ist. Allerdings ergeben sich aufgrund des nichtlinearen Optimierungsverfahrens lange Lernzeiten, was ein Nachteil des TDNN ist. Ein Problem beim TDNN ist, eine geeignete Struktur des Netzes zu finden, d.h. eine geeignete Anzahl von Schichten und der jeweiligen Neuronenanzahl (vgl. Kapitel 3.10.5).

### 8.2.1.2 Local Linear Model Tree

Wie bereits in Kapitel 3.11 erwähnt, eignet sich LOLIMOT vor allem für höherdimensionale Identifikationsaufgaben, wie dies bei nichtlinearen dynamischen Systemen der Fall ist. Dabei wird dem Problem von nichtlinearen Modellen mit Ausgangsrückkopplung in spezieller Weise Rechnung getragen. Aus Gründen der Modellgüte möchte man den Ausgangsfehler minimieren. Dies erfordert jedoch aufwändige nichtlineare Parameteroptimierungsverfahren. Deshalb möchte man aus Sicht der Parameteroptimierung den Gleichungsfehler minimieren, da in diesem Fall lineare Verfahren anwendbar sind. Das Ergebnis ist jedoch kein echtes Parallelmodell. Beim LOLIMOT–Algorithmus basiert die Parameterberechnung auf dem Gleichungsfehler (NARX–Struktur), während die Strukturoptimierung auf dem Ausgangsfehler (NOE–Struktur) beruht. Dies hat den Vorteil, dass ein lineares Verfahren zur Parameteroptimierung eingesetzt werden kann, jedoch die durch die Rückkopplung bedingte Fehlerfortpflanzung, aufgrund der Parameter-

optimierung in der Gleichungsfehleranordnung, durch die Strukturoptimierung vermieden werden kann. Das Problem der biasbehafteten Parameterberechnung bei verrauschten Signalen bleibt jedoch bestehen. Dies kann z.B. durch eine nachfolgende Parameteroptimierung in der Ausgangsfehleranordnung erfolgen. In Abb. 8.13 ist die Netzstruktur für einen nicht sprungfähigen Prozess 2. Ordnung dargestellt.



**Abb. 8.13:** Struktur von LOLIMOT zur Identifikation eines Prozesses 2. Ordnung

Der in Abb. 8.13 verwendete Multiplexer soll lediglich eine Vektorbildung der Eingangssignale andeuten. Der Regressionsvektor wird anschließend jeder Aktivierungsfunktion bzw. jedem Teilmodell zugeführt. An dem Beispiel aus Abb. 8.13 sollen noch einmal die unterschiedlichen Strukturen bei der Parameter- bzw. Strukturoptimierung und beim Betrieb als Parallelmodell verdeutlicht werden. Die Parameteroptimierung erfolgt in der NARX-Struktur, d.h. für den Regressionsvektor gilt:

$$\underline{x}^T = \left[ u[k-1], u[k-2], y[k-1], y[k-2] \right] \quad (8.6)$$

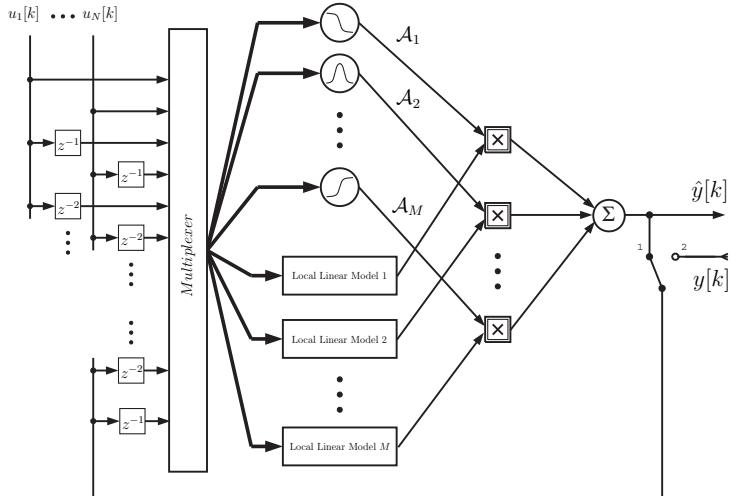
Die Strukturoptimierung hingegen basiert auf einem Gütemaß  $E$ , das in der NOE-Struktur berechnet wird:

$$E = f \left( u[k-1], u[k-2], \hat{y}[k-1], \hat{y}[k-2] \right) \quad (8.7)$$

In der NOE-Struktur wird das identifizierte Parallelmodell auch betrieben, d.h. der Vektor  $\underline{x}$  setzt sich wie folgt zusammen:

$$\underline{x}^T = [u[k-1], u[k-2], \hat{y}[k-1], \hat{y}[k-2]] \quad (8.8)$$

Bisher haben sich die Betrachtungen nur auf SISO–Systeme beschränkt. Das in Kapitel 8.2.1.1 behandelte TDNN sowie der LOLIMOT–Algorithmus sind auch in der Lage MISO–Systeme zu identifizieren. In Abb. 8.14 ist die allgemeine Struktur von LOLIMOT zur Identifikation von MISO–Systemen dargestellt.



**Abb. 8.14:** Allgemeine Struktur von LOLIMOT zur Identifikation von MISO–Systemen

Aus Abb. 8.14 wird offensichtlich, dass die Eingangsdimension des Netzes bei der Identifikation von MISO–Systemen sehr stark ansteigen kann. Um den Anstieg der Eingangsdimension des Netzes einzuschränken und zur besseren Einbringung von Vorwissen kann der Eingangsraum der Aktivierungsfunktionen und der Teilmodelle aufgeteilt werden. Der Eingangsvektor  $\underline{x}$  der lokalen linearen Modelle wird als Vektor der Regelkonklusionen und der Eingangsvektor  $\underline{z}$  der Aktivierungsfunktionen als Vektor der Regelprämissen bezeichnet. Für den Modellausgang  $\hat{y}$  gilt:

$$\hat{y} = \sum_{i=1}^M (\hat{\Theta}_{0,i} + \hat{\Theta}_{1,i} \cdot x_1 + \dots + \hat{\Theta}_{NX,i} \cdot x_{NX}) \cdot \mathcal{A}_i(\underline{z}, \underline{\xi}_i, \sigma_i), \quad (8.9)$$

Die Zugehörigkeitsfunktionen werden nach folgender Gleichung berechnet:

$$\mathcal{A}_i(\underline{z}, \underline{\xi}_i, \sigma_i) = \frac{\mu_i}{\sum_{j=1}^M \mu_j} \quad (8.10)$$

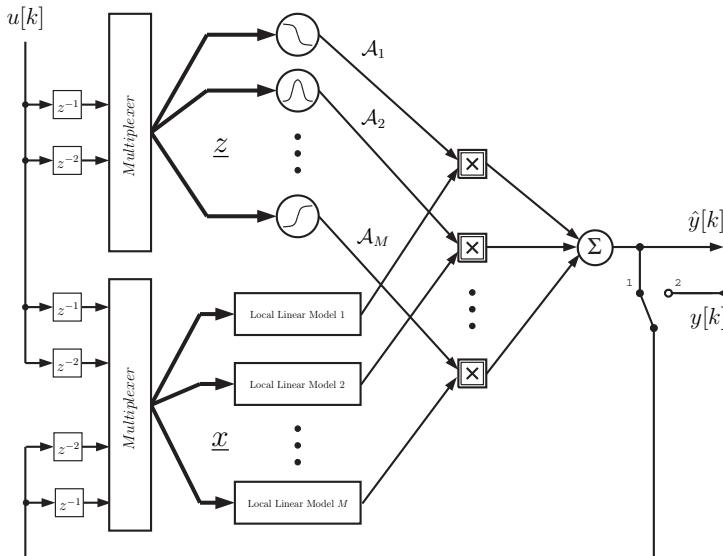
mit

$$\mu_j = \exp \left[ -\frac{1}{2} \left( \frac{(z_1 - \xi_{1,j})^2}{\sigma_{1,j}^2} + \dots + \frac{(z_N - \xi_{N,j})^2}{\sigma_{N,j}^2} \right) \right]. \quad (8.11)$$

Im allgemeinen Fall ist der Vektor der Regelkonklusionen gleich dem Vektor der Regelprämissen, d.h.  $\underline{x} = \underline{z}$ . Durch Vorwissen ist es jedoch möglich, die Dimension von  $\underline{x}$  und  $\underline{z}$  deutlich zu reduzieren. In der Regel muss der Prämissenvektor  $\underline{z}$  nicht alle zeitverzögerten Eingangssignale enthalten, so dass sich die Komplexität der Identifikationsaufgabe verringern lässt. In Abb. 8.15 ist ein Beispiel dargestellt, bei dem der Vektor der Konklusionen  $\underline{x}$  ungleich dem Vektor der Prämissen  $\underline{z}$  ist. Für die beiden Vektoren gilt:

$$\underline{x}^T = [u[k-1], u[k-2], y[k-1], y[k-2]] \quad (8.12)$$

$$\underline{z}^T = [u[k-1], u[k-2]] \quad (8.13)$$



**Abb. 8.15:** LOLIMOT mit Konklusionenvektor  $\underline{x}$  und Prämissenvektor  $\underline{z}$

Durch die Aufteilung des Eingangsvektors in einen Konklusionen- und einen Prämissenvektor lässt sich die Dimension des Eingangsraumes für die Aktivierungsfunktionen und die Teilmodelle auch für Mehrgrößensysteme in Grenzen halten, so dass man relativ schnell ein Identifikationsergebnis erhält und das Ergebnis interpretierbar bleibt.

### 8.2.2 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 7.2.2 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.

#### 8.2.2.1 Volterra–Funktionalpotenzreihe

Die Volterra–Funktionalpotenzreihe zählt zu den ersten allgemeinen Modellsätzen für nichtlineare Systeme mit einem Eingang und einem Ausgang [232]. 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 [37, 123, 131, 194, 237]. Die allgemeinste Form der Volterra–Funktionalpotenzreihe ist in Gl. (8.14) dargestellt.

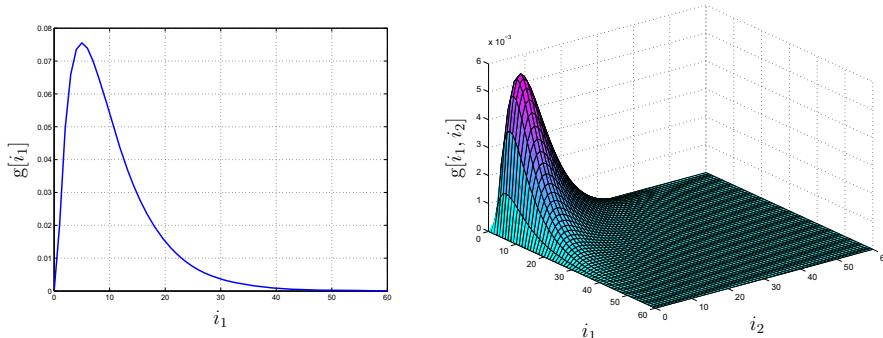
$$\begin{aligned} y[k] = & g_0 + \sum_{i_1=0}^{\infty} g[i_1] u[k - i_1] + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} g[i_1, i_2] u[k - i_1] u[k - i_2] + \dots + \\ & + \sum_{i_1=0}^{\infty} \dots \sum_{i_q=0}^{\infty} g[i_1, \dots, i_q] u[k - i_1] \dots u[k - i_q] \end{aligned} \quad (8.14)$$

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 und stellen die zu adaptierenden Parameter dar. 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. 8.16 veranschaulicht.

Die erste Summe in Gl. (8.14) 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 8.1.2, wie z.B. dem Hammerstein– und dem Wiener–Modell, wird sich in Kapitel 8.2.2.2 zeigen, dass die vollbesetzten Kerne für  $q \geq 2$  symmetrisch sind. Es gilt z.B. für die Volterra–Kerne zweiten und dritten Grades:

$$\begin{aligned} g[i_1, i_2] &= g[i_2, i_1] \\ g[i_1, i_2, i_3] &= g[i_1, i_3, i_2] = g[i_2, i_1, i_3] = \dots \end{aligned} \quad (8.15)$$

Diese Eigenschaft reduziert den Aufwand erheblich, da z.B. bei dem Volterra–Kern zweiter Ordnung aus Abb. 8.16 nur die Kernelemente der Hauptdiagonale und die Kernelemente links oder rechts der Hauptdiagonale zu identifizieren



**Abb. 8.16:** Volterra–Kern ersten (links) und zweiten Grades (rechts)

sind. Allgemein reduziert sich für den Kern  $q$ -ten Grades der Rechenaufwand auf den  $q!$ -ten Teil. Diese Symmetrie wirkt sich in Gl. (8.14) 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:

$$\begin{aligned} y[k] = g_0 + \sum_{i_1=1}^m g[i_1] u[k - i_1] + \sum_{i_1=1}^m \sum_{i_2=i_1}^m g[i_1, i_2] u[k - i_1] u[k - i_2] + \dots + \\ + \sum_{i_1=1}^m \dots \sum_{i_q=i_{q-1}}^m g[i_1, \dots, i_q] u[k - i_1] \dots u[k - i_q] \end{aligned} \quad (8.16)$$

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

$$g_0 = g_0 \quad g[i_1] = g[i_1] \quad (8.17)$$

Für den Volterra–Kern zweiten Grades gilt beispielsweise, dass die Elemente auf der Hauptdiagonalen ( $i_1 = i_2$ ) gleich sind, während alle anderen Elemente doppelt so groß sind (vgl. Abb. 8.16 rechts):

$$g[i_1, i_2] = g[i_1, i_2] \quad \forall \quad i_1 = i_2 \quad g[i_1, i_2] = 2 \cdot g[i_1, i_2] \quad \forall \quad i_1 \neq i_2 \quad (8.18)$$

Anschaulich gesprochen gilt für einen Volterra–Kern zweiten Grades, dass nur Einträge oberhalb der Hauptdiagonalen summiert werden. Die nicht summierten

Einträge unterhalb der Diagonalen werden dadurch berücksichtigt, dass die Einträge oberhalb der Hauptdiagonalen mit zwei multipliziert werden.

Allgemein werden alle Einträge mit der Anzahl der nicht summierten Elemente multipliziert. Diese entspricht der Anzahl der Permutationen der Kernindizes, wie in Gleichung (8.15) ersichtlich. Damit gilt allgemein für einen Kern q-ten Grades:

$$g[i_1 \dots i_q] = g[i_1 \dots i_q] \quad \forall i_1 = \dots = i_q \quad \text{und} \quad g[i_1 \dots i_q] = q! \cdot g[i_1 \dots i_q] \quad \text{sonst.} \quad (8.19)$$

Wird Gl. (8.16) in Vektorschreibweise dargestellt, ergibt sich eine klare Trennung zwischen den unbekannten Parametern, d.h. den Volterra-Kernen und den bekannten Eingangssignalen. Diese bewußt an das GRNN angepasste Schreibweise wird gewählt, um die bereits eingeführten Lernverfahren analog anwenden zu können.

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.20)$$

$\underline{\mathcal{A}}_{dyn}[k]$  wird dabei als dynamischer Aktivierungsvektor bezeichnet und enthält im Gegensatz zum GRNN keine Basisfunktionen sondern nur Eingangssignale.  $\underline{\Theta}$  ist der optimale Parametervektor und beinhaltet den Beharrungswert  $g_0$  und die einzelnen Elemente der Volterra-Kerne. Damit sind alle trainierbaren Parameter wie beim GRNN in  $\underline{\Theta}$  zusammengefaßt. Die Anzahl der Parameter  $p$  des Parametervektors ist abhängig von der Antwortlänge  $m$  und dem Grad  $q$  und ergibt sich zu  $p = \binom{m+q}{q}$ . Der dynamische Aktivierungsvektor  $\underline{\mathcal{A}}_{dyn}[k]$  und der Parametervektor  $\underline{\Theta}$  setzen sich wie folgt zusammen:

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[ 1, u[k-1], \dots, u[k-m], u^2[k-1], u[k-1]u[k-2], \dots, \right. \\ &\quad \left. u^2[k-m], \dots, u^q[k-m] \right] \end{aligned} \quad (8.21)$$

$$\underline{\Theta}^T = \left[ g_0, g[1], \dots, g[m], g[1, 1], g[1, 2], \dots, g[m, m], \dots, g[m, \dots, m] \right]$$

Auf die Parameteranzahl wird in Kapitel 8.2.2.4 noch genauer eingegangen. Zunächst soll gezeigt werden, wie das Hammerstein- und das Wiener-Modell, als blockorientierte Modelle, in der Volterra-Funktionalpotenzreihe enthalten sind.

### 8.2.2.2 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. 8.17 sind beide Modelle dargestellt.

Beim Hammerstein-Modell, befindet sich der nichtlineare Teil der Strecke vor dem linearen Teil. Die Eingangssignale werden durch die statische Nichtlinearität auf einen Ausgangsbereich abgebildet, diese Ausgangssignale werden



**Abb. 8.17:** Hammerstein–Modell (links) und Wiener–Modell (rechts)

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:

$$\begin{aligned} v(u) &= \mathcal{NL}(u) = a_0 + a_1 u + \dots + a_q u^q & y[k] &= \sum_{i=1}^m h[i] v[k-i] \\ y[k] &= a_0 \sum_{i=1}^m h[i] + a_1 \sum_{i=1}^m h[i] u[k-i] + \dots + a_q \sum_{i=1}^m h[i] u^q [k-i] \end{aligned} \quad (8.22)$$

Wird Gl. (8.22) dem vereinfachten Ansatz der Volterra–Funktionalpotenzreihe gegenübergestellt, kann festgestellt werden, dass nur Kernelemente mit gleichen Indizes ungleich Null sind:

$$\begin{aligned} g_0 &= a_0 \sum_{i=1}^m h[i] & g[i] &= a_1 h[i] & \dots & g[i, \dots, i] &= a_q h[i] \\ y[k] &= g_0 + \sum_{i=1}^m g[i] u[k-i] + \dots + \sum_{i=1}^m g[i, \dots, i] u^q [k-i] \end{aligned} \quad (8.23)$$

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 (8.24) gezeigt ist. Die Anzahl der zu bestimmenden Parameter ergibt sich zu  $p = q \cdot m + 1$ .

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.24)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[ 1, u[k-1], \dots, u[k-m], u^2[k-1], \dots, u^2[k-m], \dots, u^q[k-m] \right] \\ \underline{\Theta}^T &= \left[ g_0, g[1], \dots, g[m], g[1, 1], \dots, g[m, m], \dots, g[m, \dots, m] \right] \end{aligned}$$

Im Gegensatz zum Hammerstein–Modell befindet sich beim Wiener–Modell die statische Nichtlinearität hinter dem linearen Streckenteil (vgl. Abb. 8.17

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:

$$\begin{aligned} v[k] &= \sum_{i=1}^m h[i] u[k-i] & y(v) = \mathcal{NL}(v) = a_0 + a_1 v + \dots + a_q v^q \\ y[k] &= a_0 + a_1 \sum_{i_1=1}^m h[i_1] u[k-i_1] + \dots + \\ &+ a_q \sum_{i_1=1}^m \sum_{i_2=1}^m \dots \sum_{i_q=1}^m h[i_1] \dots h[i_q] u[k-i_1] u[k-i_2] \dots u[k-i_q] \end{aligned} \quad (8.25)$$

Wird ein Koeffizientenvergleich zwischen der obigen Gleichung und Gl. (8.16) durchgeführt, kann festgestellt werden, dass das Wiener–Modell genau mit der Volterra–Funktionalpotenzreihe übereinstimmt. Die Kerne der vereinfachten Volterra–Funktionalpotenzreihe aus Gl. (8.16) können beschrieben werden durch:

$$\begin{aligned} g_0 &= a_0 & g[i_1] &= a_1 h[i_1] \\ g[i_1, i_2] &= a_2 h[i_1] h[i_2] \quad \forall \quad i_1 = i_2 & g[i_1, i_2] &= 2 \cdot a_2 h[i_1] h[i_2] \quad \forall \quad i_1 \neq i_2 \\ && \vdots & \\ g[i_1, i_2, \dots, i_q] &= a_q h[i_1] h[i_2] \dots h[i_q] \end{aligned}$$

$$\begin{aligned} y[k] &= g_0 + \sum_{i_1=1}^m g[i_1] u[k-i_1] + \dots + \\ &+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \dots \sum_{i_q=i_{q-1}}^m g[i_1, i_2, \dots, i_q] u[k-i_1] u[k-i_2] \dots u[k-i_q] \end{aligned} \quad (8.26)$$

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

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.27)$$

$$\begin{aligned} \underline{\mathcal{A}}_{dyn}^T[k] &= \left[ 1, u[k-1], \dots, u[k-m], \dots, u[k-m] \dots u[k-m] \right] \\ \underline{\Theta}^T &= \left[ g_0, g[1], \dots, g[m], \dots, g[m, \dots, m] \right] \end{aligned}$$

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

### 8.2.2.3 Eigenschaften der Volterra–Funktionalpotenzreihe

Der Ansatz der diskreten Volterra–Funktionalpotenzreihe zur Identifikation nichtlinearer dynamischer Systeme weist folgende Eigenschaften auf:

- Die Volterra–Funktionalpotenzreihe ist ein allgemeiner nichtlinearer Ansatz, der wenig Vorkenntnisse über das zu identifizierende System erfordert.
- Blockorientierte nichtlineare Modelle, wie das Hammerstein– oder Wiener–Modell, können mittels der Volterra–Funktionalpotenzreihe beschrieben werden. Auch Mischmodelle, wie z.B. das Hammerstein–Wiener–Modell, sind in der Volterra–Funktionalpotenzreihe enthalten. Die Beschreibung von Mehrgrößensystemen kann durch die Volterra–Funktionalpotenzreihe nach Gl. (8.14) nicht erfolgen.
- Der Ansatz ist linear in den Parametern und das Ausgangssignal wird nur aus einer gewichteten Summe von Eingangssignalen gebildet. Daraus ergeben sich zwei wesentliche Besonderheiten dieses Ansatzes:
  1. Dadurch, dass nur das Ausgangssignal und nicht der Regressionsvektor mit Störungen behaftet sein kann, sind die Voraussetzungen für die Anwendung des einfachen Least–Squares–Algorithmus erfüllt, wodurch eine gute und schnelle Konvergenz der Parameteridentifikation erzielt werden kann [131].
  2. Bei der Identifikation wird der Ausgangsfehler und nicht der Gleichungsfehler minimiert. Dadurch entsteht ein echt paralleles Modell.
- Beim Hammerstein– und Wiener–Modell kann die nichtlineare Kennlinie mit geringem Rechenaufwand aus den identifizierten Parametern rekonstruiert werden.
- Die Stabilität des geschätzten Modells ist garantiert. Selbst wenn das Eingangssignal den Wertebereich verlässt, in dem das Modell identifiziert wurde, bleibt der Ausgang immer begrenzt, solange die Eingangssignale begrenzt sind. Deshalb entfällt der für nichtlineare Systeme oft sehr schwierige Stabilitätsbeweis. Dieser Vorteil ist darauf zurückzuführen, dass der Ansatz ohne Ausgangsrückkopplung ist.

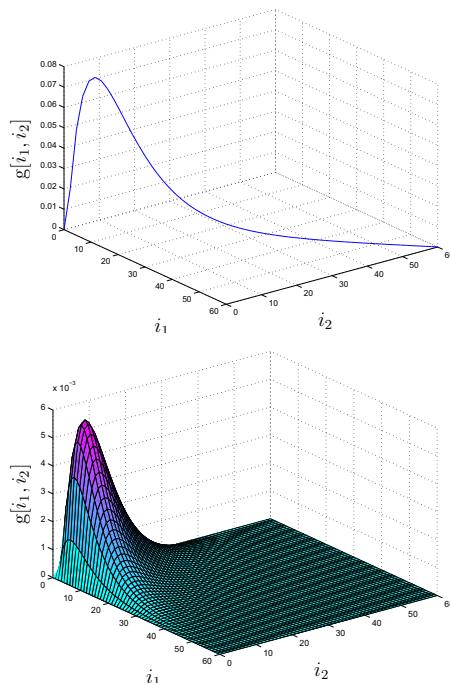
Obwohl die Volterra–Funktionalpotenzreihe einige strukturbedingte Vorteile aufweist, die sie anderen Ansätzen grundsätzlich überlegen erscheinen lässt, konnte sich die Volterra–Funktionalpotenzreihe als Modellansatz zur Identifikation nicht durchsetzen. Das liegt an der großen Anzahl unbekannter Parameter. Folgendes Zahlenbeispiel soll das verdeutlichen. Wählt man den Grad der Nichtlinearität zu  $q = 3$  und die Antwortlänge  $m = 50$ , erhält man

$$p = \binom{m+q}{q} = \binom{50+3}{3} = 23426 \quad (8.28)$$

Parameter, was unmittelbar veranschaulicht, warum sich der Ansatz in dieser Form nicht durchsetzen konnte. Die hohe Anzahl an unbekannten 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.

### 8.2.2.4 Volterra–Funktionalpotenzreihe mit Basisfunktionen

Durch die Einführung von orthonormalen Basisfunktionen lässt sich die Parameteranzahl deutlich verringern. Das identifizierte Modell wird als NOBF–Modell (Nonlinear Orthonormal Basis Function) bezeichnet. Das Prinzip der Parameterreduktion wurde bereits in Kapitel 7.2.2.2 erläutert und kann in analogen Weise zur Parameterreduktion der Volterra–Kerne beitragen. Dies wird am Beispiel des Hammerstein– sowie des Wiener–Modells genauer betrachtet. In Abb. 8.18 sind zwei typische Volterra–Kerne veranschaulicht.



**Abb. 8.18:** Volterra–Kern zweiten Grades eines Hammerstein–Modells (links) und eines Wiener–Modells (rechts)

Abbildung 8.18 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. 8.18 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 untersucht<sup>1)</sup>.

Bei genauer Analyse von Gl. (8.23) wird eine Besonderheit des Hammerstein–Modells deutlich. Die Volterra–Kerne des Hammerstein–Modells sind ausschließlich auf der Hauptdiagonalen besetzt. Jeder Volterra–Kern kann durch eine gewichtete Überlagerung von Basisfunktionen beschrieben werden. Durch die Einführung von 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:

$$\begin{aligned} g[i] &= \sum_{j=1}^{m_r} g_r[j] \tilde{r}_j[i] \\ g[i, i] &= \sum_{j=1}^{m_r} g_r[j, j] \tilde{r}_j[i] \quad \text{mit} \quad m_r \ll m \\ &\vdots \\ g[i, \dots, i] &= \sum_{j=1}^{m_r} g_r[j, \dots, j] \tilde{r}_j[i] \end{aligned} \tag{8.29}$$

Die Parameterreduktion durch die Einführung von Basisfunktionen führt zu:

$$\begin{aligned} y[k] &= g_0 + \sum_{i=1}^m \left( \sum_{j=1}^{m_r} g_r[j] \tilde{r}_j[i] \right) u[k-i] + \dots + \\ &\quad + \sum_{i=1}^m \left( \sum_{j=1}^{m_r} g_r[j, \dots, j] \tilde{r}_j[i] \right) u^q[k-i] \end{aligned} \tag{8.30}$$

Wird Gl. (8.30) in vektorielle Schreibweise überführt, ergibt sich:

---

<sup>1)</sup> An dieser Stelle sei angemerkt, dass auch Modellansätze für das Hammerstein– und Wiener–Modell existieren, bei denen die Übertragungsfunktion durch eine Differenzengleichung beschrieben wird [125, 127]. Dies führt zu einer deutlich geringeren Anzahl an unbekannten Parametern, allerdings treten bei diesen Ansätzen wieder die bereits beschriebenen Probleme von NARX– und NOE–Modellen auf.

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.31)$$

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

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. 8.18 rechts sowie in Gl. (8.25) 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  $\widetilde{\underline{r}}_j$  bzw. der Basisfunktionenmatrix  $\widetilde{\mathbf{R}}$  wie folgt approximiert werden:

$$\begin{aligned} g[i_1] &= \sum_{j_1=1}^{m_r} g_r[j_1] \widetilde{r}_{j_1}[i_1] \\ g[i_1, i_2] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} g_r[j_1, j_2] \widetilde{r}_{j_2}[i_2] \widetilde{r}_{j_1}[i_1] \quad \text{mit} \quad m_r < m \\ &\vdots \\ g[i_1, \dots, i_q] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \dots \sum_{j_q=j_{q-1}}^{m_r} g_r[j_1, \dots, j_q] \widetilde{r}_{j_q}[i_q] \dots \widetilde{r}_{j_1}[i_1] \end{aligned} \quad (8.32)$$

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

$$\begin{aligned} y[k] &= g_0 + \sum_{i_1=1}^m \left( \sum_{j_1=1}^{m_r} g_r[j_1] \widetilde{r}_{j_1}[i_1] \right) u[k - i_1] + \\ &+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \left( \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} g_r[j_1, j_2] \widetilde{r}_{j_2}[i_2] \widetilde{r}_{j_1}[i_1] \right) u[k - i_1] u[k - i_2] + \dots + \\ &+ \sum_{i_1=1}^m \sum_{i_2=i_1}^m \dots \sum_{i_q=i_{q-1}}^m \left( \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \dots \sum_{j_q=j_{q-1}}^{m_r} g_r[j_1, \dots, j_q] \widetilde{r}_{j_q}[i_q] \dots \widetilde{r}_{j_1}[i_1] \right) u[k - i_1] \dots u[k - i_q] \end{aligned} \quad (8.33)$$

Um Gl. (8.33) in eine vektorielle Schreibweise überführen zu können, wurde in [131] ein spezieller Rechenoperator definiert, der sämtliche Kombinationen von Verknüpfungen zwischen den Vergangenheitswerten des Eingangs multipliziert mit den orthonormalen Basisfunktionen erzeugt. Dieser Rechenoperator wird als  $*$ -Operator bezeichnet und ist für einen gegebenen Vektor  $\underline{a} \in \mathbb{R}^{n \times 1}$  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. Allgemein gilt:

$$\underline{a}^{*j} \in \mathbb{R}^{\binom{n+j-1}{j} \times 1}$$

Mit Hilfe dieses  $*$ -Operators ergibt sich für Gl. (8.33):

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.34)$$

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

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 erneut das Wiener-Modell mit einer Antwortlänge von  $m = 50$  und einer Nichtlinearität vom Grad  $q = 3$  betrachtet. Ohne die Einführung von orthonormalen Basisfunktionen waren 23426 Parameter nötig. Werden die Volterra-Kerne durch z.B. sieben Basisfunktionen ( $m_r = 7$ ) approximiert, so ergibt sich die Anzahl der unbekannten Parameter zu  $p = \binom{7+3}{3} = 120$ . Diese deutliche Parameterreduktion ermöglicht erst den Einsatz der Volterra-Funktionalpotenzreihe zur Identifikation nichtlinearer dynamischer Systeme.

### 8.2.2.5 Allgemeiner Ansatz für Wiener– und Hammerstein–Modelle

Durch die Einführung von Basisfunktionen ist die ursprüngliche Allgemeingültigkeit des Ansatzes der Volterra–Funktionalpotenzreihe teilweise verloren gegangen.

Vergleicht man den quadratischen Kern eines Wiener–Modells (Abb. 8.18 rechts) mit dem quadratischen Kern eines Hammerstein–Modells (Abb. 8.18 links) so könnte man annehmen, dass der quadratische Kern des Hammerstein–Modells, bei dem nur die Diagonale besetzt ist, im vollbesetzten Wiener–Kern enthalten ist. Für den Fall, dass die einzelnen Parameter der Volterra–Funktionalpotenzreihe geschätzt werden, also keine Reduktion der Parameteranzahl durch Basisfunktionen stattfindet, ist diese Annahme korrekt. Bei der Identifikation eines Hammerstein–Modells, würden in diesem Fall nur die Parameter bzw. die Elemente der Volterra–Kerne, die sich auf der Diagonalen der Kerne befinden von Null verschieden sein. Alle Kern–Elemente, die sich nicht auf der Diagonalen befinden, sind exakt gleich Null. Da dieses Vorgehen nur theoretisch und aufgrund der zu großen Parameteranzahl nicht praktisch möglich ist, werden Basisfunktionen verwendet. Zur Beschreibung von Wiener–Modellen werden die Wiener–Kerne in  $i_1$ - und  $i_2$ –Richtung durch eine Überlagerung von Basisfunktionen angenähert. Mit dieser Überlagerung ist es nicht möglich einen nur diagonalbesetzten Kern zu approximieren, da dieser in  $i_1$ – und  $i_2$ –Richtung unstetig ist. Aus dem Fourier–Theorem lässt sich ableiten, dass an einer unstetigen Stelle alle Frequenzen zur Approximation durch Sinusfunktionen benötigt werden. Da die Basisfunktionenanzahl beschränkt ist, muss ein allgemeingültiger Ansatz auf eine andere Weise erweitert werden.

Nachdem beim Wiener–Modell Basisfunktionen zur Beschreibung vollbesetzter Kerne in den Ansatz aufgenommen wurden, können auch die Basisfunktionen zur Beschreibung des Hammerstein–Modells in den gleichen Ansatz aufgenommen werden. Damit ist der andere Extremfall, bei dem nur die Hauptdiagonalen der Kerne approximiert werden, auch berücksichtigt.

Fasst man die Gleichungen (8.31) und (8.34) zusammen, entsteht wieder ein allgemeinerer Identifikationsansatz:

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (8.35)$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left[ 1, \underbrace{\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T}_{\text{linearer Anteil}}, \underbrace{\underline{u}^{2T}[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{u}^{qT}[k] \cdot \tilde{\mathbf{R}}^T}_{\text{Hammerstein–Anteil}}, \right. \\ \left. \underbrace{\left( \underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*2}, \dots, \left( \underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T \right)^{*q}}_{\text{Wiener–Anteil}} \right] \quad (8.36)$$

mit dem Parametervektor

$$\underline{\Theta}^T = [g_0, \underline{\Theta}_{lin}^T, \underline{\Theta}_{2,Ham}^T, \dots, \underline{\Theta}_{q,Ham}^T, \underline{\Theta}_{2,Wien}^T, \dots, \underline{\Theta}_{q,Wien}^T] \quad (8.37)$$

Da in beiden Ansätzen (Gl. (8.31) und Gl. (8.34)) der Beharrungswert  $g_0$  und der lineare Teil  $\underline{\Theta}_{lin}$  in gleicher Weise enthalten sind, treten diese, im hier vorgestellten allgemeinen Ansatz, nur einmal auf.

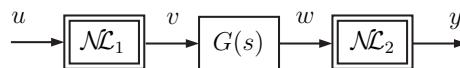
Abhängig von der Anzahl der verwendeten Basisfunktionen  $m_r$  und dem Grad  $q$  des Approximationspolynoms der Nichtlinearität ergibt sich hier die Parameteranzahl zu

$$p = \binom{m_r + q}{q} + (q - 1) \cdot m_r \quad (8.38)$$

Mit dem kombinierten Wiener– und Hammerstein–Modell kann eine Vielzahl technischer Prozesse identifiziert werden [237].

### 8.2.2.6 Erweiterung des Identifikationsansatzes

Wie bereits in Kapitel 8.2.2.5 angedeutet, geht durch die Einführung von Basisfunktionen die Allgemeingültigkeit des Volterra–Ansatzes teilweise verloren. Zwar wurde mit dem Ansatz nach Gl. (8.35), der Hammerstein– und Wiener–Anteile zusammenfasst, eine relativ allgemeine Modellstruktur festgelegt, aber dennoch gibt es Prozesse, die neben den vollbesetzten Kernen (Wiener–Anteile) und den nur diagonalbesetzten Kernen (Hammerstein–Anteile) auch schwachbesetzte Kerne aufweisen, bei denen nur ganz bestimmte Elemente von Null verschieden sind. Ein solcher Prozess ist in Abb. 8.19 dargestellt.



**Abb. 8.19:** Hammerstein–Wiener–Modell

Ähnlich wie in Kapitel 8.2.2.5 sind auch in diesem Fall die Basisfunktionen für Wiener–Anteile nicht geeignet, um die schwachbesetzten Kerne zu approximieren. Da bei den schwachbesetzten Kernen ganz bestimmte Elemente von Null verschieden sind, ist ein anderes Vorgehen bei der Approximation durch Basisfunktionen notwendig. Um für einen solchen Prozess die diskrete Volterra–Funktionalpotenzreihe zu berechnen, wird wie in Kapitel 8.2.2.2 das erste Polynom  $NLL_1$  in die Faltungssumme für das lineare System eingesetzt. Das gewonnene Ergebnis wird anschließend mit dem zweiten Polynom  $NLL_2$  verrechnet. Dies stellt die grundsätzliche Vorgehensweise bei der Erweiterung des Ansatzes dar. Die genaue Herleitung hierzu ist in [131] angegeben. An dieser Stelle soll jedoch nicht weiter darauf eingegangen werden. Es soll der Hinweis genügen, dass für sog. Mischsysteme eine Erweiterung des Ansatzes nötig ist.

### 8.2.2.7 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 werden.<sup>2)</sup> 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. 8.20 ist das Prinzip der Rekonstruktion am Beispiel eines Hammerstein–Modells dargestellt.

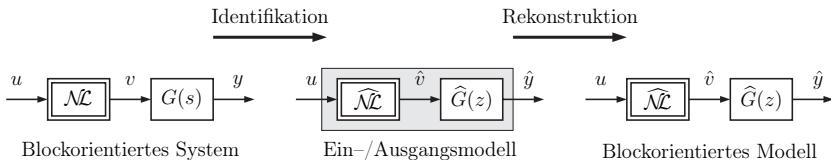


Abb. 8.20: Prinzip der Rekonstruktion einer blockorientierten Modellstruktur

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  $\widehat{\mathcal{N}}$ , linearisiert werden. Außerdem hat ein rekonstruiertes Modell Vorteile hinsichtlich der Rechenzeit. Im Folgenden wird die Rekonstruktion der Struktur des Hammerstein– und des Wiener–Modells genauer betrachtet.

### Rekonstruktion der Hammerstein–Modellstruktur

Das Identifikationsergebnis beschreibt zunächst nur das Ein-/Ausgangsverhalten des Systems. Das Ein-/Ausgangsverhalten ist durch die geschätzten Parameterwerte  $\underline{\Theta}$  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 orthonormaler Basisfunktionen  $m_r$  zu  $p = q \cdot m_r + 1$ . Der identifizierte Parametervektor sieht allgemein wie folgt aus:

$$\underline{\Theta}^T = [\hat{g}_0, \hat{\Theta}[1], \dots, \hat{\Theta}[m_r], \hat{\Theta}[1, 1], \dots, \hat{\Theta}[m_r, m_r], \dots, \hat{\Theta}[m_r, \dots, m_r]] \quad (8.39)$$

Dieser Vektor besitzt entsprechend Gl. (8.31) eine gewisse Struktur, d.h. die Elemente dieses Vektors lassen sich in Gruppen unterschiedlicher Zugehörigkeit

<sup>2)</sup> Diese Aussage gilt insbesondere bei der Einführung orthonormaler Basisfunktionen zur Parameterreduktion.

einteilen. Der erste Parameter beschreibt den Gleichanteil  $g_0$  des Systems, die weiteren  $m_r$  Parameter sind die geschätzten Gewichte  $\hat{\Theta}_i$  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:

$$\hat{\Theta}^T = \left[ \hat{g}_0, \hat{\Theta}_1^T, \dots, \hat{\Theta}_q^T \right] \quad \hat{\Theta}_i \in \mathbb{R}^{m_r \times 1} \quad \text{mit} \quad i = 1 \dots q \quad (8.40)$$

Die Gewichtsfolgen der einzelnen Volterra-Kerne können durch die gewichtete Überlagerung der orthonormalen Basisfunktionen rekonstruiert werden. Für die einzelnen Volterra-Kerne ergibt sich aus Gleichung (8.29):

$$\hat{g}[i] = \sum_{j=1}^{m_r} \hat{\Theta}[j] \tilde{r}_j[i] \quad \hat{g}[i, i] = \sum_{j=1}^{m_r} \hat{\Theta}[j, j] \tilde{r}_j[i] \quad \dots \quad \hat{g}[i, \dots, i] = \sum_{j=1}^{m_r} \hat{\Theta}[j, \dots, j] \tilde{r}_j[i] \quad (8.41)$$

Aus den rekonstruierten Volterra-Kernen können durch einen Vergleich mit der allgemeinen Beschreibung eines Hammerstein-Modells nach Gl. (8.22), 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  $G(s)$  zugeordnet werden kann, wird die Annahme getroffen, dass der Verstärkungsfaktor von  $G(s)$  gleich Eins ist, d.h.  $\sum_{i=1}^m h[i] = 1$ . Für diesen Fall können die Koeffizienten der statischen Nichtlinearität berechnet werden:

$$\hat{a}_0 = \hat{g}_0 \quad \hat{a}_1 = \sum_{i=1}^m \hat{g}[i] \quad \hat{a}_2 = \sum_{i=1}^m \hat{g}[i, i] \quad \dots \quad \hat{a}_q = \sum_{i=1}^m \hat{g}[i, \dots, i] \quad (8.42)$$

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.

$$\hat{h}_k[i] = \frac{1}{\hat{a}_k} \cdot \underbrace{\hat{g}[i, \dots, i]}_k \quad \text{mit } k = 1 \dots q \quad (8.43)$$

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 [134]. 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.

Ausgehend von der Faltungssumme

$$y[k] = \sum_{i=1}^m h[i]u[k-i]$$

sollen die Koeffizienten der diskreten Übertragungsfunktion

$$G(z) = \frac{\hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} + \dots + \hat{b}_n z^{-n}}{1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \dots + \hat{a}_n z^{-n}}$$

bestimmt werden. Dazu wird  $\bar{y}[k] = \mathcal{Z}^{-1}\{Y(z)\} = \mathcal{Z}^{-1}\{G(z)U(z)\}$  gebildet und das Fehlermaß  $(y[k] - \bar{y}[k])^2$  durch geeignete Wahl der Koeffizienten  $\hat{a}_i, \hat{b}_i, i = 1 \dots n$  minimiert. Hierfür kann beispielsweise der in Kapitel (4.2.2) besprochene Recursive Least Squares Algorithmus verwendet werden. Als Lösung des überbestimmten Gleichungssystems ergeben sich die Koeffizienten der rationalen zeitdiskreten Übertragungsfunktion. Der beschriebene Algorithmus zur Rekonstruktion der Hammerstein–Modellstruktur ist noch einmal in Abb. 8.21 veranschaulicht.

Durch die oben beschriebene Rekonstruktion der Struktur konnte das identifizierte Hammerstein–Modell in die statische Nichtlinearität und das lineare dynamische System aufgeteilt werden. Mit der Kenntnis der statischen Nichtlinearität kann der Prozess durch Vorschalten einer inversen Nichtlinearität einfach linearisiert werden. Somit sind wieder die bekannten linearen Regelstrategien anwendbar.

## Rekonstruktion der Wiener–Modellstruktur

Ausgangspunkt für die Rekonstruktion der Wiener–Modellstruktur ist wiederum der geschätzte Parametervektor  $\hat{\Theta}$ . Für das Wiener–Modell ergibt sich eine Parameteranzahl abhängig vom Grad  $q$  der Nichtlinearität  $\mathcal{NL}$  und der Anzahl orthonormaler Basisfunktionen  $m_r$  zu  $p = \binom{m_r+q}{q}$ . Der identifizierte Parametervektor sieht allgemein wie folgt aus:

$$\hat{\Theta}^T = [\hat{g}_0, \hat{\Theta}[1], \dots, \hat{\Theta}[m_r], \hat{\Theta}[1, 1], \hat{\Theta}[1, 2], \dots, \hat{\Theta}[m_r, m_r], \dots, \hat{\Theta}[m_r, \dots, m_r]] \quad (8.44)$$

Dieser Vektor besitzt nach Gl. (8.34) ebenfalls eine gewisse Struktur. Die Vektoren  $\hat{\Theta}_i$  in Gl. (8.45) beinhalten die Gewichte der orthonormalen Basisfunktionen zur Rekonstruktion des Volterra–Kerns  $i$ –ten Grades:

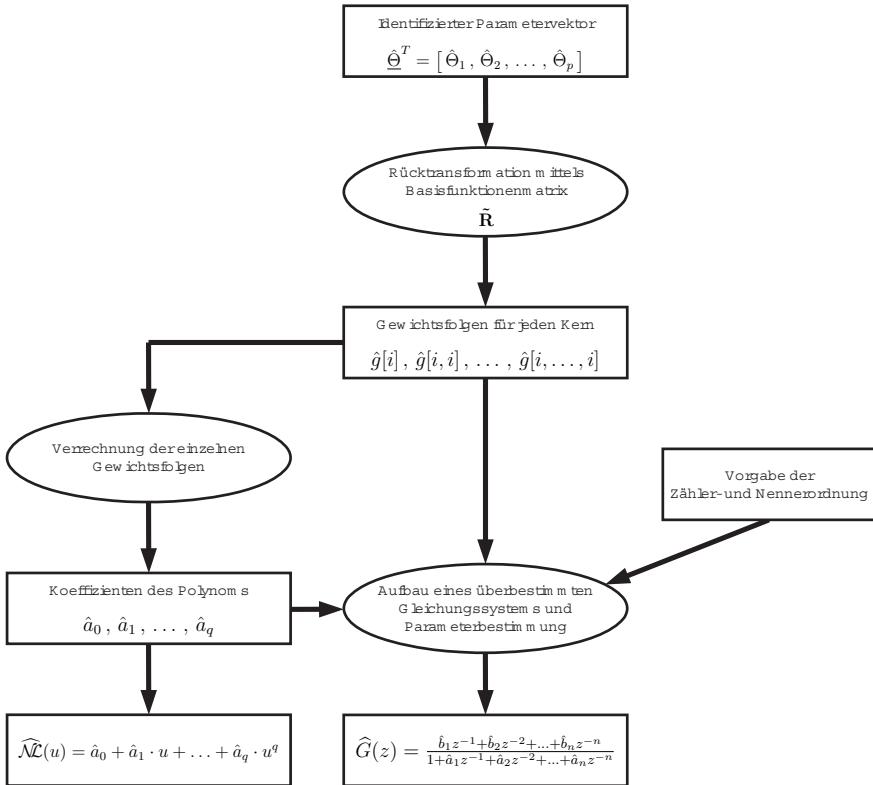


Abb. 8.21: Rekonstruktion der Hammerstein-Modellstruktur

$$\hat{\Theta}^T = \left[ \hat{g}_0, \hat{\Theta}_1^T, \dots, \hat{\Theta}_q^T \right] \quad \text{mit} \quad \hat{\Theta}_i \in \mathbb{R}^{[(m_r+i) - (m_{r+1}-1)] \times 1} \quad i = 1 \dots q \quad (8.45)$$

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:

$$\begin{aligned}
\hat{g}[i_1] &= \sum_{j_1=1}^{m_r} \hat{\Theta}[j_1] \tilde{r}_{j_1}[i_1] \\
\hat{g}[i_1, i_2] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \hat{\Theta}[j_1, j_2] \tilde{r}_{j_2}[i_2] \tilde{r}_{j_1}[i_1] \\
&\vdots \\
\hat{g}[i_1, \dots, i_q] &= \sum_{j_1=1}^{m_r} \sum_{j_2=j_1}^{m_r} \dots \sum_{j_q=j_{q-1}}^{m_r} \hat{\Theta}[j_1, \dots, j_q] \tilde{r}_{j_q}[i_q] \dots \tilde{r}_{j_1}[i_1]
\end{aligned} \tag{8.46}$$

Aus den rekonstruierten Volterra-Kernen können durch einen Vergleich mit der allgemeinen Beschreibung eines Wiener-Modells nach Gl. (8.25), 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:

$$\begin{aligned}
\hat{a}_0 &= \hat{g}_0 & \hat{a}_1 &= \sum_{i_1=1}^m \hat{g}[i_1] \\
\hat{a}_2 &= \sum_{i_1=1}^m \sum_{i_2=i_1}^m \hat{g}[i_1, i_2] & \dots & \hat{a}_q &= \sum_{i_1=1}^m \dots \sum_{i_q=i_{q-1}}^m \hat{g}[i_1, \dots, i_q]
\end{aligned} \tag{8.47}$$

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:

$$\hat{h}[i] = \frac{1}{\hat{a}_1} \cdot \hat{g}[i_1] \tag{8.48}$$

Mit den Gleichungen (8.47) und (8.48) 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.

### 8.2.3 Anregungssignale zur Identifikation

Einen entscheidenden Einfluss auf die Güte der Identifikation hat die Systemanregung. Mit einem konstanten Eingangssignal kann logischerweise keine Dynamik identifiziert werden. Die Suche nach einem für die Identifikationsaufgabe optimalen Eingangssignal gestaltet sich in der Regel als schwierig. Prinzipiell müssen jedoch immer folgende Forderungen erfüllt sein:

1. Der Eingangsraum der statischen Nichtlinearität bzw. des nichtlinearen dynamischen Systems muss vollständig angeregt werden. Anschaulich bedeutet dies, dass nur dort gelernt werden kann, wo auch Messdaten vorhanden sind. Diese Forderung ist gerade bei hochdimensionalen Identifikationsaufgaben schwierig zu erfüllen.
2. Das Anregesignal muss mindestens so viele Frequenzen enthalten, wie unbekannte Parameter im System zu bestimmen sind. Ein einfaches Sinusignal mit einer Frequenz ist in der Regel nicht ausreichend. Der Fehler geht zwar im Allgemeinen gegen Null und auch die Parameter des Identifikationsmodells konvergieren, jedoch handelt es sich in der Regel um eine lokale Lösung, die lediglich für diese spezielle Anregung gilt. Das bedeutet, dass diese Lösung keine Allgemeingültigkeit besitzt und die Modellgüte für andere Anregesignale sehr schlecht ist.

Im Folgenden sollen an zwei Beispielen die grundsätzlichen Eigenschaften, die Anregesignale erfüllen sollten, verdeutlicht werden. In Abb. 8.22 ist die Anregung eines Hammerstein-Prozesses dargestellt.

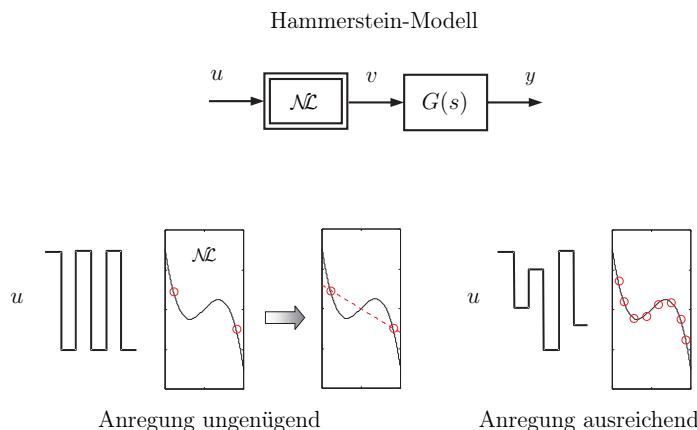
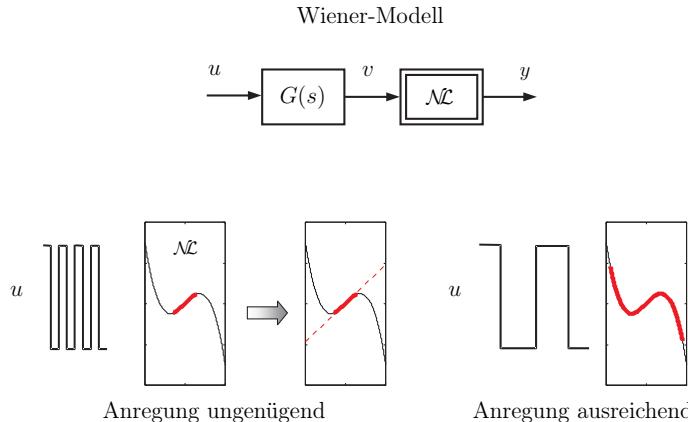


Abb. 8.22: Anregung eines Hammerstein-Prozesses

Wird als Anregung ein Rechtecksignal mit konstanter Amplitude gewählt, so wird die Nichtlinearität nur an zwei Stützstellen angeregt. Mathematisch ist es jedoch nicht möglich, durch die Vorgabe von nur zwei Punkten eine kubische Funktion, wie in diesem Beispiel vorgegeben, zu bestimmen. Dafür sind mindestens vier Punkte notwendig. Durch die Vorgabe von zwei Punkten ist nur eine Gerade eindeutig bestimmt. Dies wird bei der gewählten Anregung vom Identifikationsalgorithmus auch so angenommen und es entsteht ein Modell, das mit

dem zu identifizierenden Prozess nicht übereinstimmt. Es ist demnach sinnvoll, den Prozess mit einem Rechtecksignal unterschiedlicher Amplitude anzuregen.

Bei der Anregung eines Wiener-Prozesses entsteht ein anderes Problem, was in Abb. 8.23 schematisch dargestellt ist.

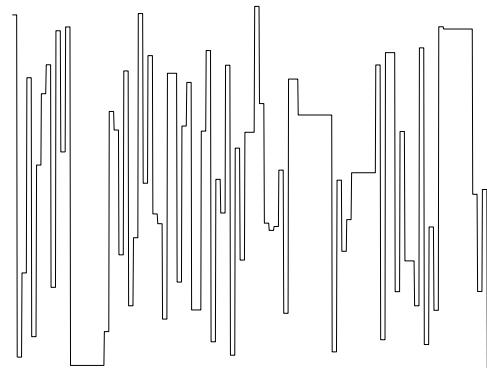


**Abb. 8.23:** Anregung eines Wiener-Prozesses

Die statische Nichtlinearität wird durch das vorgeschaltete lineare dynamische System angeregt. Bei zu kurzer Haltezeit der Sprunganregung wird die gewählte Eingangsamplitude durch das lineare System, abhängig von den Zeitkonstanten von  $G(s)$ , mehr oder weniger stark gedämpft, so dass die nachfolgende statische Nichtlinearität möglicherweise nur in einem sehr kleinen Bereich angeregt wird. Auch in diesem Fall wird ein Modell geschätzt, das außerhalb des angeregten Bereichs nicht mit dem vorgegebenen Prozess übereinstimmt. Die Haltezeit sollte abhängig von der dominanten Zeitkonstante des linearen dynamischen Systems in jedem Fall so groß gewählt werden, dass der gesamte Bereich der statischen Nichtlinearität angeregt wird.

Da man in der Realität selten im Voraus weiß, welche Prozessstruktur vorliegt, ist es sinnvoll ein Anregesignal zu verwenden, das universell geeignet ist. Ein solches Anregesignal ist das amplitudenmodulierte Pseudo-Rausch-Binär-Signal (APRBS) [111, 44], das in Abb. 8.24 dargestellt ist.

Durch das rechteckförmige Anregesignal mit unterschiedlicher Haltezeit und Amplitude wird garantiert, dass alle relevanten Frequenzen und Amplituden des Prozesses ausreichend angeregt werden. Die Realisierung dieses Signals erfolgt mittels rückgekoppelter Schieberegister [111, 112]. Ein Vergleich verschiedener Anregesignale hinsichtlich ihrer Eignung zur Identifikation ist in [117] zu finden.



**Abb. 8.24:** Amplitudenmoduliertes Pseudo–Rausch–Binär–Signal

### 8.3 Zusammenfassung

In diesem Kapitel wurde ein Einblick in die Identifikation nichtlinearer dynamischer Systeme gegeben. Ausgehend von der Identifikation linearer dynamischer Systeme wurden verschiedene Ansätze zur Identifikation nichtlinearer dynamischer Systeme vorgestellt. Ein wichtiges Unterscheidungsmerkmal war dabei der Grad an strukturellem Vorwissen, der von dem zu identifizierenden System vorhanden war.

In diesem Kapitel wurde speziell auf die Identifikationsansätze mit externer Dynamikmodellierung eingegangen. Dabei kann zwischen Modellen mit und ohne Ausgangsrückkopplung unterschieden werden. Die nichtlinearen Modelle mit Ausgangsrückkopplung benötigen wenig strukturelles Vorwissen, was zu Ergebnissen führt, die nur bedingt physikalisch interpretierbar sind. Bei den nichtlinearen Modellen ohne Ausgangsrückkopplung wurde die Volterra–Funktionalpotenzreihe als eine allgemeine Beschreibung nichtlinearer dynamischer Systeme vorgestellt. Die große Parameteranzahl der Volterra–Funktionalpotenzreihe machte die Einführung von orthonormalen Basisfunktionen erforderlich. Es wurde gezeigt, wie blockorientierte nichtlineare Systeme, wie das Hammerstein– oder Wiener–Modell mit diesem Ansatz identifiziert werden können. Am Beispiel des Hammerstein–Modells und des Wiener–Modells erfolgte eine Rekonstruktion der blockorientierten Struktur. Abschließend wurde die Bedeutung der Anregung für die Identifikation erläutert und zwingend erforderliche Eigenschaften, die ein Anregesignal erfüllen muss, dargelegt.

Zusammenfassend kann festgehalten werden, dass es einen allgemeingültigen Ansatz, der zur Identifikation aller nichtlinearen dynamischen Systeme geeignet ist, nicht gibt. Im Einzelfall muss immer geprüft werden, welcher Ansatz am besten geeignet ist, die gestellte Identifikationsaufgabe zu erfüllen. Viele unter-

schiedliche Aspekte müssen berücksichtigt werden, bevor man sich letztendlich für ein bestimmtes Verfahren entscheidet.

## 9 Beobachterentwurf für Systeme mit dynamischen Nichtlinearitäten

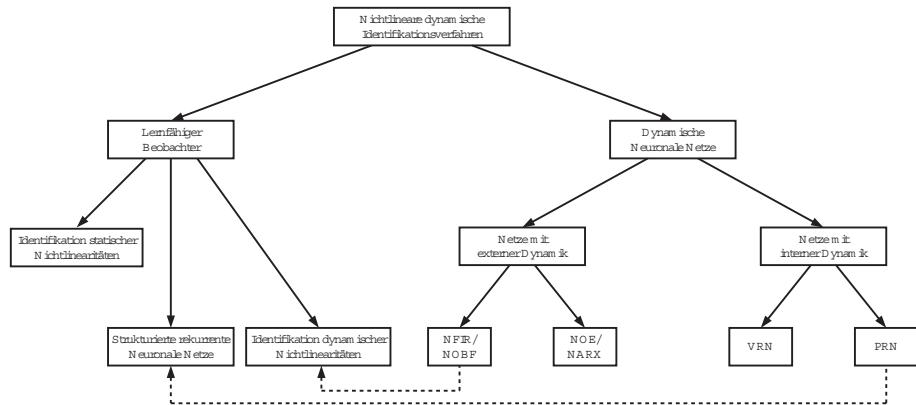
In Kapitel 8.2.2 wurde ausführlich die Identifikation von Hammerstein- und Wiener-Modell auf Basis der Volterra-Funktionalpotenzreihe behandelt. Dabei wurde davon ausgegangen, dass sowohl das Eingangssignal als auch das Ausgangssignal unmittelbar zur Verfügung stehen. In der Regel sind diese blockorientierten Systeme jedoch in ein komplexeres Gesamtsystem eingebettet, so dass das Eingangssignal nicht direkt vorgegeben bzw. das Ausgangssignal nicht direkt gemessen werden kann.

Für den Fall einer unbekannten statischen Nichtlinearität wurde für dieses Problem der lernfähige Beobachter (vgl. Kapitel 5) entwickelt. Dieser erlaubt es, statische Nichtlinearitäten innerhalb eines bekannten linearen dynamischen Systems zu identifizieren. Die Nachteile des lernfähigen 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 bei komplexeren Systemen in der Praxis ein. Aus diesem Grund konzentrierten sich nachfolgende Forschungsarbeiten auf dem Gebiet des lernfähigen Beobachters darauf, die ursprünglich getroffenen Einschränkungen aufzuheben.

In Kapitel 6 wurden neben den statischen Nichtlinearitäten auch die linearen Parameter des in seiner Struktur vollständig bekannten Systems identifiziert. Hierzu wurde das Blockschaltbild eines Systems in ein strukturiertes rekurrentes Neuronales Netz (SRNN) überführt.

In diesem Kapitel soll der lernfähige Beobachter in der Richtung erweitert werden, dass auch sog. dynamische Nichtlinearitäten identifiziert werden können. Anschaulich bedeutet dies, dass nicht mehr die gesamte Struktur des Systems exakt bekannt sein muss, sondern dass für ein Teilsystem eine grobe Strukturkenntnis ausreichend ist.<sup>1)</sup> In Abb. 9.1 sind die unterschiedlichen Erweiterungen des lernfähigen Beobachters veranschaulicht. Dynamische Nichtlinearitäten sind nichtlineare blockorientierte Systeme, wie sie in den Kapiteln 8.1.2 und 8.2.2 schon genauer betrachtet wurden. Hierbei handelte es sich jedoch ausschließlich um blockorientierte Systeme mit einem Eingang und einem Ausgang. Es sind natürlich auch blockorientierte Systeme mit mehreren Eingangssignalen und einem Ausgang denkbar. Auch solche Systeme können mit entsprechenden Er-

<sup>1)</sup> Theoretisch ist es auch möglich, Teilsysteme ohne jegliche Vorkenntnisse über die Struktur zu identifizieren. Dies führt in der Praxis jedoch zu einer sehr hohen Parameteranzahl.



**Abb. 9.1:** Erweiterungen des Neuronalen Beobachters

weiterungen durch die Volterra–Funktionalpotenzreihe beschrieben werden. Dies ist jedoch noch Thema von aktuellen Forschungsarbeiten. An dieser Stelle sei angemerkt, dass die Beschreibung von Mehrgrößensystemen auf der Basis der Volterra–Funktionalpotenzreihe möglich ist, aber dass solche Systembeschreibungen bereits bei relativ einfachen Mehrgrößensystemen sehr komplex werden können. Eine allgemeingültige Beschreibung von Mehrgrößensystemen wird dadurch sehr schwierig. Darauf wird in Kapitel 9.1 noch einmal eingegangen, wo auch eine genaue Definition des Begriffs der dynamischen Nichtlinearität erfolgt.

Die grundlegende Idee der Erweiterung des lernfähigen 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 unbekannte nichtlineare dynamische Teilsystem stellt die dynamische Nichtlinearität dar. Das grobe Strukturvorwissen 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 lernfähige Beobachter zur Identifikation dynamischer Nichtlinearitäten ist somit immer dann vorteilhaft einzusetzen, wenn die exakte Modellierung eines unbekannten nichtlinearen Teilsystems aufgrund 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 lernfähige Beobachter zur Identifikation statischer Nichtlinearitäten ist als Spezialfall im erweiterten Beobachterentwurf enthalten.

## 9.1 Systeme mit dynamischen Nichtlinearitäten

Bevor 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 beispielsweise in den Abbildungen 9.2 und 9.3 für zwei Eingangsgrößen dargestellt sind, verstanden.

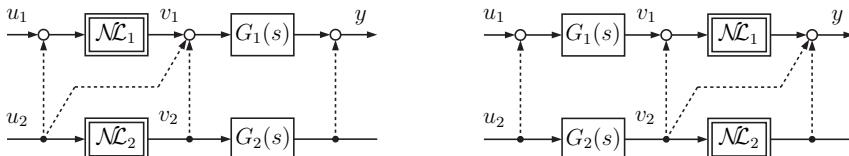


Abb. 9.2: Linear verkoppelte Hammerstein- (links) und Wiener-Modelle (rechts)

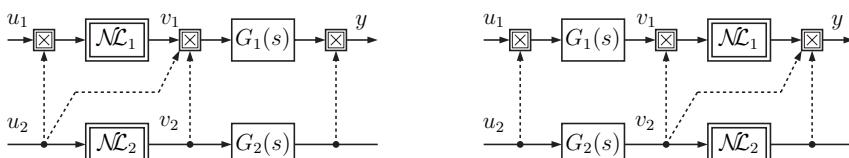
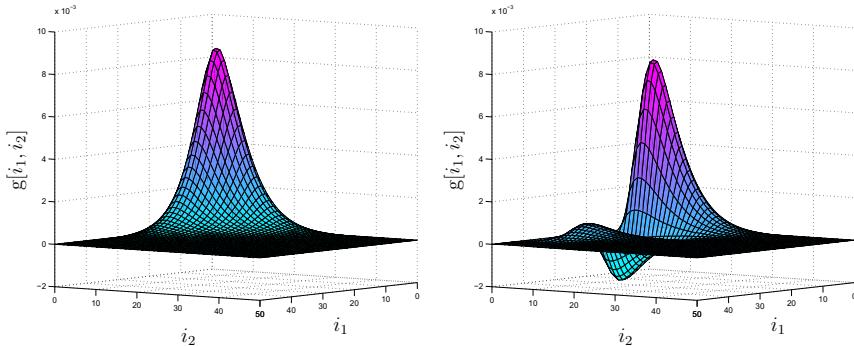


Abb. 9.3: Nichtlinear verkoppelte Hammerstein- (links) und Wiener-Modelle (rechts)

Bei der Identifikation blockorientierter Mehrgrößensysteme wie sie in den Abbildungen 9.2 und 9.3 dargestellt sind, können neben linearen, diagonalbesetzten und symmetrischen Volterra-Kernen (vgl. Kapitel 8.2.2) auch unsymmetrische Volterra-Kerne auftreten. In Abb. 9.4 sind ein symmetrischer Volterra-Kern und ein unsymmetrischer Volterra-Kern zweiter Ordnung einander gegenübergestellt.



**Abb. 9.4:** Symmetrischer Volterra–Kern zweiter Ordnung (links) und unsymmetrischer Volterra–Kern zweiter Ordnung (rechts)

Der unsymmetrische Volterra–Kern ist in  $i_1$ – und in  $i_2$ –Richtung nicht mehr symmetrisch. Er entsteht durch die Multiplikation zweier unterschiedlicher Impulsantworten, wie dies bei der Beschreibung von Mehrgrößensystemen auftreten kann. Die Identifikation von unsymmetrischen Volterra–Kernen kann ebenfalls mit Hilfe von orthonormalen Basisfunktionen erfolgen [92].

Nachdem der Begriff der dynamischen Nichtlinearität veranschaulicht wurde, wird nun ein System mit dynamischer Nichtlinearität definiert. Als System mit separierbarer dynamischer Nichtlinearität wird eine Strecke bezeichnet, die sich nach folgender Zustandsdarstellung beschreiben lässt:<sup>2)</sup>

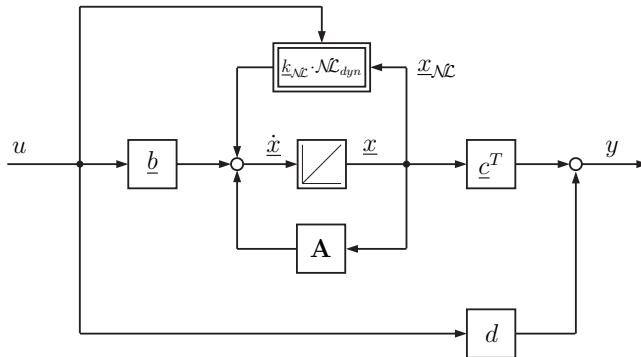
$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A} \cdot \underline{x} + \underline{b} \cdot u + k_{\mathcal{N}\mathcal{L}} \cdot \mathcal{N}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \\ y &= \underline{c}^T \cdot \underline{x} + d \cdot u\end{aligned}\tag{9.1}$$

Der Vektor  $k_{\mathcal{N}\mathcal{L}}$  bezeichnet einen konstanten Einkopplungsvektor. In Abb. 9.5 ist das Blockschaltbild eines Systems mit separierbarer dynamischer Nichtlinearität veranschaulicht.

In Gl. (9.1) bzw. Abb. 9.5 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 Streckenanteils,
- $\underline{b} \in \mathbb{R}^{n \times 1}$  den Einkopplungsvektor des Systemeingangs,
- $\underline{x}_{\mathcal{N}\mathcal{L}}$  den Vektor der Zustandsgrößen von denen die dynamische Nichtlinearität abhängt,

<sup>2)</sup> Hier wird ausschließlich der Fall einer SISO–Strecke betrachtet.



**Abb. 9.5:** Blockschaltbild eines Systems mit dynamischer Nichtlinearität

- $\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u)$  die dynamische Nichtlinearität,
- $k_{\mathcal{N}\mathcal{L}} \in \mathbb{R}^{n \times 1}$  den Einkopplungsvektor der dynamischen Nichtlinearität,
- $c \in \mathbb{R}^{n \times 1}$  den Auskopplungsvektor,
- $d$  den Durchgriff des Systemeingangs auf den Systemausgang und
- $y$  den skalaren Systemausgang.

In Gl. (9.1) ist deutlich die Trennung zwischen dem bekannten linearen Teilsystem und der unbekannten dynamischen Nichtlinearität zu erkennen.<sup>3)</sup> 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  $G(s)$  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

<sup>3)</sup> Es wird deshalb auch von einer separierbaren Nichtlinearität gesprochen.

linearen Zustandsdarstellung zugeordnet werden, jedoch wären die linearen Parameter nicht vollständig bekannt. Das bedeutet, dass der lernfähige 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 Kapitel 6 das SRNN vorgestellt, das somit auf dieses Problem angewendet werden könnte.

## 9.2 Beobachterentwurf

Zur Anwendung der im Folgenden beschriebenen Beobachtertheorie werden zunächst alle notwendigen Voraussetzungen noch einmal zusammengefasst<sup>4)</sup>:

- Das betrachtete System ist gemäß Gl. (9.1) bzw. Abb. 9.5 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 zustandsbeobachtbar.<sup>5)</sup> Dies ist genau dann der Fall, wenn die Beobachtbarkeitsmatrix  $\mathbf{Q}_{obs}$  regulär ist, d.h. wenn gilt:

$$\det(\mathbf{Q}_{obs}) \neq 0 \quad \text{mit} \quad \mathbf{Q}_{obs} = [\underline{c}, \mathbf{A}^T \underline{c}, \dots, (\mathbf{A}^T)^{(n-1)} \underline{c}] \quad (9.2)$$

- Der Einkopplungsvektor  $k_{\mathcal{N}\mathcal{L}}$  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 genau dann erfüllt, wenn gilt:

$$H_S(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot k_{\mathcal{N}\mathcal{L}} \neq 0 \quad \forall s \geq 0 \quad (9.3)$$

- 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.

---

<sup>4)</sup> Diese Voraussetzungen decken sich größtenteils mit denen des lernfähigen Beobachters zur Identifikation statischer Nichtlinearitäten [221].

<sup>5)</sup> Dies muss im Falle eines nicht messbaren Eingangsraumes der dynamischen Nichtlinearität vorausgesetzt werden.

Die nachfolgenden Ausführungen zum Beobachterentwurf erfolgen in enger Anlehnung an [137, 221]. Der Beobachterentwurf erfolgt dabei in einer kontinuierlichen Darstellungsform. Im Gegensatz dazu geht der eigentliche Identifikationsalgorithmus von 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–Approximation.<sup>6)</sup>

### 9.2.1 Beobachterentwurf bei messbarem Eingangsraum der dynamischen Nichtlinearität

Zunächst wird davon ausgegangen, dass der Eingangsraum der dynamischen Nichtlinearität  $\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{N}}, u)$  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. (9.1) wird ein Zustandsbeobachter nach Luenberger angesettzt, der um den Approximationsalgorithmus für die dynamische Nichtlinearität  $\widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{N}}, u)$  erweitert ist:<sup>7)</sup>

$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A} \cdot \dot{\underline{x}} + \underline{b} \cdot u + \underline{k}_{\mathcal{NL}} \cdot \widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{N}}, u) - \underline{l} \cdot e \\ \hat{y} &= \underline{c}^T \cdot \dot{\underline{x}} + d \cdot u\end{aligned}\tag{9.4}$$

Der Beobachterfehler ist dabei definiert zu:

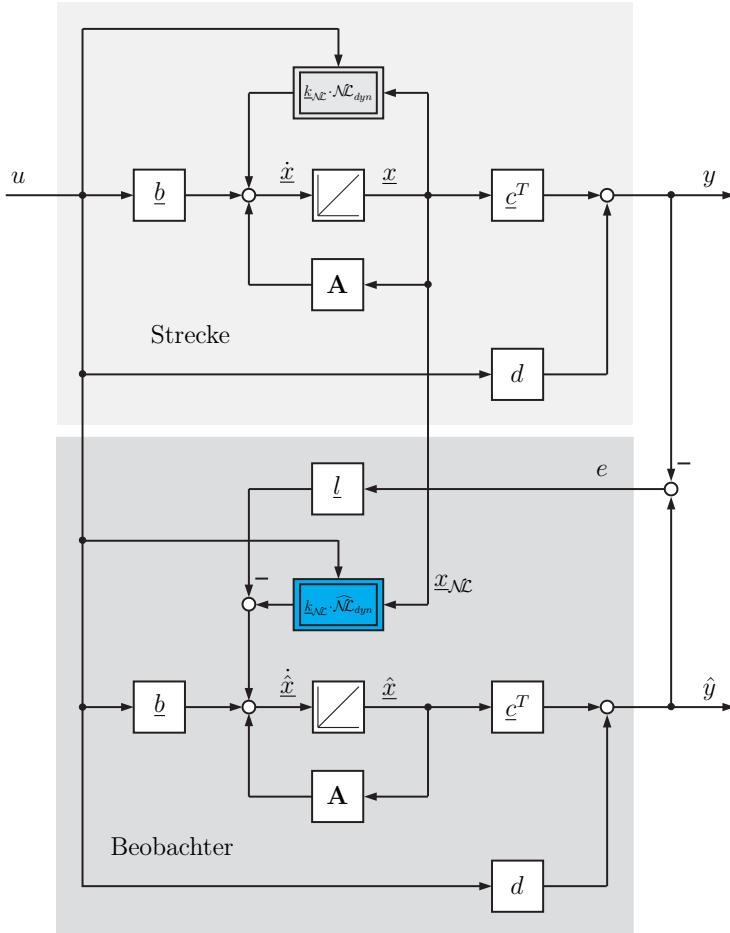
$$e = \hat{y} - y\tag{9.5}$$

Für die Dimensionierung des Beobachtervektors  $\underline{l}$  gelten alle bekannten Einstellvorschriften für den linearen Beobachterentwurf [142] (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. 9.6 ist die resultierende Struktur von System und Beobachter in Zustandsdarstellung verdeutlicht. Zur Ableitung eines stabilen Lerngesetzes ist die Bestimmung der Fehlerübertragungsfunktion  $H(s)$  nötig. Die Fehlerübertragungsfunktion

---

<sup>6)</sup> Durch Verwendung der Euler–Vorwärts–Approximation könnte die zeitkontinuierliche Darstellungsform des Beobachters sehr einfach diskretisiert werden und somit wäre der formale Widerspruch gelöst.

<sup>7)</sup> Bei diesem Ansatz wird die dynamische Nichtlinearität als Eingang des linearen Teilsystems interpretiert, so dass weiter mit den bekannten linearen Methoden gearbeitet werden kann [137].



**Abb. 9.6:** Zustandsbeobachter bei messbarem Eingangsraum der dynamischen Nichtlinearität

$$H(s) = \frac{e(s)}{e_{NL}(s)} = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + \underline{l} \cdot \underline{c}^T)^{-1} \cdot \underline{k}_{NL} \quad (9.6)$$

beschreibt das Übertragungsverhalten des Beobachters vom Angriffspunkt der dynamischen Nichtlinearität zum Systemausgang. Der Fehler  $e_{NL}$  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 5 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 5 gegeben [158].

Für die Ableitung eines stabilen Lerngesetzes ist eine zwingende Voraussetzung, dass der Identifikationsalgorithmus für die dynamische Nichtlinearität linear in den unbekannten Parametern ist. 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:

$$\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) = \underline{\Theta}^T \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \quad (9.7)$$

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. Der Ausgang des Approximationsalgorithmus berechnet sich zu:

$$\widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) = \widehat{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \quad \text{siehe Gl. (8.20)} \quad (9.8)$$

Für den Beobachterfehler  $e$  ergibt sich unter der Annahme, dass sich  $\underline{\Theta}$  nur langsam ändert:

$$e(s) = H(s) \cdot \mathcal{L} \left\{ \left( \widehat{\underline{\Theta}}^T - \underline{\Theta}^T \right) \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \quad (9.9)$$

Wird der Parameterfehlervektor  $\Phi = \widehat{\underline{\Theta}} - \underline{\Theta}$  eingesetzt, folgt die in der Literatur [158] bekannte Fehlergleichung:

$$e(s) = H(s) \cdot \mathcal{L} \{ \Phi^T \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \} \quad (9.10)$$

Erfüllt die Fehlerübertragungsfunktion  $H(s)$  die SPR–Bedingung, kann zur Adaption der unbekannten Parameter das Fehlermodell 3 nach [158] verwendet werden. Für die Adoptionsgleichung gilt im Zeitbereich:<sup>8)</sup>

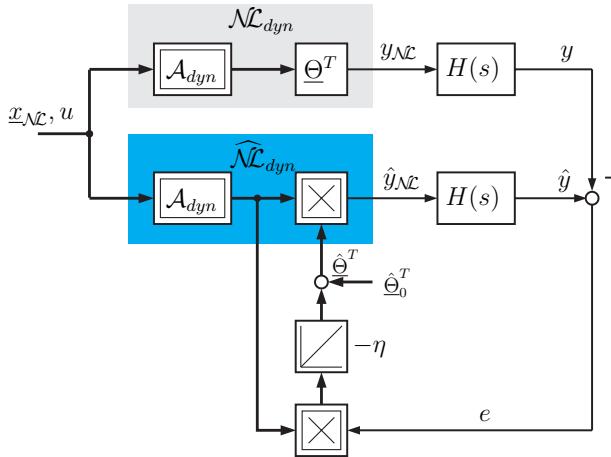
$$\dot{\underline{\Phi}} = \dot{\widehat{\underline{\Theta}}} = -\eta \cdot e \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \quad \eta > 0 \quad (9.11)$$

In Abb. 9.7 ist das Fehlermodell 3 noch einmal veranschaulicht.<sup>9)</sup> 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 [158] als Fehlermodell 4 bezeichnete Lerngesetz verwendet werden. Für dieses Lerngesetz gilt:

<sup>8)</sup> Es gilt  $e(s) = \mathcal{L}\{e\}$ .

<sup>9)</sup> In den Abbildungen 9.7 und 9.8 werden aus Gründen einer besseren Übersicht die Abkürzungen  $y_{\mathcal{N}\mathcal{L}}(s) = \mathcal{L}\{\mathcal{NL}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u)\}$  und  $\hat{y}_{\mathcal{N}\mathcal{L}}(s) = \mathcal{L}\{\widehat{\mathcal{NL}}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u)\}$  eingeführt.



**Abb. 9.7:** Lerngesetz nach Fehlermodell 3 zur Adaption von  $\widehat{\mathcal{N}}_{dyn}$

$$\begin{aligned} \dot{\underline{\Theta}} &= \dot{\underline{\Theta}} = -\eta \cdot \epsilon \cdot \mathcal{L}^{-1} \left\{ \underbrace{H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\underline{x}_{NL}, u) \}}_{\text{verzögerte dynamische Aktivierung}} \right\} \\ &= -\eta \cdot \epsilon \cdot \mathcal{L}^{-1} \{ \mathcal{A}_{Vdyn}(s) \} \quad \eta > 0 \end{aligned} \quad (9.12)$$

Das Lerngesetz nach Fehlermodell 4 wird auch als Verfahren der verzögerten Aktivierung bezeichnet [137]. Im Fall des Fehlermodells 4 wird der Beobachterfehler  $e = e_1$  gesetzt und ein zusätzlicher Fehler  $e_2$  eingeführt wird. Zur Adaption der Parameter wird der erweiterte Fehler  $\epsilon$  gebildet, der sich aus dem Beobachterfehler  $e_1$  und einem zusätzlichen Fehler  $e_2$  zusammensetzt. Für den erweiterten Fehler  $\epsilon$  ergibt sich somit:<sup>10)</sup>

$$\epsilon(s) = e_1(s) + e_2(s) \quad (9.13)$$

mit<sup>11)</sup>

$$e_2(s) = \underline{\Theta}^T \cdot H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\underline{x}_{NL}, u) \} - H(s) \cdot \mathcal{L} \{ \underline{\Theta}^T \cdot \mathcal{A}_{dyn}(\underline{x}_{NL}, u) \} \quad (9.14)$$

Das Fehlermodell 4 ist in Abb. 9.8 veranschaulicht.

Die verzögerte dynamische Aktivierung bewirkt, dass der dynamische Aktivierungsvektor  $\mathcal{A}_{dyn}(\underline{x}_{NL}, u)$  so lange verzögert wird, bis die Auswirkungen des Fehlers  $e_{NL}$  im Beobachterfehler  $e_1$  sichtbar werden. Der erweiterte Fehler  $\epsilon$  wird gebildet, um die dynamischen Auswirkungen der Adaption der unbekannten Parameter im Fehler zu kompensieren.

<sup>10)</sup> Es gilt  $\epsilon(s) = \mathcal{L}\{\epsilon\}$ .

<sup>11)</sup> Es wird angenommen, dass sich der Parametervektor  $\underline{\Theta}$  nur sehr langsam ändert.

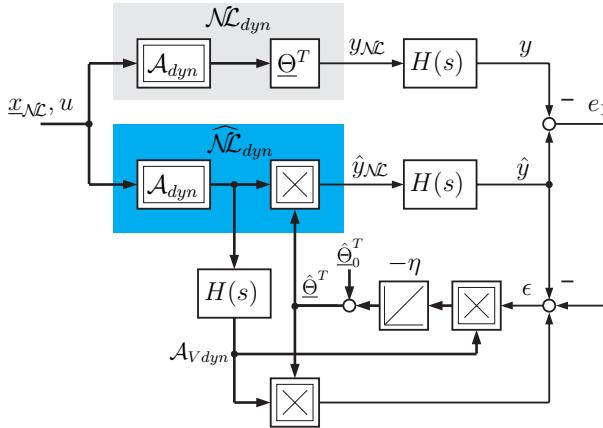


Abb. 9.8: Lerngesetz nach Fehlermodell 4 zur Adaption von  $\widehat{\mathcal{N}}_{dyn}$

### 9.2.2 Beobachterentwurf bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität

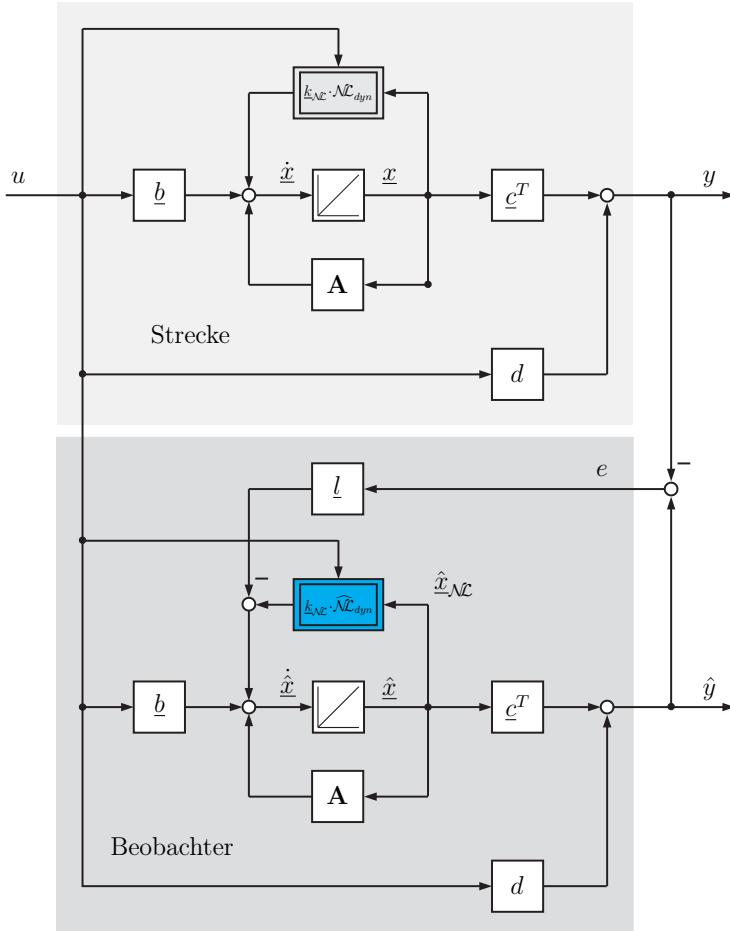
Im Gegensatz zu Kapitel 9.2.1 ist im Folgenden die vollständige Messbarkeit des Eingangsraumes der dynamischen Nichtlinearität  $\mathcal{N}_{dyn}(\underline{x}_{NL}, u)$  nicht mehr gefordert. In diesem Fall müssen die beobachteten Signale  $\hat{x}_{NL}$  als Eingangssignale für die Identifikation der dynamischen Nichtlinearität verwendet werden. Hierzu ist, wie am Anfang von Kapitel 9.2 bereits ausgeführt wurde, die vollständige Zustandsbeobachtbarkeit eine zwingende Voraussetzung. Wird ein Zustandsbeobachter analog zu Kapitel 9.2.1 entworfen, gilt:

$$\begin{aligned}\dot{\underline{x}} &= \mathbf{A} \cdot \hat{x} + \underline{b} \cdot u + \underline{k}_{NL} \cdot \widehat{\mathcal{N}}_{dyn}(\hat{x}_{NL}, u) - \underline{l} \cdot e \\ \hat{y} &= \underline{c}^T \cdot \hat{x} + d \cdot u\end{aligned}\quad (9.15)$$

Der Identifikationsalgorithmus  $\widehat{\mathcal{N}}_{dyn}(\hat{x}_{NL}, u)$  hat in Gl. (9.15) den Systemeingang  $u$  sowie die beobachteten Systemzustände  $\hat{x}_{NL}$  als Eingangssignale. In Abb. 9.9 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:

$$\widehat{\mathcal{N}}_{dyn}(\hat{x}_{NL}, u) \neq \hat{\Theta}^T \cdot \mathcal{A}_{dyn}(x_{NL}, u) \quad (9.16)$$

Der Grund dafür ist, dass der Eingangsraum der dynamischen Nichtlinearität nicht mehr als Messgröße, sondern nur noch als beobachtete Größe vorliegt. Somit



**Abb. 9.9:** Zustandsbeobachter bei nicht messbarem Eingangsraum der dynamischen Nichtlinearität

kann der dynamische Aktivierungsvektor  $\mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u)$  nicht bestimmt werden. Während der Identifikation gilt folglich:

$$\mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \neq \mathcal{A}_{dyn}(\hat{\underline{x}}_{\mathcal{N}\mathcal{L}}, u) \quad (9.17)$$

Um die Stabilität der bekannten Fehlermodelle 3 und 4 dennoch zu beweisen, werden formal sog. virtuelle Parameter eingeführt (siehe Kapitel 5.3.2). Für die virtuellen Parameter kann folgender Ansatz gemacht werden:

$$\mathcal{N}\mathcal{L}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) = \underline{\Theta}^T \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) = \check{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\hat{\underline{x}}_{\mathcal{N}\mathcal{L}}, u) \quad (9.18)$$

Die virtuellen Parameter  $\check{\Theta}$  beschreiben die Bewegung der dynamischen Nichtlinearität im Beobachterzustandsraum  $\hat{x}_{\mathcal{N}\mathcal{L}}$ . Sie sind zeitvariant und ändern sich, wenn sich  $\underline{x}_{\mathcal{N}\mathcal{L}}$  und  $\hat{x}_{\mathcal{N}\mathcal{L}}$  und somit auch die Aktivierungen  $\mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u)$  und  $\mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u)$  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

$$\hat{x} \rightarrow \underline{x} \Rightarrow \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \rightarrow \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \quad (9.19)$$

$$\hat{x} \rightarrow \underline{x} \Rightarrow \check{\Theta} \rightarrow \underline{\Theta} \quad (9.20)$$

gilt. Es muss nun gezeigt werden, wann dies der Fall ist<sup>12)</sup>. Mit Gl. (9.18) folgt für den Beobachterfehler:

$$\begin{aligned} e(s) &= H(s) \cdot \mathcal{L} \left\{ \widehat{\mathcal{N}}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) - \mathcal{N}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \\ &= H(s) \cdot \mathcal{L} \left\{ \hat{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) - \underline{\Theta}^T \cdot \mathcal{A}_{dyn}(\underline{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \end{aligned} \quad (9.21)$$

$$= H(s) \cdot \mathcal{L} \left\{ \hat{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) - \check{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \quad (9.22)$$

$$= H(s) \cdot \mathcal{L} \left\{ (\hat{\underline{\Theta}}^T - \check{\underline{\Theta}}^T) \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \right\}$$

Der Parameterfehlervektor wird definiert zu:

$$\underline{\Phi} = \hat{\underline{\Theta}} - \check{\underline{\Theta}} \quad (9.23)$$

Im Falle eines nicht messbaren Eingangsraumes gilt folglich:

$$e(s) = H(s) \cdot \mathcal{L} \left\{ \underline{\Phi}^T \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \quad (9.24)$$

Für diese Fehlergleichung kann wieder die aus [158] bekannte global stabile Adaptionsgleichung

$$\dot{\underline{\Phi}} = -\eta \cdot \epsilon \cdot \mathcal{L}^{-1} \{ H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \} \} \quad (9.25)$$

mit dem zu verwendenden Fehler<sup>13)</sup>

$$\epsilon(s) = e(s) + \hat{\underline{\Theta}}^T \cdot H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \} - H(s) \cdot \mathcal{L} \left\{ \hat{\underline{\Theta}}^T \cdot \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \right\} \quad (9.26)$$

angewandt werden. Infolge der Zeitvarianz der virtuellen Parameter kann an dieser Stelle aber nicht gefolgert werden, dass  $\dot{\underline{\Phi}} = \dot{\hat{\underline{\Theta}}}$  ist. Im Folgenden wird daher das Adoptionsgesetz

$$\dot{\check{\underline{\Theta}}} = -\eta_{virt} \cdot \epsilon \cdot \mathcal{L}^{-1} \{ H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}\mathcal{L}}, u) \} \} \quad (9.27)$$

verwendet, was für die Änderung des Parameterfehlervektors bedeutet:

<sup>12)</sup> Zur Beweisführung soll das Fehlermodell 4 betrachtet werden.

<sup>13)</sup> Es gilt  $\epsilon(s) = \mathcal{L}\{\epsilon\}$ .

$$\dot{\underline{\Phi}} = \dot{\underline{\Theta}} - \dot{\underline{\Theta}} = - \left[ \eta_{virt} \cdot \epsilon \cdot \mathcal{L}^{-1} \{ H(s) \cdot \mathcal{L} \{ \mathcal{A}_{dyn}(\hat{x}_{\mathcal{N}}, u) \} \} + \dot{\underline{\Theta}} \right] \quad (9.28)$$

Aus den Gleichungen (9.25) und (9.28) 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  $\dot{\underline{\Theta}}$  gegenüber  $\dot{\underline{\Theta}}$  überwiegt. Da die Lernschrittweite  $\eta_{virt}$  positiv sein muss, ergibt sich eine Bedingung der Art:

$$0 < \eta_{min} < \eta_{virt} < \eta_{max} \quad (9.29)$$

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  $\eta_{min}$  und  $\eta_{max}$  ist jedoch bisher nicht bekannt.

Zusammenfassend kann somit festgestellt werden, dass das in Kapitel 9.2.1 vorgestellte Verfahren auch bei nicht messbaren Zuständen eingesetzt werden kann.

### 9.3 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 7.2.2 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 global integrierenden Systemen nicht abklingt. Diese Problematik bleibt selbstverständlich bei den nichtlinearen Modellen ohne Ausgangsrückkopplung erhalten.

In [131] 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 konkretes Beispiel wird ein global integrierendes Hammerstein–Modell untersucht.

In Abb. 9.10 ist das global integrierende Hammerstein–Modell, das im Folgenden betrachtet wird, dargestellt. Entsprechend Abb. 9.6 kann das Gesamtsystem

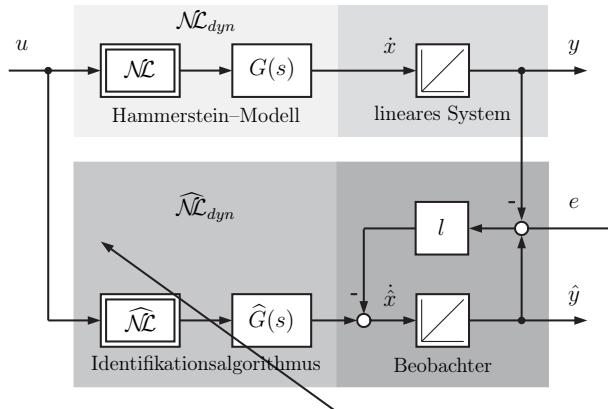


Abb. 9.10: Global integrierendes Hammerstein–Modell

in eine unbekannte 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 besteht.<sup>14)</sup> Das global integrierende Hammerstein–Modell kann entsprechend Gl. (9.1) wie folgt im Zustandsraum formuliert werden:

$$\dot{x} = \mathcal{NL}_{dyn}(u) \quad y = x \quad (9.30)$$

Aus Abb. 9.10 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.

<sup>14)</sup> Die Zeitkonstante des Integrators ist Eins und somit bekannt.

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 gilt<sup>15)</sup>:

$$\dot{\hat{x}} = \widehat{\mathcal{NL}}_{dyn}(u) - l \cdot e \quad \hat{y} = \hat{x} \quad (9.31)$$

Für die Identifikation der dynamischen Nichtlinearität soll im Folgenden ein Funktionsapproximatoransatz für das Hammerstein–Modell verwendet werden. Das bedeutet, dass anstelle des Polynoms ein GRNN zur Beschreibung der statischen Nichtlinearität verwendet wird. Dies ist vorteilhaft, wenn die statische Nichtlinearität nur bedingt durch ein Polynom beschrieben werden kann, wie dies z.B. bei Sättigungscharakteristiken der Fall ist.

Durch das Einsetzen der Gleichung des GRNN in die Faltungssumme ergibt sich die Systembeschreibung eines Hammerstein–Modells bei Approximation der Nichtlinearität durch ein GRNN<sup>16)</sup>:

$$y[k] = \Theta_{\mathcal{NL},1} \sum_{i=1}^m h[i] \mathcal{A}_1(u[k-i]) + \dots + \Theta_{\mathcal{NL},q} \sum_{i=1}^m h[i] \mathcal{A}_q(u[k-i]) \quad (9.32)$$

Ein Vergleich von Gleichung (9.32) mit der Volterra–Funktionalpotenzreihe nach Gleichung (8.16) 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. (9.32) 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:

$$g_1[i] = \Theta_{\mathcal{NL},1} \cdot h[i] \quad g_2[i] = \Theta_{\mathcal{NL},2} \cdot h[i] \quad \dots \quad g_q[i] = \Theta_{\mathcal{NL},q} \cdot h[i] \quad (9.33)$$

Bei den Gewichtsfolgen in Gl. (9.33) 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. (9.32) nicht nötig, da ein Systemoffset vom GRNN automatisch gelernt wird. So ergibt sich die Volterra–Reihe<sup>17)</sup> für Gl. (9.32) zu:

<sup>15)</sup> Die für die Identifikation erforderliche Sichtbarkeit der Nichtlinearität kann leicht überprüft werden.

<sup>16)</sup> Die Gewichte des GRNN sollen im Folgenden mit  $\Theta_{\mathcal{NL}}$  bezeichnet werden.

<sup>17)</sup> In diesem Identifikationsansatz wird der Begriff Volterra–Reihe verwendet, um eine Abgrenzung zu Volterra–Funktionalpotenzreihen herzustellen.

$$y[k] = \sum_{i=1}^m g_1[i] \mathcal{A}_1(u[k-i]) + \dots + \sum_{i=1}^m g_q[i] \mathcal{A}_q(u[k-i]) \quad (9.34)$$

Die Anzahl der unbekannten Parameter in Gl. (9.34) beträgt  $p = q \cdot m$ . Die Anzahl der Stützwerte entspricht formal dem Grad der Nichtlinearität. Der Parameter  $g_0$  für den Systemoffset kann eingespart werden. Auch bei diesem 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. (8.31) zu:

$$y[k] = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (9.35)$$

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

Für die Approximation der Gewichtsfolgen durch orthonormale Basisfunktionen muss für jeden Stützwert 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:

$$\underline{\Theta}^T = [\underline{\Theta}_1^T, \underline{\Theta}_2^T, \dots, \underline{\Theta}_q^T] \quad (9.36)$$

Die Dimensionen der verwendeten Vektoren und Matrizen in den Gleichungen (9.35) und (9.36) sind abhängig von der Antwortlänge  $m$ , der Anzahl verwendeter Basisfunktionen  $m_r$  und der Stützstellenanzahl  $q$  des GRNN. Es gilt:

$$\begin{array}{ll} \tilde{\mathbf{R}} & \in \mathbb{R}^{m_r \times m} \\ \underline{\mathcal{A}}_1^T, \underline{\mathcal{A}}_2^T, \dots, \underline{\mathcal{A}}_q^T & \in \mathbb{R}^{1 \times m} \\ \underline{\Theta}_1, \underline{\Theta}_2, \dots, \underline{\Theta}_q & \in \mathbb{R}^{m_r \times 1} \\ \underline{\Theta} & \in \mathbb{R}^{(q \cdot m_r) \times 1} \end{array} \quad (9.37)$$

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

Für den Identifikationsalgorithmus des global integrierenden Hammerstein-Modells gilt somit:

$$\widehat{\mathcal{NL}}_{dyn}(u) = \widehat{\underline{\Theta}}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad \text{mit} \quad \underline{\mathcal{A}}_{dyn}^T[k] = \left[ \underline{\mathcal{A}}_1^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \underline{\mathcal{A}}_q^T[k] \cdot \tilde{\mathbf{R}}^T \right] \quad (9.38)$$

Der Identifikationsalgorithmus ist linear in den unbekannten Parametern  $\hat{\Theta}$ , so dass abhängig von der Fehlerübertragungsfunktion  $H(s)$ , eines der in Kapitel 9.2 vorgestellten Fehlermodelle angewendet werden kann. Für das global integrierende Hammerstein–Modell ergibt sich die Fehlerübertragungsfunktion  $H(s)$  nach Gl. (9.6) wie folgt:

$$H(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A} + l \cdot \underline{c}^T)^{-1} \cdot \underline{k}_{\mathcal{NL}} = \frac{1}{s + l} \quad (9.39)$$

Aus Gl. (9.39) 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 kann<sup>18)</sup>.

Anhand eines Simulationsbeispiels soll die Funktionsweise der Identifikation in Beobachterstruktur verdeutlicht werden. Gegeben ist das folgende Hammerstein–Modell:

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

Die Zeitkonstanten von  $G(s)$  sind auf einen Simulationszeitschritt von 1s normiert. Die statische Nichtlinearität wurde als Arcustangensfunktion gewählt, da diese Funktion in der Praxis häufig vorkommt<sup>19)</sup>. 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 9.1 verwendet.

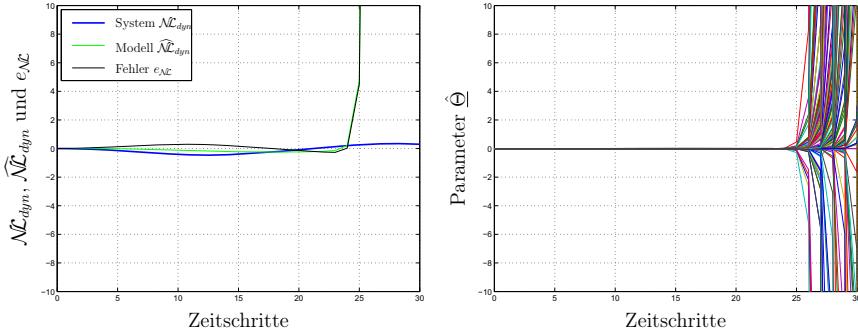
Einstellwerte	Erläuterung
$q = 21$	Stützstellenanzahl des GRNN
$\sigma_{norm} = 0.4$	normierte Standardabweichung des GRNN
$m = 100$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 21.3$	Formfaktor
$l = 0$ bzw. $l = 1$	Beobachterrückführung
$p = m_r \cdot q = 210$	Parameteranzahl
$h = 0.5s$	Abtastzeit

**Tabelle 9.1:** Einstellwerte für die Identifikation eines global integrierenden Hammerstein–Modells

In Abb. 9.11 sind die Ergebnisse bei einer Identifikation ohne Beobachterrückführkoeffizienten, d.h. bei  $l = 0$ , dargestellt.

<sup>18)</sup> Es empfiehlt sich jedoch aufgrund kürzerer Konvergenzzeiten auch in diesem Fall Fehlermodell 4 anzuwenden.

<sup>19)</sup> Sättigungscharakteristiken und Reibkennlinien werden häufig als Arcustangensfunktionen modelliert.



**Abb. 9.11:** Identifikation ohne Beobachter — Identifikationsverlauf (links) und Verlauf der Parameter (rechts)

In Abb. 9.11 links ist der Verlauf des Ausgangs der dynamischen Nichtlinearität  $\mathcal{N}\mathcal{L}_{dyn}(u)$  sowie des Ausgangs des Identifikationsalgorithmus  $\widehat{\mathcal{N}}\mathcal{L}_{dyn}(u)$  dargestellt. Zusätzlich ist der Fehler  $e_{\mathcal{N}\mathcal{L}}$  zwischen der vorgegebenen und identifizierten dynamischen Nichtlinearität veranschaulicht. Dieser Fehler

$$e_{\mathcal{N}\mathcal{L}} = \widehat{\mathcal{N}}\mathcal{L}_{dyn} - \mathcal{N}\mathcal{L}_{dyn} \quad (9.40)$$

dient ausschließlich zur Visualisierung des Lernergebnisses. Die Adaption der Parameter erfolgt mit dem Fehler  $e$ . Wie in Abb. 9.11 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.

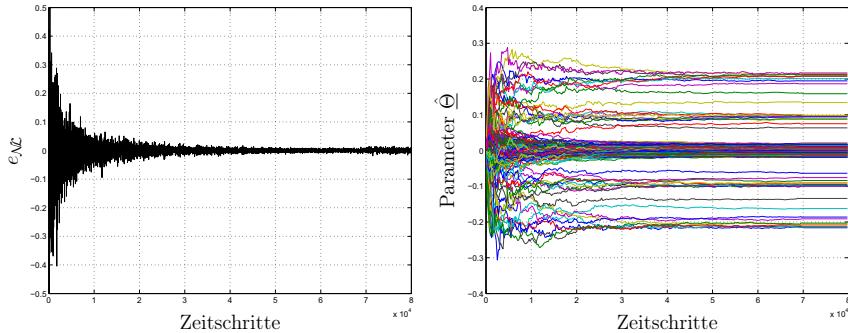
Im Gegensatz dazu zeigt Abb. 9.12 den Fehler- und Parameterverlauf einer stabilen Identifikation in Beobachterstruktur.

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.

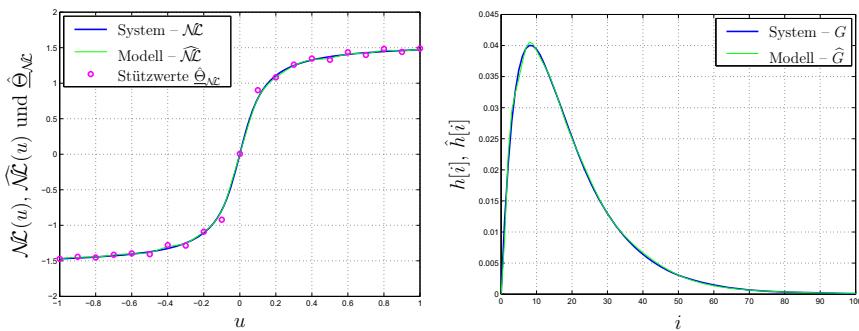
In Abb. 9.12 ist zu erkennen, dass der Fehler  $e_{\mathcal{N}\mathcal{L}}$  sehr klein wird und die Parameter  $\underline{\Theta}$  relativ gut konvergieren. Ein kleiner Restfehler, der auf die begrenzte Anzahl an Stützwerten und Basisfunktionen zurück zu führen ist, bleibt jedoch, so dass die Parameter nicht auf einen exakten Wert konvergieren können. Dies führt auch dazu, dass der Fehler im Parallelauf leicht ansteigt. Dadurch wird jedoch nicht der Erfolg der Identifikation geschmälert, da ohne die vorgestellte Beobachterstruktur überhaupt keine Identifikation möglich gewesen wäre.

In Abb. 9.13 ist die identifizierte statische Nichtlinearität sowie die identifizierte Impulsantwort dargestellt.

Auch die Lernergebnisse für die statische Nichtlinearität und die Impulsant-



**Abb. 9.12:** Identifikation mit Beobachter — Fehlerverlauf (links) und Konvergenz der Parameter (rechts)



**Abb. 9.13:** Identifizierte statische Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

wort verdeutlichen, dass sich die vorgestellte Beobachterstruktur hervorragend zur Identifikation von global integrierenden Systemen eignet.

## 9.4 Beobachterentwurf und Identifikation — Simulationsbeispiel

Nachdem in Kapitel 9.3 global integrierende Systeme als einfachste Form einer Identifikation von dynamischen Nichtlinearitäten mit Hilfe der vorgestellten Beobachtertheorie untersucht wurden, wird in diesem Kapitel ein praxisnahes Simulationsbeispiel betrachtet. Das betrachtete System mit dynamischer Nichtlinearität und der zugehörige Beobachter einschließlich Identifikationsalgorithmus

ist in Abb. 9.14 veranschaulicht.

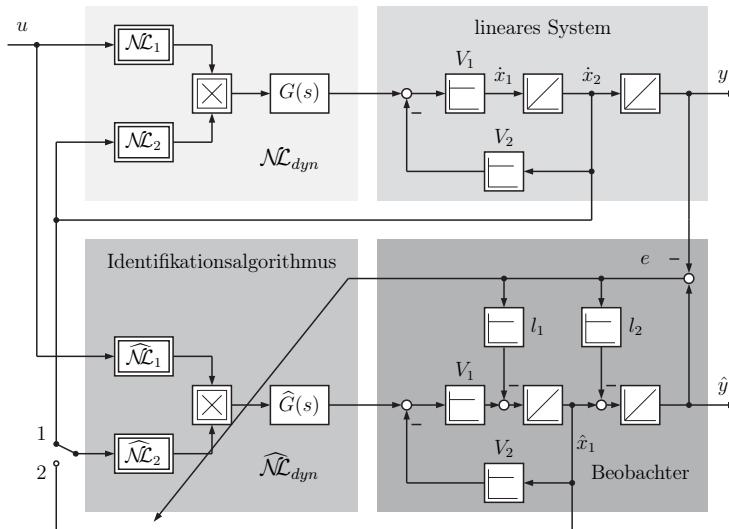


Abb. 9.14: Identifikationsbeispiel

Das System kann entsprechend Gl. (9.1) im Zustandsraum wie folgt dargestellt werden:

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\underline{\dot{x}}} = \underbrace{\begin{bmatrix} -V_1 \cdot V_2 & 0 \\ 1 & 0 \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\underline{x}} + \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\underline{b}} \cdot u + \underbrace{\begin{bmatrix} V_1 \\ 0 \end{bmatrix}}_{\underline{k}_{\mathcal{N}\mathcal{L}}} \cdot \mathcal{L}_{dyn}(x_1, u) \quad (9.41)$$

$$y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\underline{c}^T} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

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}\_{\mathcal{N}\mathcal{L}}\$ der dynamischen Nichtlinearität. Der Einkopplungsvektor \$\underline{b}\$ des Systemeingangs 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 9.5 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 ist.<sup>20)</sup> 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 wirkt.<sup>21)</sup>

Als dynamische Nichtlinearität wird ein Hammerstein–Modell mit einer linearen Übertragungsfunktion und zwei multiplikativ verknüpften Nichtlinearitäten betrachtet. Die Eingangsgrößen sind der Systemeingang  $u$  und der Zustand  $x_{\mathcal{N}} = x_1$ . 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 bestimmt werden. 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 Teilsystems<sup>22)</sup> voraus. Für die Übertragungsfunktion  $H_S(s)$  zur Überprüfung der Sichtbarkeit gilt entsprechend Gl. (9.3):

$$H_S(s) = \underline{c}^T \cdot (s\mathbf{E} - \mathbf{A})^{-1} \cdot \underline{k}_{\mathcal{N}} = \frac{V_1}{s^2 + s \cdot V_1 \cdot V_2} \neq 0 \quad (9.42)$$

Somit ist die dynamische Nichtlinearität am Systemausgang sichtbar und kann mit dem Beobachterfehler  $e$  gelernt werden.

Für die Beobachtbarkeit des linearen Teilsystems gilt:

$$\det \mathbf{Q}_{obs} = \det [\underline{c}, \mathbf{A}^T \underline{c}] = \det \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = -1 \neq 0 \quad (9.43)$$

---

<sup>20)</sup> Der Parameter  $V_1$  stellt das reziproke Trägheitsmoment dar.

<sup>21)</sup> Die 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.

<sup>22)</sup> Zustandsbeobachtbarkeit muss nur vorausgesetzt werden, wenn der Eingangsraum der dynamischen Nichtlinearität nicht messbar ist.

Damit sind alle Voraussetzungen für den Beobachterentwurf erfüllt. Der Beobachter kann im Falle eines nicht messbaren Eingangsraumes der dynamischen Nichtlinearität entsprechend Gl. (9.15) wie folgt beschrieben werden:

$$\begin{aligned} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} &= \begin{bmatrix} -V_1 \cdot V_2 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} V_1 \\ 0 \end{bmatrix} \cdot \widehat{\mathcal{N}}\mathcal{L}_{dyn}(\hat{x}_1, u) - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \cdot e \\ \hat{y} &= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \end{aligned} \quad (9.44)$$

Die Gleichungen (9.44) gelten für den Fall, dass der Eingangsraum der dynamischen Nichtlinearität nicht messbar ist (vgl. Abb. 9.14. Fall 2). Ist im Gegensatz dazu der Eingangsraum messbar (vgl. Abb. 9.14, Fall 1), so ist die dynamische Nichtlinearität von  $x_1$  und  $u$  abhängig, d.h.  $\widehat{\mathcal{N}}\mathcal{L}_{dyn}(x_1, u)$ . 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 9.3 mit einem zweidimensionalen GRNN verwendet. Somit gilt für den Identifikationsalgorithmus:

$$\widehat{\mathcal{N}}\mathcal{L}_{dyn}(\hat{x}_1, u) = \underline{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (9.45)$$

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

Zur Adaption der unbekannten Parameter  $\hat{\underline{\Theta}}$  muss die Fehlerübertragungsfunktion  $H(s)$  nach Gl. (9.6) bestimmt werden. Für  $H(s)$  ergibt sich:

$$H(s) = \frac{e(s)}{e_{\mathcal{N}\mathcal{L}}(s)} = \frac{V_1}{s^2 + s \cdot (V_1 \cdot V_2 + l_2) + (l_1 + V_1 \cdot V_2 \cdot l_2)} \quad (9.46)$$

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:

$$\begin{aligned} \mathcal{N}\mathcal{L}(x_1, u) &= \mathcal{N}\mathcal{L}_1(u) \cdot \mathcal{N}\mathcal{L}_2(x_1) = [0.5 \cdot \arctan(5 \cdot u)] \cdot [\arctan(5 \cdot x_1)^2 + 1] \\ G(s) &= \frac{1}{21 \cdot s^2 + 10 \cdot s + 1} \end{aligned}$$

Für die Parameter des bekannten linearen Teilsystems gilt:

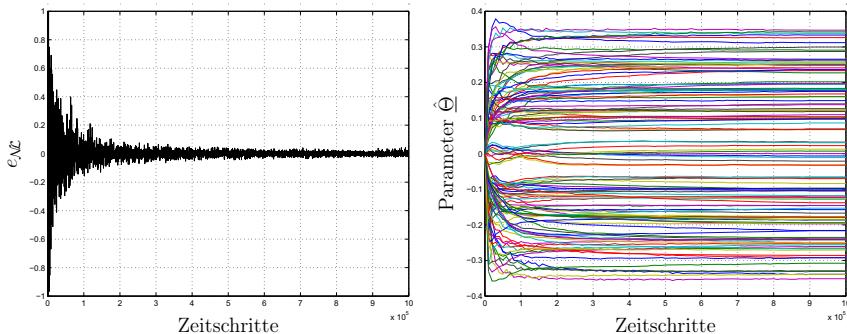
Einstellwerte	Erläuterung
$q = 11^2 = 121$	Stützstellenanzahl
$\sigma_{norm1} = \sigma_{norm2} = \sigma_{norm} = 0.6$	normierte Standardabweichung des GRNN
$m = 50$	Antwortlänge
$m_r = 10$	Basisfunktionenanzahl
$\zeta = 10.65$	Formfaktor
$\underline{l}^T = [0.52, 1.02]$	Beobachterrückführungen
$p = m_r \cdot q = 1210$	Parameteranzahl
$\eta = 5$	Lernschrittweite
$h = 1s$	Abtastzeit

**Tabelle 9.2:** Einstellwerte für die Identifikation

$$V_1 = 0.02$$

$$V_2 = 3.85$$

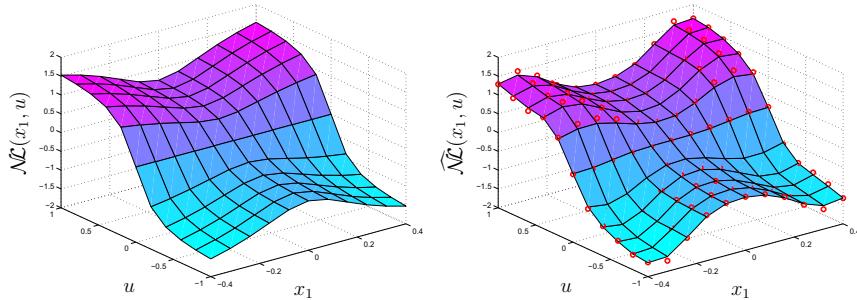
Für die Identifikation werden die Einstellwerte nach Tabelle 9.2 verwendet. Die Simulationsdauer beträgt insgesamt  $10^6$  Zeitschritte. Nach  $9 \cdot 10^5$  Zeitschritten wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird durch ein Zufallssignal im Bereich  $u \in [-1; 1]$  angeregt. Zunächst erfolgt die Identifikation unter der Annahme, dass  $x_1$  messbar ist. In Abb. 9.15 ist der Fehlerverlauf (links) und die Konvergenz ausgewählter Parameter (rechts) dargestellt.



**Abb. 9.15:** Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz ausgewählter Parameter (rechts)

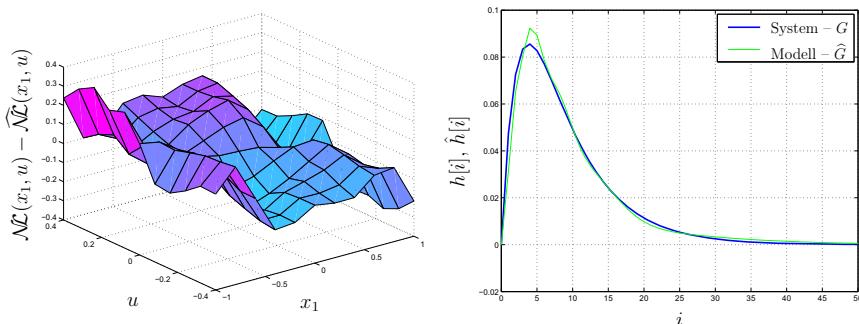
Es ist zu erkennen, dass der Fehler  $e_{NL}$  klein wird und die Parameter konvergieren. Auch im Parallelbetrieb steigt der Fehler nicht merklich an. Allerdings bleibt aufgrund der relativ geringen Anzahl an Basisfunktionen und der begrenzten Stützwertanzahl ein Restfehler. In Abb. 9.16 sind die vorgegebene und die

identifizierte statische Nichtlinearität einander gegenüber gestellt. Die statische



**Abb. 9.16:** Vergleich zwischen vorgegebener (links) und identifizierter (rechts) statischer Nichtlinearität

Nichtlinearität wird durch die beiden GRNN gut identifiziert. Da 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. 9.17 ist der Approximationsfehler der statischen Nichtlinearität und eine identifizierte Impulsantwort veranschaulicht.

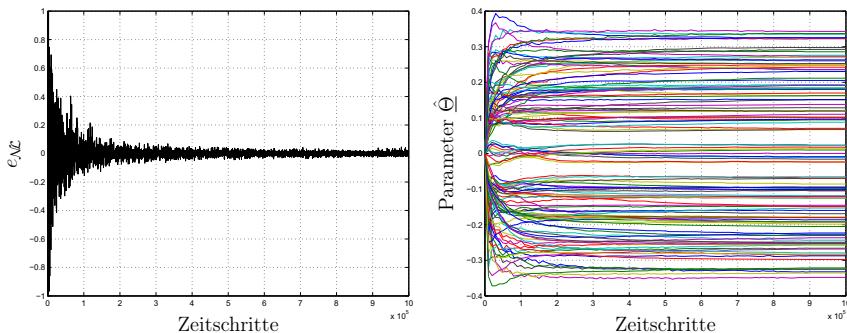


**Abb. 9.17:** Identifikationsfehler der statischen Nichtlinearität (links) und identifizierte Impulsantwort (rechts)

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.

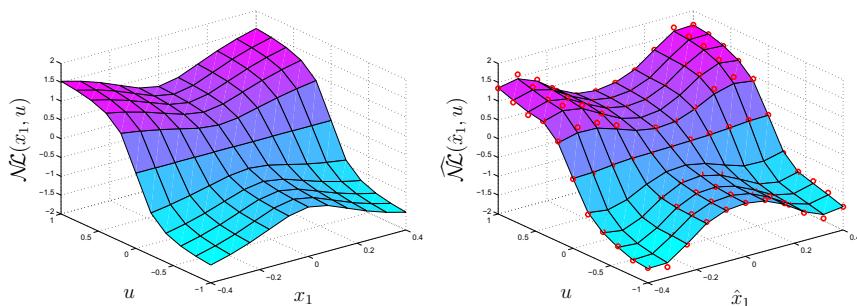
Dennoch ist das Identifikationsergebnis im Hinblick auf den sehr kleinen Fehler im Parallelbetrieb des Modells (vgl. Abb. 9.15 links) als sehr gut zu bewerten.

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. 9.18 ist der Fehlerverlauf (links) und die Konvergenz ausgewählter Parameter (rechts) dargestellt. Es ist zu erkennen, dass der Fehler  $e_{\mathcal{N}}$  ebenfalls



**Abb. 9.18:** Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz ausgewählter Parameter (rechts)

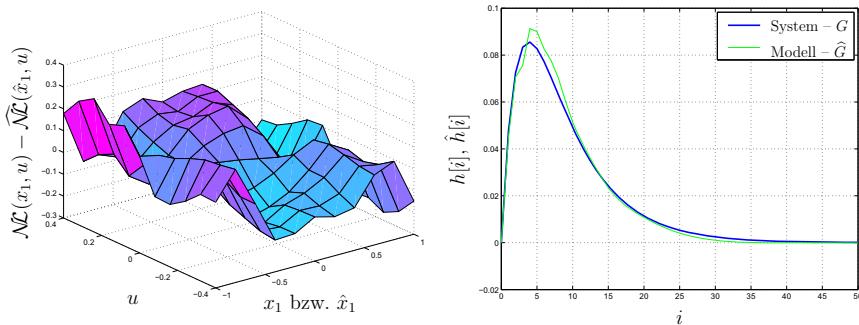
klein wird und die Parameter konvergieren. In Abb. 9.19 sind die vorgegebene und die identifizierte statische Nichtlinearität einander gegenüber gestellt.



**Abb. 9.19:** 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. 9.20 links ist der Approximationsfehler der statischen Nichtlinearität und rechts die identifizierte Impulsantwort veranschaulicht.



**Abb. 9.20:** 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.

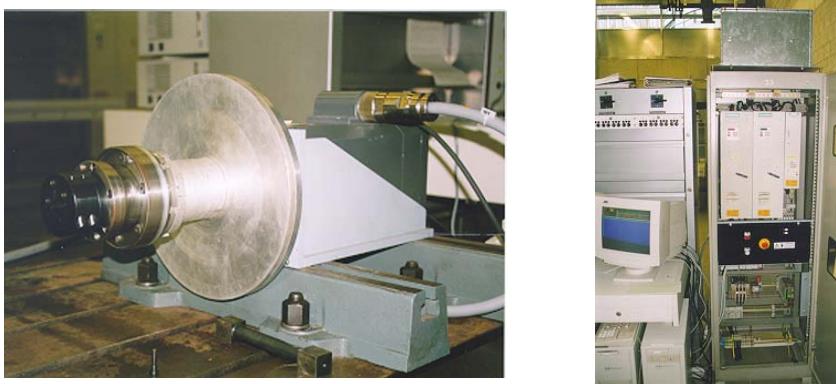
## 9.5 Identifikation eines mechatronischen Antriebssystems

Als Anwendungsbeispiel soll ein nichtlineares mechatronisches Antriebssystem identifiziert werden. Das betrachtete Antriebssystem ist in Abb. 9.21 dargestellt.

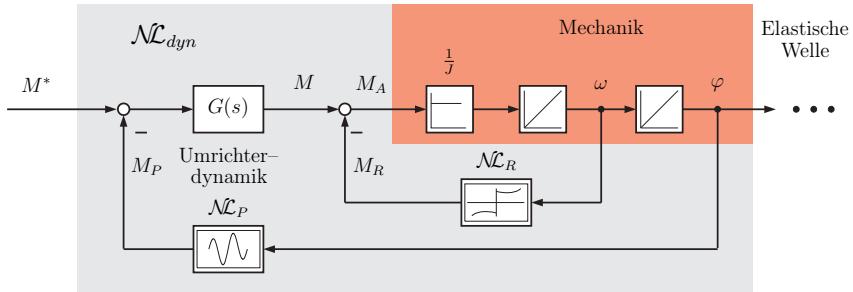
Das gesamte System besteht aus einer permanenterregten Synchronmaschine mit einem integrierten Encoder zur Rotorlage erfassung und einem zugehörigen Umrüchter.

Aus Voruntersuchungen [5] ergab sich das in Abb. 9.22 dargestellte Blockschaltbild des Systems. 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 [206].<sup>23)</sup> Wird im Gegensatz dazu für die Regelung die antriebsseitige Drehzahl bzw. Position

<sup>23)</sup> Diese Problematik ist in der Literatur auch unter dem Stichwort *Zweimassensystem* bzw. *Mehrmassensystem* zu finden.



**Abb. 9.21:** Komponenten des Antriebssystems



**Abb. 9.22:** Blockschaltbild des betrachteten Antriebssystems

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 Zweimassen-systems ist eine Identifikation mit einem strukturierten rekurrenten Netz [88].<sup>24)</sup> Hierzu muss jedoch das Eingangsmoment  $M$ , das in der Regel nicht messbar ist, bekannt sein. Dieses Moment  $M$  ist aufgrund der nichtlinearen Umrichterodynamik nicht identisch mit dem vorgegebenen Sollmoment  $M^*$ . Zur Berechnung

<sup>24)</sup> Die Federsteifigkeit einer elastischen Welle kann näherungsweise durch eine Messung bestimmt werden. Die Dämpfung dagegen kann nur äußerst schwierig durch Messungen quantifiziert werden.

des Momentes  $M$  muss ein Modell bestimmt werden, das die nichtlineare Umrichterdynamik hinreichend genau beschreibt. Dies stellt die Motivation für das Anwendungsbeispiel dar.

In Abb. 9.22 ist zu erkennen, dass das bekannte lineare Teilsystem aus einer Integratorkette 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 dynamischen Nichtlinearität beruht auf der Annahme, dass der Momentenaufbau mit Hilfe einer Übertragungsfunktion modelliert werden kann.<sup>25)</sup> 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 Fluxes zu einem Fehlverhalten des feldorientierten Drehmomentregelkreises. Dieses Fehlverhalten drückt sich in einer maschinenlageabhängigen Drehmomentschwankung aus.<sup>26)</sup> Als weitere statische Nichtlinearität ist die winkelgeschwindigkeitsabhängige Lagerreibung  $\mathcal{NL}_R(\omega)$  der Maschine zu berücksichtigen. Die Lagerreibung wirkt der Bewegung entgegen und ist bei  $\omega = 0 \frac{\text{rad}}{\text{s}}$  aufgrund der Haftreibung unstetig. Wird das System entsprechend Gl. (9.1) dargestellt, ergibt sich:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \mathcal{NL}_{dyn}(\underline{x}, u) \\ y = [0 \ 1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{mit} \quad \underline{x}^T = [\omega, \varphi] \quad \text{und} \quad u = M^* \quad (9.47)$$

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.

### 9.5.1 Identifikation in der Simulationsumgebung

Für den Beobachterentwurf wird bei diesem Anwendungsbeispiel von einem nicht messbaren Eingangsraum der dynamischen Nichtlinearität ausgegangen. Diese

<sup>25)</sup> In der Übertragungsfunktion sind die dynamischen Effekte der feldorientierten Drehmomentregelung und des Momentenaufbaus in der Maschine selbst zusammengefasst.

<sup>26)</sup> Diese Abhängigkeit konnte bei einfachen Vorversuchen an der Anlage festgestellt werden.

Annahme wird so getroffen, da die Winkelgeschwindigkeit  $\omega$ , die sich durch Differentiation der gemessenen Rotorlage  $\varphi$  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 Anwendungsbeispiel leicht gezeigt werden. Somit gilt für den Beobachter entsprechend Gl. (9.15):

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{J} \\ 0 \end{bmatrix} \cdot \widehat{\mathcal{N}}\mathcal{L}_{dyn}(\hat{x}, u) - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \cdot e \quad (9.48)$$

$$\hat{y} = [0 \ 1] \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \quad \text{mit} \quad \hat{x}^T = [\hat{\omega}, \hat{\varphi}] \quad \text{und} \quad u = M^*$$

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  $\varphi$  als Eingangsgröße. In dieses Hammerstein–Modell greift eine zweite Eingangsgröße  $M^*$  linear ein. Zusätzlich greift am Ausgang des Hammerstein–Modells eine dritte Eingangsgröße ein, die bei einer isolierten Betrachtung der dynamischen Nichtlinearität einen Durchgriff darstellt. Bisher 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:

$$\widehat{\mathcal{N}}\mathcal{L}_{dyn}(\hat{x}, u) = \hat{\Theta}^T \cdot \underline{\mathcal{A}}_{dyn}[k] \quad (9.49)$$

$$\underline{\mathcal{A}}_{dyn}^T[k] = \left[ \underbrace{\mathcal{A}_1[k], \dots, \mathcal{A}_q[k]}_{\text{Eingang } \hat{x}_1}, \underbrace{\mathcal{A}_{A,1}^T[k] \cdot \tilde{\mathbf{R}}^T, \dots, \mathcal{A}_{B,K}^T[k] \cdot \tilde{\mathbf{R}}^T}_{\text{Eingang } \hat{x}_2}, \underbrace{\underline{u}^T[k] \cdot \tilde{\mathbf{R}}^T}_{\text{Eingang } u} \right]$$

Der dynamische Aktivierungsvektor  $\underline{\mathcal{A}}_{dyn}$  besteht aus drei Teilen. Der erste Teil beschreibt die Approximation der statischen Nichtlinearität  $\widehat{\mathcal{N}}\mathcal{L}_R(\hat{x}_1)$ , 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:

$$\mathcal{A}_i[k] = \frac{\exp\left[\frac{(\hat{x}_1 - \chi_i)^2}{-2\sigma^2}\right]}{\sum_{j=1}^q \exp\left[\frac{(\hat{x}_1 - \chi_j)^2}{-2\sigma^2}\right]} \quad \text{mit} \quad i = 1 \dots q \quad (9.50)$$

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 werden.<sup>27)</sup> Der zweite Teil des dynamischen Aktivierungsvektors beschreibt ein Hammerstein–Modell mit einer periodisch statischen Nichtlinearität  $\widehat{\mathcal{NL}}_P(\hat{x}_2)$ . Zur Approximation dieser periodischen Nichtlinearität wird ein harmonisch aktviertes Netz (HANN), das in Kapitel 3.8 beschrieben wurde, eingesetzt.<sup>28)</sup> Die  $m$  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:

$$\underline{\mathcal{A}}_{A,i}^T[k] = \left[ \mathcal{A}_{A,i}(\hat{x}_2[k-1]), \dots, \mathcal{A}_{A,i}(\hat{x}_2[k-m]) \right] \quad \text{mit} \quad i = 1 \dots K \quad (9.51)$$

$$\underline{\mathcal{A}}_{B,i}^T[k] = \left[ \mathcal{A}_{B,i}(\hat{x}_2[k-1]), \dots, \mathcal{A}_{B,i}(\hat{x}_2[k-m]) \right]$$

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:

$$\underline{u}^T[k] = \left[ u[k-1], \dots, u[k-m] \right] \quad (9.52)$$

Das gesamte System sowie der zugehörige Beobachter einschließlich Identifikationsalgorithmus sind noch einmal in Abb. 9.23 veranschaulicht.

Zur Adaption der unbekannten Parameter muss die Fehlerübertragungsfunktion  $H(s)$  nach Gl. (9.6) bestimmt werden. Es ergibt sich:

$$H(s) = \frac{\frac{1}{J}}{s^2 + s \cdot l_2 + l_1} \quad (9.53)$$

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–Hélouvy [8] angenommen:

<sup>27)</sup> In Gl. (9.49) wurde dies aus Gründen der Übersichtlichkeit nicht dargestellt.

<sup>28)</sup> Der Gleichanteil im HANN wird dabei vernachlässigt, da die periodische Nichtlinearität keinen Offset besitzt.

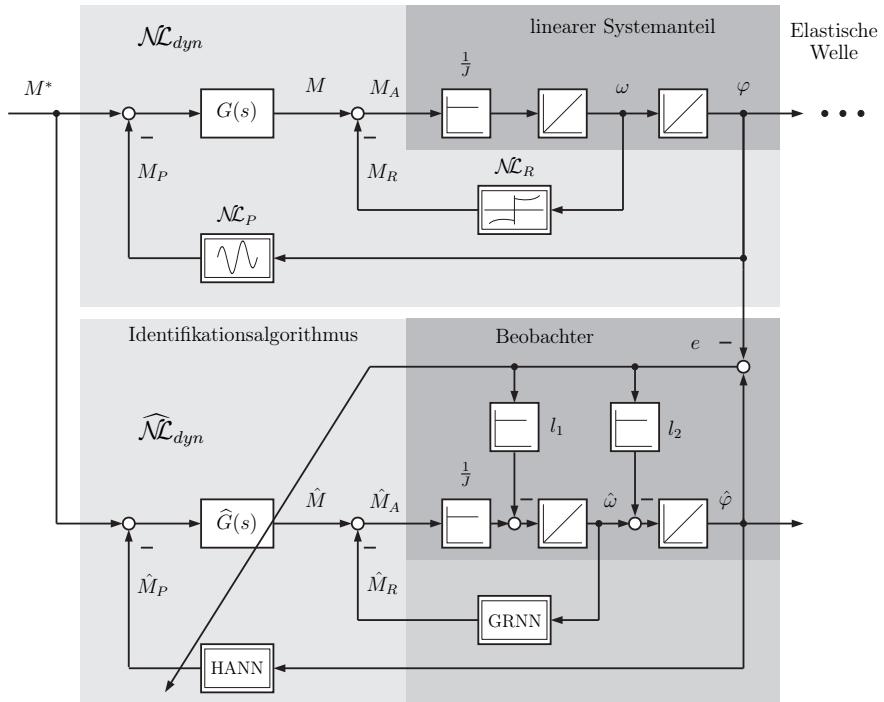


Abb. 9.23: Antriebssystem mit Beobachter und Identifikationsalgorithmus

$$\mathcal{N}\mathcal{L}_R(x_1) = M_c + (M_h + M_c) \cdot \exp \left[ - \left( \frac{x_1}{N_s} \right)^2 \right] + M_n \cdot x_1$$

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<sub>1</sub>-Verhalten angenommen:

$$G(s) = \frac{1}{s \cdot T_{lin} + 1} \quad \text{mit} \quad T_{lin} = 0.003 \text{ s}$$

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 9.3 verwendet.

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

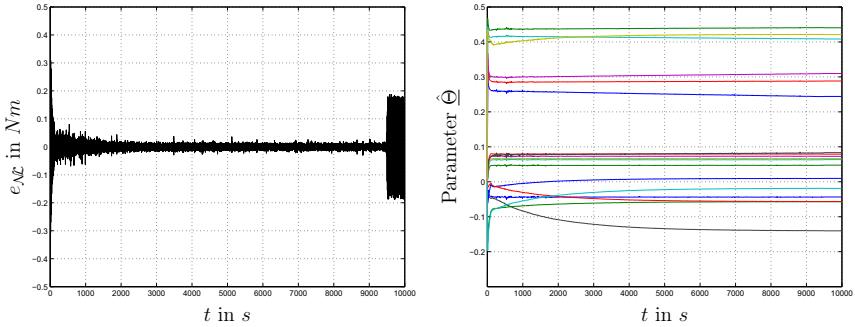
**Tabelle 9.3:** Einstellwerte für die Identifikation

Die Simulationsdauer beträgt insgesamt 10000 Sekunden. Nach 9500 Sekunden wird die Parameteradaption eingestellt und die Beobachterkoeffizienten auf Null gesetzt, so dass das identifizierte Modell als reines Parallelmodell betrieben wird. Das System wird mit einem Drehzahlregler, der als Solldrehzahl ein Zufallssignal im Bereich von  $x_1^* = \omega^* \in [-20 \frac{\text{rad}}{\text{s}}; 20 \frac{\text{rad}}{\text{s}}]$  erhält, angeregt, so dass sich ein Moment von  $u = M^* \in [-17 \text{ Nm}; 17 \text{ Nm}]$  ergibt. In Abb. 9.24 ist der Fehlerverlauf (links) und die Konvergenz der Parameter<sup>29)</sup> (rechts) dargestellt.

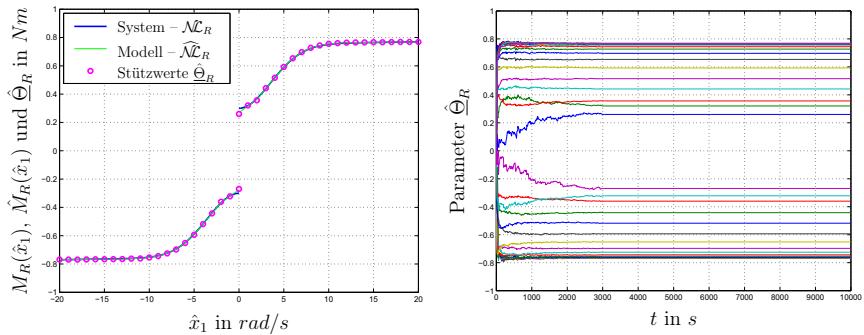
Es ist zu erkennen, dass der Fehler  $e_{\mathcal{M}}$  klein wird und die Parameter konvergieren. Allerdings steigt der Fehler im Parallelbetrieb wieder merklich an.<sup>30)</sup> Dies ist darauf zurück zu führen, dass einige Parameter sehr langsam konvergieren. Auch die geringe Basisfunktionenanzahl trägt zu diesem Fehler bei. Dennoch ist das Ergebnis als gut zu bewerten, wenn beispielsweise die statischen Nichtlinearitäten betrachtet werden. In Abb. 9.25 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 ungenügende Anregung in diesem Bereich verursacht werden. Die zugehörigen Parameter der Reibkennlinie zeigen ein sehr gutes

<sup>29)</sup> Die Parameter  $\hat{\Theta}_R$  der Reibkennlinie werden gesondert dargestellt.

<sup>30)</sup> Der Fehler ist im Parallelauf kleiner als zu Beginn der Identifikation mit aktiven Beobachter.



**Abb. 9.24:** Fehlerverlauf der dynamischen Nichtlinearität (links) und Konvergenz der Parameter (rechts)



**Abb. 9.25:** Vergleich zwischen vorgegebener und identifizierter Reibkennlinie (links) sowie Konvergenz der Parameter (rechts)

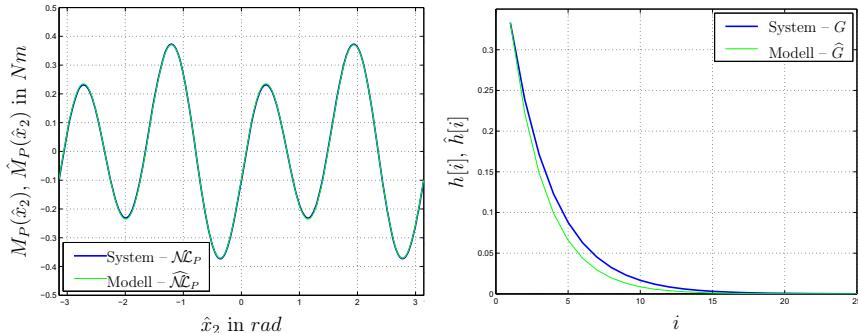
Konvergenzverhalten. Das Ergebnis der periodischen Nichtlinearität sowie der Impulsantwort ist in Abb. 9.26 veranschaulicht.

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.

### 9.5.2 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 9.4 verwendet.

Die Simulationsdauer beträgt insgesamt 1800 Sekunden. Nach 1700 Sekunden wird die Parameteradaption eingestellt und die Beobachterrückführkoeffizienten



**Abb. 9.26:** Identifikationsergebnis der periodischen Nichtlinearität (links) und der identifizierten Impulsantwort (rechts)

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

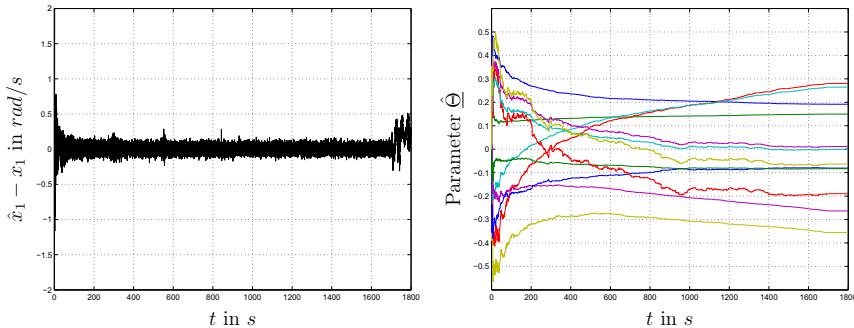
**Tabelle 9.4:** Einstellwerte für die Identifikation am realen System

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 im Bereich von  $x_1^* = \omega^* \in [-20 \frac{\text{rad}}{\text{s}}; 20 \frac{\text{rad}}{\text{s}}]$  erhält, angeregt, so dass sich ein Moment von  $u = M^* \in [-17 \text{Nm}; 17 \text{Nm}]$  ergibt.

In Abb. 9.27 ist der Fehler in der Winkelgeschwindigkeit (links) und die Konvergenz der Parameter<sup>31)</sup> (rechts) dargestellt.

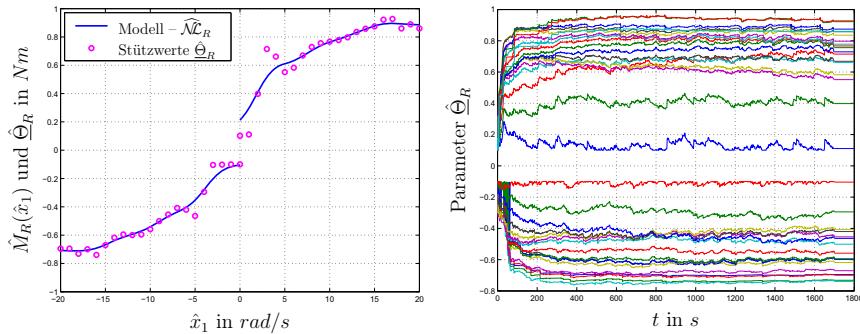
Es ist zu erkennen, dass der Winkelgeschwindigkeitsfehler  $\hat{x}_1 - x_1$  klein wird und die Parameter konvergieren. Dadurch, dass die Parameter relativ langsam konvergieren, steigt der Fehler im Parallelbetrieb wieder merklich an. Ein längerer Lernvorgang würde das Ergebnis noch verbessern. Würde lediglich die Parame-

<sup>31)</sup> Die Parameter  $\hat{\Theta}_R$  der Reibkennlinie werden wieder gesondert dargestellt.



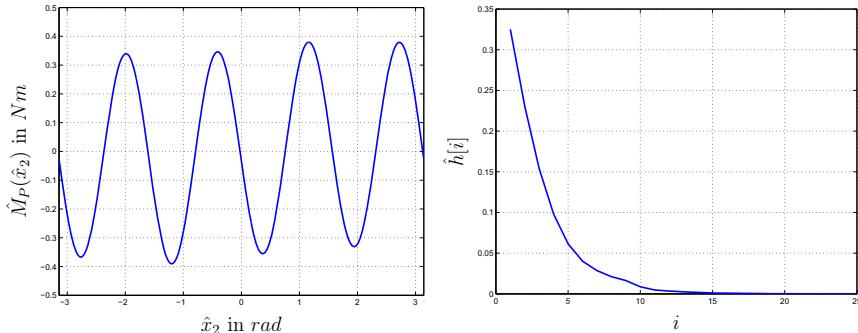
**Abb. 9.27:** Winkelgeschwindigkeitsfehler (links) und Konvergenz der Parameter (rechts)

teradaption eingestellt werden und die Korrektur der Zustände über die Beobachterrückführungen weiterhin erfolgen, wäre kein Anstieg im Fehler zu erkennen. In Abb. 9.28 links ist die identifizierte Reibkennlinie sowie rechts die Konvergenz der zugehörigen Parameter dargestellt.



**Abb. 9.28:** 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. 9.29 veranschaulicht.



**Abb. 9.29:** Identifikationsergebnis der periodischen Nichtlinearität (links) und der identifizierten Impulsantwort (rechts)

## 9.6 Zusammenfassung

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 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-Modells sowie eines weiteren Simulationsbeispiels wurde die vorgestellte Theorie veranschaulicht.

Abschließend wurde an einem mechatronischen Antriebssystem die Leistungsfähigkeit sowie die Praxisrelevanz des vorgestellten Identifikationsverfahrens verdeutlicht.

## 10 Nichtlineare Optimierung in der Systemidentifikation

Das Kapitel 4 führte in die Grundlagen der Parameteroptimierung ein. Dabei wurde für den Fall der nichtlinearen Optimierung lediglich das einfache Gradientenabstiegsverfahren betrachtet. Die nichtlineare Optimierung in der angewandten Mathematik kennt jedoch deutlich leistungsfähigere Verfahren, welche in diesem Kapitel näher untersucht werden sollen. Die Systemidentifikation mit Neuronalen Netzen führt bei  $N$  Modellparametern allgemein auf ein  $N$ -dimensionales Optimierungsproblem der Form:

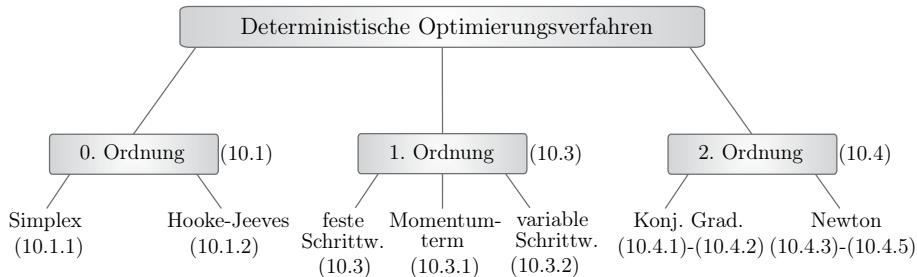
$$\min_{\hat{\Theta}} E(\hat{\Theta}), \quad \hat{\Theta} \in \mathbb{R}^N \quad (10.1)$$

Da es bei der Systemidentifikation mit Neuronalen Netzen meist keine Einschränkung im Lösungsraum gibt, müssen bei der Optimierung keine Nebenbedingungen berücksichtigt werden, und es handelt sich um ein *nichtlineares Optimierungsproblem ohne Nebenbedingungen* [44, 213, 18, 64, 177]. Für die Lösung dieser Aufgabe (10.1) existieren in der Regel keine analytischen Verfahren. Es kommen stattdessen iterative Algorithmen zum Einsatz, die entweder deterministischer oder stochastischer Natur sind. *Deterministische Optimierungsverfahren* (auch *lokale Optimierungsverfahren* genannt) werten die Fehlerfläche lokal an einem Punkt aus, während *stochastische Verfahren* (auch *globale Optimierungsverfahren* genannt) gleichzeitig mehrere Punkte berechnen, um ein Gesamtbild des Optimierungsproblems zu erfassen. Dieses Kapitel beschäftigt sich mit den deterministischen Optimierungsverfahren und orientiert sich hauptsächlich an den Ausführungen in der Arbeit [44]. Einen kurzen Einblick in die stochastischen Optimierungsverfahren gibt das nächste Kapitel 11.

Deterministische Optimierungsverfahren starten die Minimumssuche an einem einzigen Fehlerflächenpunkt und untersuchen lediglich die Fehlerfläche in der näheren Umgebung um diesen Punkt. Die Verfahren unterscheiden sich darin, welche Ableitungsberechnungen notwendig sind und wie die gewonnene Information über die Fehlerfläche letztendlich zur Parameteranpassung genutzt wird. Ist eine Ableitungsberechnung erforderlich, so muss die Fehlerfläche  $E(\hat{\Theta})$  ausreichend glatt — also ausreichend oft differenzierbar — sein. Die Algorithmen in diesem Kapitel sind *Abstiegsverfahren*. Das bedeutet, dass diese Algorithmen, ausgehend von einem Startpunkt  $\hat{\Theta}_0$  auf der Fehlerfläche, eine Folge von Parametervektoren  $\{\hat{\Theta}_k\}$  mit  $E(\hat{\Theta}_{k+1}) < E(\hat{\Theta}_k)$  generieren, bis ein Abbruchkriterium erfüllt

ist. Die Optimierungsverfahren liefern somit eine Näherung für die Lösung von (10.1). Der Abbruch bei einer erfolgreichen Optimierung erfolgt in der Praxis meist bei Unterschreiten eines vorgegebenen Wertes der Kostenfunktion, wenn also das Modell das unbekannte System ausreichend genau wiedergeben kann. Der Überblick in Abbildung 10.1 zeigt die in diesem Kapitel behandelten deterministischen Optimierungsverfahren. Bei den Verfahren unterscheidet man — je nach Ableitungsberechnung — Optimierungsverfahren 0., 1. und 2. Ordnung:

- 0. Ordnung: Die Optimierung erfordert keine Ableitungsberechnung.
- 1. Ordnung: Die Algorithmen verwenden die erste Ableitung. Die Taylorapproximation der Fehlerfläche wird nach dem ersten Glied abgebrochen.
- 2. Ordnung: Direkter oder indirekter Einsatz der Hessematrix. Die Taylorapproximation der Fehlerfläche wird nach dem zweiten Glied abgebrochen.



**Abb. 10.1:** Übersicht der deterministischen Optimierungsverfahren — Die Angaben in Klammern geben das jeweilige Kapitel an.

Das erste Kapitel 10.1 behandelt die Simplex-Methode und das Hooke-Jeeves-Tastverfahren als Vertreter der *Optimierungsverfahren 0. Ordnung*. Bevor die Verfahren mit Ableitungsberechnung in den Kapiteln 10.3 und 10.4 eingeführt werden, folgt im Kapitel 10.2 ein Exkurs in die eindimensionale Optimierung. Diese sogenannte *Liniensuche* arbeitet in vielen Optimierungsverfahren als Subroutine und trägt damit entscheidend zum Erfolg des Gesamtaufbaus bei. Die *Optimierungsverfahren 1. Ordnung* in Kapitel 10.3 beschreiben zunächst den sehr häufig eingesetzten Gradientenabstieg, der sich bei jedem Optimierungsschritt mit einer festen Schrittweite in die negative Gradientenrichtung bewegt. Eine Verbesserung des einfachen Gradientenabstiegs lässt sich durch die Hinzunahme eines Momentumterms (Kapitel 10.3.1) oder durch den Einsatz einer variablen Schrittweite (Kapitel 10.3.2) erreichen. Deutlich leistungsfähiger sind die Verfahren zweiter Ordnung in Kapitel 10.4. Diese gliedern sich in die Konjugierten Gradientenverfahren und in die Newton-Verfahren. Die *Optimierungsverfahren 2. Ordnung* bewegen sich nicht in die negative Gradientenrichtung, sondern streben zum stationären Punkt der Taylorapproximation zweiter Ordnung. Dieser

Punkt muss nicht immer ein Minimum sein, es ist auch möglich, dass es sich bei diesem Punkt um einen Sattelpunkt oder um ein Maximum handelt. Zur Gewährleistung von Stabilität verwenden die Verfahren 2. Ordnung eine untergeordnete Liniensuche oder ein Skalierungsverfahren, welches den Gültigkeitsbereich der Taylorapproximation 2. Ordnung einstellt (häufig als model-trust-region-Methode bezeichnet, mehr dazu folgt im Kapitel 10.4.5). Damit gibt es bei den Optimierungsverfahren grundsätzlich vier unterschiedliche Algorithmen: Konjugierte Gradientenabstiegsverfahren — entweder mit Liniensuche oder mit Skalierung — und Newton-Verfahren — entweder mit Liniensuche oder mit Skalierung. Eine Übersicht der behandelten Verfahren 2. Ordnung bietet Abbildung 10.22 auf Seite 353.

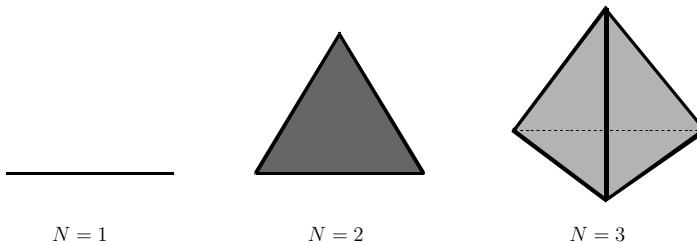
In den folgenden Ausführungen endet jedes Kapitel mit einer kurzen Zusammenfassung des jeweiligen Optimierungsverfahrens und einem einfachen Optimierungsbeispiel. Die Zusammenfassungen helfen bei der Implementierung und beim Vergleichen der vorgestellten Verfahren. Das Beispiel zeigt jeweils anschaulich den zurückgelegten Weg bei der Optimierung und die dabei durchgeföhrten Berechnungen, bevor die eher abstrakte Anwendung der Systemidentifikation mit Neuronalen Netzen folgt. Das Kapitel 10.5 bewertet die betrachteten deterministischen Optimierungsverfahren und diskutiert die Ergebnisse beim Optimierungsbeispiel. Der Übergang zur Systemidentifikation mit Neuronalen Netzen geschieht im letzten Kapitel 10.6.

## 10.1 Optimierungsverfahren 0. Ordnung

Bei den Optimierungsverfahren 0. Ordnung sind keine Ableitungsberechnungen notwendig. Diese Verfahren werten lediglich die Kostenfunktion an mehreren Punkten aus. Dies verschafft natürlich keinen guten Einblick in die tatsächliche Form der Fehlerfläche. Deshalb sind die Verfahren 0. Ordnung wenig effizient und leiden an einer langsamen Parameterkonvergenz. Diesem großen Nachteil stehen einige Vorteile gegenüber: Die Optimierungsverfahren 0. Ordnung sind leicht verständlich und einfach zu implementieren. Sie können auch dann zum Einsatz kommen, wenn eine Ableitungsberechnung der Kostenfunktion nicht möglich ist.

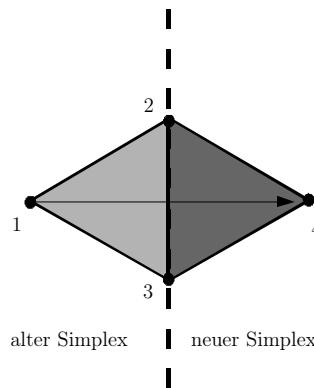
### 10.1.1 Die Simplex-Methode

Ein Simplex ist das einfachste Volumen, das einen  $N$ -dimensionalen Parameterraum aufspannen kann. Er besteht aus  $N + 1$  Ecken und ist im Eindimensionalen eine Strecke, im Zweidimensionalen ein Dreieck und im Dreidimensionalen ein Tetraeder, wie in Abbildung 10.2 dargestellt. Jeder Punkt des  $N + 1$ -dimensionalen Simplex stellt einen  $N$ -dimensionalen Gewichtsvektor im Parameterraum dar. Bei der Systemidentifikation beschreibt somit jeder Punkt ein eigenes Modell mit eigenen Parametern und eigenem der Modellgüte entsprechendem Kostenfunktionswert. In jedem Optimierungsschritt wird ein neuer Simplex bestimmt.



**Abb. 10.2:** Geometrische Interpretation eines Simplex — für  $N = 1$ : Linie, für  $N = 2$ : Dreieck und für  $N = 3$ : Tetraeder.

Das grundlegende Simplex-Verfahren von Spendley [220] ersetzt dabei den Punkt mit dem höchsten Funktionswert durch einen neuen Punkt. Der neue Punkt entsteht dabei durch Spiegelung des Punktes mit dem höchsten Funktionswert an der Hyperebene der verbleibenden Punkte. Abbildung 10.3 zeigt dieses Vorgehen für den zweidimensionalen Fall. Spendley fordert dabei, dass der neu entstandene

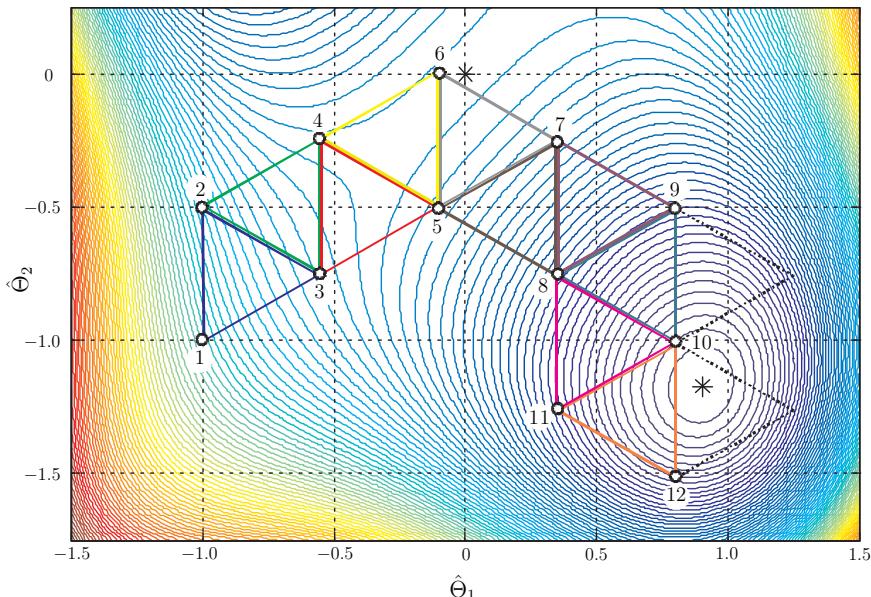


**Abb. 10.3:** Generieren eines neuen Simplex 2-3-4 für den zweidimensionalen Fall — Der Punkt 1 mit dem höchsten Kostenfunktionswert wird an der Spiegellinie (Hyperebene) der verbleibenden Punkte 2 und 3 gespiegelt.

Punkt im neuen Simplex nicht wieder der Punkt mit dem höchsten Kostenfunktionswert sein darf, um ein vorzeitiges Oszillieren zu vermeiden. Falls der neu generierte Punkt wieder den höchsten Kostenfunktionswert aufweist, wird nicht der Punkt mit dem höchsten Kostenfunktionswert gespiegelt, sondern der Punkt mit dem zweitschlechtesten Wert. Dies führt in der Nähe eines Minimums zu einer kreisförmigen Umwanderung dieses lokalen Optimums [220]. Der grundlegende Simplex-Algorithmus besteht somit aus zwei Regeln: Spiegelung des schlechtesten Simplex-Punktes an der Hyperebene der verbleibenden Punkte und Vermeidung zyklisch wiederkehrender Simplexe.

### Beispiel — Optimierung mit der Simplex-Methode

Das Höhenlinienbild in Abbildung 10.4 zeigt beispielhaft die Simplex-Optimierung einer zweidimensionalen Kostenfunktion<sup>1)</sup>. Nach geeigneter Wahl der Simplex-



**Abb. 10.4:** Beispiel einer Simplex-Optimierung — ein neuer Simplex entsteht jeweils durch Spiegeln des Punktes mit dem höchsten Kostenfunktionswert an der Hyperebene der verbleibenden Punkte. Beim Simplex 8-9-10 wird der Punkt mit dem zweit höchsten Kostenfunktionswert gespiegelt. Der Punkt 10 ist das Ergebnis der Optimierung. Die Kreuze kennzeichnen die stationären Punkte der Kostenfunktion.

Kantenlänge und Platzieren des ersten Simplex 1-2-3 auf der Fehlerfläche kann die Optimierung beginnen. In diesem ersten Simplex hat der Punkt 1 den höchsten Kostenfunktionswert und wird deshalb an der Linie 2-3 (Hyperebene) gespiegelt, es entsteht der neue Punkt 4 und damit der neue Simplex 2-3-4. Analog dazu folgen weitere Spiegelungen: Punkte 2 → 5, 3 → 6, 4 → 7, 6 → 8, 5 → 9 und 7 → 10. Beim Simplex 8-9-10 tritt beim Spiegeln des höchsten Punktes 8 der Fall auf, dass der neu generierte Punkt wieder der höchste Punkt des neuen Simplex wäre. In diesem Fall erfolgt die Spiegelung des zweithöchsten Punktes 9. Da die Optimierung mit dem Simplex 10-11-12 nicht weiter fortschreitet, ist

<sup>1)</sup> Hinweis: Bei der untersuchten Kostenfunktion handelt es sich um die nichtlineare Funktion  $E(\hat{\Theta}) = 2 \cdot \hat{\Theta}_1^4 + \hat{\Theta}_2^4 - 2 \cdot \hat{\Theta}_1^2 - 2 \cdot \hat{\Theta}_2^2 + 4 \cdot \sin(\hat{\Theta}_1 \cdot \hat{\Theta}_2) + 5$ . Da eine analytische Beschreibung der Kostenfunktion vorliegt, ist dieses Beispiel im weiteren Verlauf des Buches sehr hilfreich bei der Einführung und Veranschaulichung der Optimierungsalgorithmen.

an dieser Stelle die Optimierung beendet und ein Minimum lokalisiert. Punkt 10 hat den kleinsten Funktionswert und ist somit das Ergebnis der Optimierung.

Beim Simplex-Verfahren handelt es sich um ein lokales Optimierungsverfahren. Nur die nähere lokale Umgebung entscheidet über das Fortschreiten auf der Fehlerfläche. Die Kantenlänge des Simplex wirkt dabei wie eine Schrittweite der Optimierung. Analog zum Gradientenabstieg (siehe Kapitel 10.3) erreicht man bei einer großen Schrittweite ein schnelles Auffinden des lokalen Minimums, dieses ist jedoch dann sehr ungenau. Zur genaueren Lokalisierung des Minimums schlägt Spendley [220] vor, die bei der Optimierung berechneten Fehlerflächenpunkte in der Umgebung des Minimums für eine quadratische Approximation der Fehlerfläche zu nutzen. Eine weitere Möglichkeit für ein genaueres Auffinden des Minimums wäre, den Simplex-Algorithmus an der gefundenen Stelle mit einer reduzierten Kantenlänge neu zu starten.

In der Literatur sind zahlreiche Erweiterungen zum ursprünglichen Simplex-Verfahren bekannt [165, 1, 16]. So sieht beispielsweise das modifizierte Simplex-verfahren von Nelder und Mead [165] eine Verkleinerung sowie auch eine Vergrößerung des Simplex vor. Diese Modifikation führt zu einer schnelleren Konvergenz und zu einem genaueren Endergebnis, da sich der Simplex an die lokalen Eigenschaften der Fehlerfläche anpassen kann.

### 10.1.2 Das Hooke-Jeeves-Tastverfahren

Noch einfacher als die Simplex-Methode ist das Tastverfahren von Hooke-Jeeves [95, 183, 169, 12], das von einem zufällig gewählten Anfangspunkt  $\hat{\Theta}_0$  auf der Fehlerfläche startet. Das Verfahren besteht aus zwei grundsätzlichen Teilen: Einem Tastzyklus zur Informationsgewinnung und dem gezielten Voranschreiten durch Erfahrung. Beim ersten Teil erfolgt jeweils ein kleiner Tastschritt in eine der  $N$  Koordinatenachsen mit einer festen Tastschrittweite<sup>2)</sup>. Es ist also jeweils nur ein Element des  $N$ -dimensionalen Gewichtsvektors zu ändern und der Kostenfunktionswert an dieser Stelle auszuwerten. Steigt der Kostenfunktionswert an, wird ein weiterer Tastschritt (mit der gleichen Schrittweite) in die negative Richtung durchgeführt. Missglückt auch dieser Tastschritt, so findet keine Gewichtsanpassung in diese Koordinatenrichtung statt. Bei erfolgreichem Tasten führt der Algorithmus die gefundene Gewichtsanpassung auch tatsächlich aus, bevor die Untersuchung der nächsten Koordinatenrichtung beginnt. Waren alle Tastschritte in die  $N$  Koordinatenrichtungen erfolglos, so ist die Tastschrittweite zu groß und muss reduziert werden (z.B. halbieren). Ist die sukzessive Untersuchung in alle Koordinatenrichtungen abgeschlossen, so entsteht an der erreichten Fehlerflächenposition ein sogenannter Basispunkt  $B_k$  zum Optimierungsschritt  $k$ . Dieser Punkt dient als Ausgangspunkt für den weiteren Verlauf der Optimierung. Der erste Basispunkt  $B_0$  ist der zufällig gewählte Anfangspunkt  $\hat{\Theta}_0$ .

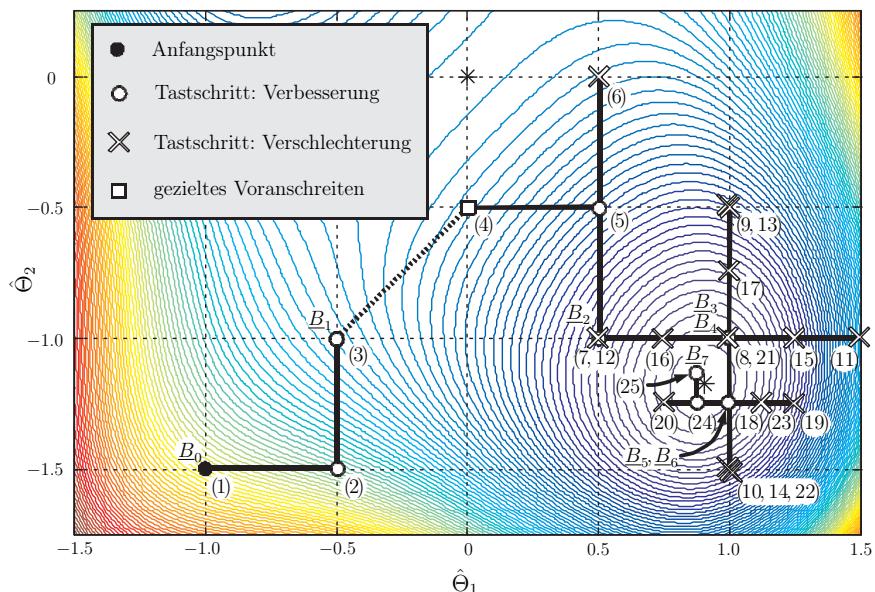
---

<sup>2)</sup> Hinweis: Die Tastschritte müssen nicht unbedingt in die Richtungen der Koordinatenachsen erfolgen. Es sind auch andere linear unabhängige Suchrichtungen möglich.

Der zweite Teil des Algorithmus nutzt die gesammelte Erfahrung aus den vorhergehenden Tastschritten aus. Das gezielte Voranschreiten testet dabei, ob ein Schritt in die Richtung  $\underline{B}_k - \underline{B}_{k-1}$  zu einem verbesserten Kostenfunktionswert führt. Dies ist zu erwarten, da die Differenz der Basispunkte  $\underline{B}_k - \underline{B}_{k-1}$  in den meisten Fällen eine gute Näherung für den negativen Gradienten darstellt. Falls das gezielte Voranschreiten zu einer Verschlechterung führt, startet sofort (also ohne gezieltem Voranschreiten) der nächste Optimierungsschritt  $k + 1$  mit einem neuen Tastzyklus.

### Beispiel — Optimierung mit dem Hooke-Jeeves-Tastverfahren

Das Höhenlinienbild in Abbildung 10.5 zeigt beispielhaft eine Hooke-Jeeves-Optimierung einer zweidimensionalen Kostenfunktion mit dem Startpunkt  $\hat{\Theta}_0 = [-1 - 1.5]^T$ . Die anfängliche Schrittweite bei den Tastzyklen ist 0.5. Die Zahlen



**Abb. 10.5:** Beispiel einer Optimierung mit dem Hooke-Jeeves-Tastverfahren — der Tastzyklus untersucht jeweils kleine Schritte in die Richtungen der Koordinatenachsen. Dem Tastzyklus folgt das gezielte Voranschreiten. Dabei wird versucht, noch einmal in die gleiche Richtung voranzuschreiten. Der Anfangspunkt der Optimierung ist  $\underline{B}_0$ , das Ergebnis ist der Punkt  $\underline{B}_7$ .

in Klammern geben die Bewegung auf der Fehlerfläche während der Optimierung an. So gehören beispielsweise die Punkte (1) bis (4) zum ersten Optimierungsschritt. Abbildung 10.5 beschreibt die ersten 7 Optimierungsschritte mit den dabei gesetzten Basispunkten. Das gezielte Voranschreiten führt nur beim ersten

Optimierungsschritt zu einer Verbesserung. In der Nähe des Minimums (d.h. bei den Optimierungsschritten 2 bis 7) sind alle Schritte mit gezieltem Voranschreiten erfolglos (all diese Schritte würden zu einem Anstieg der Kostenfunktion führen). Nach dem 4. und nach dem 6. Optimierungsschritt findet eine Halbierung der Tast-Schrittweite statt. Der letzte Basispunkt  $\underline{B}_7$  liegt nahe am gesuchten Minimum der Fehlerfläche.

Das Hooke-Jeeves-Verfahren ist recht aufwändig, da in einem Optimierungsschritt für jede Koordinatenachse (also für jeden einzustellenden Parameter) mindestens eine Funktionsauswertung durchgeführt werden muss. Kleinere Veränderungen des hier vorgestellten Verfahrens sind möglich (z.B. separate Schrittweite für jede Koordinatenrichtung; die Schrittweite in eine Koordinatenrichtung wird reduziert, falls das Tasten in diese Richtung erfolglos war). Größere Veränderungen sind jedoch nicht sinnvoll, da die größten Vorteile des hier vorgestellten Verfahrens die einfache Theorie und Implementierung sind.

Die Verfahren 0. Ordnung sind aufgrund ihrer einfachen Theorie sehr beliebt. Ein praktischer Einsatz ist jedoch nur dann zu empfehlen, falls eine Gradientenberechnung nicht möglich ist [169]. Bei der Systemidentifikation mit Neuronalen Netzen ist eine Ableitungsberechnung immer ausführbar. Deshalb werden aufgrund der schlechten Konvergenzeigenschaften die Verfahren 0. Ordnung nicht näher untersucht. Für eine ausführlichere Beschreibung der Simplex-Methode und des Hooke-Jeeves-Tastverfahrens sei auf die angegebene Literatur verwiesen.

## 10.2 Verfahren zur Liniensuche

Mehrere leistungsfähige Algorithmen der nichtlinearen Optimierung erfordern das Auffinden eines Minimums entlang einer zuvor berechneten *Suchrichtung*  $\underline{s}_k$ :

$$\eta_k = \min_{\eta} E(\hat{\Theta}_k + \eta \cdot \underline{s}_k) \quad (10.2)$$

Die Aufgabe dieser **Liniensuche** ist, einen Wert für die **Schrittweite**  $\eta_k$  zu finden, der das in Gleichung (10.2) beschriebene *eindimensionale Optimierungsproblem* im  $N$ -dimensionalen Parameterraum löst. Die Liniensuche tritt beispielsweise beim Nichtlinearen Konjugierten Gradientenverfahren (Kapitel 10.4.1) und beim Quasi-Newton-Verfahren (Kapitel 10.4.4) als zu lösendes Teilproblem auf und muss mit möglichst geringem Rechenaufwand bewältigt werden, da sie in jedem Iterationsschritt vorkommt und deshalb das Verhalten der Gesamtalgorithmen beeinflusst. Ein einfaches Verfahren zur Liniensuche wäre, kleine diskrete Schritte in die Suchrichtung  $\underline{s}_k$  zu gehen, bis der Wert der Kostenfunktion wieder ansteigt [98, 18]. Diese Methode benötigt viele Funktionsauswertungen und ist deshalb wenig effizient. Die numerische Optimierung kennt viele und deutlich leistungsfähigere Algorithmen zur Liniensuche. Diese Verfahren lassen sich prinzipiell in zwei Gruppen einteilen: Verfahren, welche lediglich die Kostenfunktion auswerten und Verfahren, welche zusätzlich die Berechnung der ersten Ableitung

der Kostenfunktion verwenden. Für die Optimierung von Neuronalen Netzen ist nach Bishop [18] die erste Gruppe ohne Ableitungsberechnung effizienter. Aus diesem Grund beschränkt sich das vorliegende Buch auf Liniensuchverfahren, welche lediglich die Kostenfunktion auswerten.

Ein Liniensuchverfahren besteht grundsätzlich aus zwei Schritten [18, 83, 79]. Zunächst gilt es, einen Bereich zu finden, indem sich das Minimum befindet. Dieser sogenannten *Intervallsuchphase* folgt eine *Intervallverkleinerungsphase*, welche das eigentliche Minimierungsproblem löst. Die Intervallsuche hat die Aufgabe, ein Grundintervall zu finden, indem es drei Punkte  $a_1 < a_2 < a_3$  entlang der Suchrichtung  $\underline{s}_k$  ermittelt, für die die folgenden Zusammenhänge<sup>3)</sup>

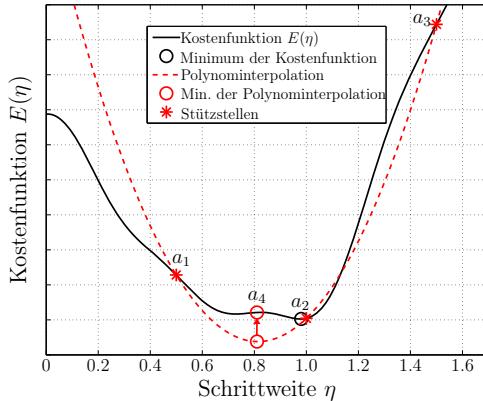
$$E(\hat{\Theta}_k + a_1 \cdot \underline{s}_k) > E(\hat{\Theta}_k + a_2 \cdot \underline{s}_k) \quad \text{und} \quad E(\hat{\Theta}_k + a_2 \cdot \underline{s}_k) < E(\hat{\Theta}_k + a_3 \cdot \underline{s}_k) \quad (10.3)$$

gelten. Wegen der Kontinuität der Fehlerfläche muss mit den Beziehungen (10.3) ein Minimum im Intervall  $[a_1 \ a_3]$  enthalten sein. Dies zeigt beispielhaft die schwarze Linie in Abbildung 10.6, bei der die Kostenfunktion abhängig von der Schrittweite  $\eta$  entlang der Suchrichtung  $\underline{s}_k$  dargestellt ist. Ein Verfahren zur Intervallsuche beschreibt das Kapitel 10.2.1.1. Die Intervallverkleinerung reduziert dieses ermittelte *Grundintervall* solange, bis das Minimum ausreichend genau gefunden ist. Bei der Intervallverkleinerung unterscheidet man zwischen *Vergleichsverfahren* und *Interpolationsverfahren*. Die Vergleichsverfahren berechnen gezielt weitere Punkte innerhalb des Grundintervalls und ermöglichen mit dieser zusätzlichen Information eine ständige Verkleinerung des Intervalls. Zu den Vergleichsverfahren zählen beispielsweise die Fibonacci-Suche und die am häufigsten verwendete [214] Goldene-Schnitt-Suche, die ausführlich im Kapitel 10.2.1.2 behandelt wird. Die Interpolationsverfahren nähern die Fehlerfläche entlang der Suchrichtung durch ein Polynom zweiten oder dritten Grades an. Durch die Auswertung der Kostenfunktion am Minimum des Polynoms ist es möglich, das Intervall fortwährend zu reduzieren [18], vergleiche dazu die Beschreibung in Abbildung 10.6.

Wie bereits oben erwähnt, bieten sich bei der Optimierung von Neuronalen Netzen ableitungsfreie Verfahren an. Insbesondere bei dynamischen Modellen ist die Gradientenberechnung mit einem erheblichen Rechenaufwand verbunden. Hagan schlägt für die Optimierung von Neuronalen Netzen vor, nach der Intervallsuche das Intervall mit dem Goldenen-Schnitt-Verfahren zu verkleinern [79]. Diese klassische Liniensuche beschreibt das Kapitel 10.2.1 genauer, da sie in diesem Buch bei vielen Optimierungen mit dem Nichtlinearen Konjugierten Gradientenverfahren (Kapitel 10.4.1) und dem Quasi-Newton-Verfahren (Kapitel 10.4.4) zum Einsatz kommt. Für alle anderen Liniensuchverfahren sei auf die Literatur [61, 174, 192] verwiesen.

---

<sup>3)</sup> Hinweis: Im weiteren Verlauf werden die Begriffe Punkt und Schrittweite synonym verwendet. Der Punkt  $a_i$  im eindimensionalen Optimierungsproblem entspricht einer Schrittweite in Richtung der Suchrichtung  $\underline{s}_k$  auf der mehrdimensionalen Fehlerfläche und liegt somit an der Stelle  $\hat{\Theta}_k + a_i \cdot \underline{s}_k$ .



**Abb. 10.6:** Intervallverkleinerung durch Polynominterpolation — Die durchgezogene schwarze Linie zeigt beispielhaft den Verlauf einer Fehlerfläche entlang der Suchrichtung. Die Kostenfunktion ist an den Punkten  $a_1$ ,  $a_2$  und  $a_3$  ausgewertet. Diese drei Punkte führen auf die gestrichelt eingezeichnete Parabel, deren Minimum  $a_4$  eine Näherung für das Minimum der Fehlerfläche darstellt. Das Minimum der Fehlerfläche lässt sich mit diesem zusätzlichen Punkt  $a_4$  auf das Intervall  $[a_4 \ a_3]$  einschränken (Erläuterungen dazu findet man in Abbildung 10.9). Im nächsten Schritt erfolgt wieder eine Fehlerflächenapproximation mit einer Parabel durch die drei Punkte  $a_4$ ,  $a_2$  und  $a_3$ . Die Intervallverkleinerung wird nun sukzessive fortgeführt.

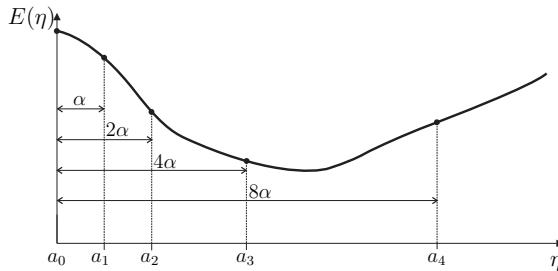
Bei allen Liniensuchverfahren besteht das Problem, dass der Anwender spezifische Parameter einstellen muss. Die richtige Einstellung dieser vom Benutzer definierten Parameter kann bei der Optimierung von Neuronalen Netzen äußerst schwierig sein, da sich das Verhalten der Fehlerfläche abhängig von der Suchrichtung deutlich ändert. Falsch eingestellte Parameter führen dazu, dass entweder die Liniensuche viel Rechenzeit benötigt oder zu ungenau arbeitet. Dies führt wieder zu einem langsamen Gesamtalgorithmus. Aus diesem Grund besteht der Wunsch nach einem Liniensuchverfahren, das ohne vordefinierte Parameter auskommt, sich also selbst der Beschaffenheit der Fehlerfläche anpasst. Das Kapitel 10.2.2 schlägt mit dem ALIS-Algorithmus ein adaptives Liniensuchverfahren vor, das ohne benutzerdefinierte Voreinstellungen auskommt und bei der Identifikation mit Neuronalen Netzen erstaunlich gut funktioniert [45].

### 10.2.1 Ein klassisches Liniensuchverfahren mit Intervallsuchphase und Intervallverkleinerungsphase

Wie bereits einführend erklärt, setzt sich die Liniensuche aus zwei Teilen zusammen: Die Intervallsuche und die Intervallverkleinerung. Die folgenden Ausführungen beschreiben die in diesem Buch verwendeten Algorithmen.

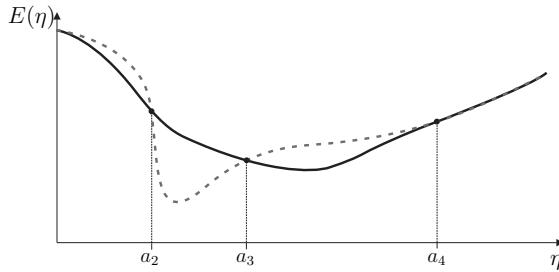
### 10.2.1.1 Die Intervallsuchphase

Die Intervallsuche hat die Aufgabe, ein anfängliches Grundintervall zu finden, das ein lokales Minimum enthält. Die Intervallsuche beginnt mit der Auswertung der Kostenfunktion an einem Anfangspunkt  $a_0$ . Dieser Punkt entspricht dem aktuellen Gewichtsvektor  $\hat{\Theta}_k$  und somit einer Schrittweite von  $\eta = 0$ . Wie in Abbildung 10.7 gezeigt, wird die Fehlerfläche nun an einem weiteren Punkt



**Abb. 10.7:** Intervallsuche zur Ermittlung des Grundintervalls — Auswertung der Kostenfunktion an den markierten Punkten. Der Abstand dieser Punkte zum ersten Punkt  $a_0$  wird jeweils verdoppelt. Vom Punkt  $a_3$  auf den Punkt  $a_4$  steigt der Kostenfunktionswert an, was bedeutet, dass im Intervall  $[a_2 \ a_4]$  ein Minimum sein muss.

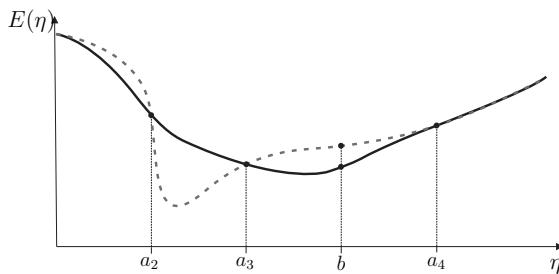
$a_1$  ausgewertet. Dieser Punkt hat die Entfernung  $\alpha$  vom ersten Punkt  $a_0$ . Das bedeutet, der Punkt  $a_1$  befindet sich auf der Fehlerfläche am Ort  $\hat{\Theta}_k + \alpha \cdot \underline{s}_k$ , was einer Schrittweite von  $\eta = \alpha$  entspricht. Der Wert der Kostenfunktion an dieser zweiten Stelle ist  $E(\hat{\Theta}_k + \alpha \cdot \underline{s}_k)$ . Es folgt eine sukzessive Berechnung weiterer Punkte  $a_i$ , wobei der Abstand der Punkte jedes mal verdoppelt wird. Die Schrittweite der  $i$ -ten Auswertung beträgt  $2^{i-1} \cdot \alpha$ . Die Intervallsuche ist beendet, wenn der Wert der Kostenfunktion beim Punkt  $a_i$  erstmalig ansteigt, also  $E(\hat{\Theta}_k + 2^{i-2} \cdot \alpha \cdot \underline{s}_k) < E(\hat{\Theta}_k + 2^{i-1} \cdot \alpha \cdot \underline{s}_k)$ . Damit ist ein lokales Minimum im Intervall  $[a_{i-2} \ a_i]$  lokalisiert. Ein Beispiel dazu zeigt Abbildung 10.7. Der Anstieg der Fehlerfläche geschieht beim Übergang von Punkt  $a_3$  auf den Punkt  $a_4$ . Es gilt  $E(\hat{\Theta}_k + 4 \cdot \alpha \cdot \underline{s}_k) < E(\hat{\Theta}_k + 8 \cdot \alpha \cdot \underline{s}_k)$ . Das Minimum muss sich also im Intervall  $[a_2 \ a_4]$  befinden. Eine weitere Einschränkung des Intervalls ist mit den durchgeführten Auswertungen nicht möglich, wie die Abbildung 10.8 zeigt. Bei den drei berechneten Punkten  $a_2 < a_3 < a_4$  entlang der Suchrichtung  $\underline{s}_k$  kann sich das Minimum entweder im Teilintervall  $[a_2 \ a_3]$  oder im Teilintervall  $[a_3 \ a_4]$  befinden. Das gefundene Intervall  $[a_{i-2} \ a_i]$  bildet das Grundintervall für den nun folgenden zweiten Teil der Liniensuche, der Intervallverkleinerung. Die Intervallverkleinerung berechnet weiterer Punkte innerhalb des gefundenen Grundintervalls, um mit einem kleiner werdenden Intervall das lokale Minimum immer weiter einzuschränken.



**Abb. 10.8:** Eine Intervallverkleinerung ist mit nur einem Zwischenpunkt im Intervall nicht möglich — Das Minimum kann sich im Teilintervall  $[a_2 \ a_3]$  (gestrichelte graue Linie) oder im Teilintervall  $[a_3 \ a_4]$  (durchgehogene schwarze Linie) befinden.

### 10.2.1.2 Die Intervallverkleinerungsphase

Wie in Abbildung 10.8 zu sehen, reicht ein einziger Punkt  $a_3$  innerhalb des Intervalls  $[a_2 \ a_4]$  nicht aus, um das gefundene Minimum genauer lokalisieren zu können. Dazu sind mindestens zwei Punkte im gefundenen Grundintervall notwendig, wie in Abbildung 10.9 dargestellt. Falls für die beiden zusätzlichen Punk-

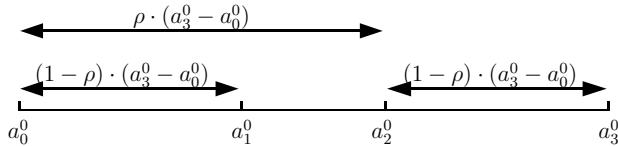


**Abb. 10.9:** Zwei Zwischenpunkte ermöglichen eine Intervallverkleinerung — für  $E(\hat{\Theta}_k + a_3 \cdot \underline{s}_k) > E(\hat{\Theta}_k + b \cdot \underline{s}_k)$  liegt das Minimum im Intervall  $[a_3 \ a_4]$  (durchgehogene schwarze Linie in Abbildung 10.9), für  $E(\hat{\Theta}_k + a_3 \cdot \underline{s}_k) < E(\hat{\Theta}_k + b \cdot \underline{s}_k)$  befindet sich im Intervall  $[a_2 \ b]$  ein Minimum (gestrichelte graue Linie).

te  $a_3$  und  $b$  im Grundintervall  $[a_2 \ a_4]$  (mit  $a_2 < a_3 < b < a_4$ ) der Zusammenhang  $E(\hat{\Theta}_k + a_3 \cdot \underline{s}_k) > E(\hat{\Theta}_k + b \cdot \underline{s}_k)$  gilt (durchgehogenen schwarzen Linie in Abbildung 10.9), muss ein lokales Minimum im Intervall  $[a_3 \ a_4]$  liegen und eine Aussage zur Intervallverkleinerung ist somit möglich. Analog dazu befindet sich für  $E(\hat{\Theta}_k + a_3 \cdot \underline{s}_k) < E(\hat{\Theta}_k + b \cdot \underline{s}_k)$  ein Minimum im Intervall  $[a_2 \ b]$ , wie im Falle der gestrichelten grauen Linie in Abbildung 10.9 gezeigt.

Es stellt sich nun die Frage, wo die beiden zusätzlichen Punkte zur Intervallverkleinerung platziert werden müssen, um den Rechenaufwand möglichst gering zu halten. Wünschenswert wäre es, nicht jedes Mal beide inneren Punkte bei einer Intervallverkleinerung neu berechnen zu müssen. Mit den beiden Intervallgrenzen

und den beiden notwendigen inneren Punkten stehen bei jedem Intervallverkleinerungsschritt vier Punkte zur Verfügung. Die Intervallverkleinerung kommt mit lediglich einem neuen Punkt aus, wenn jeweils ein Punkt einer Intervallgrenze wegfällt und die anderen drei Punkte im neuen Intervall wieder Verwendung finden. Gesucht ist somit die erforderliche Position für die beiden inneren Punkte, so dass eine Wiederverwendung von drei Punkten möglich ist. Für die nähere Be-



**Abb. 10.10:** *Teilungsverhältnisse der beiden inneren Punkte beim Grundintervall  $[a_0^0 a_3^0]$ .*

trachtung beschreibt  $\rho$  das Teilungsverhältnis der beiden inneren Punkte wie in Abbildung 10.10 dargestellt. Die hochgestellte Zahl bei den Punkten entspricht dem aktuellen Iterationsschritt der Intervallverkleinerung (eine hochgestellte 0 bezeichnet somit das Grundintervall). Nach Abbildung 10.10 unterteilt der erste innere Punkt  $a_1^0$  das Grundintervall  $[a_0^0 a_3^0]$  im Verhältnis  $(1 - \rho)$ , während der zweite innere Punkt  $a_2^0$  das Grundintervall im Verhältnis  $\rho$  teilt. Die getroffenen Annahmen lassen sich folgendermaßen formulieren:

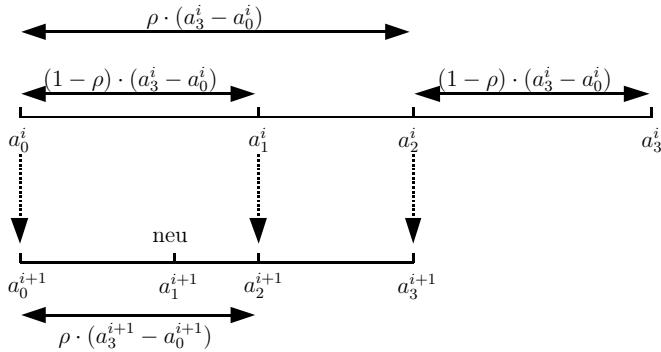
$$\begin{aligned} a_1^0 &= a_0^0 + (1 - \rho) \cdot (a_3^0 - a_0^0) = a_3^0 - \rho \cdot (a_3^0 - a_0^0) \\ a_2^0 &= a_0^0 + \rho \cdot (a_3^0 - a_0^0) = a_3^0 - (1 - \rho) \cdot (a_3^0 - a_0^0) \end{aligned} \quad (10.4)$$

Diese Verhältnisse gelten natürlich auch für einen beliebigen Intervallverkleinerungsschritt  $i$ :

$$\begin{aligned} a_1^i &= a_0^i + (1 - \rho) \cdot (a_3^i - a_0^i) \\ a_2^i &= a_0^i + \rho \cdot (a_3^i - a_0^i) \end{aligned} \quad (10.5)$$

Die Abbildung 10.11 verdeutlicht noch einmal die gewünschte Vorgehensweise bei der Intervallteilung, um drei Punkte wiederverwenden zu können. Das Intervall  $[a_0^i a_3^i]$  ist durch die beiden inneren Punkte  $a_1^i$  und  $a_2^i$  im Verhältnis  $(1 - \rho)$  bzw.  $\rho$  geteilt. Je nach Kostenfunktionswert an den beiden inneren Punkten wird im  $(i + 1)$ -ten Intervallverkleinerungsschritt einer der beiden Punkte an den Intervallgrenzen  $a_0^i$  oder  $a_3^i$  aufgegeben. In der Intervallteilung von Abbildung 10.11 fällt beispielsweise der Punkt  $a_3^i$  weg (hier soll der Zusammenhang  $E(\hat{\Theta}_k + a_1^i \cdot \underline{s}_k) < E(\hat{\Theta}_k + a_2^i \cdot \underline{s}_k)$  gelten, vergleiche Abbildung 10.9). Drei Punkte sollen erhalten bleiben:

$$a_0^{i+1} = a_0^i, \quad a_2^{i+1} = a_1^i \quad \text{und} \quad a_3^{i+1} = a_2^i \quad (10.6)$$



**Abb. 10.11:** Intervallteilung beim  $(i+1)$ -ten Intervallverkleinerungsschritt — Nur ein Punkt muss neu berechnet werden, drei Punkte bleiben im neuen Intervall  $[a_3^{i+1} - a_0^{i+1}]$  erhalten. Gesucht ist das Teilungsverhältnis  $\rho$ .

Der zweite innere Punkt des alten Intervalls  $a_2^i$  rückt somit an die hintere Grenze des neuen reduzierten Intervalls. Der erste innere Punkt des alten Intervalls  $a_1^i$  wird zum zweiten inneren Punkt im neuen Intervall  $a_2^{i+1}$ , für den dann das folgende Teilungsverhältnis gilt:

$$\begin{aligned} a_2^{i+1} - a_0^{i+1} &\stackrel{(10.5)}{=} \rho \cdot (a_3^{i+1} - a_0^{i+1}) \\ &\stackrel{(10.6)}{=} \rho \cdot (a_2^i - a_0^i) \\ &\stackrel{(10.5)}{=} \rho^2 \cdot (a_3^i - a_0^i) \end{aligned} \quad (10.7)$$

Mit den Annahmen von Gleichung (10.6) folgt weiter der Zusammenhang

$$a_2^{i+1} - a_0^{i+1} = a_1^i - a_0^i \stackrel{(10.5)}{=} (1 - \rho) \cdot (a_3^i - a_0^i) \quad (10.8)$$

Für das Teilungsverhältnis lässt sich mit den Gleichungen (10.7) und (10.8) die Bedingung

$$\rho^2 = 1 - \rho \quad (10.9)$$

ableiten mit der Lösung

$$\rho = \frac{\sqrt{5} - 1}{2} \quad (10.10)$$

Dieses gefundene Teilungsverhältnis ist bekannt als der *Goldene Schnitt* einer Strecke. Beim Goldenen Schnitt verhält sich die kleinere Teilstrecke zur größeren wie die größere Teilstrecke zur Gesamtstrecke [22, 46].

Die Intervallverkleinerung reduziert die Intervallgröße  $D_i := a_3^i - a_0^i$  bei jedem Intervallverkleinerungsschritt um den Faktor  $\rho$ :

$$\begin{aligned} D_{i+1} &= D_i \cdot \rho \\ &= D_0 \cdot \rho^{i+1} \end{aligned} \quad (10.11)$$

Als Abbruchbedingung der Intervallverkleinerung muss der Benutzer — je nach gewünschter Genauigkeit — eine Minimale Intervallgröße  $D_{min}$  angeben:

$$D_i < D_{min} \quad (10.12)$$

Für die Anzahl der für die Abbruchbedingung von Gleichung (10.12) notwendigen Intervallverkleinerungsschritte gilt:

$$\begin{aligned} D_0 \cdot \rho^i &< D_{min} \\ \rightarrow i &\geq \log_{\rho} \left( \frac{D_{min}}{D_0} \right) = \frac{\lg \left( \frac{D_{min}}{D_0} \right)}{\lg \rho} \approx -4.785 \cdot \lg \left( \frac{D_{min}}{D_0} \right) \end{aligned} \quad (10.13)$$

Die gefundenen Ergebnisse fasst der folgende Algorithmus zusammen.

### Zusammenfassung — Liniensuche mit Intervallsuchphase und anschliessender Intervallverkleinerung mit dem Goldenen-Schnitt-Verfahren

1. Festlegen der Distanz  $\alpha$  zwischen dem ersten und zweiten Punkt bei der Intervallsuche und einer gewünschten Genauigkeit  $D_{min}$  bei der Intervallverkleinerung.
2. Intervallsuche: Sukzessive Berechnung der Kostenfunktion an den Punkten  $a_i = 2^{i-1} \cdot \alpha$  entlang der Suchrichtung  $s_k$

$$E(\hat{\Theta}_k + 2^{i-1} \cdot \alpha \cdot s_k)$$

bis der Kostenfunktionswert ansteigt. Das Intervall  $[a_{i-2} \ a_i]$  ist das Grundintervall  $[a_0^0 \ a_3^0]$  für die nun folgende Intervallverkleinerung<sup>4)</sup>:

$$a_0^0 = a_{i-2} \quad \text{und} \quad a_3^0 = a_i$$

3. Berechnung der beiden inneren Punkte im Grundintervall  $[a_0^0 \ a_3^0]$  (Gleichung 10.5):

$$\begin{aligned} a_1^0 &= a_0^0 + (1 - \rho) \cdot (a_3^0 - a_0^0) \quad \text{damit} \quad E(\hat{\Theta}_k + a_1^0 \cdot s_k) \\ a_2^0 &= a_0^0 + \rho \cdot (a_3^0 - a_0^0) \quad \text{damit} \quad E(\hat{\Theta}_k + a_2^0 \cdot s_k) \end{aligned}$$

---

<sup>4)</sup> Hinweis: Falls die Distanz  $\alpha$  bei der Intervallsuche zu groß gewählt ist, kann gleich beim ersten Schritt  $i = 1$  der Funktionswert  $E(\hat{\Theta}_k + \alpha \cdot s_k)$  größer sein als  $E(\hat{\Theta}_k)$ . Die Intervallsuche kann in diesem Fall kein genauereres Ergebnis finden, und es folgt sofort die Intervallverkleinerung mit dem Grundintervall  $[0 \ \alpha]$ .

4. Intervallverkleinerung:

$$E(\hat{\Theta}_k + a_1^i \cdot \underline{s}_k) < E(\hat{\Theta}_k + a_2^i \cdot \underline{s}_k) : \\ \rightarrow a_0^{i+1} = a_0^i, \quad a_2^{i+1} = a_1^i \quad \text{und} \quad a_3^{i+1} = a_2^i$$

$$E(\hat{\Theta}_k + a_0^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_0^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + a_2^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_1^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + a_3^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_2^i \cdot \underline{s}_k)$$

→ Neuberechnung des ersten inneren Punktes (Gleichung 10.5):

$$a_1^{i+1} = a_0^{i+1} + (1 - \rho) \cdot (a_3^{i+1} - a_0^{i+1}) \\ E(\hat{\Theta}_k + a_1^{i+1} \cdot \underline{s}_k)$$

$$E(\hat{\Theta}_k + a_1^i \cdot \underline{s}_k) > E(\hat{\Theta}_k + a_2^i \cdot \underline{s}_k) : \\ \rightarrow a_0^{i+1} = a_1^i, \quad a_1^{i+1} = a_2^i \quad \text{und} \quad a_3^{i+1} = a_3^i \\ E(\hat{\Theta}_k + a_0^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_1^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + a_1^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_2^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + a_3^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + a_3^i \cdot \underline{s}_k)$$

→ Neuberechnung des zweiten inneren Punktes (Gleichung 10.5):

$$a_2^{i+1} = a_0^{i+1} + \rho \cdot (a_3^{i+1} - a_0^{i+1}) \\ E(\hat{\Theta}_k + a_2^{i+1} \cdot \underline{s}_k)$$

5. Iteration: Falls das Abbruchkriterium  $(a_3^{i+1} - a_0^{i+1}) < D_{min}$  noch nicht erfüllt ist, wiederhole den Punkt 4. Ansonsten ist  $\eta_k = (a_3^{i+1} + a_0^{i+1})/2$  die Lösung der Liniensuche.

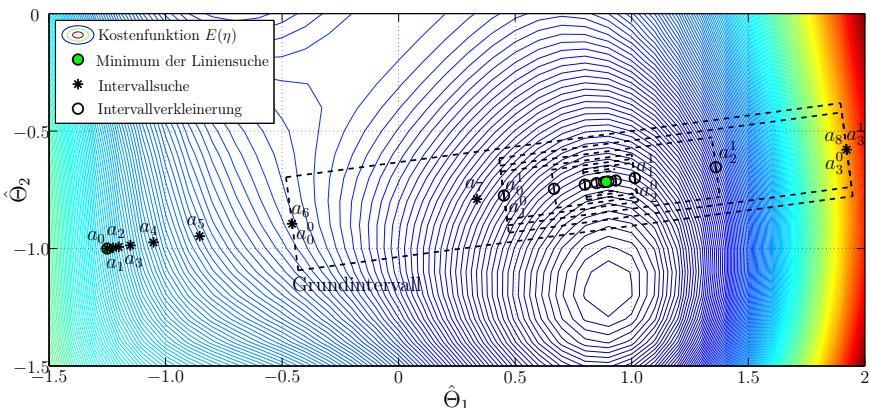
Die Verwendung der hier eingeführten Liniensuche soll im weiteren Verlauf des Buches mit dem Index ‘INT’ gekennzeichnet werden. So handelt es sich beispielsweise beim BFGS<sub>INT</sub>-Optimierungsverfahren um den in Kapitel 10.4.4.4 beschriebenen BFGS-Optimierungsalgorithmus mit einer Liniensuche durch Intervallsuche und anschließender Intervallverkleinerung mit dem Goldenen-Schnitt-Verfahren. Die Genauigkeit der ‘INT’-Liniensuche lässt sich mit wenig Aufwand verbessern, indem bereits berechnete Punkte in der Umgebung des lokalisierten Minimums für eine Polynominterpolation genutzt werden.

Beispiel — Liniensuche mit Intervallsuchphase und anschliessender Intervallverkleinerung mit dem Goldenen-Schnitt-Verfahren

Als Beispiel dient wieder die nichtlineare Funktion  $E(\hat{\Theta}) = 2 \cdot \hat{\Theta}_1^4 + \hat{\Theta}_2^4 - 2 \cdot \hat{\Theta}_1^2 - 2 \cdot \hat{\Theta}_2^2 + 4 \cdot \sin(\hat{\Theta}_1 \cdot \hat{\Theta}_2) + 5$ . Die Liniensuche startet vom Punkt  $\hat{\Theta}_k = [-1.25 \quad -1.00]^T$  in die negative Gradientenrichtung. Um günstigere Zahlenwerte zu erhalten, wird die Suchrichtung normiert<sup>5)</sup>. Für die normierte Suchrichtung ergibt sich:

$$s_k = \frac{-\underline{g}(\hat{\Theta}_k)}{\|\underline{g}(\hat{\Theta}_k)\|} = \frac{-\begin{bmatrix} -11.886 \\ -1.577 \end{bmatrix}}{11.990} = \begin{bmatrix} 0.991 \\ 0.132 \end{bmatrix}, \quad (10.14)$$

wobei der Vektor  $\underline{g}(\hat{\Theta}_k)$  den Gradienten bezeichnet, siehe Gleichung (10.25). Die benutzerdefinierten Daten werden mit  $\alpha = 0.025$  und  $D_{min} = 0.1$  relativ groß gewählt, damit die Ergebnisse in Abbildung 10.12 zu sehen sind. Die praktische Anwendung erfordert jedoch meist kleinere Werte. Die Funktionsauswertungen der



**Abb. 10.12:** Beispiel einer Liniensuche mit Intervallsuchphase und anschließender Intervallverkleinerung — Die Liniensuche startet vom Punkt  $\hat{\Theta}_k = [-1.25 \ -1.00]^T$  in die negative Gradientenrichtung.

Intervallsuche sind in Abbildung 10.12 jeweils mit einem Stern gekennzeichnet. Der Funktionswert  $E(\hat{\Theta}_k) = E([-1.250 - 1.000]^T) = 9.554$  des Anfangspunktes  $a_0$  sinkt beim Punkt  $a_1$  auf Wert  $E(\hat{\Theta}_k + \alpha \cdot s_k) = E([-1.225 - 0.997]^T) = 9.263$  ab. Die weiteren Punkte der Intervallsuche zeigt die Tabelle 10.1. Allgemein gilt: Falls der Funktionswert an der Stelle  $a_1$  bereits ansteigt, ist die gewählte Distanz  $\alpha$  zu groß und die Intervallsuche kann kein (genaueres) Ergebnis ermitteln. Der gesuchte Funktionsanstieg findet im Beispiel von Abbildung 10.12 beim Übergang

<sup>5)</sup> Hinweis: Die Normierung bezieht sich nur auf dieses Beispiel. Bei den Liniensuch-Subroutinen der Verfahren 2. Ordnung erfolgt natürlich keine Normierung der Suchrichtung.

vom Punkt  $a_7$  mit  $E(\hat{\Theta}_k + 2^6 \cdot \alpha \cdot \underline{s}_k) = E([0.336 - 0.790]^T) = 2.892$  auf den Punkt  $a_8$  mit  $E(\hat{\Theta}_k + 2^7 \cdot \alpha \cdot \underline{s}_k) = E([1.922 - 0.579]^T) = 20.768$  statt. Somit ist das Grundintervall  $[a_6 \ a_8]$  für die nun folgende Intervallverkleinerung gefunden. Abbildung 10.12 kennzeichnet die berechneten Punkte der Intervallverkleine-

Schritt $i$	0	1	2	3	4	5	6	7	8
$a_i$	0	0.025	0.050	0.100	0.200	0.400	0.800	1.600	3.200
$\hat{\Theta}_k + a_i \cdot \underline{s}_k$	-1.250 -1.000	-1.225 -0.997	-1.200 -0.993	-1.151 -0.987	-1.052 -0.974	-0.854 -0.947	-0.457 -0.895	0.336 -0.790	1.922 -0.579
$E(\hat{\Theta}_k + a_i \cdot \underline{s}_k)$	9.554	9.263	8.989	8.488	7.655	6.508	5.300	2.892	20.768

**Tabelle 10.1:** Intervallsuchphase — Die Funktionswerte der Fehlerfläche  $E(\hat{\Theta}_k + a_i \cdot \underline{s}_k)$  steigen beim Übergang vom Punkt  $a_7$  auf den Punkt  $a_8$  an.

rungsphase mit Kreisen und verdeutlicht die Intervallgröße mit gestrichelten Kästen. Der äußerste Kasten stellt das Grundintervall  $[a_0^0 \ a_3^0] = [a_6 \ a_8] = [0.8 \dots 3.2]$  dar. Zur besseren Übersicht nimmt die Kastengröße mit der Anzahl der Intervallverkleinerungsschritte ab. Die beiden inneren Punkte im Grundintervall berechnen sich zu (Gleichung (10.5))

$$a_1^0 = a_0^0 + (1 - \rho) \cdot (a_3^0 - a_0^0) = 0.8 + (1 - 0.618) \cdot (3.2 - 0.8) = 1.717$$

$$a_2^0 = a_0^0 + \rho \cdot (a_3^0 - a_0^0) = 0.8 + 0.618 \cdot (3.2 - 0.8) = 2.283$$

mit den Funktionswerten  $E(\hat{\Theta}_k + a_1^0 \cdot \underline{s}_k) = E([0.452 - 0.773]^T) = 2.465$  und  $E(\hat{\Theta}_k + a_2^0 \cdot \underline{s}_k) = E([1.012 - 0.699]^T) = 1.711$ . Da der zweite innere Punkt  $a_2^0$  zu einem kleineren Kostenfunktionswert führt als der erste innere Punkt, muss das Minimum im Intervall  $[a_0^0 \ a_3^0] = [a_0^1 \ a_3^1]$  liegen. Für dieses neue reduzierte Intervall gelten die Punkte  $a_0^1 = a_1^0$ ,  $a_1^1 = a_2^0$  und  $a_3^1 = a_3^0$ , wobei der zweite innere Punkt mit Gleichung (10.5) neu berechnet werden muss:

$$a_2^1 = a_0^1 + \rho \cdot (a_3^1 - a_0^1) = 1.717 + 0.618 \cdot (3.200 - 1.717) = 2.633$$

Die Intervallgröße des Grundintervalls  $D_0 = a_3^0 - a_0^0 = 2.4$  reduziert sich auf  $D_1 = a_3^1 - a_0^1 = 1.483$ . Die weiteren Intervallverkleinerungsschritte mit der jeweiligen Intervallgröße fasst die Tabelle 10.2 zusammen. In der Tabelle ändert sich jeweils nur ein Wert. In jedem Schritt wird entweder die obere oder die untere Intervallgrenze aufgegeben und ein innerer Punkt neu berechnet. Beim 7. Schritt ist die gewünschte Genauigkeit erreicht ( $D_7 = 0.083 < D_{min} = 0.1$ ). Das Ergebnis der Liniensuche lautet  $\eta_k = (a_3^7 - a_0^7)/2 = 2.159$  (exakte Schrittweite wäre  $\eta_k = 2.139$ ). Der in diesem Kapitel vorgestellte Liniensuchalgorithmus mit Intervallsuchphase und anschließender Intervallverkleinerung benötigt somit insgesamt 18 Funktionsauswertungen für das Minimierungsbeispiel. Für die Intervallsuche sind dabei 9 Auswertungen notwendig. Die Berechnung der inneren Punkte beim Grundintervall erfordert 2 Auswertungen und die nachfolgenden 7 Schritte der Intervallverkleinerung benötigen zusätzlich jeweils eine weitere Auswertung der Kostenfunktion.

Schritt $i$	0	1	2	3	4	5	6	7
$a_0^i$	0.800	1.717	1.717	1.717	1.933	2.067	2.067	2.118
$a_1^i$	1.717	2.283	2.067	1.933	2.067	2.150	2.118	2.150
$a_2^i$	2.283	2.633	2.283	2.067	2.150	2.201	2.150	2.169
$a_3^i$	3.200	3.200	2.633	2.283	2.283	2.283	2.201	2.201
$D_i$	2.400	1.483	0.917	0.567	0.350	0.216	0.134	0.083

**Tabelle 10.2:** Intervallverkleinerungsphase — In jedem Schritt ist nur eine Funktionsauswertung erforderlich, drei Punkte bleiben gleich. Im 7. Schritt ist die geforderte Genauigkeit  $D_{min} = 0.1$  erreicht.

### 10.2.2 Adaptives Liniensuchverfahren mit Lagrange-Interpolation

Es gibt eine große Auswahl an Liniensuchverfahren. Je nach Anwendungsfall sind die einzelnen Verfahren mehr oder weniger gut geeignet. Es besteht aber weniger das Problem, ein für die Optimierungsaufgabe unpassendes Verfahren auszusuchen, als vielmehr die Gefahr, die spezifischen Parameter des Liniensuchalgorithmus falsch einzustellen. Im Folgenden soll der *ALIS-Algorithmus (Adaptive Lagrange Interpolation Search)* vorgestellt werden, der ohne benutzerdefinierte Voreinstellungen auskommt [45].

Der ALIS-Algorithmus besteht nicht aus den beiden Phasen Intervallsuche und Intervallverkleinerung. Der Algorithmus startet sofort mit der Berechnung der Kostenfunktion an  $r+1$  Stützstellen  $\eta_j$  ( $j = 0 \dots r$ ) entlang der Suchrichtung  $s_k$ , um die Fehlerfläche mit einer *Lagrange-Interpolation* vom Grad  $r$  anzunähern [231, 45]:

$$L(\eta) = \sum_{j=0}^r L_j(\eta) \cdot E(\hat{\Theta}_k + \eta_j \cdot s_k)$$

$$\text{mit } L_j(\eta) = \prod_{h=0, h \neq j}^r \frac{\eta - \eta_h}{\eta_j - \eta_h} \quad (10.15)$$

Die zu berechnenden Punkte befinden sich alle in einem Intervall  $[0 \ \eta_{max}]$ , dessen Größe während der Optimierung an die Beschaffenheit der Fehlerfläche angepasst wird. Das Minimum der Liniensuche lässt sich mit geringem Rechenaufwand aus der Lagrange-Interpolation  $L(\eta)$  von Gleichung (10.15) bestimmen.

Beim ALIS-Algorithmus entscheidet die gefundene Position des Minimums innerhalb des Intervalls, ob die Fehlerflächenapproximation gültig ist oder nicht. Die Approximation ist ungültig, wenn sich das gefundene Minimum an der Intervallgrenze  $\eta_{max}$  befindet. Dies deutet darauf hin, dass das Minimum außerhalb des untersuchten Intervalls liegt und eine größere Schrittweite erforderlich ist. In diesem Fall vergrößert der ALIS-Algorithmus den Suchbereich so lange, bis das ermittelte Minimum in einem gültigen Bereich innerhalb des untersuchten Intervalls zu liegen kommt (Intervallvergrößerung). Außerdem ist die Approximation ungültig, falls die gefundene Schrittweite extrem klein ist im Vergleich

zur Intervallgröße, also  $\eta_k < \eta_{valid} \ll \eta_{max}$  gilt. In diesem Fall ist die verwendete Approximation — und mit ihr das berechnete Minimum — zu ungenau. Der ALIS-Algorithmus reduziert daraufhin den Suchbereich mit dem Ziel, eine genauere Approximation der Fehlerfläche entlang der Suchrichtung zu erreichen (Intervallverkleinerung). Durch die Intervallanpassung kann eine Liniensuche aus mehreren Iterationen bestehen. Die angepasste obere Intervallgrenze  $\eta_{max}$  wird nach jeder Liniensuche gespeichert, damit die Liniensuche beim nächsten Optimierungsschritt sofort mit der neu bestimmten Intervallgröße starten kann.

Das Grundintervall bei der ersten Liniensuche wird zu  $[0 \ \eta_{max}^0]$  mit  $\eta_{max}^0 = 0.5$  gesetzt, wobei der hochgestellte Index den aktuellen Iterationsschritt der Liniensuche angibt. Die Größe dieses anfänglichen Suchbereichs ist unwichtig und kann auch auf einen anderen Wert gesetzt werden, da während der Optimierung eine Adaption stattfindet<sup>6)</sup>. Als untere Gültigkeitsgrenze  $\eta_{valid}^i$  eignen sich 25 Prozent von der jeweiligen oberen Intervallgrenze  $\eta_{max}^i$ . Mit der Adaption der oberen Intervallgrenze während der Optimierung passt sich auch die untere Gültigkeitsgrenze fortwährend an. Die Approximation im Suchbereich mit dem berechneten Minimum  $\eta_k$  ist gültig, falls das Minimum im Bereich

$$\eta_{valid}^i \geq \eta_k < \eta_{max}^i \quad (10.16)$$

liegt. Bei Verwendung einer Polynominterpolation vom Grad  $r = 4$  lässt sich die Minimalstelle ausreichend genau ermitteln. Die dafür notwendigen fünf Stützstellen sind beim Grundintervall äquidistant im Suchbereich verteilt:  $\eta_0^0 = 0$ ,  $\eta_1^0 = 1/4 \cdot \eta_{max}^0$ ,  $\eta_2^0 = 2/4 \cdot \eta_{max}^0$ ,  $\eta_3^0 = 3/4 \cdot \eta_{max}^0$  und  $\eta_4^0 = \eta_{max}^0$ . Die erste Stützstelle entspricht dem Funktionswert des aktuellen Gewichtsvektors  $E(\hat{\Theta}_k)$  und ist bekannt. Somit sind für die Berechnung der ersten Lagrange-Interpolation  $L^0(\eta)$  vier Funktionsauswertungen erforderlich. Die Minimumssuche bei der Lagrange-Interpolation  $L^i(\eta)$

$$\eta_k = \min_{\eta} L^i(\eta) \quad (10.17)$$

gelingt ohne großen Rechenaufwand entweder durch Auswerten an einigen tausend Punkten oder analytisch mit Hilfe der Cardanischen Formeln [231]. Je nach gefundenem Minimum  $\eta_k$  müssen beim ALIS-Algorithmus die nun folgenden drei Fälle unterschieden werden:

**1. Fall: Die Approximation ist gültig, da Gleichung (10.16) erfüllt**  
In diesem Fall ist das Ergebnis der Liniensuche sofort verfügbar. Der ALIS-Algorithmus führt mit nur vier Kostenfunktionsauswertungen zum Ergebnis.

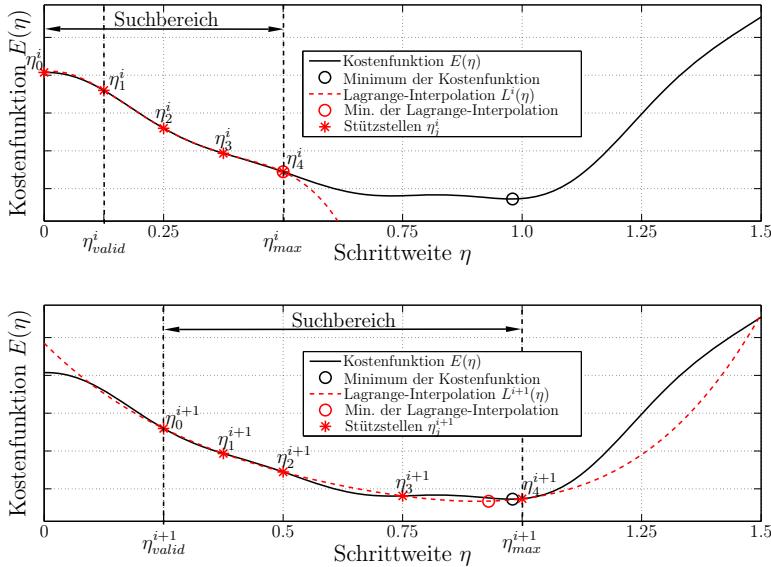
**2. Fall: Intervallvergrößerung, da  $\eta_k^i = \eta_{max}^i$  (Approximation ist ungültig)**

Falls sich das gefundene Minimum der Polynominterpolation an der Intervallgrenze  $\eta_{max}^i$  befindet, liegt das Minimum mit großer Wahrscheinlichkeit außerhalb des

---

<sup>6)</sup> Hinweis: Die Anfangsinitialisierung der Gewichte bei Modellen mit tanh-Funktionen beträgt meist  $\hat{\Theta}_0 \in [-0.5 \dots 0.5]^N$ , um Sättigungserscheinungen der Transferfunktionen zu vermeiden.

untersuchten Intervalls  $[\eta_0^i \dots \eta_{max}^i]$ . Deshalb verdoppelt der ALIS-Algorithmus — wie in Abbildung 10.13 dargestellt — die obere Intervallgrenze ( $\eta_{max}^{i+1} = 2 \cdot \eta_{max}^i$ ).



**Abb. 10.13:** Intervallvergrößerung beim ALIS-Algorithmus — Die schwarze Linie zeigt beispielhaft den Verlauf der Fehlerfläche entlang der Suchrichtung. Im oberen Bild liegt das Minimum an der Intervallgrenze  $\eta_{max}^i$ . Daraufhin verdoppelt der ALIS-Algorithmus die Intervallgrenze sowie die untere Gültigkeitsgrenze  $\eta_{valid}^i$ , wie im unteren Bild zu sehen. Die neue Lagrange-Interpolation  $L^{i+1}(\eta)$  nutzt die drei Stützpunkte  $\eta_0^i, \eta_3^i$  und  $\eta_4^i$  weiter und verwendet zusätzlich die beiden neu berechneten Punkte  $\eta_3^{i+1}$  und  $\eta_4^{i+1}$  zur Fehlerflächenapproximation.

Um mit möglichst geringem Rechenaufwand auszukommen, werden die drei oberen Punkte des alten Intervalls wieder verwendet:

$$\begin{aligned} E(\hat{\Theta}_k + \eta_0^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_0^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + \eta_1^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_3^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + \eta_2^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_4^i \cdot \underline{s}_k), \end{aligned} \quad (10.18)$$

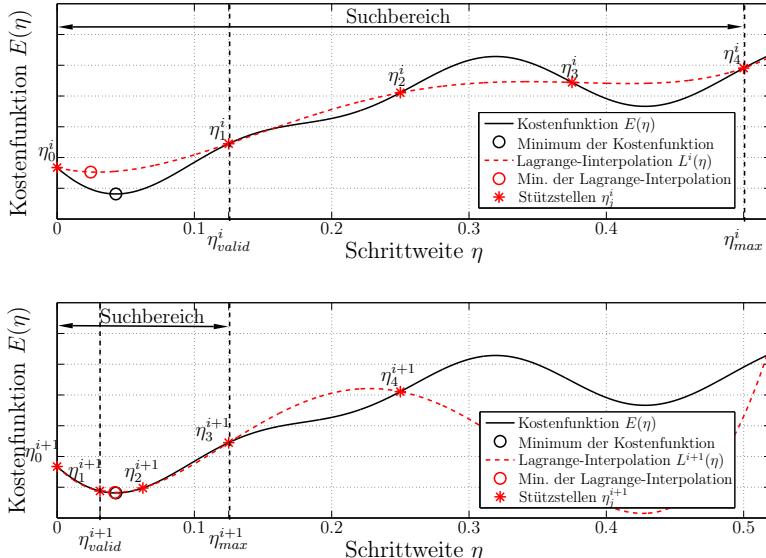
wobei diese Stützstellen im neuen Intervall bei  $\eta_0^{i+1} = 1/4 \cdot \eta_{max}^{i+1}$ ,  $\eta_1^{i+1} = 3/8 \cdot \eta_{max}^{i+1}$  und  $\eta_2^{i+1} = 1/2 \cdot \eta_{max}^{i+1}$  zu liegen kommen. Zwei Neuberechnungen sind an der neuen Intervallgrenze und zwischen der alten und neuen Intervallgrenze erforderlich:

$$\begin{aligned} \eta_4^{i+1} &= \eta_{max}^{i+1} & E(\hat{\Theta}_k + \eta_4^{i+1} \cdot \underline{s}_k) \\ \eta_3^{i+1} &= \frac{3}{4} \cdot \eta_{max}^{i+1} & E(\hat{\Theta}_k + \eta_3^{i+1} \cdot \underline{s}_k) \end{aligned} \quad (10.19)$$

Die Minimumssuche erfolgt nun mit der neuen Lagrange-Interpolation  $L^{i+1}(\eta)$  im vergrößerten Suchbereich  $[\eta_0^{i+1}, \eta_{max}^{i+1}]$ .

### 3. Fall: Intervallverkleinerung, da $\eta_k^i < \eta_{valid}^i$ (Approximation ist ungültig)

Führt das gefundene Minimum der Lagrange-Interpolation  $L^i(\eta)$  auf eine sehr kleine Schrittweite  $\eta_k^i < \eta_{valid}^i$ , so eignen sich die berechneten Stützwerte nicht, den Bereich um das Minimum ausreichend genau zu beschreiben. Wie in Abbildung 10.14 dargestellt, wird die obere Intervallgrenze geviertelt ( $\eta_{max}^{i+1} = 1/4 \cdot \eta_{max}^i$ )



**Abb. 10.14:** Intervallverkleinerung beim ALIS-Algorithmus — Die schwarze Linie zeigt beispielhaft den Verlauf der Fehlerfläche entlang der Suchrichtung. Im oberen Bild liegt das Minimum der Lagrange-Interpolation  $L^i(\eta)$  unterhalb der Gültigkeitsgrenze  $\eta_{valid}^i$ . Daraufhin viertelt der ALIS-Algorithmus die Intervallgrenze  $\eta_{max}^i$  sowie die untere Gültigkeitsgrenze  $\eta_{min}^i$ , wie im unteren Bild zu sehen. Die neue Lagrange-Interpolation  $L^{i+1}(\eta)$  nutzt die drei Stützpunkte  $\eta_0^i, \eta_1^i$  und  $\eta_2^i$  weiter und verwendet zusätzlich die beiden neu berechneten Punkte  $\eta_1^{i+1}$  und  $\eta_2^{i+1}$  zur Fehlerflächenapproximation.

und lediglich die unteren drei Punkte finden bei der neuen Approximation Verwendung:

$$\begin{aligned} E(\hat{\Theta}_k + \eta_0^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_0^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + \eta_3^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_1^i \cdot \underline{s}_k) \\ E(\hat{\Theta}_k + \eta_4^{i+1} \cdot \underline{s}_k) &= E(\hat{\Theta}_k + \eta_2^i \cdot \underline{s}_k) \end{aligned} \quad (10.20)$$

Diese drei Stützstellen befinden sich im neuen verkleinerten Intervall an den Stellen  $\eta_0^{i+1} = 0$ ,  $\eta_3^{i+1} = \eta_{max}^{i+1}$  und  $\eta_4^{i+1} = 2 \cdot \eta_{max}^{i+1}$ . Zur genaueren Beschreibung

des unteren Bereichs berechnet der ALIS-Algorithmus zwei weitere Punkte:

$$\begin{aligned}\eta_1^{i+1} &= \frac{1}{4} \cdot \eta_{max}^{i+1} & E(\hat{\Theta}_k + \eta_1^{i+1} \cdot \underline{s}_k) \\ \eta_2^{i+1} &= \frac{1}{2} \cdot \eta_{max}^{i+1} & E(\hat{\Theta}_k + \eta_2^{i+1} \cdot \underline{s}_k)\end{aligned}\quad (10.21)$$

Die Minimumssuche erfolgt nun im verkleinerten Suchbereich  $[\eta_0^{i+1} \quad \eta_{max}^{i+1}]$  mit der neuen (und um das Minimum genauerer) Lagrange-Interpolation  $L^{i+1}(\eta)$ .

### Zusammenfassung — ALIS-Algorithmus, Adaptive Liniensuche mit Lagrange Interpolation

1. Nur beim ersten Optimierungsschritt des Gesamtalgorithmus (also für  $k=0$ ) ausführen:

$$\eta_{max}^0 = 0.5 \quad \text{und} \quad \eta_{valid}^0 = \frac{1}{4} \cdot \eta_{max}^0$$

2. Berechnung der Stützpunkte des Grundintervalls  $[0 \quad \eta_{max}^0]$  ( $\eta_{max}^0$  für  $k \neq 0$  aus Speicher holen):

$$\begin{aligned}\eta_j^0 &= \frac{j}{4} \cdot \eta_{max}^0 \quad j = 0 \dots 4 \\ E(\hat{\Theta}_k + \eta_j^0 \cdot \underline{s}_k) &\quad j = 1 \dots 4 \\ \left( E(\hat{\Theta}_k + \eta_0^0 \cdot \underline{s}_k) = E(\hat{\Theta}_k) \text{ ist bekannt} \right)\end{aligned}$$

3. Approximation der Fehlerfläche entlang der Suchrichtung  $\underline{s}_k$  mittels Lagrange-Interpolation vierten Grades (Gleichung (10.15)):

$$\begin{aligned}L^i(\eta) &= \sum_{j=0}^4 L_j^i(\eta) \cdot E(\hat{\Theta}_k + \eta_j^i \cdot \underline{s}_k) \\ \text{mit } L_j^i(\eta) &= \prod_{h=0, h \neq j}^4 \frac{\eta - \eta_h^i}{\eta_j^i - \eta_h^i}\end{aligned}\quad (10.22)$$

4. Minimum der Lagrange-Interpolation  $L^i(\eta)$  im Suchbereich  $[\eta_0^i \quad \eta_{max}^i]$  finden (Gleichung (10.17)):

$$\eta_k = \min_{\eta} L^i(\eta)$$

5. Intervallanpassung:

$$\eta_k^i = \eta_{max}^i :$$

→ Intervallvergrößerung:

$$\eta_{max}^{i+1} = 2 \cdot \eta_{max}^i \quad \text{und} \quad \eta_{valid}^{i+1} = 2 \cdot \eta_{valid}^i$$

→ Alte Punkte weiterverwenden:

$$\eta_j^{i+1} = \frac{j+2}{8} \cdot \eta_{max}^{i+1} \quad j = 0 \dots 2$$

$$E(\hat{\Theta}_k + \eta_j^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + \eta_{j+2}^i \cdot \underline{s}_k) \quad j = 0 \dots 2$$

→ Neuberechnung von Stützpunkten:

$$\eta_j^{i+1} = \frac{j}{4} \cdot \eta_{max}^{i+1} \quad j = 3 \dots 4$$

$$E(\hat{\Theta}_k + \eta_j^{i+1} \cdot \underline{s}_k) \quad j = 3 \dots 4$$

→ weiter mit Schritt 3

$$\eta_k^i < \eta_{valid}^i :$$

→ Intervallverkleinerung:

$$\eta_{max}^{i+1} = \frac{1}{4} \cdot \eta_{max}^i \quad \text{und} \quad \eta_{valid}^{i+1} = \frac{1}{4} \cdot \eta_{valid}^i$$

→ Alte Punkte weiterverwenden:

$$\eta_0^{i+1} = \eta_0^i \quad \text{und} \quad E(\hat{\Theta}_k + \eta_0^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + \eta_0^i \cdot \underline{s}_k)$$

$$\eta_j^{i+1} = (j-2) \cdot \eta_{max}^{i+1} \quad j = 3 \dots 4$$

$$E(\hat{\Theta}_k + \eta_j^{i+1} \cdot \underline{s}_k) = E(\hat{\Theta}_k + \eta_{j-2}^i \cdot \underline{s}_k) \quad j = 3 \dots 4$$

→ Neuberechnung von Stützpunkten:

$$\eta_j^{i+1} = \frac{j}{4} \cdot \eta_{max}^{i+1} \quad j = 1 \dots 2$$

$$E(\hat{\Theta}_k + \eta_j^{i+1} \cdot \underline{s}_k) \quad j = 1 \dots 2$$

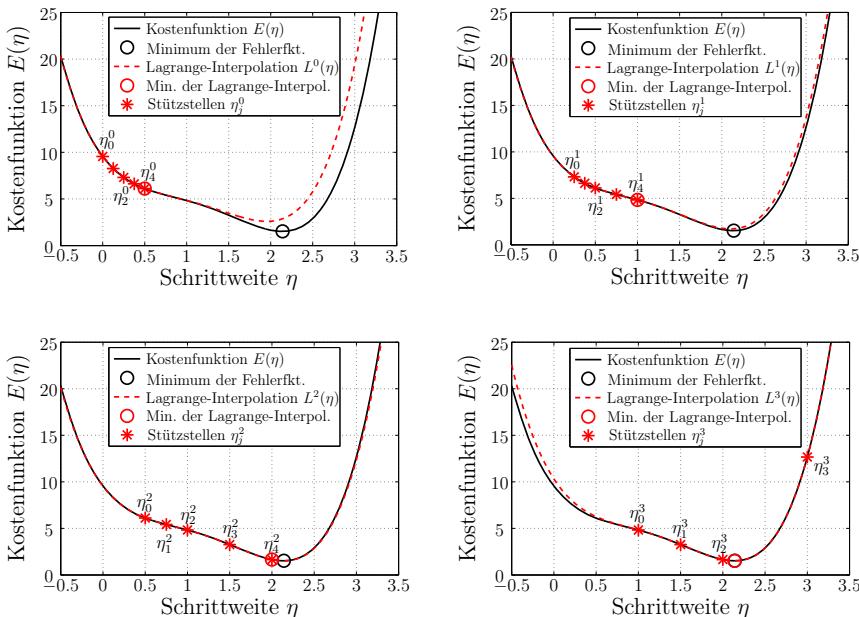
→ weiter mit Schritt 3

6. Iteration: Falls das Abbruchkriterium von Gleichung (10.16) mit  $\eta_{valid}^i \geq \eta_k < \eta_{max}^i$  noch nicht erfüllt ist, wiederhole die Punkte 3 bis 5. Ansonsten ist  $\eta_k$  die Lösung der Liniensuche ( $\eta_{max}^i$  und  $\eta_{valid}^i$  für die nächste Liniensuche zum Zeitpunkt  $k + 1$  speichern).

Die Verwendung des ALIS-Algorithmus wird im weiteren Verlauf mit dem Index ‘‘ALIS’’ gekennzeichnet. So verwendet zum Beispiel der BFGS-Optimierungsalgorithmus (siehe Kapitel 10.4.4.4)  $BFGS_{ALIS}$  den ALIS-Algorithmus zur Liniensuche.

### Beispiel — ALIS-Algorithmus, Adaptive Liniensuche mit Lagrange Interpolation

Die Liniensuche startet — wie bei der INT-Liniensuche — vom Punkt  $\hat{\underline{\theta}}_k = [-1.25 \ -1.00]^T$  in die normierte negative Gradientenrichtung  $\underline{s}_k = [0.991 \ 0.132]^T$  (siehe Gleichung (10.14)). Die schwarze Linie in Abbildung 10.15 stellt jeweils den Fehlerverlauf entlang der Suchrichtung dar. Für den ALIS-Algorithmus sind keine benutzerdefinierten Daten erforderlich. Abbildung 10.15 zeigt die berechneten



**Abb. 10.15:** Lagrange-Interpolationen des ALIS-Algorithmus für das Liniensuchbeispiel — Die schwarze Linie zeigt jeweils den Verlauf der Beispielderflächen entlang des negativen Gradienten. Im Grundintervall und im 1. und 2. Schritt liegt das Minimum der Lagrange-Interpolation an der Intervallgrenze, was zu einer Verdopplung der Intervallgrenze führt. Erst im 3. Schritt ist die Forderung  $\eta_{valid}^3 \geq \eta_k < \eta_{max}^3$  erfüllt und die Liniensuche ist gültig.

Lagrange-Interpolationen der Fehlerfläche während des Liniensuchvorgangs. Die Grafik links oben enthält die Lagrange-Interpolation  $L^0(\eta)$  des Grundintervalls mit den dazugehörigen fünf Stützstellen. Da die Fehlerfläche analytisch vorliegt, lassen sich die Funktionswerte dieser Stützstellen recht einfach berechnen zu:

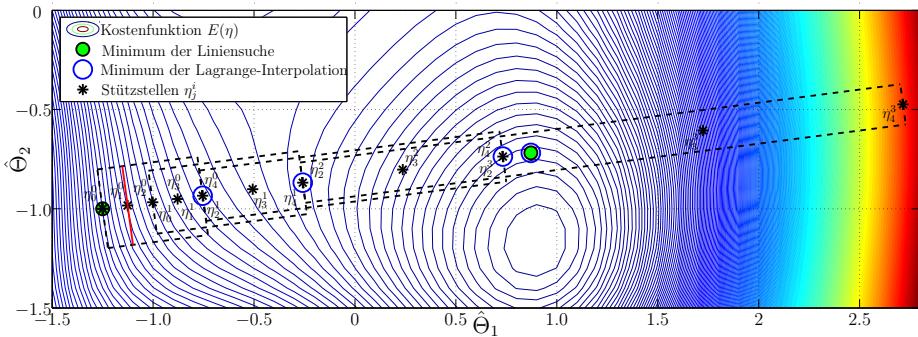
$$\begin{aligned}
 \eta_0^0 &= 0 & E(\hat{\Theta}_k + 0 \cdot \underline{s}_k) &= E([-1.250 - 1.000]^T) = 9.554 \\
 \eta_1^0 &= 0.125 & E(\hat{\Theta}_k + 0.125 \cdot \underline{s}_k) &= E([-1.126 - 0.984]^T) = 8.259 \\
 \eta_2^0 &= 0.250 & E(\hat{\Theta}_k + 0.250 \cdot \underline{s}_k) &= E([-1.002 - 0.967]^T) = 7.311 \\
 \eta_3^0 &= 0.375 & E(\hat{\Theta}_k + 0.375 \cdot \underline{s}_k) &= E([-0.878 - 0.951]^T) = 6.622 \\
 \eta_4^0 &= 0.500 & E(\hat{\Theta}_k + 0.500 \cdot \underline{s}_k) &= E([-0.755 - 0.934]^T) = 6.117 \quad (10.23)
 \end{aligned}$$

Das Minimum der Lagrange-Interpolation  $L^0(\eta)$  liegt an der Intervallgrenze  $\eta_{max}^0$ . Der ALIS-Algorithmus verdoppelt daraufhin die Intervallgrenze auf  $\eta_{max}^1 = 1$  und findet eine neue Approximation  $L^1(\eta)$  für den vergrößerten Suchbereich  $[\eta_0^1 \ \eta_{max}^1] = [0.250 \ 1]$ . Die drei oberen Punkte des Intervalls werden weiterhin verwendet:  $\eta_0^1 = \eta_2^0 = 0.250$ ,  $\eta_1^1 = \eta_3^0 = 0.375$  und  $\eta_2^1 = \eta_4^0 = 0.500$ . Zwei neue Stützstellen entstehen bei  $\eta_3^1 = 3/4 \cdot \eta_{max}^1 = 0.750$  und bei  $\eta_4^1 = \eta_{max}^1 = 1$ . Die so entstehende zweite Lagrange-Interpolation von Abbildung 10.15 kann die Fehlerfläche im Bereich des Minimums (dargestellt durch einen schwarzen Kreis) mit dem nun vergrößerten Intervall recht gut approximieren. Trotzdem sind noch zwei weitere Intervallvergrößerungsschritte erforderlich — wie in Abbildung 10.15 und in Tabelle 10.3 dargestellt — bis die Forderung  $\eta_{valid}^3 \geq \eta_k < \eta_{max}^3$  erfüllt ist ( $\eta_{valid}^3 = 0.25 \cdot \eta_{max}^3 = 1$ ). Das gefundene Minimum der Lagrange-Interpolation  $\eta_k = \min_{\eta} L^3(\eta) = 2.139$  stimmt exakt mit dem Minimum der Fehlerfläche entlang der Suchrichtung überein. Der gesamte Verlauf der ALIS-Liniensuche ist in Ab-

Schritt $i$	0	1	2	3
$\eta_0^i$	0.000	0.250	0.500	1.000
$\eta_1^i$	0.125	0.375	0.750	1.500
$\eta_2^i$	0.250	0.500	1.000	2.000
$\eta_3^i$	0.375	0.750	1.500	3.000
$\eta_4^i$	0.500	1.000	2.000	4.000
$\min_{\eta} L^i(\eta)$	$\eta_{max}$	$\eta_{max}$	$\eta_{max}$	2.139

**Tabelle 10.3:** Stützstellen der Lagrange-Interpolation — Bei jeder Intervallanpassung sind zwei neue Stützstellen erforderlich, drei Punkte bleiben gleich. Beim 3. Schritt gilt  $\eta_{valid}^3 \geq \eta_k < \eta_{max}^3$  und die Liniensuche ist erfolgreich.

bildung 10.16 zusammengefasst. Die Minimumssuche des Beispiels erfordert vier Schritte mit dem ALIS-Algorithmus und insgesamt 11 Funktionsauswertungen. Für die Lagrange-Interpolation  $L^0(\eta)$  des Grundintervalls sind 5 Funktionsauswertungen notwendig. Die drei Intervallvergrößerungsschritte benötigen jeweils zwei zusätzliche Auswertungen der Kostenfunktion. Verglichen mit dem Linien-suchalgorithmus von Kapitel 10.2.1 kommt der ALIS-Algorithmus mit weniger Funktionsauswertungen aus und arbeitet außerdem noch viel genauer. Die Intervallanpassung des ALIS-Algorithmus erweist sich besonders bei den komplexen Fehlerflächen von Neuronalen Netzen als vorteilhaft, wie die Identifikationen in Kapitel 10.6 zeigen.



**Abb. 10.16:** Beispiel einer Liniensuche mit ALIS — Die Liniensuche startet vom Punkt  $\hat{\Theta}_k = [-1.25 \ - 1.00]^T$  in die negative Gradientenrichtung.

### 10.3 Optimierungsverfahren 1. Ordnung

Aufgabe eines Optimierungsverfahrens ist, die Parameter des Modells so zu verändern, dass der Wert der Kostenfunktion  $E(\hat{\Theta})$  abnimmt. Bei den deterministischen Optimierungsverfahren startet die Minimumsuche von einem einzigen Punkt auf der Fehlerfläche. Durch die Parameterveränderungen wandert der aktuelle Punkt auf der Fehlerfläche hinab bis zu einem stationären Minimum. Besteht sich die Optimierung zum  $k$ -ten Optimierungsschritt an der Stelle  $\hat{\Theta}_k$ , so entsteht mit diesen Parametern ein Modell, welches den Fehler  $E(\hat{\Theta}_k)$  zur Folge hat. Mit Hilfe der Taylorapproximation der Fehlerfläche ist es möglich, geeignete Algorithmen zur Minimumsuche auf der Fehlerfläche zu entwickeln. Während im Folgenden lediglich die Taylorapproximation 1. Ordnung für die Herleitung des Gradientenabstiegsverfahrens verwendet wird, nutzt das Kapitel 10.4 die exaktere Taylorapproximation 2. Ordnung aus.

Als Entwicklungspunkt der Taylorapproximation dient die aktuelle Position auf der Fehlerfläche  $\hat{\Theta}_k$  zum  $k$ -ten Optimierungsschritt:

$$E^*(\hat{\Theta}) = E(\hat{\Theta}_k) + g^T(\hat{\Theta}_k) \cdot (\hat{\Theta} - \hat{\Theta}_k), \quad (10.24)$$

wobei

$$\underline{g}(\hat{\Theta}_k) = \frac{dE(\hat{\Theta}_k)}{d\hat{\Theta}_k} \quad (10.25)$$

den Gradienten der Fehlerfläche am Fehlerflächenpunkt  $\hat{\Theta}_k$  bezeichnet.

Ziel des Optimierungsalgorithmus ist es nun, eine Bedingung für den nächsten Punkt  $\hat{\Theta}_{k+1}$  zu finden, so dass der Kostenfunktionswert einen kleineren Wert annimmt:

$$E(\hat{\Theta}_{k+1}) < E(\hat{\Theta}_k) \quad (10.26)$$

Vom aktuellen Punkt  $\hat{\underline{\Theta}}_k$  kommt man durch den Optimierungsschritt  $\Delta\hat{\underline{\Theta}}_k$  zum nächsten Punkt  $\hat{\underline{\Theta}}_{k+1}$  auf der Fehlerfläche:

$$\hat{\underline{\Theta}}_{k+1} = \hat{\underline{\Theta}}_k + \Delta\hat{\underline{\Theta}}_k \quad (10.27)$$

Der Funktionswert an dieser Stelle lässt sich näherungsweise durch die Taylorapproximation 1. Ordnung berechnen durch:

$$E(\hat{\underline{\Theta}}_{k+1}) \approx E^*(\hat{\underline{\Theta}}_{k+1}) = E(\hat{\underline{\Theta}}_k) + \underline{g}^T(\hat{\underline{\Theta}}_k) \cdot \Delta\hat{\underline{\Theta}}_k \quad (10.28)$$

Damit die Abstiegsbedingung von Gleichung (10.26) erfüllt ist, muss der zweite Term auf der rechten Seite von Gleichung (10.28) negativ sein:

$$\underline{g}^T(\hat{\underline{\Theta}}_k) \cdot \Delta\hat{\underline{\Theta}}_k < 0 \quad (10.29)$$

Ein einfaches Optimierungsverfahren entsteht, wenn man beim  $k$ -ten Optimierungsschritt eine positive *Schrittweite*  $\eta_k$  in eine geeignete *Suchrichtung*  $\underline{s}_k$  voranschreitet:

$$\Delta\hat{\underline{\Theta}}_k = \eta_k \cdot \underline{s}_k \quad (10.30)$$

Mit der Forderung von Gleichung (10.29) gilt für die Suchrichtung  $\underline{s}_k$

$$\underline{g}^T(\hat{\underline{\Theta}}_k) \cdot \underline{s}_k < 0 \quad (10.31)$$

Jede Suchrichtung  $\underline{s}_k$ , welche die Bedingung von Gleichung (10.31) erfüllt, ist eine sogenannte *Abstiegsrichtung*. Bei ausreichend kleiner Schrittweite  $\eta_k$  ist gewährleistet, dass mit einem Optimierungsschritt in eine Abstiegsrichtung nach Gleichung (10.30) ein Fehlerflächenpunkt mit geringerem Funktionswert gefunden wird. Der Abstieg ist am größten, wenn das Skalarprodukt aus Gleichung (10.31) den negativsten Wert annimmt. Dieser Fall tritt bei einer Suche in die negative Gradientenrichtung ein:

$$\underline{s}_k = -\underline{g}(\hat{\underline{\Theta}}_k) \quad (10.32)$$

Diese Überlegungen führen zu einem der bekanntesten Optimierungsverfahren für nichtlineare Probleme, dem *Gradientenabstiegsverfahren*, welches an dieser Stelle kurz zusammengefasst werden soll:

### Zusammenfassung — Gradientenabstieg

1. Festlegen einer kleinen konstanten Lernschrittweite  $\eta$  (z.B.  $\eta = 0.01$ ).
2. Anfangsinitialisierung: Wahl eines geeigneten Startpunktes  $\hat{\underline{\Theta}}_0$  auf der Fehlerfläche<sup>7)</sup>.

---

<sup>7)</sup> Hinweis: Bei der Anfangsinitialisierung darf der Betrag der Gewichte nicht zu groß sein, damit sich die nichtlinearen Transferfunktionen nicht in Sättigung befinden.

3. Gradientenberechnung: Berechnung der Ableitung der Kostenfunktion nach den Parametern  $\underline{g}(\hat{\Theta}_k)$  am aktuellen Punkt  $\hat{\Theta}_k$ .
4. Neuen Punkt  $\hat{\Theta}_{k+1}$  in Richtung des negativen Gradienten ermitteln:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \eta \cdot \underline{g}(\hat{\Theta}_k) \quad (10.33)$$

5. Iteration: Falls das Abbruchkriterium (Wert der Kostenfunktion erreicht oder maximale Identifikationszeit) noch nicht erfüllt ist, wiederhole die Punkte 3 und 4.

Das Gradientenabstiegsverfahren ist numerisch ineffizient. Es verfügt lediglich über eine lineare Konvergenz [213, 187, 18], siehe Gleichung (10.98). Eine ausführliche Konvergenzanalyse und weiterführende theoretische Betrachtungen sind beispielsweise in [187, 18] zu finden.

### Beispiel — Optimierung mit dem Gradientenabstieg

Abschließend soll die Optimierung mit dem Gradientenabstiegsverfahren anhand der einfachen nichtlinearen Funktion  $E(\hat{\Theta}) = 2 \cdot \hat{\Theta}_1^4 + \hat{\Theta}_2^4 - 2 \cdot \hat{\Theta}_1^2 - 2 \cdot \hat{\Theta}_2^2 + 4 \cdot \sin(\hat{\Theta}_1 \cdot \hat{\Theta}_2) + 5$  gezeigt werden. Die feste Lernrate  $\eta$  beträgt zunächst 0.01. Die Optimierung startet beim Punkt  $\hat{\Theta}_0 = [-1.25 \quad -1.00]^T$ . Der Gradient an dieser Stelle berechnet sich zu

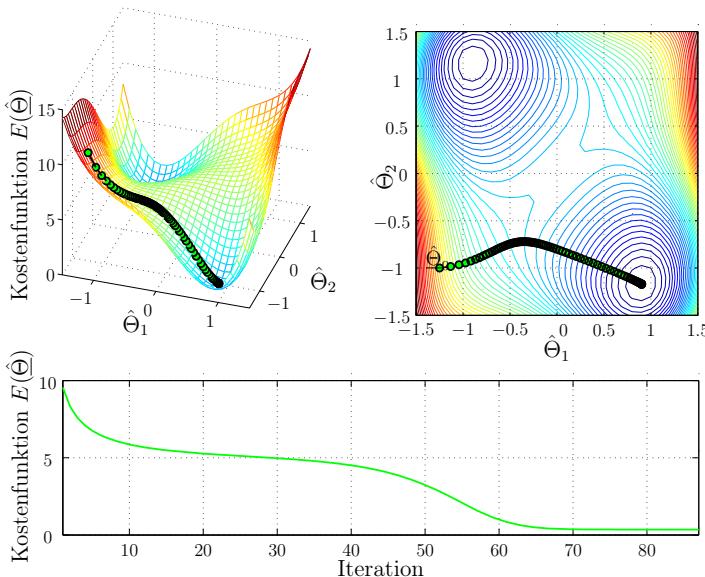
$$\underline{g}(\hat{\Theta}_0) = \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix}.$$

Damit lautet der erste Optimierungsschritt entsprechend Gleichung (10.33):

$$\begin{aligned} \hat{\Theta}_1 &= \hat{\Theta}_0 - \eta \cdot \underline{g}(\hat{\Theta}_0) \\ &= \begin{bmatrix} -1.25 \\ -1.00 \end{bmatrix} - 0.01 \cdot \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix} = \begin{bmatrix} -1.13 \\ -0.98 \end{bmatrix} \end{aligned}$$

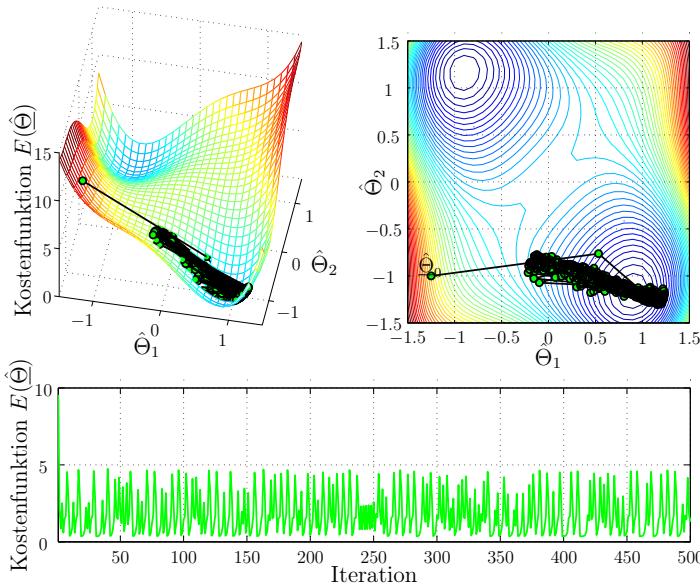
Analog zu dieser Berechnung folgen weitere Optimierungsschritte, bis die Abbruchbedingung<sup>8)</sup> erfüllt ist. Zeichnet man die berechneten Punkte in das Höhenlinienbild von Abbildung 10.17 ein, so entsteht eine Trajektorie. Da der Gradient in die Richtung des steilsten Anstieges zeigt, steht er senkrecht auf einer an diesem Punkt verlaufenden Höhenlinie. Erwartungsgemäß verläuft die Trajektorie beim Gradientenabstiegsverfahren mit klein gewählter Schrittweite immer senkrecht zu den Höhenlinien. Wie in Abbildung 10.17 zu sehen, funktioniert die Optimierung mit dem Gradientenabstieg bei einer Lernschrittweite von  $\eta = 0.01$  recht gut. Zu Beginn ist der Gradient — und damit auch der Lernfortschritt

<sup>8)</sup> Hinweis: Das hier betrachtete Fehlerflächenbeispiel hat zwei globale Minima mit einem Funktionswert von jeweils  $E(\hat{\Theta}^*) = 0.3524$ . Dieser Wert dient als Abbruchkriterium für die Optimierung.



**Abb. 10.17:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren und einer Lernschrittweite von  $\eta = 0.01$  — Bei kleinen Gradienten ist der Lernfortschritt gering. Nach 87 Iterationen ist das Minimum erreicht.

— am größten. Der relativ kleine Gradient im Bereich zwischen der 10. und 40. Iteration führt zu einer geringen Abnahme der Kostenfunktion. In der Nähe des Minimums bei  $\hat{\Theta} = [0.9 \ -1.17]^T$  ist der Gradient annähernd Null und der Algorithmus konvergiert nur noch sehr langsam. Eine größere Lernschrittweite verursacht größere Optimierungsschritte, nicht aber unbedingt eine schnellere Konvergenz der Parameter, wie in Abbildung 10.18 veranschaulicht. Dieser Optimierungsverlauf entsteht bei einer Lernschrittweite von  $\eta = 0.15$ . Die Lernschrittweite ist für das Fehlerflächenbeispiel zu groß gewählt, und es kommt zu oszillierenden Bewegungen um das Minimum. Bei einer weiteren geringfügigen Erhöhung der Lernschrittweite wird die Parameteroptimierung sogar instabil. In Abbildung 10.19 ist ein Optimierungsversuch mit einer Lernschrittweite von  $\eta = 0.2$  zu sehen. Der Wert der Kostenfunktion nimmt nur beim ersten Optimierungsschritt ab. Bei der 5. Iteration steigt die Kostenfunktion sehr stark an. In der Praxis ist es manchmal sehr schwierig, eine passende Lernschrittweite  $\eta$  zu finden. Bei dem gezeigten Beispiel gelingt eine erfolgreiche Optimierung mit  $\eta = 0.01$ . Im Falle der Identifikation mit Neuronalen Netzen kann sich das Verhalten der Fehlerfläche an unterschiedlichen Punkten sehr verändern. Dies führt häufig dazu, dass der einfache Gradientenabstieg mit fester Lernschrittweite keine oder nur sehr schlechte Ergebnisse liefert. Die folgenden Ausführungen



**Abb. 10.18:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren und einer Lernschrittweite von  $\eta = 0.15$  — Die Lernschrittweite ist für das Beispiel zu groß gewählt. Das Minimum wird in den ersten 500 Schritten nicht erreicht. Es entsteht eine oszillierende Bewegung.

beschreiben einige Modifikationen des einfachen Gradientenabstiegs, die eine erfolgreichere Optimierung ermöglichen.

### 10.3.1 Gradientenabstieg mit Momentumterm

Wie im Optimierungsbeispiel von Abbildung 10.17 gesehen, verläuft die Optimierung bei kleinen Gradienten recht langsam. Durch das Einführen eines sogenannten *Momentumterms* lässt sich der Lernfortschritt in diesen Bereichen deutlich verbessern [179, 18]. Der Momentumterm berücksichtigt die vorangegangene Gewichtsänderung  $\Delta\hat{\Theta}_{k-1}$ . Diese wird einfach zum bekannten Gradientenabstieg von Gleichung (10.33) hinzugefügt:

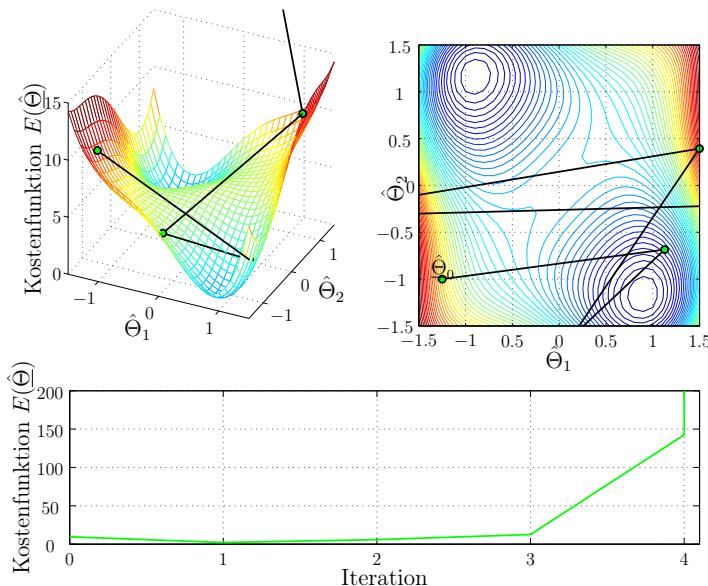
$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \eta \cdot \underline{g}(\hat{\Theta}_k) + \alpha \cdot \Delta\hat{\Theta}_{k-1} \quad (10.34)$$

mit  $\Delta\hat{\Theta}_{k-1} = \hat{\Theta}_k - \hat{\Theta}_{k-1}$  (vergleiche Gleichung (10.27)) und  $0 \leq \alpha < 1$ . Der Optimierungsverlauf beim Gradientenabstieg von Abbildung 10.17 ähnelt einer Kugel, welche durch eine Gravitationskraft ins Minimum rollt. Bei einem großen Gradienten bewegt sich die Kugel schnell, bei einem kleinen Gradienten langsam. Der Momentumterm von Gleichung (10.34) wirkt bei dieser Modellvorstellung

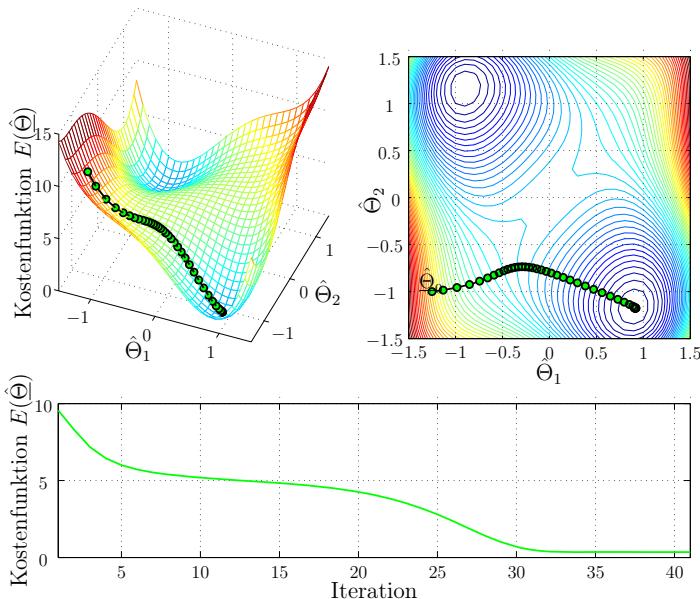
wie eine Art Trägheit der Kugel, die hilft, flache Fehlerflächenstellen schneller zu überwinden.

### Beispiel — Optimierung mit Gradientenabstieg und Momentumterm

Die Auswirkung eines Momentumterms von  $\alpha = 0.5$  auf die Optimierung des Fehlerflächenbeispiels zeigt Abbildung 10.20. Die Simulation erfolgt mit einer Lernschrittweite von  $\eta = 0.01$ , so dass ein direkter Vergleich mit den Ergebnissen aus Abbildung 10.17 möglich ist. Die Addition vorangegangener Gewichtsänderungen führt, wie erwartet, zu einer Beschleunigung in den Bereichen mit kleinen Gradienten. Nach 41 Iterationen ist das Minimum bei  $\hat{\Theta} = [0.9 \ -1.17]^T$  erreicht. Verglichen mit dem Gradientenabstieg ohne Momentumterm hat sich die Anzahl der notwendigen Optimierungsschritte mehr als halbiert. Jedoch muss der Anwender beim Gradientenabstieg mit Momentumterm einen zusätzlichen Parameter definieren. Ein zu groß gewählter Momentumterm wirkt sich wieder negativ auf die Parameterkonvergenz aus. Auch hier eignet sich wieder die Modellvorstellung einer hinabrollenden Kugel mit Trägheit (eine Kugel mit großer Trägheit braucht lange, bis sie zur Ruhe kommt). Abbildung 10.21 stellt den Op-



**Abb. 10.19:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren und einer Lernschrittweite von  $\eta = 0.2$  — Nur beim ersten Optimierungsschritt sinkt der Fehler. Danach steigt der Wert der Kostenfunktion  $E(\hat{\Theta})$  immer schneller an, die Optimierung wird instabil.



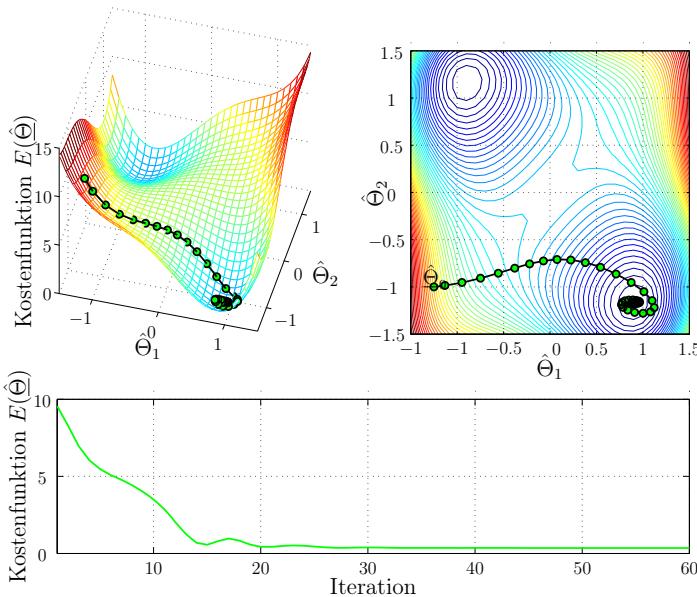
**Abb. 10.20:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren bei einer Lernschrittweite von  $\eta = 0.01$  und einem Momentumterm von  $\alpha = 0.5$  — Der Momentumterm führt zu schnellerer Konvergenz in Fehlerflächenbereichen mit kleinen Gradienten. Nach 41 Iterationen ist das Minimum erreicht.

timierungsverlauf bei einem Momentumterm von  $\alpha = 0.8$  dar. Der Algorithmus umkreist das Minimum, bevor er es nach 60 Iterationen erreicht.

### 10.3.2 Gradientenabstieg mit variabler Lernschrittweite

Die Simulationen haben gezeigt, welche Probleme beim Gradientenabstiegsverfahren auftreten, wenn die Lernschrittweite für die Optimierungsaufgabe falsch eingestellt ist. Während eine zu kleine Lernschrittweite zu einer sehr langsamem Parameterkonvergenz führt, kann es bei zu großen Werten zu oszillierendem Verhalten oder sogar zu Instabilität kommen. Aus diesem Grund wäre es wünschenswert, wenn sich die Lernschrittweite der Optimierungsaufgabe anpasst.

Ein einfacher Algorithmus zur Anpassung der Lernschrittweite lässt sich wie folgt formulieren: Falls die Kostenfunktion bei einem Optimierungsschritt ansteigt, ist die Lernschrittweite des Gradientenabstiegs zu groß und kann reduziert werden (Multiplikation der Lernschrittweite mit einem Faktor  $0 < \rho < 1$ ). Nimmt der Wert der Kostenfunktion ab, so lässt sich durch Vergrößern der Lernschrittweite eine schnellere Konvergenz erreichen (Multiplikation der Lernschrittweite mit einem Faktor  $\kappa > 1$ ) [79].



**Abb. 10.21:** Optimierungsbeispiel mit dem Gradientenabstiegsverfahren bei einer Lernschrittweite von  $\eta = 0.01$  und einem Momentumterm von  $\alpha = 0.8$  — Der Momentumterm ist für das Beispiel zu groß gewählt. Es entsteht eine kreisende Bewegung um das Minimum. Erst nach 60 Iterationen ist das Minimum erreicht.

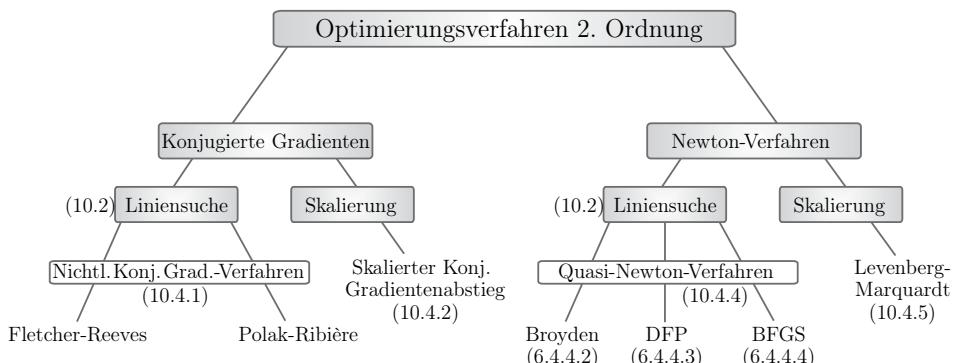
In der Literatur findet man viele weitere Gradientenabstiegsverfahren mit variabler Lernschrittweite: Beim delta-bar-delta-Algorithmus [116] hat beispielsweise jedes einzustellende Gewicht eine eigene Lernschrittweite. Erfährt ein Gewicht über mehrere Optimierungsschritte nur Gewichtsänderungen in eine Richtung, so erfolgt eine Vergrößerung der entsprechenden Lernschrittweite. Alterniert die Gewichtsänderung, so wird die zu dem Gewicht gehörende Lernschrittweite reduziert. Der SuperSAB-Algorithmus von Tollenaere [225] arbeitet ähnlich wie der delta-bar-delta-Algorithmus. Der Algorithmus verfügt jedoch über komplexere Einstellregeln für die Lernschrittweiten. Fahlman [47] geht beim sogenannten Quickprop-Algorithmus einen anderen Weg. Er nimmt an, dass die Fehlerfläche für jeden Parameter durch eine quadratische Funktion darstellbar ist. Die quadratische Approximation und die Optimierung erfolgt dabei für jeden Parameter unabhängig. Ein großer Nachteil dieser Algorithmen sind die vielen benutzerdefinierten Daten, welche zum Teil sehr stark die Effizienz der Optimierung beeinflussen. So kommt es vor, dass Optimierungsaufgaben, welche mit einfacherem Gradientenanstieg (nur ein Parameter einstellbar) lösbar wären, nicht mehr gelöst werden können [79].

Die beste Lernschrittweite kann durch eine Liniensuche für jeden Optimierungsschritt ermitteln werden. Wie in Kapitel 10.2 gesehen, ist dazu aber ein

relativ großer Rechenaufwand erforderlich, der beim Gradientenabstieg nicht zu rechtfertigen ist. Es lässt sich zeigen, dass beim Gradientenabstieg mit Liniensuche in der Nähe eines Minimums (hier ist auch eine nichtlineare Fehlerfläche annähernd quadratisch) aufeinander folgende Gradienten immer senkrecht zueinander stehen. Dies führt trotz Liniensuche zu einer deutlich langsameren Konvergenz als bei den Verfahren zweiter Ordnung, die im Folgenden näher behandelt werden.

## 10.4 Optimierungsverfahren 2. Ordnung

Die Verfahren 2. Ordnung machen direkt oder indirekt von der 2. Ableitung der Fehlerfläche Gebrauch. Durch die Verwendung der Taylorapproximation 2. Ordnung verfügen diese Methoden über eine genauere Beschreibung der Fehlerfläche als die Verfahren 1. Ordnung, was mit einer deutlich schnelleren Parameterkonvergenz einhergeht. Abbildung 10.22 gibt einen Überblick über die gängigsten Verfahren und über den Aufbau des Kapitels 10.4. Die Methoden 2. Ordnung untergliedern sich grob in Konjugierte Gradientenverfahren und in Newton-Verfahren. Damit die Algorithmen stabil arbeiten, gibt es die beiden unterschiedlichen Ansätze der Liniensuche und der Skalierung. Bei den Verfahren der Konjugierten Gradienten führt die Stabilisierung mit Liniensuche zu den in Kapitel 10.4.1 eingeführten Nichtlinearen Konjugierten Gradientenverfahren. Das entsprechende Verfahren mit Skalierungsalgorithmus bezeichnet man als Skalierten Konjugierten Gradientenabstieg und wird im Kapitel 10.4.2 besprochen. Während die in Kapitel 10.4.4 vorgestellten Quasi-Newton-Verfahren aufgrund der durchgeföhrten Liniensuche stabil arbeiten, nutzt der Levenberg-Marquardt-Algorithmus von Kapitel 10.4.5 wieder eine Skalierung.



**Abb. 10.22:** Übersicht der deterministischen Optimierungsverfahren 2. Ordnung — Die Angaben in Klammern geben das jeweilige Kapitel an.

### 10.4.1 Das Nichtlineare Konjugierte Gradientenverfahren

Der Rechenaufwand der *Methode der Nichtlinearen Konjugierten Gradienten* (NKG) ist nur geringfügig größer als der beim einfachen Gradientenabstieg von Kapitel 10.3. Der NKG-Algorithmus erfordert keine Matrixspeicherung und verwendet lediglich die erste Ableitung, erreicht aber trotzdem eine viel schnellere Parameterkonvergenz als die Verfahren 1. Ordnung. Das ursprünglich von Hestenes und Stiefel [85] vorgeschlagene *Konjugierte Gradientenverfahren* (KG) ist eine iterative Methode zum Lösen linearer Gleichungssysteme<sup>9)</sup>. Der erste NKG-Algorithmus wurde von Fletcher und Reeves in den 1960er Jahren eingeführt [55]. Es war eines der ersten Verfahren zur Lösung von großen nichtlinearen Optimierungsproblemen [174]. In den Folgejahren kamen viele Varianten zu diesem ursprünglichen Verfahren hinzu, die teilweise bessere Eigenschaften im praktischen Einsatz zeigen. Die Herleitung des NKG-Algorithmus ist nicht schwierig, aber sehr aufwändig. Eine für den Ingenieur verständliche Herleitung findet sich in [214][44]. Diese dort durchgeführten elementaren Überlegungen verdeutlichen, weshalb der NKG-Algorithmus mit der Berechnung der ersten Ableitung auskommt, obwohl er zu den Optimierungsverfahren 2. Ordnung zählt. Der folgende Algorithmus fasst die Rechenschritte beim Nichtlinearen Konjugierten Gradientenverfahren zusammen [214][44]:

#### Zusammenfassung — Nichtlinearer Konjugierter Gradientenabstieg

1. Anfangsinitialisierung: Wahl eines geeigneten Startpunktes  $\hat{\Theta}_0$  auf der Fehlerfläche
2. Suchrichtung für den ersten Optimierungsschritt bestimmen:

$$\underline{s}_0 = -\underline{g}(\hat{\Theta}_0)$$

3. Neue Schrittweite berechnen (Liniensuche nach Kapitel 10.2):

$$\eta_k = \min_{\eta} E(\hat{\Theta}_{k+1}) = \min_{\eta} E(\hat{\Theta}_k + \eta \cdot \underline{s}_k)$$

4. Optimierungsschritt ausführen:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \eta_k \cdot \underline{s}_k$$

---

<sup>9)</sup> Hinweis: Das Auffinden des Minimums einer quadratischen Funktion ist das gleiche mathematische Problem wie das Lösen eines linearen Gleichungssystems. Das KG-Verfahren findet in  $N$  Schritten das Minimum einer quadratischen Funktion.

5. Gradient an der neuen Stelle ermitteln:

$$\underline{g}(\hat{\Theta}_{k+1})$$

6. Mit dem Gram-Schmidt-Konjugationsverfahren eine neue konjugierte Suchrichtung berechnen [214][44]:

$$\beta_{k+1} = -\frac{\underline{g}(\hat{\Theta}_{k+1})^T \cdot \underline{g}(\hat{\Theta}_{k+1})}{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)} \quad (10.35)$$

$$\underline{s}_{k+1} = -\underline{g}(\hat{\Theta}_{k+1}) - \beta_{k+1} \cdot \underline{s}_k$$

Der Faktor  $\beta_{k+1}$  bezeichnet dabei den Konjugationskoeffizienten.

7. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 3 bis 6.

Da der NKG-Algorithmus nicht in das Minimum der Fehlerfläche, sondern in das globale Minimum der genäherten quadratischen Form konvergiert, ist es erforderlich, den Algorithmus nach  $N$  Iterationen neu zu starten (neu starten heißt, man berechnet wieder die Suchrichtung mit Hilfe einer Gradientenberechnung, führt also wieder den 2. Schritt des NKG-Algorithmus aus). Ein Neustart bedeutet, dass zu einer neuen genäherten quadratischen Form übergegangen wird.

Die Bestimmung der Konjugationskoeffizienten  $\beta_{k+1}$  im 6. Schritt des NKG-Algorithmus entspricht der von Fletcher und Reeves [55] vorgeschlagenen Vorgehensweise. Die Literatur kennt noch weitere Berechnungsvorschriften für  $\beta_{k+1}$  [192, 183, 169, 85], welche für quadratische Fehlerflächen exakt die gleichen Berechnungen ausführen, bei der Optimierung nichtlinearer Fehlerflächen jedoch unterschiedliche Ergebnisse liefern. Im Folgenden soll außer der bereits eingeführten Formel von Fletcher und Reeves (10.35) die Berechnungsvorschrift von Polak und Ribiére [181] vorgestellt werden, da diese für die nichtlineare Optimierung am effektivsten arbeitet [213, 214, 183]. Berechnet man im 6. Schritt des NKG-Algorithmus die Konjugationskoeffizienten  $\beta_{k+1}$  nach der Formel von Polak und Ribiére [181]

$$\beta_{k+1} = -\frac{\underline{g}(\hat{\Theta}_{k+1})^T \cdot (\underline{g}(\hat{\Theta}_{k+1}) - \underline{g}(\hat{\Theta}_k))}{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}, \quad (10.36)$$

so kann der Neustart des NKG-Algorithmus entfallen: Falls die konjugierten Suchrichtungen zur Fehlerfläche (bzw. zur aktuellen Taylorapproximation 2. Ordnung) passen, stehen die beiden Gradienten  $\underline{g}(\hat{\Theta}_k)$  und  $\underline{g}(\hat{\Theta}_{k+1})$  senkrecht aufeinander [214][44] und die Berechnung der Konjugationskoeffizienten nach Polak und Ribiére (10.36) geht über in die bekannte Form von Gleichung (10.35). Sind

jedoch die konjugierten Suchrichtungen nicht mehr geeignet, die aktuelle Fehlerfläche zu beschreiben, so ist der Optimierungsfortschritt gering und die beiden aufeinanderfolgenden Gradienten  $\underline{g}(\hat{\Theta}_k)$  und  $\underline{g}(\hat{\Theta}_{k+1})$  sind sehr ähnlich<sup>10)</sup>. Für diesen Fall startet der NKG-Algorithmus automatisch neu, da mit Gleichung (10.36) der Konjugationskoeffizient  $\beta_{k+1} \approx 0$  ist und somit die neue Suchrichtung  $\underline{s}_{k+1} = -\underline{g}(\hat{\Theta}_{k+1}) - \beta_{k+1} \cdot \underline{s}_k$  wie im 2. Schritt des NKG-Algorithmus berechnet wird.

### Beispiel — Optimierung mit dem Nichtlinearen Konjugierten Gradientenabstieg

Das zu lösende Optimierungsproblem ist wieder die Minimierung der einfachen Beispieldehlerfläche von Abbildung 10.23. Der hier zum Einsatz kommende Algorithmus berechnet die Konjugationskoeffizienten nach Polak und Ribiére (Gleichung (10.36)), die Liniensuche übernimmt der in Kapitel 10.2.2 vorgeschlagene ALIS-Algorithmus. Die NKG-Optimierung startet vom Anfangspunkt  $\hat{\Theta}_0 = [-1.25 \ -1.00]^T$  mit einer Liniensuche in die negative Gradientenrichtung  $\underline{s}_0 = [11.89 \ 1.58]^T$  (berechnet in Gleichung (10.14)). Diese erste Liniensuche wurde bereits bei dem Liniensuchbeispiel in Kapitel 10.2.2 auf Seite 343 durchgeführt, mit dem Ergebnis<sup>11)</sup>  $\eta_0 = 2.139/11.99 = 0.178$ . Mit dieser Schrittweite lautet der erste Optimierungsschritt (Punkt 4 des NKG-Algorithmus):

$$\hat{\Theta}_1 = \hat{\Theta}_0 + \eta_0 \cdot \underline{s}_0 = \begin{bmatrix} -1.25 \\ -1.00 \end{bmatrix} + 0.178 \cdot \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix} = \begin{bmatrix} 0.87 \\ -0.72 \end{bmatrix} \quad (10.37)$$

Zur Bestimmung der neuen Suchrichtung ist der Gradient am Punkt  $\hat{\Theta}_1$  erforderlich, für den

$$\underline{g}(\hat{\Theta}_1) = \begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} \quad (10.38)$$

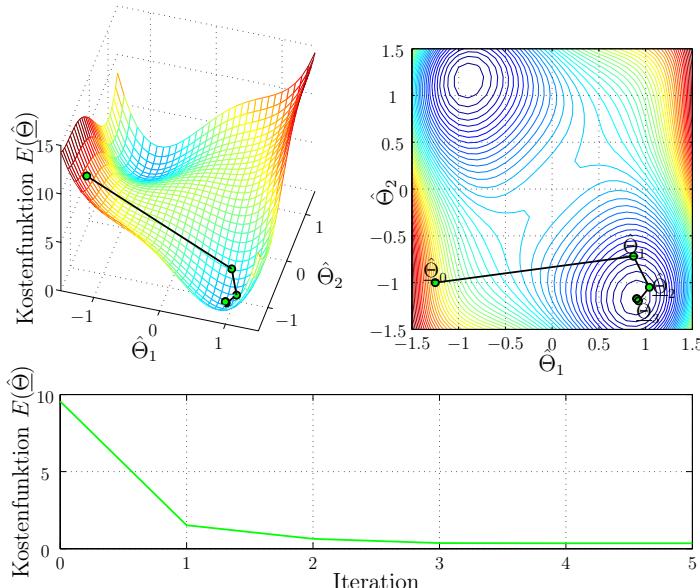
gilt. Nach Polak und Ribiére (Gleichung (10.36)) berechnet sich schließlich die neue Suchrichtung  $\underline{s}_1$  zu (Gradient  $\underline{g}(\hat{\Theta}_0)$  ist bereits aus Gleichung (10.14) bekannt):

$$\begin{aligned} \beta_1 &= -\frac{\underline{g}(\hat{\Theta}_1)^T \cdot (\underline{g}(\hat{\Theta}_1) - \underline{g}(\hat{\Theta}_0))}{\underline{g}(\hat{\Theta}_0)^T \cdot \underline{g}(\hat{\Theta}_0)} = -\frac{[-0.56 \ 4.21] \cdot \left( \begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} - \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix} \right)}{[-11.89 \ -1.58] \cdot \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix}} \\ &= -0.125 \\ \underline{s}_1 &= -\underline{g}(\hat{\Theta}_1) - \beta_1 \cdot \underline{s}_0 = -\begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} + 0.125 \cdot \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix} = \begin{bmatrix} 2.05 \\ -4.01 \end{bmatrix} \end{aligned} \quad (10.39)$$

<sup>10)</sup> Hinweis: Der Algorithmus verwendet eine nicht mehr aktuelle quadratischen Form zur Beschreibung der Fehlerfläche, er hängt fest und müsste neu gestartet werden.

<sup>11)</sup> Hinweis: Das Beispiel zur ALIS-Liniensuche auf Seite 343 nutzt für eine anschauliche Darstellung eine normierte Suchrichtung. Aus diesem Grund ist das Ergebnis der ALIS-Liniensuche  $\eta_k = 2.139$  zu groß und muss ebenfalls durch den Betrag  $\|\underline{s}_0\| = 11.99$  geteilt werden, um die in Gleichung (10.14) getroffene Normierung rückgängig zu machen.

Die Liniensuche entlang dieser Suchrichtung  $s_1$  ergibt den nächsten Gewichtsvektor  $\hat{\Theta}_2$ . Analog dazu folgen die weiteren Optimierungsschritte, welche in der Abbildung 10.23 und in der Tabelle 10.4 zusammengefasst sind.



**Abb. 10.23:** Optimierungsbeispiel mit dem Nichtlinearen Konjugierten Gradientenabstieg — Die Berechnung der Konjugationskoeffizienten erfolgt nach Polak und Ribiére. Für die Liniensuche kommt der ALIS-Algorithmus zum Einsatz. Nach nur 5 Iterationen ist das Minimum erreicht.

Schritt $k$	0	1	2	3	4	5
$\hat{\Theta}_k$	-1.25 -1.00	0.87 -0.72	1.04 -1.05	0.92 -1.19	0.91 -1.17	0.90 -1.17
$s_k$	11.89 1.58	2.05 -4.01	-2.23 -2.81	-0.40 0.51	-0.04 -0.04	-
$\eta_k$	0.178	0.083	0.052	0.048	0.061	-
$\eta_{max}$	0.5	0.125	0.125	0.125	0.125	-
ALIS	5	6	4	4	4	0

**Tabelle 10.4:** Optimierungsbeispiel mit dem Nichtlinearen Konjugierten Gradientenabstieg — Der ALIS-Algorithmus führt nur eine Intervallanpassung bei der zweiten Liniensuche im Schritt  $k = 1$  durch. In der letzten Zeile steht die Anzahl der für die Liniensuche erforderlichen Funktionsauswertungen.

Der hier verwendete NKG<sub>ALIS</sub>-Algorithmus mit Berechnung der Konjugationskoeffizienten nach Polak und Ribiére benötigt nur 5 Iterationen, um das Mi-

nimum des Fehlerflächenbeispiels zu erreichen. Für die ALIS-Liniensuche sind insgesamt 23 Funktionsauswertungen<sup>12)</sup> erforderlich.

#### 10.4.2 Das Skalierte Konjugierte Gradientenverfahren

Möller [155, 154] beschreibt eine Abwandlung des NKG-Algorithmus, um die meist aufwändige Liniensuche zu vermeiden. Falls die Fehlerfläche eine quadratische Form — mit der Hessematrix  $\mathbf{H}(\hat{\Theta}_k)$  — hat, so berechnet sich die für das Minimum notwendige Schrittweite zu [214][44]:

$$\eta_k = \frac{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}{\underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k} \quad (10.40)$$

Beim sogenannten *Skalierten Konjugierten Gradienten-Algorithmus* (SKG) soll die Schrittweitenberechnung nach Gleichung (10.40) — die eigentlich nur für quadratische Fehlerflächen gilt — mit Hilfe einer Skalierung auch für die Optimierung von nichtlinearen Fehlerflächen herangezogen werden: Für die Berechnung der Schrittweite nach Gleichung (10.40) ist die Hessematrix  $\mathbf{H}(\hat{\Theta}_k)$  notwendig. Da eine explizite Berechnung der Hessematrix mit einem relativ großen Aufwand verbunden ist, gilt es zunächst, eine geeignete Approximation zu finden. Betrachtet man zunächst die Richtungsableitung in die Richtung  $\underline{r}$ , so ist mit dem Differenzenquotient aus zwei Kostenfunktionswerten

$$\underline{g}(\hat{\Theta}_k)^T \cdot \underline{r} \approx \frac{E(\hat{\Theta}_k + \sigma \cdot \underline{r}) - E(\hat{\Theta}_k)}{\sigma} \quad (10.41)$$

und einem kleinen Wert für  $\sigma$  eine gute Approximation möglich. In ähnlicher Weise lässt sich der Ausdruck  $\mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k$  im Nenner von Gleichung (10.40) näherungsweise durch einen Differenzenquotient aus zwei Gradienten darstellen:

$$\mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k \approx \frac{\underline{g}(\hat{\Theta}_k + \sigma \cdot \underline{s}_k) - \underline{g}(\hat{\Theta}_k)}{\sigma} = \underline{\xi}_k \quad (10.42)$$

Für  $\sigma \xrightarrow{\lim} 0$  konvergiert die Näherung von Gleichung (10.42) gegen den tatsächlichen Wert von  $\mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k$  [155, 154, 60]. Beim SKG wird also nicht die Hessematrix approximiert, sondern das mit wenig Aufwand zu berechnende Produkt aus Hessematrix und Suchrichtung. Für die Approximation von Gleichung (10.42) ist eine weitere Gradientenberechnung am Fehlerflächenpunkt  $\hat{\Theta}_k + \sigma \cdot \underline{s}_k$  erforderlich. Damit sich der Abstand dieses Punktes zum bereits berechneten Punkt an der Stelle  $\hat{\Theta}_k$  nicht verändert, wird der Abstand durch eine Normierung konstant gehalten. Damit folgt für die Näherung von Gleichung (10.42):

---

<sup>12)</sup> Hinweis: Wie in Tabelle 10.4 zu sehen, führt der ALIS-Algorithmus bei der ersten Liniensuche keine Intervallanpassung durch und benötigt nur 5 Funktionsauswertungen. Beim Liniensuchbeispiel auf Seite 343 sind aufgrund der Normierung bei der Suchrichtung 3 Intervallvergrößerungen und 11 Funktionsauswertungen notwendig.

$$\begin{aligned}\sigma_k &= \frac{\sigma}{\|\underline{s}_k\|} \\ \xi_k &= \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k \approx \frac{\underline{g}(\hat{\Theta}_k + \sigma_k \cdot \underline{s}_k) - \underline{g}(\hat{\Theta}_k)}{\sigma_k}\end{aligned}\quad (10.43)$$

Møller [155, 154] empfiehlt, den Wert von  $\sigma$  so klein zu wählen, wie es die Rechengenauigkeit zulässt. In diesem Fall (z.B.  $\sigma \leq 10^{-4}$ ) ist laut Møller der Parameter  $\sigma$  nicht ausschlaggebend für die Konvergenz des Optimierungsalgorithmus. Da bei nichtlinearen Fehlerflächen die positive Definitheit der Hessematrix nicht immer gewährleistet ist, kann der Nenner in Gleichung (10.40) auch negativ werden und die ermittelte Gewichtsanpassung zu einer Vergrößerung des Kostenfunktionswertes führen. Aus diesem Grund wird zur Hessematrix ein Vielfaches der Einheitsmatrix addiert:

$${}^*H(\hat{\Theta}_k) = \mathbf{H}(\hat{\Theta}_k) + \mu_k \cdot E \quad (10.44)$$

Dieser Vorgang heißt *Skalierung*. Eine ausführliche Beschreibung dazu folgt im Kapitel 10.4.5 bei der Optimierung mit dem Levenberg-Marquardt-Verfahren. Durch die Skalierung verändert sich die Schrittweitenberechnung von Gleichung (10.40) zu

$$\begin{aligned}\eta_k &= \frac{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}{\underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k + \mu_k \cdot \|\underline{s}_k\|^2} \\ &\stackrel{(10.42)}{=} \frac{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}{\underline{s}_k^T \cdot \xi_k + \mu_k \cdot \|\underline{s}_k\|^2} \\ &= \frac{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}{\delta_k},\end{aligned}\quad (10.45)$$

mit der Definition  $\delta_k := \underline{s}_k^T \cdot \xi_k + \mu_k \cdot \|\underline{s}_k\|^2$  für den Nenner. Das Vorzeichen des Nenners  $\delta_k$  von Gleichung (10.45) gibt an, ob die gerade gültige Näherung der Hessematrix positiv definit ist. Für  $\delta_k \leq 0$  wird der *Skalierungsfaktor*  $\mu_k$  vergrößert (neuer Skalierungsfaktor:  ${}^*\mu_k$ ), bis der Nenner einen positiven Wert annimmt (neuer Wert des Nenners:  ${}^*\delta$ ). Der Zusammenhang zwischen den neuen und alten Werten lautet:

$${}^*\delta_k = \delta_k + ({}^*\mu_k - \mu_k) \cdot \|\underline{s}_k\|^2 \quad (10.46)$$

Damit ist der neue Nenner  ${}^*\delta_k$  positiv für

$${}^*\mu_k > \mu_k - \frac{\delta_k}{\|\underline{s}_k\|^2} \quad (10.47)$$

Møller schlägt in [155, 154] vor, den neuen Skalierungsfaktor  ${}^*\mu_k$  auf den Wert

$${}^*\mu_k = 2 \cdot \left( \mu_k - \frac{\delta_k}{\|\underline{s}_k\|^2} \right) \quad (10.48)$$

zu setzen. Mit dieser Wahl (10.48) und der Gleichung (10.46) folgt für die Neuberechnung des Nenners:

$$\begin{aligned} {}^*\delta_k &= \delta_k + 2 \cdot \mu_k \cdot \|\underline{s}_k\|^2 - 2 \cdot \delta_k - \mu_k \cdot \|\underline{s}_k\|^2 \\ &= -\delta_k + \mu_k \cdot \|\underline{s}_k\|^2 \end{aligned} \quad (10.49)$$

Durch den neuen Skalierungsfaktor  ${}^*\mu_k$  nach Gleichung (10.48) ist der Nenner  ${}^*\delta_k$  nun positiv. Die Schrittweitenberechnung von Gleichung (10.45) verwendet den mit Gleichung (10.49) berechneten Nenner.

Bisher wurde lediglich eine Art Vorskalierung durchgeführt, damit die Hessematrix positiv definit ist. Je nach Qualität der Fehlerflächenapproximation kann der Skalierungsfaktor einen direkten Einfluss auf die Wirkungsweise des Algorithmus nehmen. Da sich der Skalierungsfaktor  $\mu_k$  von einem Schritt zum nächsten verändern darf, kann er einen Vertrauensbereich — eine sogenannte *model trust region* — einstellen. Ist die quadratische Approximation gut geeignet, die Fehlerfläche zu beschreiben, so muss der Skalierungsfaktor  $\mu_k$  klein sein. Damit ist der Vertrauensbereich groß und die Schrittweitenbestimmung nach Gleichung (10.40) (bzw. nach Gleichung (10.45)) funktioniert sehr gut. Stimmt die ermittelte Hessematrix — und damit die quadratische Approximation der Fehlerfläche — nicht mit der tatsächlichen Fehlerflächenform überein, so ist ein großer Skalierungsfaktor  $\mu_k$  erforderlich. In diesem Fall ist die Schrittweite (und der Vertrauensbereich für die quadratische Fehlerflächenapproximation) sehr klein und die Hessematrix hat keinen Einfluss mehr auf die Schrittweitenbestimmung. Auch die Konjugation der Suchrichtungen zueinander geht verloren, so dass die Optimierung vergleichbar ist mit dem in Kapitel 10.3 vorgestellten Gradientenabstieg mit einer kleinen Lernschrittweite von  $\eta_k = 1/\mu_k$ .

Die grundsätzliche Idee des SKG ist mit den Formeln (10.42) und (10.45) gegeben. Im Folgenden muss ein Gütekriterium für die quadratische Fehlerflächenapproximation gefunden werden. Außerdem ist eine Bedingung für die Veränderung des Skalierungsfaktor  $\mu_k$  notwendig. Ein geeignetes Gütekriterium für die Qualität der gefundenen Taylorapproximation 2. Ordnung zur Beschreibung der Fehlerfläche ist [53, 155, 18]:

$$\Upsilon_k = \frac{E(\hat{\Theta}_k) - E(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}{E(\hat{\Theta}_k) - E^*(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}, \quad (10.50)$$

wobei  $E^*(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)$  die ermittelte Taylorapproximation 2. Ordnung am Entwicklungspunkt  $\hat{\Theta}^* = \hat{\Theta}_k$  ist:

$$E^*(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k) = E(\hat{\Theta}_k) + \underline{g}(\hat{\Theta}_k)^T \cdot \eta_k \cdot \underline{s}_k + \frac{1}{2} \cdot \eta_k^2 \cdot \underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k \quad (10.51)$$

Setzt man die Taylorapproximation von Gleichung (10.51) in das Gütekriterium von Gleichung (10.50) ein, so gilt:

$$\begin{aligned}
\Upsilon_k &= \frac{E(\hat{\Theta}_k) - E(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}{E(\hat{\Theta}_k) - E(\hat{\Theta}_k) - \underline{g}(\hat{\Theta}_k)^T \cdot \eta_k \cdot \underline{s}_k - \frac{1}{2} \cdot \eta_k^2 \cdot \underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k} \\
&= \frac{E(\hat{\Theta}_k) - E(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}{-\underline{g}(\hat{\Theta}_k)^T \cdot \eta_k \cdot \underline{s}_k - \frac{1}{2} \cdot \eta_k \cdot \left( -\frac{\underline{s}_k^T \cdot \underline{g}(\hat{\Theta}_k)}{\underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k} \right) \cdot \underline{s}_k^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k} \\
&= -2 \cdot \frac{E(\hat{\Theta}_k) - E(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}{\eta_k \cdot \underline{s}_k^T \cdot \underline{g}(\hat{\Theta}_k)}
\end{aligned} \tag{10.52}$$

Falls das Gütekriterium  $\Upsilon_k$  nahe bei 1 liegt, ist die quadratische Approximation der Fehlerfläche sehr gut und das SKG Verfahren kann den Skalierungsfaktor  $\mu_k$  reduzieren (z.B.: falls  $\Upsilon_k > 0.75$  dann  $\mu_{k+1} = \mu_k/2$  [18, 53]). Umgekehrt zeigt ein kleines Gütekriterium  $\Upsilon_k$  an, dass die Approximation am aktuellen Fehlerflächenpunkt  $\hat{\Theta}_k$  nicht in der Lage ist, die tatsächliche Fehlerfläche zu beschreiben. Das SKG-Verfahren muss den Skalierungsfaktor  $\mu_k$  erhöhen (z.B.: falls  $\Upsilon_k < 0.25$  dann  $\mu_{k+1} = 4 \cdot \mu_k$  [18, 53]).

### Zusammenfassung — Skalierter Konjugierter Gradientenabstieg

1. Anfangsinitialisierung: Wahl eines geeigneten Startpunktes  $\hat{\Theta}_0$  auf der Fehlerfläche und kleine Werte festlegen für  $\sigma$  und den Skalierungsfaktors  $\mu_0$ .

2. Suchrichtung für den ersten Optimierungsschritt bestimmen:

$$\underline{s}_0 = -\underline{g}(\hat{\Theta}_0)$$

3. Neue Approximation für das Produkt aus Hessematrix und Suchrichtung ermitteln (Gleichung (10.43)):

$$\begin{aligned}
\sigma_k &= \frac{\sigma}{\|\underline{s}_k\|} \\
\underline{\xi}_k &= \mathbf{H}(\hat{\Theta}_k) \cdot \underline{s}_k \approx \frac{\underline{g}(\hat{\Theta}_k + \sigma_k \cdot \underline{s}_k) - \underline{g}(\hat{\Theta}_k)}{\sigma_k}
\end{aligned}$$

4. Nenner der Schrittweitenformel berechnen (Gleichung (10.45)):

$$\delta_k := \underline{s}_k^T \cdot \underline{\xi}_k + \mu_k \cdot \|\underline{s}_k\|^2$$

5. Vorskalierung, damit Hessematrix positiv definit (bzw. Nenner  $\delta_k$  positiv):

$\delta_k \leq 0$  :  $\rightarrow$  Skalierungsfaktor  $\mu_k$  anpassen (Gleichung (10.48)):

$${}^*\mu_k = 2 \cdot \left( \mu_k - \frac{\delta_k}{\|\underline{s}_k\|^2} \right)$$

$\rightarrow$  Neuberechnung des Nenners (Gleichung (10.49)):

$${}^*\delta_k = -\delta_k + \mu_k \cdot \|\underline{s}_k\|^2$$

$\delta_k > 0$  :  $\rightarrow$  keine Veränderung notwendig:

$${}^*\mu_k = \mu_k$$

$${}^*\delta_k = \delta_k$$

6. Berechnung der Schrittweite (Gleichung (10.45)):

$$\eta_k = \frac{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}{{}^*\delta_k}$$

7. Gütekriterium bestimmten (Gleichung (10.52)):

$$\Upsilon_k = -2 \cdot \frac{E(\hat{\Theta}_k) - E(\hat{\Theta}_k + \eta_k \cdot \underline{s}_k)}{\eta_k \cdot \underline{s}_k^T \cdot \underline{g}(\hat{\Theta}_k)}$$

8. Validierung und Skalierung:

$\Upsilon_k \geq 0$  :  $\rightarrow$  Fehlerflächenapproximation erfolgreich, Optimierungsschritt

$$\text{ausführen: } \hat{\Theta}_{k+1} = \hat{\Theta}_k + \eta_k \cdot \underline{s}_k$$

$\rightarrow$  Gradient  $\underline{g}(\hat{\Theta}_{k+1})$  an der neuen Stelle ermitteln

$\rightarrow$  Neue Suchrichtung nach Polak und Ribiére berechnen (Gleichung (10.36)):

$$\beta_{k+1} = -\frac{\underline{g}(\hat{\Theta}_{k+1})^T \cdot (\underline{g}(\hat{\Theta}_{k+1}) - \underline{g}(\hat{\Theta}_k))}{\underline{g}(\hat{\Theta}_k)^T \cdot \underline{g}(\hat{\Theta}_k)}$$

$$\underline{s}_{k+1} = -\underline{g}(\hat{\Theta}_{k+1}) - \beta_{k+1} \cdot \underline{s}_k$$

→ Skalierung:

für  $\Upsilon_k > 0.75$  Vertrauensbereich vergrößern mit  $\mu_{k+1} = \hat{\mu}_k / 2$

für  $\Upsilon_k < 0.25$  Vertrauensbereich verkleinern mit  $\mu_{k+1} = 4 \cdot \hat{\mu}_k$

sonst Vertrauensbereich beibehalten mit  $\mu_{k+1} = \hat{\mu}_k$

→ weiter mit Schritt 3

$\Upsilon_k < 0$  : → Fehlerflächenapproximation nicht angemessen, keinen Optimierungsschritt ausführen:  $\hat{\Theta}_{k+1} = \hat{\Theta}_k$

→ Skalierung:

Vertrauensbereich verkleinern mit  $\mu_{k+1} = 4 \cdot \hat{\mu}_k$

→ im nächsten Schritt ist keine Neuberechnung der Hessematrix

erforderlich:  $\underline{\xi}_{k+1} = \underline{\xi}_k$

auch die Suchrichtung bleibt gleich:  $\underline{s}_{k+1} = \underline{s}_k$

→ weiter mit Schritt 4

9. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 3 bis 8.

Møller startet den SKG-Algorithmus nach  $N$  Iterationen neu. Dies ist jedoch nicht erforderlich, da die Suchrichtungsbestimmung nach Polak und Ribiére (siehe Gleichung (10.36)) einen automatischen Neustart durchführt. Deshalb verzichtet die SKG-Realisierung in diesem Buch auf den Neustart. Eine ausführliche Erklärung dazu findet man im Text auf Seite 355 nach Gleichung (10.36).

### Beispiel — Optimierung mit dem Skalierten Konjugierten Gradientenabstieg

Wie die vorangegangenen Optimierungsverfahren soll auch der SKG die Beispieldifferenzfläche minimieren. Für das Optimierungsbeispiel ist eine Distanz von  $\sigma = 0.1$  gut geeignet. Der anfängliche Skalierungsfaktor wird zu  $\mu_0 = 0.1$  gesetzt. Die erste Suchrichtung beim Startpunkt  $\hat{\Theta}_0 = [-1.25 \ -1.00]^T$  ergibt sich zu:

$$\underline{s}_0 = -\underline{g}(\hat{\Theta}_0) = \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix} \quad \|\underline{s}_0\| = 11.99$$

Nach der Bestimmung des zusätzlichen Gradienten an der Stelle  $\hat{\Theta}_0 + \sigma_0 \cdot \underline{s}_0$

$$\underline{g}(\hat{\Theta}_0 + \sigma_0 \cdot \underline{s}_0) = \underline{g}\left(\begin{bmatrix} -1.25 \\ -1.00 \end{bmatrix} + \frac{0.1}{11.99} \cdot \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix}\right) = \underline{g}\left(\begin{bmatrix} -1.151 \\ -0.987 \end{bmatrix}\right) = \begin{bmatrix} -9.26 \\ -1.84 \end{bmatrix}$$

kann die Approximation für das Produkt aus Hessematrix und Suchrichtung (Gleichung (10.43)) erfolgen:

$$\begin{aligned} \sigma_0 &= \frac{\sigma}{\|\underline{s}_0\|} = \frac{0.1}{11.99} \\ \underline{\xi}_0 &= \mathbf{H}(\hat{\Theta}_0) \cdot \underline{s}_0 \approx \frac{\underline{g}(\hat{\Theta}_0 + \sigma_0 \cdot \underline{s}_0) - \underline{g}(\hat{\Theta}_0)}{\sigma_0} \\ &\approx \left(\begin{bmatrix} -9.26 \\ -1.84 \end{bmatrix} - \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix}\right) \cdot \frac{11.99}{0.1} = \begin{bmatrix} 315.34 \\ -31.17 \end{bmatrix} \end{aligned}$$

Da die Beispieldifferenzfläche als Funktion vorliegt, ist eine exakte Kontrollrechnung möglich, welche mit  $\mathbf{H}(\hat{\Theta}_0) \cdot \underline{s}_0 = [347.7 \ -38.2]^T$  recht ähnliche Werte liefert. Die gefundene Approximation führt mit Gleichung (10.45) auf den Nenner der Schrittweitenformel:

$$\delta_0 = \underline{s}_0^T \cdot \underline{\xi}_0 + \mu_0 \cdot \|\underline{s}_0\|^2 = [11.89 \ 1.58]^T \cdot \begin{bmatrix} 315.34 \\ -31.17 \end{bmatrix} + 0.1 \cdot 11.99^2 = 3715$$

Wegen  $\delta_0 > 0$  ist die Hessematrix positiv definit, was keine Vorskalierung notwendig macht ( ${}^*\delta_0 = \delta_0$ ). Somit berechnet sich die erste Schrittweite nach Gleichung (10.45) zu:

$$\eta_0 = \frac{\underline{g}(\hat{\Theta}_0)^T \cdot \underline{g}(\hat{\Theta}_0)}{{}^*\delta_0} = \frac{[-11.89 \ -1.58] \cdot \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix}}{3715} = 0.039$$

Das Gütekriterium (Gleichung (10.52)) erfordert eine weitere Funktionsauswertung an der Stelle  $\hat{\Theta}_1 = \hat{\Theta}_0 + \eta_0 \cdot \underline{s}_0$ :

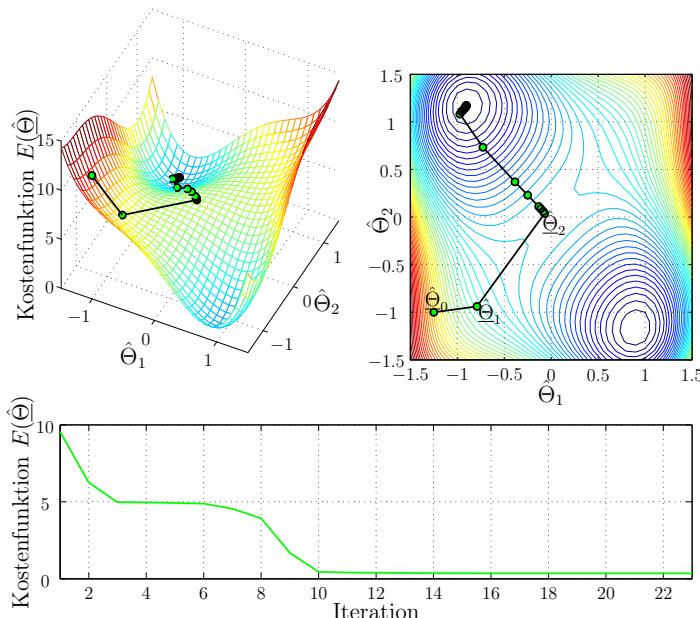
$$\hat{\Theta}_1 = \hat{\Theta}_0 + \eta_0 \cdot \underline{s}_0 = \begin{bmatrix} -1.25 \\ -1.00 \end{bmatrix} + 0.039 \cdot \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix} = \begin{bmatrix} -0.79 \\ -0.93 \end{bmatrix}$$

$$E(\hat{\Theta}_1) = 6.25$$

$$\begin{aligned} \Upsilon_0 &= -2 \cdot \frac{E(\hat{\Theta}_0) - E(\hat{\Theta}_0 + \eta_0 \cdot \underline{s}_0)}{\eta_0 \cdot \underline{s}_0^T \cdot \underline{g}(\hat{\Theta}_0)} \\ &= -2 \cdot \frac{9.55 - 6.25}{0.039 \cdot [11.89 \ 1.58] \cdot \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix}} = 1.18 \end{aligned}$$

Das Gütekriterium  $\Upsilon_0$  ist positiv, die Berechnungen führen somit auf eine erfolgreiche Fehlerflächenapproximation. Der Optimierungsschritt zum Punkt  $\hat{\Theta}_1$

kann ausgeführt werden (siehe Abbildung 10.24). Die folgenden beiden Punkte entsprechen dem NKG-Algorithmus: Berechnung des Gradienten an der neuen Stelle  $\hat{\Theta}_1$  (vergleiche Gleichung (10.38)) und Bestimmen der neuen Suchrichtung nach Polak und Ribiére mit Gleichung (10.36) (vergleiche Gleichung (10.39)). Das Gütekriterium erlaubt mit  $\Upsilon_0 > 0.75$  eine Vergrößerung des Vertrauensbereichs durch  $\mu_1 = \mu_0/2 = 0.05$ . Diese abschließende Skalierung beendet den ersten Optimierungsschritt.



**Abb. 10.24:** Optimierungsbeispiel mit dem Skalierten Konjugierten Gradientenabstieg — Nach 23 Iterationen ist das Minimum erreicht.

Abbildung 10.24 fasst den weiteren Optimierungsverlauf zusammen. Der SKG-Algorithmus erreicht nach 23 Optimierungsschritten das Minimum der Fehlerfläche. Auffällig hierbei ist, dass zunächst Kurs auf den Sattelpunkt in der Mitte der Fehlerfläche genommen wird, bevor das Verfahren zum hinteren Minimum bei  $\hat{\Theta}_{23} = [-0.90 \ 1.17]^T$  konvergiert.

#### 10.4.3 Das Newton-Verfahren

Die Herleitung des *Newton-Verfahrens* ist vergleichsweise einfach und geht wieder von einer Taylorapproximation 2. Ordnung der Fehlerfläche aus. Als Entwicklungspunkt der Taylorapproximation dient der aktuelle Punkt  $\hat{\Theta}_k$  auf der Fehlerfläche zum  $k$ -ten Optimierungsschritt:

$$E^*(\hat{\Theta}) = E(\hat{\Theta}_k) + \underline{g}^T(\hat{\Theta}_k) \cdot (\hat{\Theta} - \hat{\Theta}_k) + \frac{1}{2} \cdot (\hat{\Theta} - \hat{\Theta}_k)^T \cdot \mathbf{H}(\hat{\Theta}_k) \cdot (\hat{\Theta} - \hat{\Theta}_k) \quad (10.53)$$

Die grundlegende Idee bei allen Newton-Verfahren ist, den stationären Punkt der quadratischen Fehlerflächenapproximation aufzuspüren. Ziel ist also, eine Gewichtsänderung  $(\hat{\Theta} - \hat{\Theta}_k)$  zu finden, die vom Fehlerflächenpunkt  $\hat{\Theta}_k$  aus zu einem Punkt mit minimalem Kostenfunktionswert führt. Für diese Bedingung muss die erste Ableitung der Fehlerfläche nach der Gewichtsänderung  $(\hat{\Theta} - \hat{\Theta}_k)$  Null sein:

$$\begin{aligned} \frac{dE^*(\hat{\Theta})}{d(\hat{\Theta} - \hat{\Theta}_k)} &= 0 \\ &= \underline{g}(\hat{\Theta}_k) + \mathbf{H}(\hat{\Theta}_k) \cdot (\hat{\Theta} - \hat{\Theta}_k) \end{aligned} \quad (10.54)$$

Was zu einer notwendigen Gewichtsänderung von

$$\hat{\Theta} - \hat{\Theta}_k = -\mathbf{H}(\hat{\Theta}_k)^{-1} \cdot \underline{g}(\hat{\Theta}_k) \quad (10.55)$$

führt. Die Richtung des Vektors von Gleichung (10.55) wird häufig als *Newton-Suchrichtung* bezeichnet und zeigt nicht in die Richtung des negativen Gradien-ten, sondern auf den stationären Punkt der quadratischen Fehlerflächenapproximation. Der Newton-Algorithmus lautet somit:

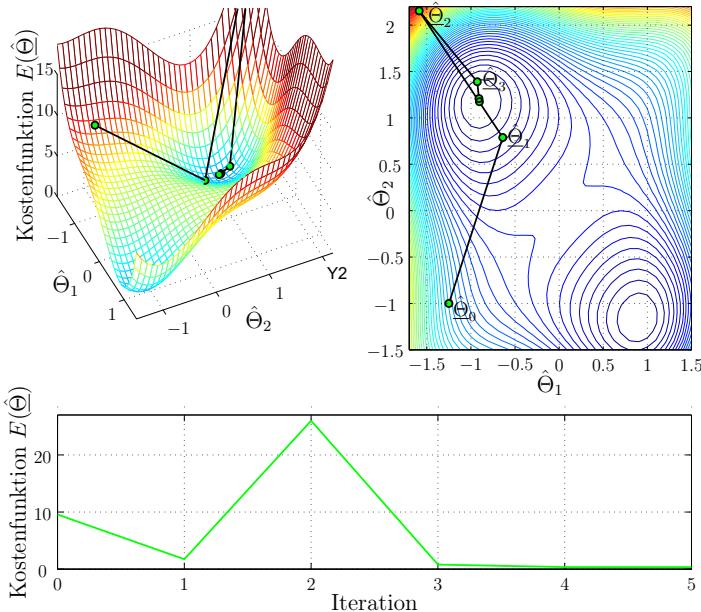
$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \mathbf{H}(\hat{\Theta}_k)^{-1} \cdot \underline{g}(\hat{\Theta}_k) \quad (10.56)$$

#### 10.4.3.1 Konvergenz des Newton-Verfahrens

Falls die Fehlerfläche eine quadratische Form hat — was in der Umgebung eines Minimums sehr gut zutrifft — konvergiert das Newton-Verfahren in einem Schritt. Bei einer nichtlinearen Fehlerfläche sind mehrere Iterationen notwendig. Im Falle einer positiv definiten Hessematrix besitzt das Newton-Verfahren eine Konvergenzrate (siehe Gleichung (10.100)) zweiter Ordnung [169, 192, 97].

Ein Problem bei der direkten Implementierung des Newton-Verfahrens ist, dass der Algorithmus nicht unbedingt in ein Minimum konvergieren muss. Der Algorithmus findet stationäre Punkte, was bedeutet, dass auch Maximalstellen oder Sattelpunkte mögliche Optimierungsergebnisse darstellen. Diese Problematik lässt sich mit der eingeführten Beispieldehlerfläche zeigen. Während die Optimierung bei einem Startpunkt von  $\hat{\Theta}_0 = [-1.25 \ -1.00]^T$  nach 5 Schritten das Minimum  $\hat{\Theta}_6 = [-0.90 \ 1.17]^T$  findet (siehe Abbildung 10.25), konvergiert der Newton-Algorithmus bei einem Startpunkt von  $\hat{\Theta}_0 = [-1.25 \ -1.25]^T$  auf den Sattelpunkt  $\hat{\Theta} = [0 \ 0]^T$  zwischen den beiden globalen Minima (siehe Abbildung 10.26). Für ein Minimum gilt die zusätzliche Forderung, dass die Hessematrix  $\mathbf{H}(\hat{\Theta}_{k+1})$  am gefundenen stationären Punkt  $\hat{\Theta}_{k+1}$  positiv definit sein muss.

Der Newton-Algorithmus von Gleichung (10.56) ist für die Minimumssuche auf der Fehlerfläche nicht geeignet. Bei der praktischen Umsetzung ist eine



**Abb. 10.25:** Optimierungsbeispiel mit dem Newton-Verfahren bei einem Startpunkt von  $\underline{\hat{\Theta}}_0 = [-1.25 \ -1.00]^T$  — Der Newton-Algorithmus findet nach 5 Schritten das Minimum.

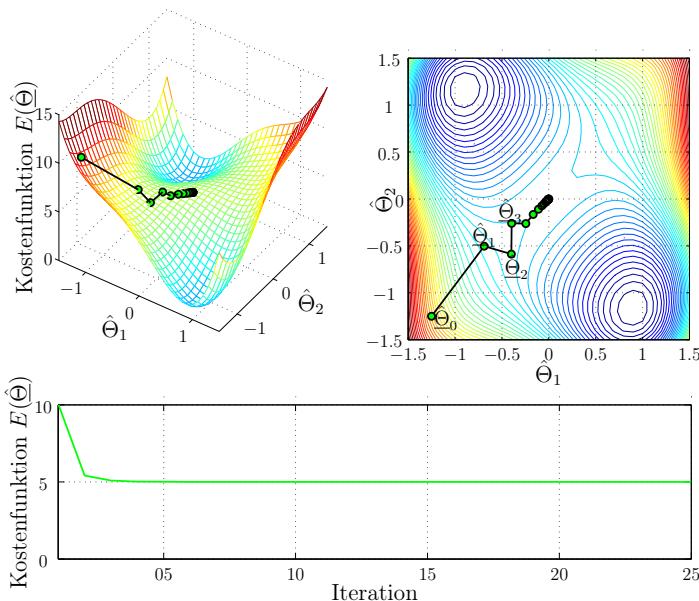
zusätzliche stabilisierende Maßnahme notwendig. Hier gibt es prinzipiell zwei unterschiedliche Lösungsansätze: Der erste Ansatz ist die Skalierung der Hessematrix. Diese Maßnahme versucht, die Hessematrix gezielt so zu verändern, dass sie positiv definit wird. Die Skalierung kommt beim Levenberg-Marquardt-Algorithmus in Kapitel 10.4.5 zum Einsatz. Hinweis: Auch das SKG-Verfahren im Kapitel 10.4.2 nutzte bereits ein ähnliches Skalierungsverfahren. Die zweite Möglichkeit ist, den Abstieg auf der Fehlerfläche — wie beim Algorithmus des NKG in Kapitel 10.4.1 — über eine Minimumssuche zu gewährleisten.

#### 10.4.3.2 Hessematrixberechnung beim Newton-Verfahren

Ein weiteres Problem bei der direkten Implementierung des Newton Verfahrens ist die sehr aufwändige Berechnung der zweiten Ableitung [97, 18]. Auch für dieses Problem hat die nichtlineare Optimierung zwei Lösungen gefunden. Die erste Lösung ist, die Hessematrix — wie in [44] gezeigt — über die Jacobimatrix anzunähern. Der Newton-Algorithmus von Gleichung (10.56) geht damit über zu

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \left( J(\hat{\Theta}_k)^T \cdot J(\hat{\Theta}_k) \right)^{-1} \cdot J(\hat{\Theta}_k)^T \cdot \underline{e}(\hat{\Theta}_k) \quad (10.57)$$

und wird im Folgenden, wie in der Literatur üblich [213, 183, 79], als *Gauss-Newton-Verfahren* bezeichnet. Die zweite Lösung ist, die inverse Hessematrix



**Abb. 10.26:** Optimierungsbeispiel mit dem Newton-Verfahren bei einem Startpunkt von  $\hat{\Theta}_0 = [-1.25 \ -1.25]^T$  — Der Newton-Algorithmus konvergiert auf den Sattelpunkt zwischen den beiden globalen Minima.

schrittweise über eine Aufdatierungsformel anzunähern. Diesen Weg gehen die *Inversen Quasi-Newton Verfahren*<sup>13)</sup>. Ein weiterer Vorteil der Inversen Quasi-Newton-Verfahren ist, dass die aufwändige Berechnung der Matrixinversion entfällt. Jedoch ist bei jedem Optimierungsschritt wieder eine Minimumssuche in die gefundene Newton-Richtung erforderlich.

#### 10.4.4 Die Quasi-Newton-Verfahren

Wie bereits oben erwähnt, kann das Konvergenzproblem beim Newton-Algorithmus durch eine Minimumssuche in die gefundene Newton-Richtung gelöst werden. Trotzdem kommt das klassische Newton-Verfahren nach Gleichung (10.56) in der Praxis kaum zum Einsatz, da die Berechnung der Hessematrix — insbesondere bei einer großen Parameteranzahl  $N$  — eine große Herausforderung darstellt. Einen erheblichen Rechenaufwand erfordert auch die Invertierung der  $N \times N$ -dimensionalen Hessematrix.

In der Praxis findet man das Newton-Verfahren deshalb meist nur in modifizierter Form. Die Inversen Quasi-Newton-Verfahren in diesem Unterkapitel

<sup>13)</sup> Hinweis: Für das Quasi-Newton-Verfahren sind auch die Bezeichnungen Variable Metrik-methode [183, 192, 169] und Sekanten-Methode [213] in Gebrauch.

bauen iterativ eine Näherung für die inverse Hessematrix

$$\underline{G}(\hat{\Theta}_k) = \mathbf{H}(\hat{\Theta}_k)^{-1} \quad (10.58)$$

auf, wobei dazu lediglich die erste Ableitung notwendig ist. Die *Direkten Quasi-Newton-Verfahren* approximieren die Hessematrix, weshalb bei der Optimierung eine Matrixinversion notwendig ist<sup>14)</sup>. Die Approximation beginnt mit einer beliebigen positiv definiten Matrix und wird über eine *Aufdatierungsformel* im Laufe der Optimierung immer genauer. Je nach Aufdatierungsformel erhält man verschiedene Quasi-Newton-Verfahren, welche alle die im Folgenden beschriebene Quasi-Newton-Bedingung erfüllen. Die anschließenden Ausführungen stellen die gängigsten Aufdatierungsformeln vor. Konvergenzanalysen und weiterführende theoretische Betrachtungen sind in [61, 46] zu finden.

#### 10.4.4.1 Die Quasi-Newton-Bedingung

Zur Vermeidung des hohen Rechenaufwandes beim Newton-Verfahren soll zunächst eine Approximation für die Hessematrix gefunden werden. Im eindimensionalen Fall ist eine Näherung für die zweite Ableitung der Kostenfunktion  $h(\hat{\Theta}_{k+1}) = \nabla^2 E(\hat{\Theta}_{k+1})$  recht einfach über den Differenzenquotienten möglich:

$$h(\hat{\Theta}_{k+1}) \approx \frac{g(\hat{\Theta}_{k+1}) - g(\hat{\Theta}_k)}{\hat{\Theta}_{k+1} - \hat{\Theta}_k}, \quad (10.59)$$

wobei  $g(\hat{\Theta}_k) = \underline{g}(\hat{\Theta}_k)$  und  $g(\hat{\Theta}_{k+1}) = \underline{g}(\hat{\Theta}_{k+1})$  die einfachen Ableitungen der Kostenfunktion an den Punkten  $\hat{\Theta}_k$  und  $\hat{\Theta}_{k+1}$  sind. In ähnlicher Weise ist eine Approximation im mehrdimensionalen Fall für die Hessematrix durchführbar. Nimmt man eine quadratische Fehlerfläche an, so ist mit Gleichung (10.56) — ausgehend von zwei unterschiedlichen Punkten  $\hat{\Theta}_k$  und  $\hat{\Theta}_{k+1}$  auf der Fehlerfläche — der stationäre Punkt  $\hat{\underline{\Theta}}^*$  jeweils nach einem Optimierungsschritt gefunden:

$$\hat{\underline{\Theta}}^* = \hat{\Theta}_k - \mathbf{H}(\hat{\Theta}_k)^{-1} \cdot \underline{g}(\hat{\Theta}_k) \quad (10.60)$$

$$\hat{\underline{\Theta}}^* = \hat{\Theta}_{k+1} - \mathbf{H}(\hat{\Theta}_{k+1})^{-1} \cdot \underline{g}(\hat{\Theta}_{k+1}) \quad (10.61)$$

Bei einer quadratischen Fehlerfläche wäre die Hessematrix konstant:

$$\mathbf{H}_{k+1} = \mathbf{H}(\hat{\Theta}_k) = \mathbf{H}(\hat{\Theta}_{k+1}) \quad (10.62)$$

Diese Bedingung ist näherungsweise auch bei nichtlinearen Fehlerflächen erfüllt, da sich das Krümmungsverhalten meist nicht abrupt ändert. Die Differenz aus den beiden Gleichung (10.61) und (10.60) führt mit der Annahme einer konstanten Hessematrix (10.62) zu einer Approximation für die Hessematrix

---

<sup>14)</sup> Hinweis: Durch die notwendige Matrixinversion bei den Direkten Quasi-Newton-Verfahren geht der eigentliche Vorteil dieser Verfahren verloren. Deshalb wird im Folgenden fast ausschließlich das Inverse Quasi-Newton-Verfahren betrachten und auf den Zusatz “invers” verzichtet.

$$\hat{\Theta}_{k+1} - \hat{\Theta}_k = \mathbf{H}_{k+1}^{-1} \cdot \left( \underline{g}(\hat{\Theta}_{k+1}) - \underline{g}(\hat{\Theta}_k) \right), \quad (10.63)$$

da alle übrigen Werte von Gleichung (10.63) bekannt sind. Nach der Multiplikation mit der Hessematrix von links entsteht schließlich die sogenannte *Quasi-Newton-Bedingung*<sup>15)</sup>:

$$\mathbf{H}_{k+1} \cdot \left( \hat{\Theta}_{k+1} - \hat{\Theta}_k \right) = \underline{g}(\hat{\Theta}_{k+1}) - \underline{g}(\hat{\Theta}_k), \quad (10.64)$$

welche sich mit den Abkürzungen (vergleiche Gleichung (10.27))

$$\Delta\Theta_k = \hat{\Theta}_{k+1} - \hat{\Theta}_k \quad (10.65)$$

und

$$\Delta\underline{g}_k = \underline{g}(\hat{\Theta}_{k+1}) - \underline{g}(\hat{\Theta}_k) \quad (10.66)$$

noch einfacher darstellen lässt:

$$\mathbf{H}_{k+1} \cdot \Delta\Theta_k = \Delta\underline{g}_k \quad (10.67)$$

Mit der inversen Hessematrix nach Gleichung (10.58) gilt außerdem der Zusammenhang:

$$\Delta\Theta_k = \mathcal{G}_{k+1} \cdot \Delta\underline{g}_k \quad (10.68)$$

$\mathcal{G}_{k+1}$  und  $\mathbf{H}_{k+1}$  sind Näherungen für die inverse Hessematrix  $\mathcal{Q}(\hat{\Theta}_{k+1})$  und die Hessematrix  $\mathbf{H}(\hat{\Theta}_{k+1})$  am Fehlerflächenpunkt  $\hat{\Theta}_{k+1}$ . Die Hessematrix in der Quasi-Newton-Bedingung von Gleichung (10.64) ist nicht eindeutig. Bei einer positiv definiten Hessematrix sind  $N(N+1)/2$  unbekannte Matrixeinträge vorhanden, zu deren Bestimmung stehen aber lediglich  $N$  Gleichungen zur Verfügung. Das gleiche gilt für die inverse Hessematrix in der Quasi-Newton-Bedingung von Gleichung (10.68). Es gibt also mehrere Möglichkeiten, zu gegebenen Vektoren  $\Delta\underline{g}_k$  und  $\Delta\Theta_k$  eine (inverse) Hessematrix zu finden, welche die Quasi-Newton-Bedingung erfüllt. Eine Bestimmung der (inversen) Hessematrix in einem Schritt ist somit nicht möglich. Die (inverse) Hessematrix kann jedoch iterativ durch mehrere Schritte approximiert werden. Dazu addiert man zur inversen Hessematrix  $\mathcal{G}_k$  einen Korrekturterm  $\Delta\mathcal{G}_k$ :

$$\mathcal{G}_{k+1} = \mathcal{G}_k + \Delta\mathcal{G}_k \quad (10.69)$$

Diese allgemeine Aufdatierungsformel muss die Quasi-Newton-Bedingung enthalten. Setzt man die allgemeine Aufdatierungsformel (10.69) in die Quasi-Newton-Bedingung von Gleichung (10.68) ein, so folgt:

$$\Delta\Theta_k = \mathcal{G}_k \cdot \Delta\underline{g}_k + \Delta\mathcal{G}_k \cdot \Delta\underline{g}_k \quad (10.70)$$

Alle Quasi-Newton-Verfahren erfüllen die Quasi-Newton-Bedingung. Je nach Konstruktion der Aufdatierungsformel entstehen unterschiedliche Verfahren. Die nun folgenden Unterkapitel stellen die bekanntesten und effizientesten Formeln kurz vor.

---

<sup>15)</sup> Hinweis: In der Literatur findet man auch häufig die Bezeichnung *Sekantengleichung* oder *Quasi-Newton-Gleichung* [61, 174].

#### 10.4.4.2 Die Aufdatierungsformel von Broyden

Zunächst soll der Korrekturterm der Aufdatierungsformel (10.69) vom Rang 1 sein:

$$\underline{G}_{k+1} = \underline{G}_k + \alpha \cdot \underline{u} \cdot \underline{u}^T \quad (10.71)$$

Die Korrekturmatrix  $\Delta \underline{G}_k = \alpha \cdot \underline{u} \cdot \underline{u}^T$  entsteht dabei aus einem dyadischen Produkt des noch näher zu bestimmenden Vektors  $\underline{u}$  und der noch unbekannten Konstanten  $\alpha$ . Zur Approximation der (inversen) Hessematrix muss die Quasi-Newton-Bedingung eingehalten werden. Die Rang-Eins-Aufdatierung von Gleichung (10.71) eingesetzt in die Quasi-Newton-Bedingung von Gleichung (10.68) ergibt:

$$\Delta \underline{\Theta}_k = \underline{G}_k \cdot \Delta \underline{g}_k + \alpha \cdot \underline{u} \cdot \underline{u}^T \cdot \Delta \underline{g}_k \quad (10.72)$$

Nach Broyden kann der Vektor  $\underline{u}$  in Gleichung (10.72) zu

$$\underline{u} = \Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k \quad (10.73)$$

und die Konstante  $\alpha$  zu

$$\alpha = \frac{1}{(\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k)^T \cdot \Delta \underline{g}_k} \quad (10.74)$$

gewählt werden [23]. Dies lässt sich durch einfaches Einsetzen der ausgewählten Werte in die Gleichung (10.72) überprüfen:

$$\begin{aligned} \Delta \underline{\Theta}_k &= \underline{G}_k \cdot \Delta \underline{g}_k + \frac{(\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k) \cdot (\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k)^T \cdot \Delta \underline{g}_k}{(\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k)^T \cdot \Delta \underline{g}_k} \\ &= \underline{G}_k \cdot \Delta \underline{g}_k + \Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k \\ &= \Delta \underline{\Theta}_k \end{aligned} \quad (10.75)$$

Damit folgt mit den Gleichungen (10.71), (10.73) und (10.74) für die Aufdatierungsformel nach Broyden:

$$\underline{G}_{k+1} = \underline{G}_k + \frac{(\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k) \cdot (\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k)^T}{(\Delta \underline{\Theta}_k - \underline{G}_k \cdot \Delta \underline{g}_k)^T \cdot \Delta \underline{g}_k} \quad (10.76)$$

#### 10.4.4.3 Die DFP-Aufdatierungsformel

Der Name dieser Aufdatierungsformel geht auf die Entdecker Davidon [34], Fletcher und Powell [54] zurück. Der Korrekturterm ist nun vom Rang 2. Die Aufdatierungsformel lautet:

$$\underline{G}_{k+1} = \underline{G}_k + \alpha \cdot \underline{u} \cdot \underline{u}^T + \beta \cdot \underline{v} \cdot \underline{v}^T \quad (10.77)$$

Die Korrekturmatrixt  $\Delta G_k = \alpha \cdot \underline{u} \cdot \underline{u}^T + \beta \cdot \underline{v} \cdot \underline{v}^T$  ist vom Rang 2 und entsteht aus zwei Matrizen mit Rang 1. Die beiden Vektoren  $\underline{u}$  und  $\underline{v}$  sowie die beiden Konstanten  $\alpha$  und  $\beta$  sind noch näher zu bestimmen. Zur Approximation einer geeigneten inversen Hessematrix ist wieder die Quasi-Newton-Bedingung (10.68) einzuhalten:

$$\Delta \Theta_k = G_k \cdot \Delta g_k + \alpha \cdot \underline{u} \cdot \underline{u}^T \cdot \Delta g_k + \beta \cdot \underline{v} \cdot \underline{v}^T \cdot \Delta g_k \quad (10.78)$$

Wählt man die beiden Konstanten zu

$$\alpha = \frac{1}{\underline{u}^T \cdot \Delta g_k} \quad (10.79)$$

und

$$\beta = -\frac{1}{\underline{v}^T \cdot \Delta g_k} \quad (10.80)$$

so vereinfacht sich die Bedingung von Gleichung (10.78) zu

$$\Delta \Theta_k = G_k \cdot \Delta g_k + \underline{u} - \underline{v}, \quad (10.81)$$

womit für die beiden Vektoren

$$\underline{v} = G_k \cdot \Delta g_k \quad (10.82)$$

und

$$\underline{u} = \Delta \Theta_k \quad (10.83)$$

gelten muss. Mit diesen Konstanten und Vektoren geht die Aufdatierungsformel von Gleichung (10.77) über zu:

$$G_{k+1} = G_k + \frac{\Delta \Theta_k \cdot \Delta \Theta_k^T}{\Delta \Theta_k^T \cdot \Delta g_k} - \frac{G_k \cdot \Delta g_k \cdot \Delta g_k^T \cdot G_k}{(G_k \cdot \Delta g_k)^T \cdot \Delta g_k} \quad (10.84)$$

#### 10.4.4.4 Die BFGS-Aufdatierungsformel

Das BFGS-Verfahren wurde praktisch zeitgleich von Broyden [24], Fletcher [52], Goldfarb [66] und Shanno [212] nach dem DFP-Verfahren entdeckt und gilt zur Zeit als effizientestes Quasi-Newton-Verfahren [219, 174, 61]. Interessant dabei ist, dass alle vier Autoren die BFGS-Aufdatierungsformel auf einem etwas anderen Weg herleiten [61, 219, 3].

Analog zu den Aufdatierungsformeln (10.71) und (10.77) für die Inversen Quasi-Newton-Verfahren ist es auch für die Direkten Quasi-Newton-Verfahren möglich, mit Hilfe einer Korrekturmatrixt vom Rang 1 oder Rang 2 eine neue Matrix  $H_{k+1}$  aus der Matrix  $H_k$  zu bestimmen. Die unbekannten Koeffizienten der Aufdatierungsformeln ermittelt man dann mit Hilfe der Quasi-Newton-Bedingung von Gleichung (10.67) (analog zu Gleichung (10.68) bei den inversen

Verfahren). Die BFGS-Aufdatierungsformeln (also direkt und invers) entstehen aus den DFP-Aufdatierungsformeln, indem das Tripel  $(\mathbf{H}_k, \Delta\underline{\Theta}_k, \Delta\underline{g}_k)$  durch das Tripel  $(\mathcal{G}_k, \Delta\underline{g}_k, \Delta\underline{\Theta}_k)$  ersetzt wird und umgekehrt [61, 97, 174], vergleiche dazu die Gleichungen (10.67) und (10.68). Deshalb bezeichnet man die DFP- und die BFGS-Aufdatierungsformeln als zueinander dual.

Durch Vertauschen von  $\Delta\underline{\Theta}_k$  und  $\Delta\underline{g}_k$  in der inversen DFP-Aufdatierungsformel (10.84) und Ersetzen von  $\mathcal{G}_k$  durch  $\mathbf{H}_k$  entsteht die direkte BFGS-Aufdatierungsformel:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\Delta\underline{g}_k \cdot \Delta\underline{g}_k^T}{\Delta\underline{g}_k^T \cdot \Delta\underline{\Theta}_k} - \frac{\mathbf{H}_k \cdot \Delta\underline{\Theta}_k \cdot \Delta\underline{\Theta}_k^T \cdot \mathbf{H}_k}{(\mathbf{H}_k \cdot \Delta\underline{\Theta}_k)^T \cdot \Delta\underline{\Theta}_k} \quad (10.85)$$

Diese Formel (10.85) muss nun invertiert werden, um die gesuchte inverse BFGS-Aufdatierungsformel zu erhalten. Nach zweimaliger Anwendung der Sherman-Morrison-Woodbury-Formel [174, 97] folgt für die Inverse von Gleichung (10.85):

$$\begin{aligned} \mathcal{G}_{k+1} &= \mathcal{G}_k + \left( 1 + \frac{\Delta\underline{g}_k^T \cdot \mathcal{G}_k \cdot \Delta\underline{g}_k}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \right) \cdot \frac{\Delta\underline{\Theta}_k \cdot \Delta\underline{\Theta}_k^T}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \\ &\quad - \frac{\Delta\underline{\Theta}_k \cdot \Delta\underline{g}_k^T \cdot \mathcal{G}_k + \mathcal{G}_k \cdot \Delta\underline{g}_k \cdot \Delta\underline{\Theta}_k^T}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \end{aligned} \quad (10.86)$$

Eine Gegenrechnung zeigt, dass das Produkt der beiden Aufdatierungsformeln der Gleichungen (10.85) und (10.86) die Einheitsmatrix ergibt. Die Untersuchungen im weiteren Verlauf verwenden nur noch das leistungsfähige inverse BFGS-Verfahren, da es in den meisten Fällen dem DFP-Verfahren überlegen ist [183, 192, 169].

### Zusammenfassung — Das Quasi-Newton-Verfahren (BFGS)

1. Anfangsinitialisierung: Wahl einer positiv definiten inversen Hessematrix  $\mathcal{G}_0$  (z.B.  $\mathcal{G}_0 = \underline{E}$ , Suche startet in die negative Gradientenrichtung) und eines geeigneten Startpunktes  $\hat{\Theta}_0$  auf der Fehlerfläche. Berechnung des Gradienten  $\underline{g}(\hat{\Theta}_0)$ .
2. Newton-Suchrichtung bestimmen (Gleichung (10.55)):

$$\underline{s}_k = -\mathcal{G}_k \cdot \underline{g}(\hat{\Theta}_k)$$

3. Schrittweite berechnen (Liniensuche nach Kapitel 10.2):

$$\eta_k = \min_{\eta} E(\hat{\Theta}_{k+1}) = \min_{\eta} E(\hat{\Theta}_k + \eta \cdot \underline{s}_k)$$

damit gilt  $\Delta\underline{\Theta}_k = \eta_k \cdot \underline{s}_k$ .

4. Optimierungsschritt ausführen:

$$\hat{\underline{\Theta}}_{k+1} = \hat{\underline{\Theta}}_k + \eta_k \cdot \underline{s}_k$$

5. Gradient an der neuen Stelle  $\underline{g}(\hat{\underline{\Theta}}_{k+1})$  ermitteln und damit  $\Delta\underline{g}_k = \underline{g}(\hat{\underline{\Theta}}_{k+1}) - \underline{g}(\hat{\underline{\Theta}}_k)$  bestimmen.
6. Neue Näherung für die inverse Hessematrix berechnen mit der BFGS-Aufdatierungsformel (Gleichung (10.86)):

$$\begin{aligned} \mathcal{G}_{k+1} &= \mathcal{G}_k + \left( 1 + \frac{\Delta\underline{g}_k^T \cdot \mathcal{G}_k \cdot \Delta\underline{g}_k}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \right) \cdot \frac{\Delta\underline{\Theta}_k \cdot \Delta\underline{\Theta}_k^T}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \\ &\quad - \frac{\Delta\underline{\Theta}_k \cdot \Delta\underline{g}_k^T \cdot \mathcal{G}_k + \mathcal{G}_k \cdot \Delta\underline{g}_k \cdot \Delta\underline{\Theta}_k^T}{\Delta\underline{\Theta}_k^T \cdot \Delta\underline{g}_k} \end{aligned}$$

7. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 2 bis 6.

Für die Algorithmen der anderen Quasi-Newton-Verfahren ist lediglich im 6. Schritt die Aufdatierungsformel durch die Gleichung (10.76) für das Broyden oder durch die Gleichung (10.84) für das DFP Verfahren zu ersetzen. Die Quasi-Newton-Verfahren arbeiten durch die Liniensuche stabil und konvergieren superlinear [213, 174], vergleiche Gleichung (10.99). Die DFP-Aufdatierungsformel (10.84) und die BFGS-Aufdatierungsformel (10.86) sehen relativ kompliziert aus. Dennoch ist der Rechenaufwand sehr viel geringer als beim Newton-Verfahren (10.56), das eine Matrixinversion erforderlich macht. Bei genauer Liniensuche liefert die DFP- und die BFGS-Aufdatierungsformel die gleiche Folge von Parametervektoren  $\{\hat{\underline{\Theta}}_k\}$  mit  $E(\hat{\underline{\Theta}}_{k+1}) < E(\hat{\underline{\Theta}}_k)$ . Bei ungenauer Liniensuche erweist sich bei praktischen Aufgabenstellungen das BFGS-Verfahren als robuster [177].

### **Beispiel — Optimierung mit dem Quasi-Newton-Verfahren (BFGS)**

Im Folgenden soll das Quasi-Newton-Verfahren die zu Beginn eingeführte Beispieldatenfläche minimieren. Die Aufdatierung der inversen Hessematrix erfolgt mit der BFGS-Formel von Gleichung (10.86), die Liniensuche übernimmt der in Kapitel 10.2.2 vorgeschlagene ALIS-Algorithmus. Die Optimierung startet vom Anfangspunkt  $\hat{\underline{\Theta}}_0 = [-1.25 \ -1.00]^T$  mit einer Liniensuche in die Newton-Richtung, die beim ersten Optimierungsschritt dem negativen Gradienten entspricht, da die inverse Hessematrix mit der Einheitsmatrix initialisiert wird ( $\mathcal{G}_0 = \mathcal{E}$ ). Diese Berechnungen wurden bereits beim NKG-Beispiel auf Seite 356 durchgeführt mit den Ergebnissen  $\underline{s}_0 = [11.89 \ 1.58]^T$  und  $\eta_0 = 0.178$ . Wie in Gleichung (10.37) lautet somit der erste Optimierungsschritt des BFGS-Verfahrens

$$\begin{aligned}\hat{\underline{\Theta}}_1 &= \hat{\underline{\Theta}}_0 + \eta_0 \cdot \underline{s}_0 = \begin{bmatrix} -1.25 \\ -1.00 \end{bmatrix} + 0.178 \cdot \begin{bmatrix} 11.89 \\ 1.58 \end{bmatrix} = \begin{bmatrix} 0.87 \\ -0.72 \end{bmatrix} \\ \Delta \underline{\Theta}_k &= \eta_0 \cdot \underline{s}_0 = \begin{bmatrix} 2.12 \\ 0.28 \end{bmatrix}\end{aligned}\quad (10.87)$$

Für die Gradientenberechnung an dieser Stelle gilt wie in Gleichung (10.38)

$$\begin{aligned}\underline{g}(\hat{\underline{\Theta}}_1) &= \begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} \\ \Delta \underline{g}_k &= \underline{g}(\hat{\underline{\Theta}}_1) - \underline{g}(\hat{\underline{\Theta}}_0) = \begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} - \begin{bmatrix} -11.89 \\ -1.58 \end{bmatrix} = \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix}\end{aligned}\quad (10.88)$$

Beim Vergleich des Quasi-Newton-Algorithmus (Zusammenfassung auf Seite 373) mit dem Algorithmus des NKG (Zusammenfassung auf Seite 354) fällt auf, dass sich die beiden Verfahren nur in der Berechnung der Suchrichtungen unterscheiden. Alle anderen Punkte der beiden Algorithmen stimmen überein.

Für die Berechnung der neuen Newton-Suchrichtung  $\underline{s}_1$  ist die inverse Hessematrix  $\mathcal{G}_1$  erforderlich, die sich mit der BFGS-Aufdatierungsformel (Gleichung (10.86)) und den Gleichungen (10.87) und (10.88) berechnet zu:

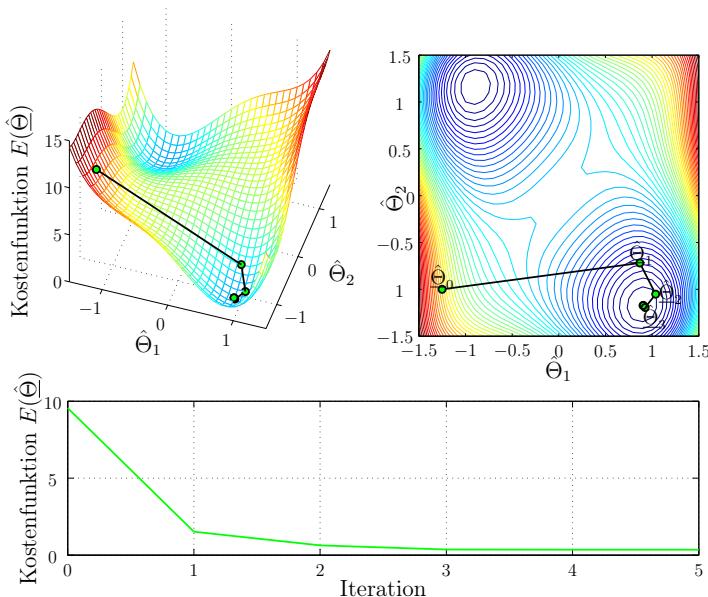
$$\begin{aligned}\mathcal{G}_1 &= \mathcal{G}_0 + \left( 1 + \frac{\Delta \underline{g}_k^T \cdot \mathcal{G}_0 \cdot \Delta \underline{g}_k}{\Delta \underline{\Theta}_k^T \cdot \Delta \underline{g}_k} \right) \cdot \frac{\Delta \underline{\Theta}_k \cdot \Delta \underline{\Theta}_k^T}{\Delta \underline{\Theta}_k^T \cdot \Delta \underline{g}_k} \\ &\quad - \frac{\Delta \underline{\Theta}_k \cdot \Delta \underline{g}_k^T \cdot \mathcal{G}_0 + \mathcal{G}_0 \cdot \Delta \underline{g}_k \cdot \Delta \underline{\Theta}_k^T}{\Delta \underline{\Theta}_k^T \cdot \Delta \underline{g}_k} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \left( 1 + \frac{\begin{bmatrix} 11.33 & 5.79 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix}}{\begin{bmatrix} 2.12 & 0.28 \end{bmatrix} \cdot \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix}} \right) \cdot \frac{\begin{bmatrix} 2.12 \\ 0.28 \end{bmatrix} \cdot \begin{bmatrix} 2.12 & 0.28 \end{bmatrix}}{\begin{bmatrix} 2.12 & 0.28 \end{bmatrix} \cdot \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix}} \\ &\quad - \frac{\begin{bmatrix} 2.12 \\ 0.28 \end{bmatrix} \cdot \begin{bmatrix} 11.33 & 5.79 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix} \cdot \begin{bmatrix} 2.12 & 0.28 \end{bmatrix}}{\begin{bmatrix} 2.12 & 0.28 \end{bmatrix} \cdot \begin{bmatrix} 11.33 \\ 5.79 \end{bmatrix}} \\ &= \begin{bmatrix} 0.408 & -0.433 \\ -0.433 & 0.896 \end{bmatrix}\end{aligned}$$

Damit folgt mit Gleichung (10.55) die neue Suchrichtung

$$\underline{s}_1 = -\mathcal{G}_1 \cdot \underline{g}(\hat{\underline{\Theta}}_1) = - \begin{bmatrix} 0.408 & -0.433 \\ -0.433 & 0.896 \end{bmatrix} \cdot \begin{bmatrix} -0.56 \\ 4.21 \end{bmatrix} = \begin{bmatrix} 2.05 \\ -4.01 \end{bmatrix},$$

welche exakt mit der in Gleichung (10.39) berechneten ersten Suchrichtung des NKG-Beispiels übereinstimmt. Die Minimumssuche entlang dieser neuen Newton-Suchrichtung  $\underline{s}_1$  ergibt den nächsten Gewichtsvektor  $\hat{\underline{\Theta}}_2$ . Analog dazu folgen die

weiteren Optimierungsschritte, welche in der Abbildung 10.27 und in der Tabelle 10.5 zusammengefasst sind.



**Abb. 10.27:** Optimierungsbeispiel mit dem Quasi-Newton-Verfahren — Die Aufdauerung der inversen Hessematrix erfolgt nach der BFGS-Formel. Für die Liniensuche kommt der ALIS-Algorithmus zum Einsatz. Nach nur 5 Iterationen ist das Minimum erreicht.

Schritt $k$	0	1	2	3	4	5
$\hat{\Theta}_k$	-1.25 -1.00	0.87 -0.72	1.04 -1.05	0.92 -1.19	0.91 -1.17	0.90 -1.17
$s_k$	11.886 1.577	2.051 -4.013	-0.349 -0.440	-0.030 0.038	-0.002 -0.002	-
$\eta_k$	0.178	0.083	0.326	0.621	1.037	-
$\eta_{max}$	0.5	0.125	0.5	1	2	-
ALIS	5	6	8	6	6	0

**Tabelle 10.5:** Optimierungsbeispiel mit dem Quasi-Newton-Verfahren — Der ALIS-Algorithmus führt bei der zweiten Liniensuche im Schritt  $k = 1$  eine Intervallverkleinerung und in den folgenden Schritten jeweils eine Intervallvergrößerung durch. In der letzten Zeile steht die Anzahl der für die Liniensuche erforderlichen Funktionsauswertungen.

Die berechneten Zahlen stimmen auf zwei Stellen nach dem Komma mit den Ergebnissen des NKG-Beispiels auf Seite 356 überein, wie die 5 Optimierungsschritte in den Tabellen 10.5 und 10.4 zeigen. Somit sind auch die Bilder 10.27 und 10.23 annähernd gleich. Man beachte, dass die Suchrichtungen in den Tabellen 10.5 und 10.4 unterschiedliche Beträge haben, jedoch immer in die gleiche Richtung weisen. Die unterschiedlichen Beträge führen schließlich dazu, dass der ALIS-Algorithmus bei der BFGS-Optimierung und bei der NKG-Optimierung eine jeweils andere Minimierungsaufgaben zu lösen hat. Für die ALIS-Liniensuche sind bei diesem Beispiel mit BFGS-Optimierung insgesamt 31 Funktionsauswertungen erforderlich. Beim 5. Optimierungsschritt ( $k = 4$ ) stimmt die Taylorapproximation 2. Ordnung sehr genau mit der tatsächlichen Form der Fehlerfläche überein. Dies erkennt man daran, dass die Liniensuche mit  $\eta_k = 1.037$  fast bei eins liegt.

Schritt $k$	0	1	2
$G_k$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.408 & -0.433 \\ -0.433 & 0.896 \end{bmatrix}$	$\begin{bmatrix} 0.094 & 0.055 \\ 0.055 & 0.191 \end{bmatrix}$
Schritt $k$	3	4	5
$G_k$	$\begin{bmatrix} 0.055 & -0.009 \\ -0.009 & 0.090 \end{bmatrix}$	$\begin{bmatrix} 0.048 & 0.005 \\ 0.005 & 0.064 \end{bmatrix}$	$\begin{bmatrix} 0.049 & 0.007 \\ 0.007 & 0.065 \end{bmatrix}$

**Tabelle 10.6:** Approximation der inversen Hessematrix während der Optimierung — Die Näherung startet mit der Einheitsmatrix und stimmt am Ende recht gut mit der wahren inversen Hessematrix am lokalen Minimum (vergleiche Gleichung (10.89)) überein.

Abschließend soll die Aufdatierung der inversen Hessematrix untersucht werden. Tabelle 10.6 zeigt, wie sich die Matrix während der Optimierung entwickelt. Für die Hessematrix gilt am Minimum  $\hat{\Theta}_5 = [0.9 \ -1.17]^T$

$$\mathbf{H}(\hat{\Theta}_5) = \begin{bmatrix} 20.41 & -1.75 \\ -1.75 & 15.37 \end{bmatrix},$$

deren Inverse sich recht einfach berechnen lässt zu [231]:

$$G(\hat{\Theta}_5) = \mathbf{H}(\hat{\Theta}_5)^{-1} = \frac{1}{311} \begin{bmatrix} 15.37 & 1.75 \\ 1.75 & 20.41 \end{bmatrix} = \begin{bmatrix} 0.049 & 0.006 \\ 0.006 & 0.066 \end{bmatrix} \quad (10.89)$$

Bereits beim Schritt  $k = 4$  nähert sich die Approximation  $G_4$  in Tabelle 10.6 der wahren inversen Hessematrix an der Minimalstelle.

#### 10.4.5 Levenberg-Marquardt-Algorithmus

Der *Levenberg-Marquardt-Algorithmus* (LM) [138, 149] ist eine weitere Variante des Newton-Algorithmus. Er ist eine Erweiterung des in Gleichung (10.57)

beschriebenen Gauss-Newton-Verfahrens und ist immer dann anwendbar, wenn die Berechnung der Kostenfunktion nach dem summierten quadratischen Fehler erfolgt. Der LM-Algorithmus verwendet — wie das Gauss-Newton-Verfahren von Gleichung (10.57) — eine Näherung für die Hessematrix. Bei der Bestimmung des Gradienten kommt ebenfalls die Jacobimatrix zum Einsatz.

Das Gauss-Newton-Verfahren (10.57) hat gegenüber dem Newton-Verfahren den Vorteil, dass es keine zweiten Ableitungen der Fehlerfläche benötigt. Jedoch ist es möglich, dass die approximierte Hessematrix  $\mathbf{H} = \mathcal{J}^T \cdot \mathcal{J}$  nicht invertierbar ist. Um dieses Problem zu vermeiden, addiert das LM-Verfahren ein Vielfaches der Einheitsmatrix  $\underline{E}$  zur ursprünglichen Hessematrix:

$${}^*\underline{\mathcal{H}}(\hat{\Theta}_k) = \mathbf{H}(\hat{\Theta}_k) + \mu \cdot \underline{E}, \quad (10.90)$$

wobei mit dem *Skalierungsfaktor*  $\mu$  die Eigenwerte der Hessematrix einstellbar sind, um eine Invertierbarkeit zu erreichen. Sei  $\lambda_i$  der  $i$ -te Eigenwert zum Eigenvektor  $\underline{z}_i$ , so gilt:

$$\begin{aligned} {}^*\underline{\mathcal{H}}(\hat{\Theta}_k) \cdot \underline{z}_i &= (\mathbf{H}(\hat{\Theta}_k) + \mu \cdot \underline{E}) \cdot \underline{z}_i = \mathbf{H}(\hat{\Theta}_k) \cdot \underline{z}_i + \mu \cdot \underline{z}_i \\ &= \lambda_i \cdot \underline{z}_i + \mu \cdot \underline{z}_i = (\lambda_i + \mu) \cdot \underline{z}_i \end{aligned} \quad (10.91)$$

Diese Überlegung zeigt, dass die durch *Skalierung* entstandene Matrix  ${}^*\underline{\mathcal{H}}(\hat{\Theta}_k)$  die gleichen Eigenvektoren  $\underline{z}_i$  besitzt wie die ursprüngliche Matrix  $\mathbf{H}(\hat{\Theta}_k)$ . Die Matrix  ${}^*\underline{\mathcal{H}}(\hat{\Theta}_k)$  hat nach Gleichung (10.91) die Eigenwerte  $(\lambda_i + \mu)$ . Durch Vergrößerung des Skalierungsfaktors  $\mu$  ist es somit möglich, die Eigenwerte so lange zu erhöhen, bis für alle Eigenwerte  $(\lambda_i + \mu) > 0$  erfüllt ist. Die Matrix  ${}^*\underline{\mathcal{H}}(\hat{\Theta}_k)$  ist dann positiv definit und somit invertierbar.

Die Gewichtsanpassung nach LM entsteht durch Einsetzen der skalierten Hessematrix von Gleichung (10.90) in das Gauss-Newton-Verfahren nach Gleichung (10.57) [80, 192, 169]:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \left( \mathcal{J}(\hat{\Theta}_k)^T \cdot \mathcal{J}(\hat{\Theta}_k) + \mu_k \cdot \underline{E} \right)^{-1} \cdot \mathcal{J}(\hat{\Theta}_k)^T \cdot \underline{\varepsilon}(\hat{\Theta}_k) \quad (10.92)$$

Abhängig vom Skalierungsfaktor  $\mu_k$  zeigt der LM-Algorithmus (10.92) ein unterschiedliches Verhalten: Für sehr kleine Werte des Skalierungsfaktors  $\mu_k$  nähert sich das Verfahren dem Gauss-Newton-Algorithmus von Gleichung (10.57) an. Für sehr große Werte hingegen ist der Einfluss der ursprünglichen Hessematrix  $\mathbf{H} = \mathcal{J}^T \cdot \mathcal{J}$  vernachlässigbar und für die Gewichtsanpassung folgt

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k - \frac{1}{\mu_k} \cdot \mathcal{J}(\hat{\Theta}_k)^T \cdot \underline{\varepsilon}(\hat{\Theta}_k), \quad (10.93)$$

was dem aus Gleichung (10.33) bekannten Gradientenabstiegsverfahren entspricht, mit einer Lernschrittweite von  $\eta = 1/\mu_k$ .

Die Anpassung des Skalierungsfaktors  $\mu_k$  in Gleichung (10.92) erfolgt automatisch nach folgender Vorgehensweise: Der Skalierungsfaktor startet mit einem

kleinen Wert (z.B.  $\mu_0 = 0.01$  [192, 79, 169]). Je nach Erfolg der durchgeführten Gewichtsanpassung wird der aktuelle Skalierungsfaktor  $\mu_k$  vergrößert oder verkleinert. War ein Optimierungsschritt erfolgreich, so kann die approximierte Hessematrix (bzw. die Taylorapproximation 2. Ordnung) das aktuelle Gebiet der Fehlerfläche gut beschreiben. Man kann der gefundenen Näherung vertrauen<sup>16)</sup> und den aktuellen Skalierungsfaktors  $\mu_k$  verkleinern. Dies geschieht mit einer Skalierungskonstanten  $\vartheta$  (z.B.  $\vartheta = 10$  [192, 79, 169]):

$$\mu_{k+1} = \frac{\mu_k}{\vartheta} \quad (10.94)$$

War hingegen die Gewichtsanpassung nicht erfolgreich — das heißt der Optimierungsschritt führte zu keiner Verbesserung ( $E(\hat{\underline{\Theta}}_k) \geq E(\hat{\underline{\Theta}}_{k-1})$ ) — so stimmt die ermittelte Taylorapproximation 2. Ordnung nicht mit der tatsächlichen Fehlerfläche überein. Es erfolgt eine Vergrößerung des aktuellen Skalierungsfaktors  $\mu_k$  durch Multiplikation mit der Skalierungskonstanten  $\vartheta$ :

$$\mu_{k+1} = \mu_k \cdot \vartheta \quad (10.95)$$

Bei einem größer werdenden Skalierungsfaktor nutzt das LM-Verfahren die Information aus der ungenauen Taylorapproximation 2. Ordnung immer weniger und verwendet das garantiert konvergierende Gradientenabstiegsverfahren. Der hier zum Einsatz kommende Gradientenabstieg verfügt sogar über eine Art Schrittweitensteuerung, da durch die Skalierung die Schrittweite sukzessive zurückgenommen wird.

### Zusammenfassung — Levenberg-Marquardt-Verfahren

1. Anfangsinitialisierung: Wahl eines Skalierungsfaktors  $\mu_0$ , einer Skalierungskonstanten  $\vartheta$  und eines geeigneten Startpunktes  $\hat{\underline{\Theta}}_0$  auf der Fehlerfläche (in diesem Buch gilt für alle Optimierungen:  $\mu_0 = 3$  und  $\vartheta = 10$ ).
2. Jacobimatrix berechnen:

$$\underline{J}(\hat{\underline{\Theta}}_k) = \frac{\partial \underline{e}(\hat{\underline{\Theta}}_k)}{\partial \hat{\underline{\Theta}}}$$

3. Gewichtsänderung ermitteln (Gleichung (10.92)):

$$\hat{\underline{\Theta}}_{k+1} = \hat{\underline{\Theta}}_k - \left( \underline{J}(\hat{\underline{\Theta}}_k)^T \cdot \underline{J}(\hat{\underline{\Theta}}_k) + \mu_k \cdot \underline{E} \right)^{-1} \cdot \underline{J}(\hat{\underline{\Theta}}_k)^T \cdot \underline{e}(\hat{\underline{\Theta}}_k)$$

4. Kostenfunktionswert  $E(\hat{\underline{\Theta}}_{k+1})$  an der neuen Position  $\hat{\underline{\Theta}}_{k+1}$  bestimmen (summiert quadratischer Fehler notwendig).

---

<sup>16)</sup> Hinweis: In der englischsprachigen Literatur findet man deshalb häufig die Bezeichnung *region of trust* oder auch *trusted region algorithm* [192, 18].

5. Validierung und Skalierung:

$$E(\hat{\Theta}_{k+1}) < E(\hat{\Theta}_k) : \quad \rightarrow \text{Optimierungsschritt erfolgreich}$$

$\rightarrow$  Gewichtsänderung beibehalten

$$\rightarrow \mu_{k+1} = \frac{\mu_k}{\vartheta}$$

$\rightarrow$  weiter mit Schritt 2

$$E(\hat{\Theta}_{k+1}) \geq E(\hat{\Theta}_k) : \quad \rightarrow \text{Optimierungsschritt fehlgeschlagen}$$

$\rightarrow$  Gewichtsänderung rückgängig machen

$$\rightarrow \mu_{k+1} = \mu_k \cdot \vartheta$$

$\rightarrow$  weiter mit Schritt 3

6. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 2 bis 5.

Insgesamt ist der LM-Algorithmus ein Verfahren, das sehr geschickt zwischen der schnellen Konvergenz des Gauss-Newton-Verfahrens (etwa bei einer sehr guten quadratischen Beschreibung der Fehlerfläche in der Nähe eines Minimums) und der garantierten Konvergenz des Gradientenabstiegs (bei einer ungeeigneten quadratischen Fehlerflächenapproximation, welche besonders bei weiter Entfernung vom Minimum möglich ist) variiert. Durch die notwendige Matrixinversion ist das Verfahren jedoch nur für kleinere Modelle geeignet.

## 10.5 Zusammenfassung: Deterministische Optimierungsverfahren

Die in diesem Kapitel beschriebenen Optimierungsverfahren sind lokale Abstiegsverfahren. Das heißt, sie starten die Minimumssuche von einem einzigen Punkt der Fehlerfläche und generieren eine Folge von Parametervektoren  $\{\hat{\Theta}_k\}$  mit  $E(\hat{\Theta}_{k+1}) < E(\hat{\Theta}_k)$ , bis eine Abbruchbedingung erfüllt ist. Während der Optimierung werten die Algorithmen unterschiedliche Eigenschaften der Fehlerfläche aus. Die Verfahren 0. Ordnung berechnen den Wert der Kostenfunktion, führen jedoch keine Ableitungsberechnung durch. Während die Verfahren 1. Ordnung die erste Ableitung der Kostenfunktion verwenden, nutzen die Verfahren 2. Ordnung zusätzlich direkt oder indirekt die zweite Ableitung der Kostenfunktion. Bei den Verfahren 2. Ordnung muss nur beim original Newton-Verfahren eine direkte Berechnung der zweiten Ableitung erfolgen, alle anderen Verfahren approximieren diese Information mit zunehmender Genauigkeit während der Optimierung.

Für die Bewertung der Optimierungsverfahren eignen sich mehrere Kriterien:

- Konvergenz(geschwindigkeit) der Parameter
- Rechen- und Speicheraufwand
- Aufwand bei der Implementierung

Die folgenden Ausführungen vergleichen die in diesem Kapitel vorgestellten Optimierungsverfahren im Hinblick auf diese Bewertungskriterien.

### 10.5.1 Konvergenz der Parameter

Bei der Parameterkonvergenz gibt es einen theoretischen und einen eher praktischen bzw. problembezogenen Aspekt. Die theoretische Betrachtungsweise behandelt die folgenden Ausführungen. Sie untersuchen allgemein die lokale *Konvergenzrate* eines Optimierungsverfahrens. Diese wird auch als Ordnung der Konvergenz bezeichnet [231]. Die praktische bzw. problembezogene Betrachtungsweise behandelt das Kapitel 10.6. Dabei interessiert die Zeit, die ein Optimierungsverfahren für die Lösung eines vorgegebenen Problems benötigt. Im Falle der Systemidentifikation ist das die Zeit, die ein Optimierungsverfahren braucht, um ein Modell mit einer gewünschten Genauigkeit zu finden.

Ein Optimierungsalgorithmus konvergiert, falls er eine Folge von Parametervektoren  $\{\hat{\underline{\Theta}}_k\}$  liefert, mit

$$\lim_{k \rightarrow \infty} \hat{\underline{\Theta}}_k = \hat{\underline{\Theta}}^* \quad (10.96)$$

wobei  $\hat{\underline{\Theta}}^*$  ein globales oder lokales Minimum der Fehlerfläche darstellt. Die Konvergenz ist von der Ordnung  $o$ , wenn mit einer Konstanten  $K > 0$  der folgende Zusammenhang gilt [231]:

$$\|\hat{\underline{\Theta}}_{k+1} - \hat{\underline{\Theta}}^*\| < K \cdot \|\hat{\underline{\Theta}}_k - \hat{\underline{\Theta}}^*\|^o \quad (10.97)$$

Bei den Optimierungsverfahren unterscheidet man die folgenden Spezialfälle bei der Konvergenz [97, 231]: Ein Optimierungsverfahren konvergiert *lokal linear*, falls

$$\|\hat{\underline{\Theta}}_{k+1} - \hat{\underline{\Theta}}^*\| < K \cdot \|\hat{\underline{\Theta}}_k - \hat{\underline{\Theta}}^*\|^1 \quad (10.98)$$

zutrifft, die Konvergenz ist *lokal superlinear*, wenn

$$\|\hat{\underline{\Theta}}_{k+1} - \hat{\underline{\Theta}}^*\| < K \cdot \|\hat{\underline{\Theta}}_k - \hat{\underline{\Theta}}^*\|^{1+\alpha} \quad (10.99)$$

für ein kleines  $\alpha$  mit  $0 < \alpha < 1$  erfüllt ist. Die schnellstmögliche Konvergenzrate bei der nichtlinearen Optimierung erreicht der Newton-Algorithmus von Kapitel 10.4.3. Er konvergiert *lokal quadratisch*, so dass die folgende Beziehung angegeben werden kann [169]:

$$\|\hat{\underline{\Theta}}_{k+1} - \hat{\underline{\Theta}}^*\| < K \cdot \|\hat{\underline{\Theta}}_k - \hat{\underline{\Theta}}^*\|^2 \quad (10.100)$$

Alle anderen Verfahren 2. Ordnung konvergieren lokal superlinear. Der Gradientenabstieg von Kapitel 10.3 konvergiert lediglich linear.

### 10.5.2 Rechen- und Speicheraufwand

Beim Gradientenabstieg, bei den NKG-Verfahren und beim SKG steigt der Rechen- und Speicheraufwand jeweils linear mit der Netzgröße ( $\mathcal{O}(N)$ ). Alle Newton-Verfahren — also der original Newton-Algorithmus, die Quasi-Newton-Verfahren und LM — müssen die Hessematrix ablegen, was einen quadratischen Speicheraufwand zur Folge hat ( $\mathcal{O}(N^2)$ ). Der Rechenaufwand bei den Quasi-Newton-Verfahren nimmt — aufgrund der Matrixmultiplikation — quadratisch mit der Netzgröße zu ( $\mathcal{O}(N^2)$ ). Dies führt dazu, dass bei großen Modellen die Quasi-Newton-Verfahren deutlich mehr Rechenleistung für eine Iteration benötigen als die NKG-Verfahren. Noch ungünstiger schneiden bei großen Modellen die Verfahren ab, bei denen die Hessematrix invertiert werden muss. Die Matrixinversion geht mit einem kubischen Rechenaufwand ein ( $\mathcal{O}(N^3)$ ) und findet beim original Newton- und beim LM-Algorithmus statt. Die hier beschriebenen Zusammenhänge und noch weitere Details findet man in [169, 183, 18, 155]. Diese Überlegungen sagen nichts über den Erfolg einer Optimierung bzw. die Leistungsfähigkeit eines Optimierungsverfahrens bei einer bestimmten Anwendung aus. Wie die aufwändigen Studien in [44] zeigen, arbeiten gerade die rechenintensiven Verfahren besonders effektiv und führen so mit wenigen Iterationen schon zu guten Ergebnissen.

### 10.5.3 Aufwand bei der Implementierung

Der Gradientenabstieg lässt sich mit Abstand am einfachsten implementieren. Das ist jedoch der einzige Vorteil dieses — bei der Systemidentifikation mit Neuronalen Netzen immer noch beliebten — Verfahrens. Ohne der Subroutine Liniensuche sind die Verfahren mit Skalierung aufwändiger zu implementieren als NKG und Quasi-Newton. Schließt man die Liniensuche mit ein, so ist die Situation genau umgekehrt.

### 10.5.4 Ergebnisse des Optimierungsbeispiels

Das in diesem Kapitel untersuchte Optimierungsbeispiel diente in erster Linie der Veranschaulichung der einzelnen Verfahren. Dennoch lassen sich einige wichtige Erkenntnisse zusammenfassen: Der Gradientenabstieg arbeitet weniger effektiv als die Verfahren 2. Ordnung. Eine Konvergenz ist nur bei ausreichend kleiner Schrittweite gewährleistet. Bei zu groß gewählter Schrittweite entsteht oszillierendes Verhalten oder sogar Instabilität. Bei dem betrachteten Beispiel haben NKG nach Polak-Ribi  re und Quasi-Newton nach BFGS einen fast identischen Optimierungsverlauf und können das Beispiel am besten lösen. Beim SKG empfiehlt M  ller, den Wert von  $\sigma$  so klein zu wählen, wie es die Rechengenauigkeit zulässt. Außerdem behauptet M  ller, dass die Wahl des Parameters  $\sigma$  nicht ausschlaggebend ist für die Konvergenz des Optimierungsalgorithmus [155, 154]. Beides

kann mit dem untersuchten Beispiel nicht beobachtet werden: Der Optimierungserfolg ist vom gewählten  $\sigma$  abhängig. Außerdem funktioniert die Optimierung des zweidimensionalen Beispiels für  $\sigma = 0.1$  deutlich besser als bei kleineren Werten. Die direkte Implementierung des Newton-Verfahrens hat das Problem, dass die Optimierung auch zu einem Sattelpunkt oder Maximum der Fehlerfläche führen kann. Aus diesem Grund ist das Newton-Verfahren in seiner Originalform nicht für die Minimumssuche geeignet.

Die Verfahren NKG und Quasi-Newton benötigen in jeder Iteration eine Liniensuche. Der Optimierungserfolg dieser Verfahren hängt deutlich vom Liniensuchergebnis und damit von den benutzerdefinierten Einstellungen beim Liniensuchalgorithmus ab [18]. Der in diesem Kapitel beschriebene ALIS-Algorithmus passt sich während der Optimierung dem Problem automatisch an und kommt damit ohne benutzerabhängige Parameter aus. Dies erleichtert die Anwendung der NKG und Quasi-Newton-Verfahren erheblich und vermeidet Einstellungsfehler. Die Liniensuche muss bei den Quasi-Newton Verfahren nicht so genau arbeiten wie bei den NKG-Verfahren [18, 169]. Beim NKG-Algorithmus ist eine exakte Liniensuche essentiell, damit ein korrektes System aus konjugierten Suchrichtungen und orthogonalen Gradienten entstehen kann [214][44].

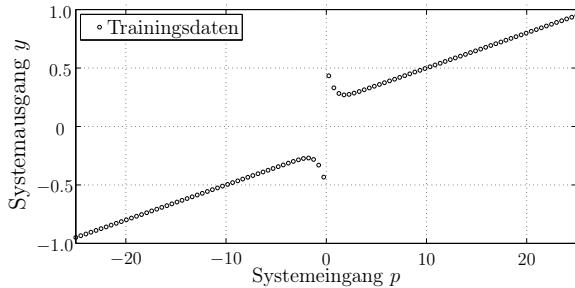
## 10.6 Identifikationsbeispiele

Die Systemidentifikation mit Neuronalen Netzen führt meist zu einem nichtlinearen Parameteroptimierungsproblem. Wie erfolgreich die einzelnen Optimierungsverfahren bei der Lösung dieser Aufgabe sind, lässt sich nur experimentell bestimmen [213]. Aus diesem Grund werden in diesem Unterkapitel zwei Identifikationsbeispiele gezeigt, wobei alle zuvor eingeführten Optimierungsverfahren jeweils das gleiche Problem lösen müssen.

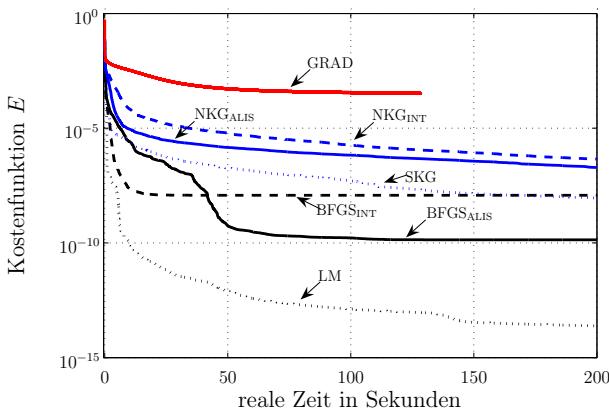
### 10.6.1 Identifikation einer statischen Reibkennlinie

Bei der ersten Untersuchung wird die in Abbildung 10.28 dargestellte Reibkennlinie mit einem 1-4-4-1-MLP ( $N = 33$  Gewichte) identifiziert. Die Reibkennlinie ist in Form von 100 äquidistant verteilten Trainingsdaten im Bereich  $[-25 \dots 25]$  gegeben.

Abbildung 10.29 stellt die Konvergenzverläufe während der Identifikation dar. Der überlegene LM-Algorithmus kann das Optimierungsproblem am schnellsten lösen und findet ein Modell mit sehr kleinem Fehlerfunktionswert. Die Newton-Verfahren (also LM und die beiden BFGS-Verfahren) finden deutlich schneller eine Lösung als die Konjugierten Gradientenverfahren (also SKG und die beiden NKG-Verfahren). Die ALIS-Liniensuche verläuft erfolgreicher als die INT-Liniensuche. Der einfache Gradientenabstieg ist für die Identifikation mit Neuronalen Netzen die schlechteste Wahl.



**Abb. 10.28:** Reibkennlinie, gegeben in Form von 100 äquidistant verteilten Trainingsdaten.

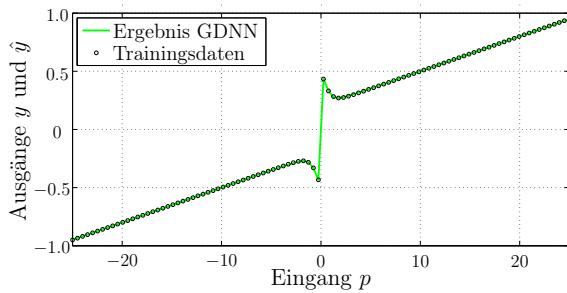


**Abb. 10.29:** Identifikation der Reibkennlinie von Abbildung 10.28 mit einem 1-4-4-1-MLP.

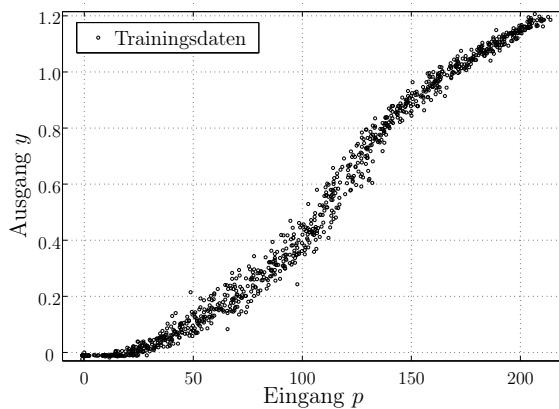
Abbildung 10.30 zeigt, wie gut das MLP-Modell die Reibkennlinie nach einer LM-Identifikation von nur 0.5 Sekunden wiedergeben kann (der Fehler beträgt dabei  $E = 10^{-6}$ ).

### 10.6.2 Identifikation von stark verrauschten Messdaten

Analog zur Untersuchung der statischen Reibkennlinie in Kapitel 10.6.1 folgt nun die Identifikation von 1000 verrauschten Messdaten, welche in Abbildung 10.31 dargestellt sind. Bei der Identifikation wird ein 1-5-5-1-MLP verwendet. Zur Einstellung der 46 Gewichte kommen wieder alle besprochenen Optimierungsverfahren zum Einsatz. Einen Vergleich der Konvergenzverläufe sieht man in Abbildung 10.32. Durch den Rauschanteil erreicht der Fehler keine kleineren Werte als  $E = 10^{-3}$ . Der eigentliche Modellfehler (ohne Rauschen) nimmt bei der

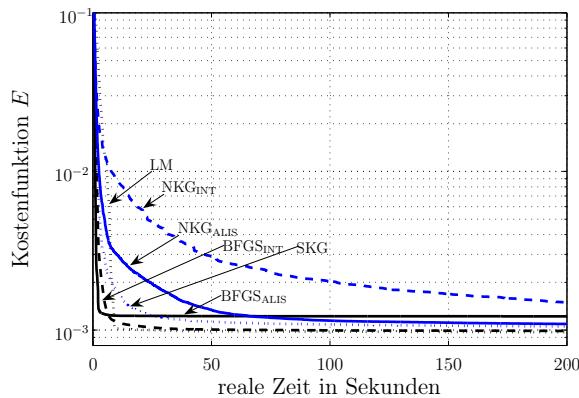


**Abb. 10.30:** Validierung der Reibkennlinie nach einer LM-Identifikation von 0.5 Sekunden.

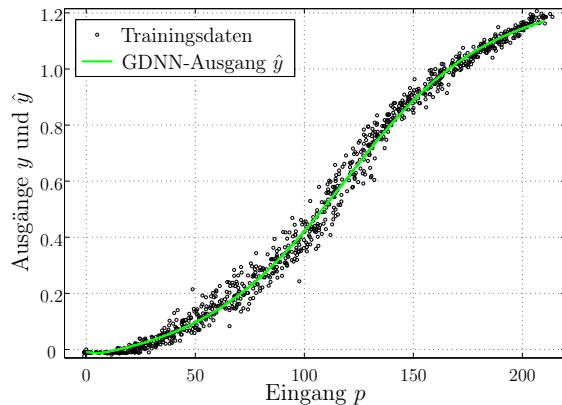


**Abb. 10.31:** Gegeben: 1000 verrauschte Trainingsdaten.

Identifikation jedoch kleinere Werte als  $E = 10^{-3}$  an und es entstehen brauchbare Modelle, wie in Abbildung 10.33 veranschaulicht. Abbildung 10.33 zeigt, wie gut das MLP-Modell die Reibkennlinie nach einer LM-Identifikation von nur 3 Sekunden wiedergeben kann (der Fehler dabei beträgt  $E = 1.1 \cdot 10^{-3}$ ).



**Abb. 10.32:** Konvergenzverläufe bei der Identifikation der verrauschten Messdaten von Abbildung 10.31 mit einem 1-5-5-1-MLP.



**Abb. 10.33:** Validierung der identifizierten Kennlinie mit den gegebenen Messdaten nach einer LM-Identifikation von 3 Sekunden.

# 11 Stochastische Optimierungsverfahren

Das Kapitel 10 behandelte die lokalen Minimumsuche auf der Fehlerfläche. Dabei besteht bei allen lokalen bzw. deterministischen Verfahren das Problem, dass die Optimierung vorzeitig in einem schlechten lokalen Minimum enden kann. Eine gängige Praxis zur Vermeidung schlechter lokaler Minima bei der Identifikation ist, ein lokales Lernverfahren an mehreren Punkten der Fehlerfläche (also mit unterschiedlichen Anfangsinitialisierungen) zu starten und anschließend das beste Ergebnis auszusuchen [82, 79, 169]. Dieses Vorgehen löst jedoch nicht das eigentliche Problem der lokalen Lernverfahren. Dieses Kapitel untersucht deshalb stochastische Optimierungsverfahren, welche gezielt Zufallsgrößen einsetzen, um global die beste Lösung zu finden. Im Vergleich zu den in Kap. 10 beschriebenen lokalen Optimierungsverfahren, die deterministischer Natur sind und von einem Startpunkt aus immer den gleichen Weg auf der Fehlerfläche berechnen, sind die Prozesse bei den stochastischen Verfahren nicht reproduzierbar.

Die in diesem Kapitel behandelten stochastischen Algorithmen orientieren sich alle an Phänomenen der Natur: Die Grundidee beim Simulated Annealing (Kap. 11.1) ist der Abkühlungsprozess von geschmolzenen Festkörpern. Bei den Evolutionären Algorithmen handelt es sich um mathematische Modelle, welche die Evolution des Lebens simulieren — Kap. 11.2 beschreibt mit den Evolutionsstrategien einen Vertreter aus dieser Gruppe. Unter Swarm Intelligence versteht man Verfahren, die das Schwarmverhalten von Tieren nachahmen. Stellvertretend für diese Gattung stellt das Kap. 11.3 das Particle Swarm Optimization Verfahren vor. Nach dieser allgemeinen Einführung in die stochastischen Algorithmen folgt in Kap. 11.4 der Übergang zur Systemidentifikation. Dabei stellt sich heraus, dass die global agierenden Verfahren nur bedingt für die Systemidentifikation mit Neuronalen Netzen geeignet sind.

## 11.1 Simulated Annealing

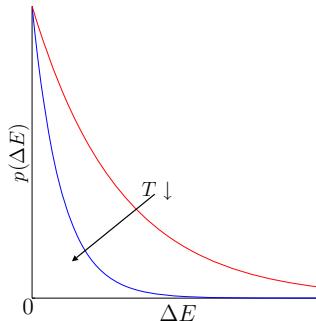
### Physikalischer Hintergrund

Kühlt man einen geschmolzenen Festkörper ab, so ändert sich der Aggregatzustand von flüssig auf fest. Je nach Geschwindigkeit der Abkühlung treten dabei unterschiedliche Formen der Kristallisation auf, welche mit verschiedenen Energieniveaus verbunden sind. Eine reine Kristallstruktur bedeutet das Erreichen

eines absoluten Energieminimums und ist nur durch sehr langsames Abkühlen möglich. Zu schnelles Abkühlen führt zu einer unreinen Kristallstruktur und damit zu einem Verharren in einem lokalen Energieminimum. Die Teilchen in der Schmelze belegen nicht nur die jeweils energetisch günstigste Position. Mit einer bestimmten temperaturabhängigen Wahrscheinlichkeit nehmen die Teilchen auch um  $\Delta E$  energetisch schlechtere Positionen ein. Diese Wahrscheinlichkeit beschreibt die folgende Boltzmann-Verteilung [63]

$$p(\Delta E, T) = g \cdot \exp\left(-\frac{\Delta E}{k_B T}\right) \quad (11.1)$$

dargestellt in Abbildung 11.1. Die Variable  $g$  bezeichnet dabei das statistische



**Abb. 11.1:** Boltzmannverteilung suboptimaler Teilchenenergien — Bei abnehmender Temperatur sinkt die Wahrscheinlichkeit, dass ein Teilchen eine um  $\Delta E$  schlechtere Positionen einnimmt.

Gewicht eines bestimmten Energieniveaus und  $k_B$  die Boltzmann-Konstante. Mit abnehmender Temperatur nimmt die Streuung der Energieniveaus ab und immer mehr energetisch günstige Zustände gewinnen die Oberhand, bis schließlich für  $T \rightarrow 0$  nur noch das niedrigste Energieniveau möglich ist.

### Umsetzung in einen Optimierungsalgorithmus

Die Ursprünge des Simulated Annealing Algorithmus gehen bis in die Fünfziger Jahren des vergangenen Jahrhunderts zurück. Metropolis entwickelte einen Algorithmus zur simulativen Nachbildung des beschriebenen thermodynamischen Prozesses [151], auf dessen Grundlage Kirkpatrick et al. [124] und Cerny [32] in den Achtziger Jahren unabhängig voneinander Simulated Annealing als stochastisches Optimierungsverfahren vorstellten.

Die Grundidee dabei war, eine zufällige Suchbewegung im Lösungsraum durch einen Steuerungsparameter so zu beeinflussen, dass zunächst weite Gebiete möglicher Lösungen untersucht werden und im Verlauf des Optimierungsprozesses zunehmend bevorzugt Schritte ausgeführt werden, die zu einer Verbesserung der Zielfunktion führen. Da lange Zeit auch verschlechternde Schritte zugelas-

sen werden, ergibt sich die Möglichkeit, erreichte lokale Minima wieder verlassen zu können, um letztlich das globale Minimum zu finden.

In Anlehnung an die Abkühlung eines geschmolzenen Festkörpers steuert eine virtuelle Temperatur  $T_k$  (zum Optimierungsschritt  $k$ ) den Optimierungsprozess. Ausgehend von einem zufällig initialisierten Punkt  $\underline{\Theta}_0$  auf der Fehlerfläche wird bei jeder Iteration ein normalverteilt zufälliger Schritt  $\Delta \hat{\Theta}_k = \underline{z}_k(\sigma)$  probeweise ausgeführt und auf die Veränderung des Fehlerflächenwertes hin untersucht ( $\underline{z}_k(\sigma)$  ist hierbei ein Vektor aus normalverteilten Zufallszahlen mit Mittelwert 0 und Standardabweichung  $\sigma$ ). Bei einer Verbesserung des Kostenfunktionswertes  $\Delta E_k = E(\hat{\Theta}_k + \Delta \hat{\Theta}_k) - E(\hat{\Theta}_k) < 0$  wird der Schritt immer ausgeführt, bei einer Verschlechterung  $\Delta E_k > 0$  nur mit einer von der sich langsam verringernden virtuellen Temperatur  $T_k$  abhängigen Wahrscheinlichkeit gemäß<sup>1)</sup>

$$p_k = \exp\left(-\frac{\Delta E_k}{T_k}\right) \quad \text{mit} \quad \Delta E_k > 0 \quad (11.3)$$

Für sehr große Temperaturen  $T_k \rightarrow \infty$  führt der Algorithmus Verschlechterungen mit einer Wahrscheinlichkeit von  $p_k \rightarrow 1$  weitgehend unabhängig von  $\Delta E_k$  aus. Bei abnehmender Temperatur sinkt die Wahrscheinlichkeit für große Verschlechterungen (siehe Abbildung 11.1). Für sehr kleine Temperaturen  $T_k \rightarrow 0$  werden Verschlechterungen mit einer Wahrscheinlichkeit von  $p_k \rightarrow 0$  nicht mehr berücksichtigt.

Der Simulated Annealing Algorithmus sucht also zunächst bei hoher virtueller Temperatur die Fehlerfläche durch stochastisches Umherwandern ab, bis mit zunehmender Abkühlung langsam immer mehr (und am Ende ausschließlich) Schritte zugelassen werden, die zu einer Verbesserung führen.

### Kühlschema

Eine entscheidende Bedeutung kommt einer geeigneten Verringerung der virtuellen Temperatur im Laufe des Optimierungsprozesses zu. Die Folge der Temperaturveränderungen wird als *Kühlschema* bezeichnet. Häufig verwendete Kühl schemata sind [62]:

- Arithmetische Abkühlung:  $T_{k+1} = T_k - T_{red}$ , mit einer konstanten Temperaturreduktion  $T_{red}$
- Geometrische Abkühlung:  $T_{k+1} = \kappa \cdot T_k$ , mit einem konstanten Abkühl faktor  $\kappa < 1$  (z.B.  $\kappa = 0.99$ )

---

<sup>1)</sup> Hinweis: Zur Berechnung der Ausführungswahrscheinlichkeit von Schritten im Lösungsraum kann alternativ auch die Fermifunktion

$$p_k = \frac{1}{1 + \exp\left(\frac{\Delta E_k}{T_k}\right)}$$

Verwendung finden [27, 222], die ein ähnliches Verhalten des Algorithmus bewirkt und gleichermaßen auf verbessernde und verschlechternde Schritte anwendbar ist.

## Zusammenfassung — Simulated Annealing

1. Anfangsinitialisierung: Wahl einer geeigneten Standardabweichung  $\sigma$  (bestimmt die Größenordnung der Zufallsschritte), eines großen Wertes für die anfängliche virtuelle Temperatur  $T_0$ , eines passenden Abkühlfaktors  $\kappa < 1$ , eines sinnvollen Suchbereichs und eines Startpunktes  $\underline{\Theta}_0$  auf der Fehlerfläche. Berechnung des Kostenfunktionswertes  $E(\underline{\Theta}_0)$  am Startpunkt.
2. Vektor  $\underline{z}_k$  aus normalverteilten Zufallszahlen mit Streuung  $\sigma$  generieren. Damit berechnet sich die zufällige Positionsänderung zu:

$$\hat{\underline{\Theta}}_{k+1} = \underline{\Theta}_k + \underline{z}_k$$

Kontrolle, ob der Suchbereich verlassen wird, falls nicht: Kostenfunktion an dieser zufälligen Stelle  $E(\hat{\underline{\Theta}}_{k+1})$  berechnen.

3. Verbesserung/Verschlechterung der zufälligen Positionsänderung ermitteln

$$\Delta E_k = E(\hat{\underline{\Theta}}_{k+1}) - E(\underline{\Theta}_k)$$

und auswerten:

$\Delta E_k \leq 0$  : Zufällige Positionsänderung ausführen ( $p_k = 1$ )

$\Delta E_k > 0$  : Berechnung der Ausführungswahrscheinlichkeit (Gleichung (11.3)) für die zufällig bestimmte Positionsänderung:

$$p_k = \exp\left(-\frac{\Delta E_k}{T_k}\right)$$

Erzeugung einer gleichverteilten Zufallszahl  $r_k \in [0 \ 1]$

Falls  $r_k \leq p_k$  erfüllt ist: Ausführen der zufälligen Positionsveränderung

Ansonsten: Zufällige Positionsveränderung rückgängig

machen  $\hat{\underline{\Theta}}_{k+1} = \underline{\Theta}_k$

4. Abkühlung der Temperatur gemäß Kühlschema:  $T_{k+1} = \kappa \cdot T_k$
5. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 2 bis 4.

Neben dem hier beschriebenen Basisalgorithmus gibt es noch viele weitere Varianten von Simulated Annealing. Um die Rechenzeit zu verkürzen, wurde in [222] das Fast Simulated Annealing vorgeschlagen. Dabei sind die zufälligen Gewichtsänderungen nicht gaußverteilt, wie beim klassischen Simulated Annealing, sondern gemäß einer Cauchy-Lorentz-Verteilung vorzunehmen. Dueck et al. entwickelte mit dem Treshold Accepting und dem Sinflut-Algorithmus vereinfachte Varianten des Simulated Annealing. Näheres dazu findet man in [38, 39, 62, 172].

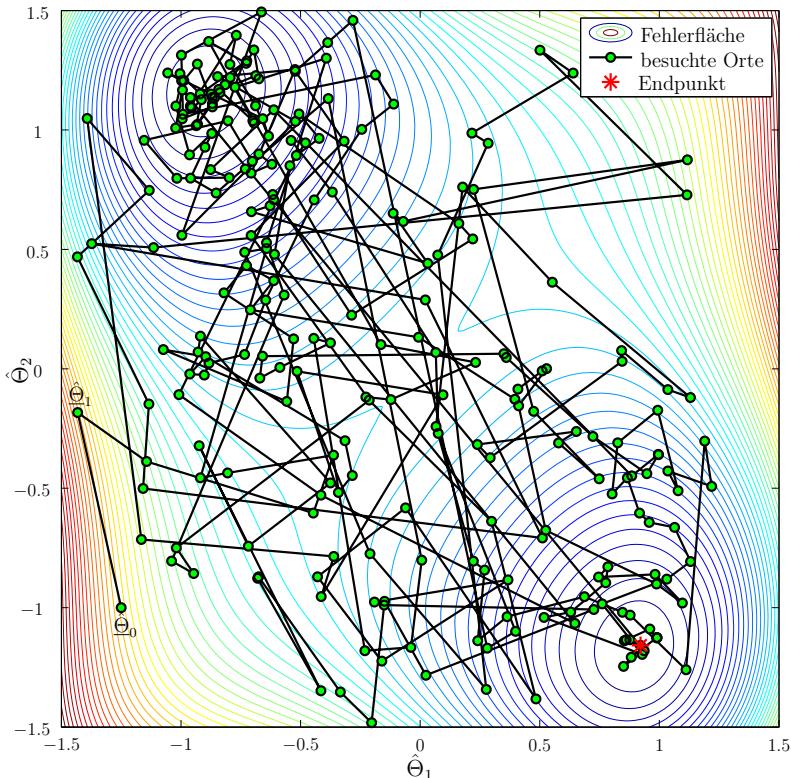
### Beispiel — Optimierung mit Simulated Annealing

Dieser Abschnitt zeigt ein einfaches Optimierungsbeispiel mit dem Verfahren Simulated Annealing. Als zu optimierende Fehlerfläche dient wieder die in Kap. 10.1 eingeführte Funktion. Abgesucht werden soll lediglich ein Parameterbereich von  $-1.5$  bis  $+1.5$ . In diesem Bereich befinden sich die beiden globalen Minima. Für die passende Einstellung der benutzerdefinierten Parameter sind einige Vorversuche erforderlich, da es keine allgemeingültigen Einstellregeln gibt. Stochastische Optimierungsverfahren reagieren meist empfindlich auf Veränderungen in den benutzerdefinierten Parametern. In dem hier vorgestellten Beispiel eignen sich normalverteilte Zufallszahlen mit einer Streuung von  $\sigma = 0.5$  für die zufällige Positionsänderung. Die anfängliche virtuelle Temperatur  $T_0 = 10$  wird durch einen Abkühlfaktor von  $\kappa = 0.99$  angemessen reduziert. Die Optimierung startet beim Punkt  $\hat{\Theta}_0 = [-1.25 \ -1.00]^T$ . Abbruchbedingung des Optimierungsbeispiels ist das Erreichen von 500 Iterationen.

Der Optimierungsverlauf zusammen mit dem Höhenlinienbild ist in Abbildung 11.2 dargestellt. Schon beim ersten Optimierungsschritt erkennt man, dass es sich hier um ein stochastisches Optimierungsverfahren handeln muss, da der erste Schritt mit  $E(\hat{\Theta}_1) > E(\hat{\Theta}_0)$  zu einem schlechteren Kostenfunktionswert führt (wohingegen lokal deterministische Verfahren nur Abstiegsrichtungen ansteuern). Durch die hohe virtuelle Temperatur am Anfang der Optimierung (vergleiche den oberen Verlauf von Abbildung 11.3) führt der Simulated Annealing Algorithmus Verschlechterungen mit einer relativ hohen Wahrscheinlichkeit (mittleres Bild von Abbildung 11.3) aus. Beim Optimierungsverlauf in Abbildung 11.2 ist auffällig, dass zunächst verstärkt das globale Minimum  $\hat{\Theta} = [-0.90 \ 1.17]^T$  untersucht wird, bevor sich schließlich die Optimierung auf das globale Minimum bei  $\hat{\Theta} = [0.90 \ -1.17]^T$  konzentriert.

Der Verlauf der virtuellen Temperatur in Abbildung 11.3 startet bei  $T_0 = 10$  und endet bei  $T_{500} = \kappa^{500} \cdot T_0 = 0.0066$ . Im mittleren Bild von Abbildung 11.3 sind alle berechneten Ausführungswahrscheinlichkeiten angetragen. Verbesserungen ergeben eine Ausführungswahrscheinlichkeit von  $p_k = 1$  und sind ebenfalls dargestellt. Während es bei den ersten 20 Optimierungsschritten hohe Ausführungswahrscheinlichkeiten trotz vieler Verschlechterungen gibt, finden ab  $k = 460$  nur noch Verbesserungen statt.

Da stochastische Optimierungsverfahren auch zu Verschlechterungen führen können, ist es üblich, den bisher besten Fehlerflächenpunkt zu speichern. Die-

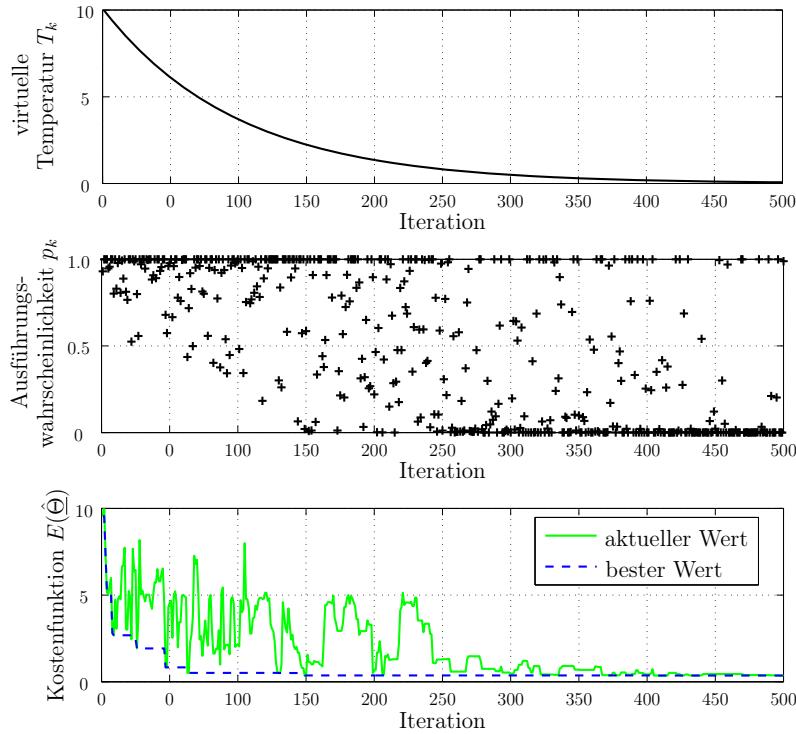


**Abb. 11.2:** Optimierungsbeispiel mit Simulated Annealing — Die Optimierung wird nach 500 Iterationen abgebrochen und bietet eine gute Lösung.

ser Punkt steht dann am Ende der Optimierung als Lösung zur Verfügung. Der durchgezogene Verlauf im unteren Bild von Abbildung 11.3 zeigt den aktuellen Fehlerflächenverlauf mit vielen Verschlechterungen, die jedoch im Laufe der Optimierung durch die geringer werdende virtuelle Temperatur immer moderater ausfallen. Der gestrichelte Verlauf stellt die bisher beste Lösung gegenüber.

## 11.2 Evolutionsstrategien

Die *Evolutionsstrategien* (ES) sind wichtige Vertreter der evolutionären Algorithmen. Sie wurden in den sechziger Jahren des letzten Jahrhunderts von Ingo Rechenberg an der technischen Universität Berlin begründet [182] und später von ihm und von vielen anderen Wissenschaftlern (insbesondere von Hans-Paul Schwefel) weiterentwickelt. Die Evolutionsstrategien verwenden eine reellwertige Codierung, da die Lösung ingenieurtechnischer Aufgaben im Mittelpunkt steht.



**Abb. 11.3:** Optimierungsbeispiel mit Simulated Annealing — Das obere Bild zeigt den Verlauf der virtuellen Temperatur. Im mittleren Bild sind die Ausführungswahrscheinlichkeiten während der Optimierung eingetragen. Das untere Bild vergleicht den aktuellen Kostenfunktionswert mit der bisher besten Lösung.

Somit sind diese Verfahren zur Optimierung von Neuronalen Netzen einsetzbar.

### Evolutionäre Algorithmen, ein Überblick

Evolutionäre Algorithmen sind Modelle der Evolution, mit deren Hilfe Optimierungsprobleme gelöst werden können. Vorbild dabei ist die biologischen Evolution, welche in einem seit vier Milliarden Jahren andauernden Evolutionsprozess (bzw. Optimierungsprozess) äußerst leistungsfähige und an die jeweilige Lebensbedingung perfekt angepasste Lebewesen hervorgebracht hat.

Bei den Evolutionären Algorithmen unterscheidet man im Wesentlichen zwischen den Evolutionsstrategien und den Genetischen Algorithmen, die weitgehend unabhängig voneinander entstanden. Während die in Deutschland entwickelten Evolutionsstrategien nach Rechenberg [182] hauptsächlich die Lösung praktischer Aufgaben verfolgte, befassten sich Holland [94] und Goldberg [65] bei

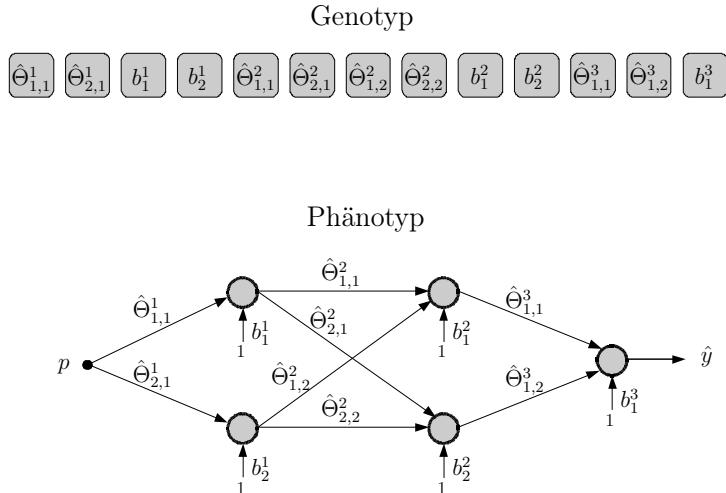
der Entwicklung der *Genetischen Algorithmen* in den USA vor allem mit der Frage, auf welche Art und Weise bei der biologischen Evolution die genetischen Daten gespeichert werden und wie die jeweiligen Prozesse mit dieser genetischen Information operieren. Die Entwicklung der beiden evolutionsorientierten Verfahren fand lange Zeit unabhängig voneinander statt. Erst seit 1985 findet ein stetiger Austausch von Ideen und Konzepten auf regelmäßigen gemeinsamen Konferenzen statt. Anfang der 90er Jahre wurde der Begriff “Evolutionäre Algorithmen” als gemeinsame Bezeichnung für die Algorithmen der beiden verschiedenen Schulen eingeführt [180, 36].

Ein entscheidender Punkt in der Evolution ist die Codierung aller Merkmale eines Lebewesens (bzw. eines Individuums) in den Genen. Bei der Optimierung stellt ein *Individuum* eine mögliche Lösung des Optimierungsproblems dar. Im speziellen Fall der Systemidentifikation mit Neuronalen Netzen verkörpert ein Neuronales Netz ein Individuum der Population mit einer ganz bestimmten Gewichtskonstellation. Der Gewichtsvektor als Punkt auf der Fehlerfläche entspricht bei der Systemidentifikation mit Evolutionären Algorithmen einer Erbinformation, den Genen des Individuums. In diesem Zusammenhang unterscheidet man, wie in Abbildung 11.4 gezeigt, den Phänotyp und den Genotyp. Der *Phänotyp* zeigt das Erscheinungsbild des Individuums auf der Problemebene (in unserem Beispiel das Neuronale Netz mit der individuellen Fähigkeit, das System nachzubilden zu können). Der *Genotyp* beschreibt die Codierung des Phänotyps in den Genen.

Mehrere Individuen bilden eine *Population*. Eine Population besteht aus einer Vielzahl möglicher Lösungen (unterschiedlicher Qualität) für das Optimierungsproblem. Möchte man den zeitlichen Bezug berücksichtigen, so spricht man von einer Generation. Eine *Generation* ist eine Population zu einem ganz bestimmten Zeitpunkt während des Evolutionsprozesses. Ziel bei der Optimierung ist, die Qualität der Population fortwährend zu verbessern. Dies geschieht durch die drei biologischen Operatoren Rekombination, Mutation und Selektion. Die *Rekombination* erzeugt durch Mischen der Erbinformation von Elter-Individuen<sup>2)</sup> Nachkommen, auch Kinder genannt. Die *Mutation* verändert die Erbinformation eines Kindes und ermöglicht dadurch eine Weiterentwicklung der Population. Aus Sicht der Optimierung übernimmt die Mutation die Aufgabe, lokale Minima überwinden zu können. Die *Selektion* steuert die Suchrichtung der Evolution. Sie bestimmt, in welche Richtung sich das Erbgut verändert und letztendlich welche Phänotypen sich bei der Evolution durchsetzen. Mit der Selektion ist ein Verlust an *Diversität* (das ist die Artenvielfalt einer Population) verbunden, da gute Individuen bei der Fortpflanzung begünstigt werden. Die beiden Operatoren Rekombination und Mutation wirken gegen diesen Diversitätsverlust, da sie immer wieder neue Varianten hinzufügen [28].

---

<sup>2)</sup> Hinweis: In der Genetik ist der Begriff *Elter* als Singular von Eltern üblich, da eine Unterscheidung in Vater und Mutter häufig nicht sinnvoll ist. Viele Aussagen beziehen sich gleichermaßen auf Vater und Mutter, man spricht von einem *Elter-Individuum*.



**Abb. 11.4:** Genotyp und Phänotyp eines einfachen MLP mit den Verbindungsgewichten  $\hat{\Theta}_{i,j}^l$  und den Biasgewichten  $b_i^l$  (vergleiche Kap. 3.10) — Während der Genotyp das Individuum als Chromosom repräsentiert, beschreibt der Phänotyp das daraus resultierende Erscheinungsbild auf der Problemebene.

Der klassische Genetische Algorithmus nach Holland [94] und Goldberg [65] nutzt eine binäre Codierung. Da bei den meisten ingenieurtechnischen Optimierungsproblemen — wie auch bei der Systemidentifikation mit Neuronalen Netzen — reellwertige Parameter vorliegen, ist die Verwendung von GA mit einem ständigen (De-) Codierungsaufwand verbunden. Im Laufe der Zeit entstanden deshalb reellwertig codierte GA für verschiedene praktische Anwendungen [31, 186, 188]. Prinzipiell sind damit sowohl die Evolutionsstrategien, als auch die Genetischen Algorithmen für die Systemidentifikation mit Neuronalen Netzen anwendbar. Dennoch werden im weiteren Verlauf der Arbeit nur noch die Evolutionsstrategien als Vertreter der evolutionären Algorithmen betrachtet. Für die Theorie der genetischen Algorithmen sei auf die Literatur verwiesen [62, 172, 188, 198, 27].

### Varianten der Evolutionsstrategien

Die Varianten der Evolutionsstrategien unterscheiden sich in ihrer Komplexität. Rechenberg entwickelte eine eigene Notation, aus der ersichtlich ist, auf welche Art und Weise ein Evolutionsalgorithmus arbeiten. Die folgenden Abschnitte geben einen Überblick über die einzelnen Varianten der Evolutionsstrategien [42, 62, 128, 172, 198].

**$(1+1)$ -ES:**

In dieser einfachsten Form der ES umfasst die Population nur ein einziges Individuum. Zu Beginn der Optimierung erfolgt die Anfangsinitialisierung der Population (bzw. des Individuums) durch einen Vektor zufällig gewählter Zahlen. Eine Kopie dieses Vektors, sozusagen ein Kind, wird erzeugt und dadurch mutiert, dass auf jeden Eintrag des Vektors ein kleiner zufälliger Wert addiert wird. Von den beiden nun vorhandenen Individuen überlebt das bessere und kommt in die nächste Generation. Auf diese Weise lassen sich durch die drei Schritte Selbstreproduktion, Mutation und Selektion allmählich Individuen finden, die dem Optimierungsziel immer näher kommen. Die Notation  $(1+1)$ -ES kennzeichnet, dass jeweils ein Individuum mit seinem Nachkommen verglichen wird. Diese Strategie stellt nur eine sehr grobe Näherung der Evolution dar. Zur Varianz des Erbgutes trägt alleine die Mutation bei, die Rekombination von Erbgut bleibt völlig unberücksichtigt.

 **$(\mu + \lambda)$ -ES:**

Diese ES stellt eine Verallgemeinerung der  $(1+1)$ -ES auf Populationen von mehreren Individuen dar:  $\mu$  Elter-Individuen erzeugen  $\lambda$  mutierte Nachkommen. Es gilt  $1 \leq \mu \leq \lambda$ . Prinzipiell sind die gleichen Schritte wie bei der  $(1+1)$ -ES vorhanden: Die Anfangspopulation besteht aus  $\mu$  Vektoren zufällig gewählter Zahlen. Die  $\mu$  Elter-Individuen erzeugen mit gleicher Wahrscheinlichkeit zufällig  $\lambda$  Nachkommen durch Selbstreproduktion (also durch Clonen). Dabei ist für  $\mu < \lambda$  eine Mehrfachauswahl aus den Elter-Individuen erforderlich. Von den  $\lambda$  mutierten Nachkommen und den  $\mu$  Elter-Individuen überleben unabhängig von der Generationszugehörigkeit die  $\mu$  besten Individuen und kommen so in die nächste Generation. Dies führt dazu, dass die Qualität des besten Individuums nie schlechter werden kann, da es sicher in die nächste Generation eingeht. Bessere Individuen können so viele Generationen überleben. Ein Nachteil der  $(\mu + \lambda)$ -ES ist, dass eine vorzeitige Konvergenz auf ein lokales Minimum der Fehlerfläche eintreten kann. Dieses Problem lässt sich mit der nun folgenden  $(\mu, \lambda)$ -ES vermeiden.

 **$(\mu, \lambda)$ -ES:**

Unsterbliche Individuen, wie im Falle der  $(\mu + \lambda)$ -ES, entsprechen nicht dem Vorbild der biologischen Evolution. Deshalb werden bei der  $(\mu, \lambda)$ -ES nur noch die  $\lambda$  Nachkommen zur Auswahl der besten  $\mu$  Individuen für die Folgegeneration herangezogen, die Elterngeneration stirbt und ist in der neuen Generation nicht mehr vorhanden. Alle Individuen leben also nur noch eine Generation. Dies führt dazu, dass der Wert der Kostenfunktion der besten Individuen nicht mehr monoton fallend wie bei der  $(\mu + \lambda)$ -ES ist, sondern von Generation zu Generation sowohl nach oben als auch nach unten schwanken kann. Dadurch wird eine vorzeitige Konvergenz auf ein lokales Minimum unwahrscheinlicher. Das Ausmaß der Schwankungen hängt von der Art und Stärke der Mutation ab. Soll erst einmal nicht festgelegt werden, ob die Selektion nur aus den Nachkommen oder aus beiden Generationen vorgenommen wird, ist hierfür die Bezeichnung  $(\mu\#\lambda)$ -ES üblich.

Selektionsdruck  $s$ :

Ein charakteristisches Merkmal der Evolutionsstrategien ist der Selektionsdruck  $s$

$$s = \frac{\mu}{\lambda} \quad (11.4)$$

Der Selektionsdruck kann je nach Wahl von  $\mu$  und  $\lambda$  Werte im Bereich 0 und 1 annehmen. Ein sehr kleiner Wert bedeutet einen starken Selektionsdruck (aus den  $\lambda$  Individuen werden nur sehr wenige ausgewählt), ein großer Wert dagegen weist auf einen schwachen Selektionsdruck hin. Bei  $s = 1$  findet überhaupt keine Selektion statt, da alle Individuen in die nächste Generation übernommen werden. Für die Optimierung gilt: Je höher der Selektionsdruck, desto rascher konzentriert sich die Evolution auf ein gutes Individuum und desto größer ist die Gefahr, dass es sich dabei nur um ein lokales Minimum handelt.

$(\mu/\rho\#\lambda)$ -ES:

Im Gegensatz zu den bisher vorgestellten Evolutionsstrategien, bei denen der Lernfortschritt ausschließlich durch Selektion aus von Generation zu Generation mutierten Individuen erzielt wurde, nutzt die  $(\mu/\rho\#\lambda)$ -ES zusätzlich den genetischen Operator der Rekombination. Bei der  $(\mu/\rho\#\lambda)$ -ES laufen prinzipiell die gleichen Evolutionsschritte ab, wie bei den  $(\mu\#\lambda)$ -ES, nur dass anstelle der Selbstreproduktion zur Erzeugung der Nachkommen nun eine Rekombination stattfindet. Für die Bildung eines Nachkommen ist nun nicht mehr ein Individuum erforderlich, sondern eine Gruppe aus  $\rho$  Elter-Individuen (üblicherweise gilt  $\rho = 2$ ). Die Auswahl der  $\lambda$  mal  $\rho$  Elter-Individuen erfolgt wieder zufällig mit gleicher Wahrscheinlichkeit. Zur Erzeugung von Nachkommen gibt es eine Vielzahl von Rekombinationsstrategien. Sehr häufig kommen die arithmetische und die diskrete Rekombination vor. Die *arithmetische Rekombination* führt jeweils eine arithmetische Mittelung der an gleicher Position befindlichen reellen Zahlen der zu rekombinierenden Vektoren durch. So entsteht beispielsweise bei  $\rho = 2$  aus den beiden Vektoren der Elter-Individuen  $\hat{\Theta}_1 = [1 \ 2 \ 3 \ 4]$  und  $\hat{\Theta}_2 = [7 \ 8 \ 1 \ 2]$  durch arithmetische Rekombination der Nachkomme  $\hat{\Theta}_3 = [4 \ 5 \ 2 \ 3]$ . Die *diskreten Rekombination* übernimmt für  $\rho = 2$  die Vektoreneinträge unverändert aus einem der beiden Elter-Individuen. Eine zufällig mit 0 oder 1 besetzte Maske entscheidet, von welchem Elter-Individuum eine bestimmte Position des Nachkommenvektors besetzt werden darf. Die Maske  $m = [1 \ 1 \ 0 \ 0]$  soll beispielhaft eine Rekombination der beiden Elter-Individuen  $\hat{\Theta}_1$  und  $\hat{\Theta}_2$  zeigen: Eine 1 in der Maske übernimmt die entsprechende Position aus dem Elter-Individuum  $\hat{\Theta}_1$ , eine 0 verwendet den entsprechenden Eintrag aus dem Elter-Individuum  $\hat{\Theta}_2$ . Somit entsteht der Nachkomme  $\hat{\Theta}_3 = [1 \ 2 \ 1 \ 2]$ .

Mutation:

Der zentrale genetische Operator der Evolutionsstrategien ist die Mutation. Mutierte Nachkommen entstehen durch die Addition eines Vektors aus normalverteilten Zufallszahlen:

$$\hat{\Theta}_{\text{mutiert}} = \hat{\Theta} + \underline{N}(0, \sigma) \quad (11.5)$$

Dabei ist  $\underline{N}(0, \sigma)$  ein Vektor mit normalverteilten Zufallszahlen mit Mittelwert 0 und Standardabweichung  $\sigma$ . Diese Mutation hat den Nachteil, dass die Standardabweichung während des Evolutionsprozesses konstant bleibt. Weiterentwicklungen der ES führten zu einer *adaptiven Schrittweitensteuerung*. Dabei erfolgt eine Anpassung der Standardabweichung in Abhängigkeit vom Erfolg oder Misserfolg der Mutation. Bei den Untersuchungen in [36] brachte eine adaptive Schrittweitensteuerung jedoch keine Verbesserung bei der Systemidentifikation mit Neuronalen Netzen. Deshalb wird im weiteren Verlauf nur die  $(\mu/\rho\#\lambda)$ -Evolutionsstrategie mit fest eingestellter Standardabweichung betrachtet.

### Zusammenfassung — $(\mu/\rho\#\lambda)$ -Evolutionsstrategie

#### 1. Anfangsinitialisierung:

Geeignete benutzerdefinierte Parameter festlegen: Anzahl an Elter-Individuen  $\mu$ , Anzahl an Nachkommen  $\lambda$  und Anzahl an Elter-Individuen für eine Rekombination  $\rho$ . Wahl einer passenden Standardabweichung  $\sigma$  (bestimmt die Größenordnung der Mutation) und eines sinnvollen Suchbereiches, auf den sich die Optimierung beschränkt. Zufällige Verteilung der Anfangspopulation auf der Fehlerfläche (also  $\mu$  Vektoren zufällig gewählter Zahlen erzeugen).

Für jedes Individuum der Anfangspopulation:

→ Berechnung des Kostenfunktionswertes  $E(\hat{\Theta}_0)$

#### 2. Rekombination — für alle $\lambda$ Nachkommen:

→ Zufällige Auswahl von  $\rho$  Elter-Individuen

→ Arithmetische oder diskrete Rekombination (siehe Seite 397)

#### 3. Mutation — für alle $\lambda$ Nachkommen:

→ Vektor  $\underline{N}(0, \sigma)$  aus normalverteilten Zufallszahlen erzeugen

→  $\hat{\Theta}_{mutiert,k} = \hat{\Theta}_k + \underline{N}(0, \sigma)$  (nach Gleichung (11.5))

→ Kontrolle, ob der Nachkomme den Suchbereich verlässt. Falls nicht:

Berechnung der Qualität des Nachkommen  $E(\hat{\Theta}_{mutiert,k})$

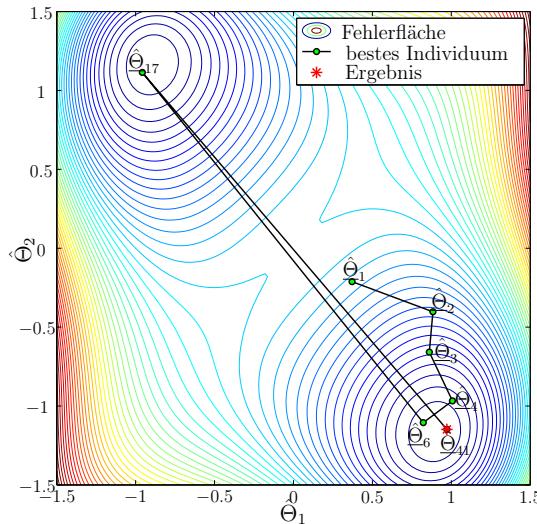
#### 4. Selektion — für alle $\lambda$ Nachkommen (bei der $(\mu/\rho, \lambda)$ -ES) bzw. für alle $\mu + \lambda$ Individuen (bei der $(\mu/\rho + \lambda)$ -ES):

→ Auswahl der  $\mu$  besten Individuen für die neue Generation

5. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 2 bis 4.

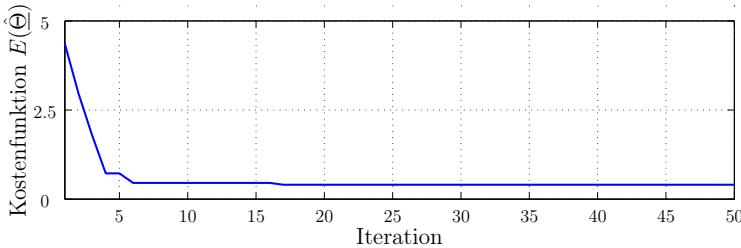
### Beispiel — Optimierung mit der $(\mu/\rho, \lambda)$ -Evolutionsstrategie

Als Beispiel dient wieder die einfache Fehlerfläche aus Kap. 10.1, wobei sich die Minimumsuche auf den Parameterbereich von  $-1.5$  bis  $+1.5$  begrenzt. Die Anfangspopulation wird zufällig innerhalb des Suchbereichs verteilt. Die Population besteht aus  $\mu = 5$  Individuen, welche in jeder Iteration  $\lambda = 10$  Nachkommen bilden. Dabei erfolgt die Rekombination mit jeweils  $\rho = 2$  zufällig ausgewählten Elter-Individuen. Diese Einstellungen führen mit Gleichung (11.4) zu einem Selektionsdruck von  $s = 0.5$ . Für das Beispiel ist eine Standardabweichung von  $\sigma = 0.5$  gut geeignet. Die Optimierung endet nach 50 Iterationen.



**Abb. 11.5:** Optimierungsbeispiel mit der  $(\mu/\rho, \lambda)$ -Evolutionsstrategie — Der Verlauf zeigt den Weg des bisher besten Individuums auf der Fehlerfläche. Dargestellt sind 50 Iterationen, die insgesamt 6 mal zu einer Verbesserung führen.

Abbildung 11.5 zeigt die zu optimierende Fehlerfläche mit der Position des bisher besten Individuums. Diese Position muss sich bei den stochastischen Optimierungsverfahren natürlich nicht bei jeder Iteration ändern. Verbesserungen auf der Fehlerfläche finden lediglich die Generationen im 2., 3., 4., 6., 17. und 41.-ten Optimierungsschritt. Der dazugehörige Wert der Kostenfunktion ist in Abbildung 11.6 angetragen.



**Abb. 11.6:** Optimierungsbeispiel mit der  $(\mu/\rho, \lambda)$ -Evolutionsstrategie — Dargestellt ist der Kostenfunktionswert der bisher besten Lösung. Die beste Lösung ändert sich insgesamt 6 mal. Da die Verbesserungen zum Teil recht gering ausfallen, sind auf dem Bild nicht alle 6 Schritte sichtbar.

Dieses Beispiel zeigt, wie auch das Beispiel bei Simulated Annealing auf Seite 391, dass das Interesse der Optimierung mehrfach zwischen den beiden globalen Minima wechselt. Das ist bei den stochastischen Optimierungsverfahren durchaus gewollt. Gerade diese Eigenschaft trägt dazu bei, dass der Optimierungsprozess aus schlechten lokalen Minima wieder herauskommt. Probleme treten jedoch bei gleich guten symmetrisch angeordneten globalen Minima auf, wie in diesem Beispiel. Befinden sich beispielsweise zwei Elter-Individuen in jeweils einem der beiden globalen Minima, so führt die Rekombination zu einem Kind, das die sehr guten Eigenschaften der beiden Elter-Individuen verliert und sich deutlich verschlechtert. Dieser Effekt tritt bei der Optimierung von Neuronalen Netzen noch viel stärker auf, da bereits kleine Netze über viele Millionen symmetrisch angeordneter globaler Minima verfügen. Eine genauere Betrachtung dieser Problematik folgt in Kap. 11.4.

### 11.3 Particle Swarm Optimization

*Particle Swarm Optimization* (PSO) wurde von Kennedy und Eberhart im Jahre 1995 vorgestellt und hat sich seither zu einer sehr populären stochastischen Optimierungsmethode entwickelt [96]. Ursprüngliches Ziel der Forschung auf diesem Gebiet war weniger die Entwicklung eines Optimierungsverfahrens als die simulative Nachbildung des Schwarmverhaltens vieler Tiere [120]. Die einzelnen Schwarmmitglieder können von Erkundungen aller anderen — z.B. auf der Suche nach Futter — profitieren. Soziale Übermittlung von Information wird so zum Vorteil in der Evolution [120].

#### Swarm Intelligence, ein Überblick

*Swarm Intelligence* (SI) ist eine Technik der künstlichen Intelligenz, die auf der Erforschung des kollektiven Verhaltens in dezentralen, selbstorganisierten Systemen begründet ist. Die Bezeichnung SI wurde 1989 von Beni, Hackwood und

Wang im Zusammenhang mit verteilten Robotersystemen mit vielen einfachen Agenten eingeführt [19]. Zwei der gegenwärtig erfolgreichsten Strategien der SI sind Particle Swarm Optimization und Ant Colony Optimization. PSO ist eine stochastische Optimierungsmethode, welche versucht, das soziale Verhalten von in Schwärmen lebenden Tieren nachzubilden. Eine Menge von Individuen (potentiellen Lösungen) erkundet durch Bewegung die zu minimierende Fehlerfläche. Durch Austausch von Information über die individuelle Qualität bewegt sich der gesamte Schwarm mit der Zeit in Richtung des Individuums mit dem geringsten Kostenfunktionswert. Durch die hohe Zahl der Schwarmmitglieder wird der Lösungsraum gründlich untersucht und im günstigsten Fall das globale Optimum gefunden. *Ant Colony Optimization* (ACO) ist ein Optimierungsalgorithmus zur Lösung anspruchsvoller kombinatorischer Probleme. Hierbei erkunden viele Individuen, sozusagen "Ameisen", mögliche Lösungen der Aufgabenstellung, z.B. durch eine optimale Route einen möglichst kurzen Weg zu finden. Sie hinterlassen dabei Spuren in Form einer virtuellen Nachbildung des Duftstoffes Pheromon und unterstützen dadurch nachfolgende Individuen, bessere Lösungen zu finden. ACO wurde 1992 von Dorigo [35] erstmals als Methode zur Lösung des bekannten Travelling Salesman Problem vorgestellt. Da das ACO-Verfahren hauptsächlich für die Lösung kombinatorischer Probleme geeignet ist und weniger für die Optimierung in der Systemidentifikation, soll in dieser Arbeit nur das PSO-Verfahren als Vertreter der Gruppe Swarm Intelligence näher betrachtet werden.

### Der PSO-Basisalgorithmus

Beim PSO-Algorithmus kann man sich einen Schwarm von Vögeln, Insekten oder Fischen vorstellen, der sich verteilt über ein bestimmtes Gebiet bewegt und dieses erkundet. Wenn ein Tier eine besonders gute Umgebung (z.B. in Bezug auf Futter, Schutz etc.) gefunden hat, ist der Rest des Schwarmes in der Lage, in diese Richtung zu folgen.

Im Simulationsmodell erhalten mehrere Individuen (potentielle Lösungen des Optimierungsproblems) zu Beginn eine zufällige Position  $\hat{\Theta}_0$  auf der  $N$ -dimensionalen Fehlerfläche und eine zufällige Geschwindigkeit  $\underline{v}_0$ . Die Individuen bewegen sich im Lösungsraum und speichern jeweils ihre beste Position, die sie im Laufe der Erkundung erreicht haben. Die Schwarmmitglieder kommunizieren diese vielversprechenden Orte und modifizieren in jeder Iteration ihre Position und ihre Geschwindigkeit nach folgender Berechnung [41, 121]:

$$\hat{\Theta}_{k+1} = \hat{\Theta}_k + \underline{v}_k \quad (11.6)$$

$$\underline{v}_{k+1} = j \cdot \underline{v}_k + c_1 r_1 (\tilde{\Theta} - \hat{\Theta}_k) + c_2 r_2 (\tilde{\Theta}^g - \hat{\Theta}_k) \quad (11.7)$$

Entscheidend ist also die Bestimmung eines neuen Geschwindigkeitsvektors  $\underline{v}_{k+1}$ , der sich aus einer Trägheitskomponente (Trägheitskonstante  $j < 1$ ), der individuellen Erfahrung (gewichtet mit  $c_1$ ) und der gemeinsamen Erfahrung (gewichtet mit  $c_2$ ) zusammensetzt.  $r_1$  und  $r_2$  sind gleichverteilte Zufallszahlen aus  $[0 \ 1]$ .  $\tilde{\Theta}$  bezeichnet die bisher beste Position des einzelnen Individuums und  $\tilde{\Theta}^g$  die bisher beste Position aus allen Schwarmmitgliedern. Jedes Individuum wird also von

einer gewichteten Kompromissposition aus individueller und gemeinsamer Erfahrung angezogen, die sich durch den Einfluss des Zufalls und durch mögliche neue Bestwerte bei jeder Iteration verändert. Die Trägheitskomponente verhindert ein zu schnelles Einschwingen auf einen bestimmten Bereich und fördert so das Erkunden noch unbekannter Bereiche. Im Laufe der Zeit bewegen sich zunehmend alle Schwarmmitglieder in Richtung eines optimalen Bereiches, wobei die Wahl der Konstanten  $j$ ,  $c_1$  und  $c_2$  einen großen Einfluss auf das Konvergenzverhalten des stochastischen Optimierungsverfahrens hat.

Bei der Kommunikationsstruktur unterscheidet man eine globale und eine lokale Variante. Die *globale Variante* verwendet zur Bestimmung der Geschwindigkeitsvektoren neben dem individuellen Optimum die zu diesem Zeitpunkt beste Position des gesamten Schwarmes  $\tilde{\Theta}^g$ . Die *lokale Variante* zieht für die Berechnung der neuen Geschwindigkeitsvektoren neben dem individuellen Optimum die bis zum Berechnungszeitpunkt beste Position einer bestimmten Anzahl von Individuen in einer definierten Umgebung heran. Die globale Variante kann als ein Spezialfall der lokalen betrachtet werden mit einer Umgebung des jeweiligen Individuums, die den gesamten restlichen Schwarm umfasst [41]. Während die globale Version vergleichsweise schnell konvergiert, da sie immer die absolut beste Position im Focus hat, aber leichter in einem lokalen Minimum steckenbleiben kann, wird für die lokale Version ein dazu gegensätzliches Verhalten berichtet [40, 41].

Abbruchkriterium der PSO-Optimierung kann entweder ein Unterschreiten einer bestimmten Fehlergrenze an der Position des Schwarmführers  $\tilde{\Theta}^g$ , das Erreichen einer bestimmten Zahl von Iterationen oder das Unterschreiten eines bestimmten Schwarmradius sein [40, 12].

Die Zahl der Veröffentlichung auf dem Gebiet PSO ist in den letzten Jahren sprunghaft angestiegen. Die vielen Veröffentlichungen brachten sehr viele Varianten zum PSO-Algorithmus hervor. Die relevanten Modifikationen haben die Begründer des PSO-Algorithmus in [41] und [96] zusammengefasst, auf die an dieser Stelle nicht näher eingegangen werden soll.

### **Zusammenfassung — Particle Swarm Optimization (globale Variante)**

1. Anfangsinitialisierung:

Wahl einer geeigneten Trägheitskonstante  $j < 1$  und Initialisieren der beiden Beschleunigungskoeffizienten  $c_1$  und  $c_2$  (gewichten die individuelle bzw. die gemeinsame Erfahrung). Festlegen einer passenden Anzahl an Individuen und eines sinnvollen Suchbereichs, auf den sich die Optimierung beschränkt.

Für jedes Individuum:

→ Zufällige Bestimmung einer Anfangsposition auf der Fehlerfläche  $\underline{\Theta}_0$  und einer zufälligen Anfangsgeschwindigkeit  $\underline{v}_0$ .

2. Beim ersten Optimierungsschritt:

Für jedes Individuum:

- Berechnung des Kostenfunktionswertes  $E(\hat{\underline{\Theta}}_0)$
- Initialisierung der besten Position:  $\tilde{\underline{\Theta}} = \hat{\underline{\Theta}}_0$
- Funktionswert speichern:  $E(\tilde{\underline{\Theta}}) = E(\hat{\underline{\Theta}}_0)$

Für das beste Individuum:

- Initialisierung der besten Position:  $\tilde{\underline{\Theta}}^g = \hat{\underline{\Theta}}_0$
- Funktionswert speichern:  $E(\tilde{\underline{\Theta}}^g) = E(\hat{\underline{\Theta}}_0)$

Fortführung der Optimierung mit  $k = 0$ .

3. Neue Positionen der Individuen berechnen nach Gleichungen (11.6).

Für jedes Individuum:

$$\rightarrow \hat{\underline{\Theta}}_{k+1} = \hat{\underline{\Theta}}_k + \underline{v}_k$$

Kontrolle, ob die Position im Suchbereich liegt.

4. Erfahrung über Fehlerfläche sammeln.

Für jedes Individuum:

- Berechnung des Kostenfunktionswertes  $E(\hat{\underline{\Theta}}_{k+1})$
- falls  $E(\hat{\underline{\Theta}}_{k+1}) < E(\tilde{\underline{\Theta}}) : \tilde{\underline{\Theta}} = \hat{\underline{\Theta}}_{k+1}$
- falls  $E(\hat{\underline{\Theta}}_{k+1}) < E(\tilde{\underline{\Theta}}^g) : \tilde{\underline{\Theta}}^g = \hat{\underline{\Theta}}_{k+1}$

5. Neue Geschwindigkeiten der Individuen berechnen nach Gleichung (11.7).

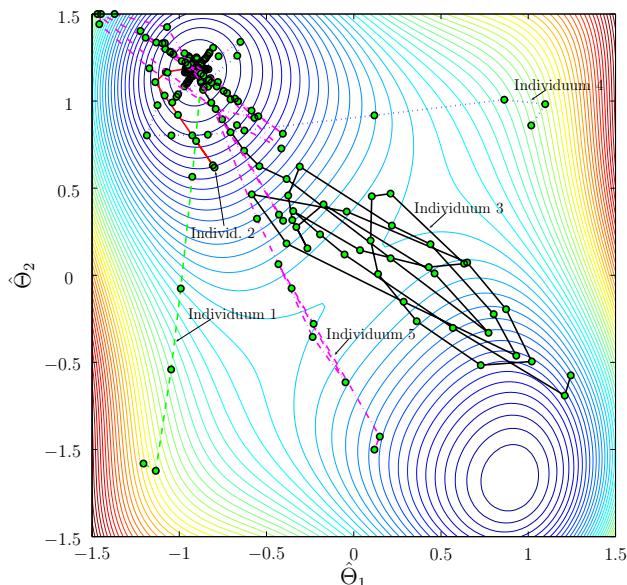
Für jedes Individuum:

$$\rightarrow \underline{v}_{k+1} = j \cdot \underline{v}_k + c_1 r_1 (\tilde{\underline{\Theta}} - \hat{\underline{\Theta}}_k) + c_2 r_2 (\tilde{\underline{\Theta}}^g - \hat{\underline{\Theta}}_k)$$

6. Iteration: Falls das Abbruchkriterium noch nicht erfüllt ist, wiederhole die Punkte 3 bis 5.

### Beispiel — Optimierung mit dem PSO-Algorithmus (globale Variante)

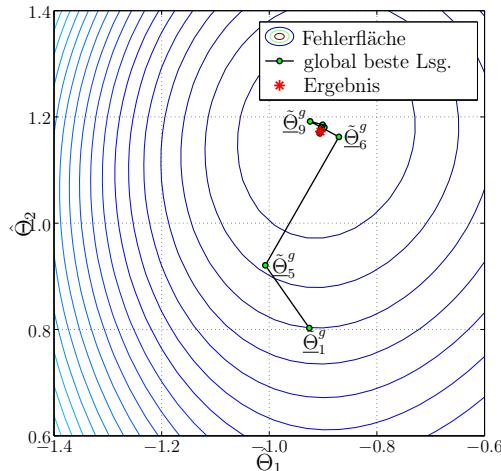
Das zu lösende Optimierungsproblem ist wieder die Minimierung der in Kap. 10.1 beschriebenen Fehlerfläche. Dabei beschränkt sich die Optimierung auf den Parameterbereich von  $-1.5$  bis  $+1.5$ , in dem sich beide globalen Minima befinden. Zunächst gilt es wieder, passende Werte für die benutzerdefinierten Parameter zu finden. Für das Beispiel eignet sich eine Trägheitskonstante von  $j < 0.9$ , die Gewichtung der individuellen Erfahrung erfolgt mit  $c_1 = 0.7$  und die gemeinsame Erfahrung geht mit  $c_2 = 0.5$  in die Berechnung ein. Die Anfangsposition und die Anfangsgeschwindigkeit der 5 Individuen wird zufällig bestimmt. Die Optimierung endet wieder nach 50 Iterationen.



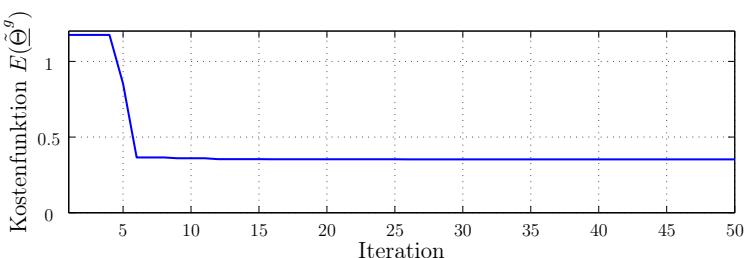
**Abb. 11.7:** Optimierungsbeispiel mit Particle Swarm Optimization — Das Individuum 2 übernimmt zunächst die Schwarmführerschaft. Während die beiden Individuen 1 und 4 mit hohen Kostenfunktionswerten dem Schwarmführer unmittelbar folgen, benötigen die beiden Individuen 3 und 5 mit sehr guten Anfangswerten viele Iterationen, bis sie sich in die Richtung des Schwarmführers bewegen.

Abbildung 11.7 zeigt, wie sich die Schwarmmitglieder während der Optimierung bewegen. Es ist deutlich zu erkennen, dass gerade die Individuen mit schlechten Funktionswerten dem Schwarmführer (das ist das Individuum, mit dem besten Funktionswert) folgen. Im Beispiel von Abbildung 11.7 übernimmt das Individuum 2 zu Beginn die Schwarmführerschaft. Die beiden Individuen 1 und 4 folgen dem Schwarmführer unverzüglich, da ihre Anfangspositionen zu hohen Kostenfunktionswerten führen. Bei den beiden Individuen 3 und 5 verge-

hen viele Iterationen, bis sie sich dem Schwarmführer anschließen. Die Ursache dafür liegt an den niedrigen Werten der Kostenfunktion gleich zu Beginn der Optimierung. Diese gesammelte Erfahrung beeinflusst über die Konstante  $c_1$  den weiteren Optimierungsverlauf.



**Abb. 11.8:** Optimierungsbeispiel mit Particle Swarm Optimization — Der Verlauf zeigt den Weg der bisher besten Lösung  $\tilde{\Theta}^g$  auf der Fehlerfläche. Dargestellt sind 50 Iterationen, die insgesamt 9 mal zu Verbesserungen führen.



**Abb. 11.9:** Optimierungsbeispiel mit Particle Swarm Optimization — Kostenfunktionswert des bisher besten Punktes  $E(\tilde{\Theta}^g)$ .

In Abbildung 11.8 ist die Position des bisher besten Fehlerflächenpunkts  $\tilde{\Theta}^g$  dargestellt. Der entsprechende Funktionswert  $E(\tilde{\Theta}^g)$  ist in Abbildung 11.9 aufgetragen. Da die Anfangsposition des Individuums 2 bereits in der Nähe des Minimums liegt, erreicht die Optimierung schon nach wenigen Schritten gute Ergebnisse.

## 11.4 Stochastische Optimierungsverfahren bei der Systemidentifikation mit Neuronalen Netzen

Es stellt sich nun die Frage, wie gut die in den Kapiteln 11.1 bis 11.3 vorgestellten Verfahren zur Systemidentifikation mit Neuronalen Netzen geeignet sind. Dazu folgt in diesem Unterkapitel ein interessantes Identifikationsbeispiel, das sich auf die Ergebnisse einer in [36] durchgeföhrten Untersuchung bezieht.

Identifiziert werden soll der in Abbildung 11.10 dargestellte Plattenaufbau bestehend aus zwei Gleichstrommotoren mit Propellern, zwischen denen eine drehbar gelagerte Plexiglasplatte eingebracht ist. Der Luftstrom, der von den beiden Propellern erzeugt wird, lenkt die Plexiglasplatte aus. Zwischen den Drehzahlen der Motoren und dem Auslenkwinkel der Platte besteht ein nichtlinearer dynamischer Zusammenhang, welcher im Folgenden für einen Motor identifiziert werden soll. Die Identifikation verwendet ein statisches MLP mit externer Dy-

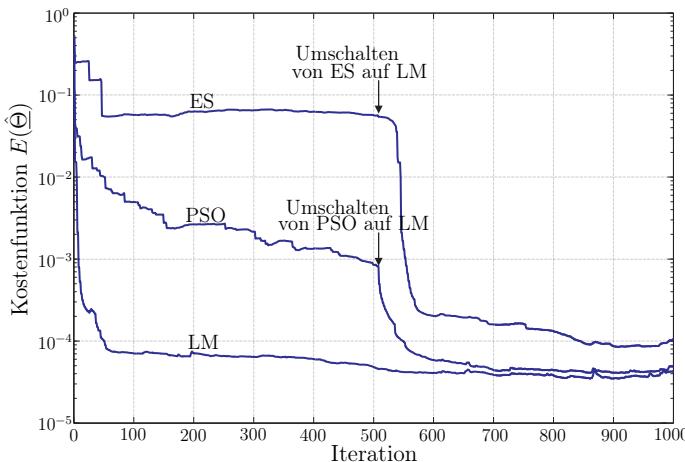


**Abb. 11.10:** Dynamische Nichtlinearität zur Identifikation: Plattenaufbau

namik in einer Seriell-Parallel Anordnung. Da Ein- und Ausgang des zu identifizierenden Systems mit ein- bis dreifacher Verzögerung in das Modell eingehen, muss das statische MLP-Netz insgesamt 6 Eingänge besitzen. Die Aufgabe der Optimierungsverfahren bei der Identifikation besteht nun darin, die insgesamt  $N = 113$  Gewichte des 6-7-7-1-MLP geeignet einzustellen. Die Abtastzeit von Modell und Optimierungsalgorithmus bei dieser Untersuchung beträgt 60 ms. Die quasi-online berechnete Fehlerfläche verfügt über 500 Trainingsdaten. Angezeigt wird das System mit einem APRB-Signal. Für die stochastischen Optimierungsverfahren gelten die folgenden Einstellungen:

- Identifikation mit der  $(\mu/\rho, \lambda)$ -Evolutionsstrategie: Die Population besteht aus  $\mu = 8$  Individuen, welche in jeder Iteration  $\lambda = 56$  Nachkommen bilden. Dabei erfolgt die Rekombination mit jeweils  $\rho = 2$  zufällig ausgewählten Elter-Individuen. Diese Einstellungen führen mit Gleichung (11.4) zu einem Selektionsdruck von  $s = 1/7$ . Die Mutation verwendet eine fest eingestellte Standardabweichung von  $\sigma = 1$ , da nach [36] eine adaptive Standardabweichung keine Verbesserung bei der Systemidentifikation mit Neuronalen Netzen bringt.

- Identifikation mit Particle Swarm Optimization Verfahren (globale Variante): Der Schwarm setzt sich aus 56 Individuen zusammen. Für die Trägheitskonstante gilt  $j = 0.5$ , die Gewichtung der individuellen Erfahrung erfolgt mit  $c_1 = 1.5$  und die gemeinsame Erfahrung geht mit  $c_2 = 1.5$  in die Berechnung ein.
- Simulated Annealing: Dieses stochastische Optimierungsverfahren eignet sich nur für Optimierungsaufgaben niedriger Dimension und ist somit nicht für die Anwendung bei Neuronalen Netzen geeignet [36].



**Abb. 11.11:** Vergleich der Fehlerverläufe bei der Identifikation des Plattenaufbaus — nach etwa 500 Iterationen erfolgt jeweils eine Umschaltung auf den deterministischen LM-Algorithmus [36].

Abbildung 11.11 vergleicht die Fehlerverläufe bei der Identifikation des Plattenaufbaus für die Optimierungsverfahren ES, PSO und LM. Die beiden stochastischen Verfahren übergeben nach etwa 500 Iterationen die bis dahin gefundene Lösung einem LM-Algorithmus, welcher die Optimierung fortsetzt (hybride Optimierungsstrategie). Das Bild zeigt, dass das deterministische LM-Verfahren viel schneller den Fehler reduziert als die stochastischen Verfahren. Da eine Iteration bei den stochastischen Verfahren erheblich länger dauert (durch die vielen Fehlerflächenauswertungen) als beim LM-Algorithmus, wäre bei einer zeitlichen Gegenüberstellung der Unterschied noch viel deutlicher. Für eine nähere Beschreibung der hier gezeigten Identifikation sei auf [36] verwiesen.

Eine viel umfangreichere Untersuchung führt Hamm in [82] durch. Er vergleicht darin 8 stochastische Optimierungsverfahren mit dem lokalen Quasi-Newton-Verfahren und kommt zu dem gleichen Resultat: Globale Optimierungsverfahren sind nicht geeignet für die Parameteroptimierung bei Neuronalen Netzen. Es ist viel aussichtsreicher, ein lokales Verfahren 2. Ordnung mit mehreren

unterschiedlichen Anfangsinitialisierungen anzuwenden. Bei den Untersuchungen in [82] könnte während einer globalen Optimierung ein lokales Optimierungsverfahren etwa 20 Neustarts durchführen.

Stochastische Verfahren leben vom Zufall und von der Vielfältigkeit der Lösungen. Dazu sind viele Fehlerflächenberechnungen erforderlich, was bei Neuronalen Netzen zu sehr langsamer Parameterkonvergenz führt. Des Weiteren existieren bei der Systemidentifikation mit Neuronalen Netzen sehr viele und exakt gleich gute Lösungen. Dieses zu lösende Optimierungsproblem entspricht nicht der biologischen Evolution und führt zu Schwierigkeiten: Bei der Swarm Intelligence können sich Individuen gegenseitig behindern, anstatt sich zu helfen, bei den Evolutionären Algorithmen tragen die Nachkommen von sehr guten Individuen nicht die guten Eigenschaften der Eltern, sondern führen in der Regel zu Verschlechterungen. Ein weiterer Punkt ist, dass bei stochastischen Optimierungsverfahren die erzielbare Genauigkeit der Lösung meist bei weitem nicht ausreicht, um bei der Systemidentifikation mit Neuronalen Netzen vernünftige Modelle zu erhalten. Die deterministischen Verfahren gehen bei der Optimierung zielgerichtet vor als die stochastischen Verfahren. Sie erreichen dadurch eine zügige Verbesserung des Kostenfunktionswertes mit nur wenigen Funktionsauswertungen. Durch die vielen guten lokalen und die vielen globalen Minima finden lokale Optimierungsverfahren häufig gute Lösungen [44]. Insgesamt stellen die stochastischen Optimierungsverfahren bei der Systemidentifikation mit Neuronalen Netzen keine Alternative zu den leistungsfähigen deterministischen Verfahren dar. Für die globale Minimumssuche bei der Systemidentifikation mit Neuronalen Netzen ist es deshalb aussichtsreicher, deterministische Verfahren zu verwenden und bei einer zum Stillstand gekommenen Optimierung einzugreifen, indem man beispielsweise die Struktur des Modells verändert. Näheres dazu findet man in [44].

# 12 Verfahren zur Regelung nichtlinearer Systeme

Bei der Regelung nichtlinearer Systeme ist eine Zustandstransformation hilfreich, die in diesem Kapitel vorgestellt wird. Es handelt sich dabei um die Transformation in die sogenannte nichtlineare Regelungsnormalform, deren Ziel es ist, gewisse strukturelle Eigenschaften des zu regelnden Systems sichtbar zu machen. Es sind dies der Relativgrad, die Ordnung und Nulldynamik des Systems. Interessant dabei ist, dass durch Transformation ein nichtlineares Teilsystem "isoliert", und anschliessend kompensiert werden kann. Nach der Kompensation liegt ein System vor, das in seinem Ein-/Ausgangsverhalten linear ist und mit bekannten Methoden geregelt werden kann. Die Kapitel 12.1 bis 12.3 stellen die notwendigen Werkzeuge zur Verfügung. Das Verfahren der E/A-Linearisierung wird in Kapitel 12.4 vorgestellt. Eine Verallgemeinerung auf das Problem der Folgeregelung auf ein Referenzsignal wird in Kapitel 12.5 besprochen. In Kapitel 12.6 wird das Problem behandelt, dass die Nichtlinearität unbekannt ist und erlernt werden muss. Kapitel 12.7 schliesslich zeigt, wie nichtlineare Systeme auch ohne strukturelle Vorkenntnisse geregelt werden können. In diesem Fall muss nicht nur die Nichtlinearität, sondern auch die Transformation, die die Nichtlinearität isoliert (und damit beherrschbar macht), erlernt werden.

## 12.1 Relativgrad und Ordnung

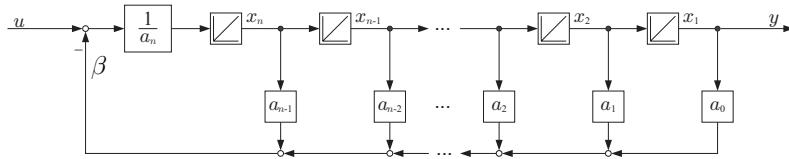
Der Relativgrad  $\delta$  eines Systems gibt Auskunft über den Abstand des regelnden Systemeingriffs vom Systemausgang, wobei dieser Abstand in Anzahl der Integratoren gemessen wird. Die Ordnung des Systems ist dagegen einfach die Anzahl *aller* Integratoren.

Für ein lineares System bedeutet dies: der Relativgrad  $\delta$  ist gleich der Differenz zwischen der Ordnung von Nenner- und Zählerpolynom der Übertragungsfunktion. Wird die Ordnung des Nenners (Anzahl der Polstellen) mit  $n \in \mathbb{N}$  bezeichnet und die Ordnung des Zählers (Anzahl der Nullstellen) mit  $m \in \mathbb{N}$ , so ergibt sich der Relativgrad

$$\delta = n - m \geq 0 \tag{12.1}$$

für ein kausales System. Obwohl nichtlineare Systeme nicht durch Pole und Nullstellen beschreibbar sind, lässt sich diese Idee auf nichtlineare Systeme übertragen, wie im folgenden gezeigt wird.

Um die systemtheoretische Bedeutung des Relativgrades zu verstehen, wird das Problem der **Kompensation systemeigener Dynamik** betrachtet. Ausgangspunkt bildet die Regelungsnormalform (RNF). Für diesen Fall ist die RNF

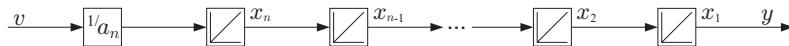


**Abb. 12.1:** Signalflussplan eines LTI-Systems in Regelungsnormalform für den Fall  $\delta = n$

in Abbildung 12.1 dargestellt. Anhand dieser Darstellung wird offensichtlich, dass durch ein geeignet gewähltes Eingangssignal  $u(t) = \beta(t)$  der Einfluss des Rückführzweiges aufgehoben werden kann. Wird ein Zustandsregler als Kompensator angesetzt, der die Linearkombination

$$u(t) = \underbrace{a_0 x_1(t) + a_1 x_2(t) + \cdots + a_{n-1} x_n(t)}_{\text{Kompensationssignal } \beta(t)} + v(t) \quad (12.2)$$

aufschaltet, wird das Signal  $\beta(t)$  kompensiert. Es verbleibt lediglich das Verhalten einer Integratorkette, die in Abbildung 12.2 gezeigt ist. Der Zustandsreg-



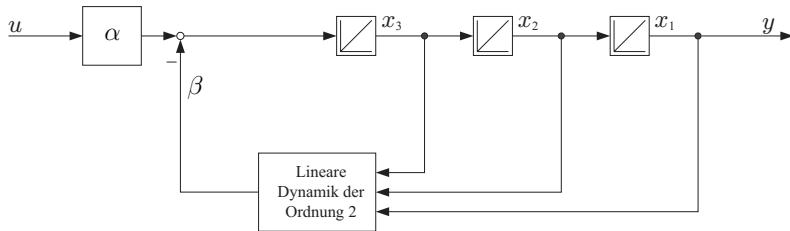
**Abb. 12.2:** Signalflussplan des kompensierten LTI-Systems in Regelungsnormalform für den Fall  $\delta = n$ . Es verbleibt durch die Aufschaltung des Zustandsreglers lediglich eine Integratorkette

ler kompensiert die charakteristische Systemdynamik vollständig und prägt dem Eingang der Kette das Signal  $v(t)$  ein. Dieser Vorgang entspricht der bekannten Tatsache, dass Zustandsregler die Eigendynamik und damit die Polstellen des Regelkreises beliebig verändern können. Die Polkonfiguration der ursprünglichen Regelstrecke wird bei geeigneter Zustandsregelung mit dem Kompensator in den Ursprung verschoben, sämtliche  $n$  Pole liegen bei  $s = 0$ . Dort befindet sich folglich eine  $n$ -fache Polstelle.

Mit kausalen Mitteln (d.h. ohne Verwendung von Differenzierern, die nicht realisierbar sind), kann die Dynamik der Integratorkette nicht weiter reduziert werden. Daher bildet eine Kette aus  $\delta$  Integratoren das Minimum an Dynamik, die die Regelstrecke beibehält.

Ein ähnliches Vorgehen kann ebenso bei Strecken angesetzt werden, die ein Zählerpolynom vom Grad  $m > 0$  besitzen und damit mindestens eine Nullstelle aufweisen. Wie in Kapitel 12.3 allgemein und in Kapitel 12.5.1 am Beispiel

ausführlich erklärt wird, bedient man sich auch hier der Byrnes-Isidori Form. Beispielhaft ist ein solches System für die Konstellation  $n = 5$  und  $m = 2$  in Abbildung 12.3 gezeigt. Der Relativgrad ergibt sich zu  $\delta = n - m = 3$ . Aus diesem Grund ist eine Kette aus 3 Integratoren im Vorwärtszweig enthalten. Die beiden restlichen Zustände befinden sich in dieser Form im Rückführzweig, wobei deren Eigendynamik von der Lage der Nullstellen der Übertragungsfunktion abhängt. Wird oben beschriebene Vorgehensweise der Kompensation angewandt, führt die



**Abb. 12.3:** Signalflussplan eines LTI-Systems in Byrnes-Isidori Form für den Fall  $n = 5$ ,  $m = 2$ . Der Relativgrad  $\delta = 3$  bedingt eine Integratorkette dritter Ordnung

Eliminierung von  $\beta(t)$  auch zur Eliminierung der Dynamik im Rückführzweig. Auch für den vorliegenden Fall, dass die Rückführung nicht nur statische Elemente (Verstärkungen), sondern dynamische Anteile (zwei Integratoren) enthält, setzt sich  $\beta(t)$  aus einer Linearkombination der Zustände zusammen und kann daher von einem Zustandsregler aufgehoben werden. Es bleibt eine Kette aus drei Integratoren, also wiederum der Vorwärtszweig, übrig. Aus Sicht der Polplatzierung bei linearen Zustandsreglern entspricht dies der folgenden Polverschiebung: von den fünf vorhandenen Polstellen der ungeregelten Strecke werden zwei auf die beiden vorhandenen Nullstellen gelegt. Dadurch entsteht eine Pol-Nullstellen-Kürzung, wodurch zwei Eigenwerte entfallen. Eine weitere Reduzierung der Dynamik ist nicht erreichbar. Die verbleibenden drei Polstellen werden wiederum in den Ursprung verschoben und beschreiben daher eine Integratorkette mit drei Elementen ( $\delta = 3$ ).

## 12.2 Nulldynamik

Ist der Relativgrad  $\delta$  kleiner als die Systemordnung  $n$ , so besitzt das System eine sogenannte Nulldynamik. Um eine bessere Vorstellung zu erhalten, welche Auswirkung die Nulldynamik eines nichtlinearen Systems im offenen und geschlossenen Regelkreis besitzt, soll zunächst untersucht werden, was eine Nulldynamik im linearen Fall bedeutet.

### 12.2.1 Nulldynamik bei linearen Systemen

Auf den Fall eines linearen Systems übertragen, bedeutet die Eigenschaft  $\delta = n$ , dass bei der Übertragungsfunktion  $G(s)$  des linearen Systems für das Zählerpolynom  $Z(s) = 1$  gilt und nur ein Nennerpolynom  $N(s)$  der Ordnung  $n$  vorhanden ist – die Übertragungsfunktion besitzt vollen Relativgrad. Wenn aber nun ein Zählerpolynom der Ordnung  $m$  vorliegt, d.h. es existieren  $m$  Nullstellen, dann verringert sich der Relativgrad zu  $\delta = n - m \geq 0$  (es gilt stets  $m \leq n$  in einem kausalen System).

Ein System mit der Übertragungsfunktion

$$G(s) = \frac{Z(s)}{N(s)} \quad (12.3)$$

mit Zähler  $Z(s)$  und Nenner  $N(s)$  soll derart zerlegt werden, dass im Vorwärtszweig eine Integratorkette der Ordnung  $\delta$  mit lediglich Rückkopplungen auf den Eingang und im Rückwärtszweig ein Subsystem der Ordnung  $m = n - \delta$  vorliegt. Hierzu wird die Polynomdivision

$$G(s)^{-1} = \frac{N(s)}{Z(s)} = Q(s) + \frac{R(s)}{Z(s)} \quad (12.4)$$

durchgeführt, womit resultiert, dass sich im Vorwärtszweig die Übertragungsfunktion

$$\frac{1}{Q(s)} \quad (12.5)$$

mit der Ordnung  $\delta$  und im Rückwärtszweig die Übertragungsfunktion

$$\frac{R(s)}{Z(s)} \quad (12.6)$$

mit der Ordnung  $m = n - \delta$  befindet. Als *Nulldynamik* wird im linearen Fall das Subsystem  $R/Z$  im Rückwärtszweig bezeichnet und existiert nur, wenn es Nullstellen gibt. Bereits an dieser Stelle ist ersichtlich, dass bei linearen Systemen die Nulldynamik durch die Nullstellen bestimmt ist. Die Position der Nullstellen ist durch das Polynom  $Z(s)$  gegeben, das sich im Nenner der Nulldynamik befindet und daher gleichzeitig die Pole der Nulldynamik festlegt. Aus diesem Grund stimmen die Nullstellen des Systems mit den Polstellen der Nulldynamik überein.

Die Wurzeln des Zählerpoly nomes  $Z(s)$  werden als Nullstellen der Übertragungsfunktion  $G(s)$  bezeichnet. Dagegen heißen die Wurzeln des Nennerpoly nom  $N(s)$  Pole von  $G(s)$  und legen die Eigendynamik des Systems fest. Es handelt sich hierbei um die Bewegung des Systems, die aufgrund der Anfangswerte innerer Systemzustände hervorgerufen wird. Anhand der Realteile der Polstellen lässt sich die Stabilität des Systems ablesen. Es ist zu betonen, dass die

Nullstellen keinerlei Bedeutung für die Eigendynamik besitzen. Der Einfluss des Zählerpolynoms  $Z(s)$  bzw. der dadurch festgelegten Nullstellen tritt ausschließlich im Zusammenhang mit der durch die Eingangsgröße erzwungenen Bewegung zu Tage – das Zählerpolynom kann als Filterung des Eingangssignales interpretiert werden. Deshalb zeigt sich die Wirkung von  $Z(s)$  vor allem bei Änderungen der Eingangsgröße, wie im nachfolgenden Abschnitt dargestellt werden soll.

### 12.2.2 Auswirkung von Nullstellen auf die Impulsantwort

Es seien zwei Übertragungsfunktionen angenommen, die sich nur im Zählerpolynom unterscheiden;  $G_1(s)$  besitzt keine Nullstelle,  $G_2(s)$  hingegen eine stabile Nullstelle bei  $s = -1$ :

$$G_1(s) = \frac{1}{(s+2)(s+3)} = \frac{1}{s+2} - \frac{1}{s+3} \quad \Rightarrow \delta = 2 \quad (12.7)$$

$$G_2(s) = \frac{s+1}{(s+2)(s+3)} = \frac{-1}{s+2} + \frac{2}{s+3} \quad \Rightarrow \delta = 1 \quad (12.8)$$

Diese beiden Systeme befinden sich in Ruhe und werden nun mit einem Dirac-Impuls beaufschlagt. Die Impulsantworten im Zeitbereich ergeben sich durch Anwenden der Laplace-Rücktransformation auf die Übertragungsfunktion. Hierbei geht die Übertragungsfunktion  $1/(s+\alpha)$  im  $s$ -Bereich über in die Funktion  $e^{-\alpha t}$  im Zeitbereich. Die Impulsantworten ergeben sich daher zu:

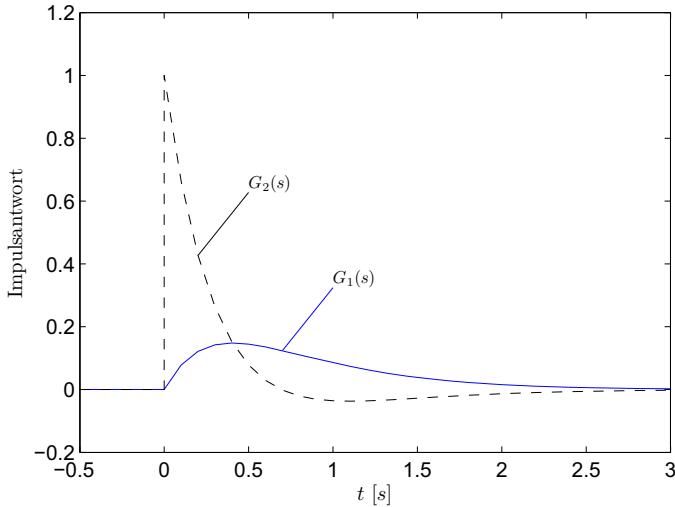
$$G_1(s) \xrightarrow{\bullet \mathcal{L}} y_1(t) = e^{-2t} - e^{-3t} \quad (12.9)$$

$$G_2(s) \xrightarrow{\bullet \mathcal{L}} y_2(t) = -e^{-2t} + 2e^{-3t} \quad (12.10)$$

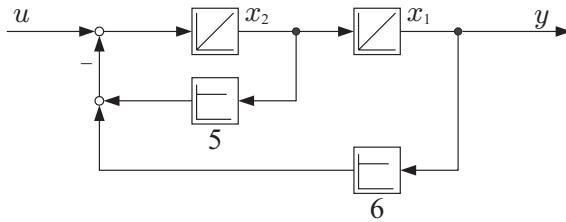
Wie ebenfalls die Simulation in Abbildung 12.4 zeigt, ist nach Aufbringen des Impulses ein markanter Unterschied im Signalverlauf zu sehen, welcher ausschließlich auf die Nullstelle zurückzuführen ist. Bemerkenswert ist vor allem die Tatsache, dass die Impulsantwort von  $G_2(s)$  keine stetige Funktion ist, diejenige von  $G_1(s)$  hingegen schon. Dieser Sachverhalt liegt im unterschiedlichen Relativgrad begründet und lässt sich wiederum mittels Byrnes-Isidori Form erklären.

Werden die beiden Beispielsysteme  $G_1(s)$  und  $G_2(s)$  nach Morse in Byrnes-Isidori Form transformiert, entstehen die Signalflusspläne, die in den Abbildungen 12.5 und 12.6 gezeigt sind.

Die zusätzliche Nullstelle bei  $G_2(s)$  reduziert den Relativgrad, weshalb der Vorwärtzweig nur aus einem Integrator besteht und die zusätzliche Dynamik im Rückführzweig angeordnet ist. Wird der Eingang von  $G_2(s)$  mit einem Dirac-Impuls erregt, so liegt dieser Impuls unverfälscht am Eingang des Integrators im Vorwärtzweig an. Weil das System voraussetzungsgemäß in der Ruhelage startet, gilt  $x_2(0) = 0$ , der Rückführzweig liefert daher keinen Beitrag für das



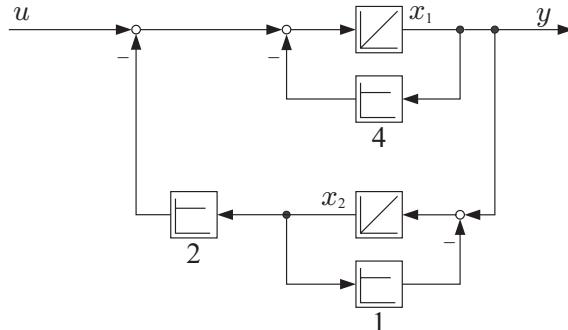
**Abb. 12.4:** Impulsantwort zweier Systeme mit gleichen Polen jedoch unterschiedlichen Nullstellen



**Abb. 12.5:** Beispielsystem  $G_1(s)$  in Byrnes-Isidori Form

Eingangssignal des Vorwärtszweiges. Ein Integrator, der mit einem Dirac beaufschlagt wird, antwortet mit einem Sprung am Ausgang. Wegen  $x_1(t) = y(t)$  ist dieser Sprung als Unstetigkeit in der Impulsantwort  $y_2(t)$  sichtbar.

Eine ähnliche Argumentation trifft für  $G_1(s)$  zu. Es ist wegen  $\delta = n$  keine Nulldynamik vorhanden, weswegen der Dirac wiederum unverfälscht am ersten Integrator des Vorwärtszweiges anliegt. Auch dessen Ausgang (der Zustand  $x_2$ ) antwortet mit einem Sprung. Da allerdings – bedingt durch die fehlende Nullstelle – ein zweiter Integrator im Vorwärtszweig liegt, wird dieses sprungförmige Signal nochmals integriert, es entsteht eine stetige Zustandsgröße  $x_1(t)$ , die sich am Ausgang zeigt.



**Abb. 12.6:** Beispielsystem  $G_2(s)$  in Byrnes-Isidori Form

### 12.2.3 Auswirkung von Nullstellen auf den geschlossenen Regelkreis

Um den von Isidori [113] geprägten Begriff der Nulldynamik (1999) für lineare und nichtlineare Systeme näher zu erläutern, wird im folgenden zuerst die Nulldynamik für lineare Systeme  $G(s)$  gemäß (12.3) weiter diskutiert und der Einfluss von Nullstellen tiefergehend untersucht.

Wie im Anschluss gezeigt wird, haben Nullstellen nicht nur einen erheblichen Einfluss auf das dynamische Ein-Ausgangsverhalten eines Systems, sondern auch auf den geschlossenen Kreis. Es lässt sich anhand eines einfachen Beispiels demonstrieren, dass eine Nullstelle im geschlossenen Regelkreis zur Instabilität führen kann.

Es ist bekannt, dass das Nennerpolynom  $N(s)$  die charakteristische Gleichung ist, die das statische Verhalten und insbesondere die dynamischen Eigenschaften des Systems beschreibt. Das Zählerpolynom  $Z(s)$  beschreibt stattdessen die Signalverarbeitung des Eingangssignals  $u(s)$  des Systems, welches das dynamische Verhalten vor allem bei Eingangssignal-Änderungen beeinflusst. Es ist weiter bekannt, dass Nullstellen von  $N(s)$  in der linken  $s$ -Halbebene ein stabiles und Pole in der rechten  $s$ -Halbebene ein instabiles Systemverhalten repräsentieren. Nullstellen stehen somit in keiner direkten Verbindung mit der Instabilität eines Systems, es sei jedoch betont, dass in einem geregelten Gesamtsystem Nullstellen mit einer entsprechenden Rückführung zu Polen in der rechten Halbebene und folglich zur Instabilität des geregelten Systems führen können. Bei nicht-minimalphasigen Systemen, die sich durch Nullstellen in der rechten Halbebene auszeichnen, ist dies beispielsweise darauf zurückzuführen, dass bei sehr hoher Reglerverstärkung  $m$  Pole in die  $m$  Nullstellen wandern. Liegen instabile Nullstellen vor, so werden Pole des geregelten Gesamtsystems instabil, was folgendes Beispiel zeigt: Das System

$$G_v(s) = \frac{(s+q)}{(s+2)(s+3)}, \quad q = -1 \quad (12.11)$$

besitzt zwei stabile Pole und eine instabile Nullstelle  $s = -q = 1$ , d.h. es liegt ein nichtminimalphasiges System, bzw. System mit instabiler Nulldynamik vor. Dieses soll nun mit einem proportionalen Regler

$$G_r(s) = -a \quad (12.12)$$

geregelt werden. Untersucht man das geregelte Gesamtsystem

$$G_w(s) = \frac{1}{\frac{1}{G_v(s)} + G_r(s)} = \frac{(s-1)}{(s+2)(s+3) + a(s-1)} \quad (12.13)$$

$$G_w(s) = \frac{s-1}{s^2 + s(5+a) + 6-a} \quad (12.14)$$

bzgl. Stabilität, so ist nach Hurwitz das Nennerpolynom zweiter Ordnung stabil, sofern alle Koeffizienten positiv sind. Dies ist nicht mehr der Fall für  $a \geq 6$ . Lage einer Nullstelle mit negativem Realteil bei  $s = -q < 0$  vor, d.h. wäre das System minimalphasig, so würde das resultierende Polynom  $s^2 + s(5+a) + 6 + aq$  für alle positiven Verstärkungen stabil sein. Dieses Beispiel macht deutlich, dass auch die Nullstellen auf das System-Verhalten einen entscheidenden Einfluss haben können bis hin zur Instabilität.

Nichtminimalphasige Systeme besitzen Nullstellen in der rechten Halbebene, wobei  $Z(s)$  kein Hurwitzpolynom ist. Anhand der Byrnes-Isidori Form wird sofort ersichtlich, dass der Rückwärtszweig in diesem Fall ein instabiles System besitzt. Bezogen auf das Gesamtsystem bedeutet dies zwar noch keine Instabilität, jedoch ergibt sich ein Gesamtsystem, welches schwieriger zu regeln ist; eine Eigenschaft ist zum Beispiel anfängliche Richtungsumkehr der Sprungantwort. Des Weiteren ist die Gefahr von Instabilität im geschlossenen Regelkreis mit einem nichtminimalphasigen System erhöht, wie obiges Beispiel aufzeigt. Diese Erkenntnis lässt sich grundsätzlich auch auf nichtlineare Systeme übertragen, d.h. nichtlineare Systeme mit einer stabilen Nulldynamik sind weitaus einfacher zu regeln. Die in den Kapiteln 12.2.5, 12.4 und 12.5.1 diskutierte Ein-Ausgangslinearisierung kann bei instabiler Nulldynamik beispielsweise nicht angewandt werden, was einfach zu verstehen ist: wird durch die Aufschaltung die Eigenschaft des instabilen Rückwärtszweiges nicht exakt kompensiert, so zeigt sich ein ansteigendes Signal im Rückwärtszweig, welches durch eine ansteigende Stellgröße nur begrenzt kompensiert werden kann. Im linearen Fall wäre für die Ein-Ausgangslinearisierung eine instabile Pol-Nullstellen-Kompensation nötig, um das Verhalten einer Integratorkette zu erreichen, was jedoch in der Praxis nicht durchführbar ist.

### 12.2.4 Unterdrückung von Eingangssignalen durch Nullstellen

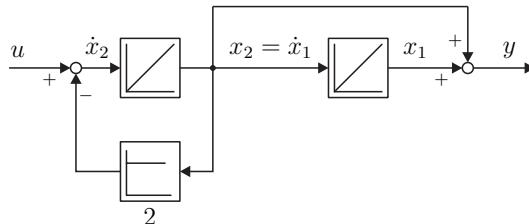
Bisher wurde ausführlich die Auswirkung von Nullstellen diskutiert, deren Bezeichnung lässt sich allerdings daraus nicht ableiten. Die Namensgebung erklärt sich aus der Tatsache, dass eine Nullstelle Eingangssignale mit bestimmter Frequenz blocken kann. Insofern ist die Nullstelle das Gegenteil des Resonanzeffektes, bei dem eine vorhandene Polstelle (bzw. ein konjugiert komplexes Polpaar) ein Eingangssignal mit der Resonanzfrequenz hoch verstärkt.

Die Situation, dass eine Übertragungsnullstelle das Eingangssignal auslöscht, soll an einem weiteren Beispiel erläutert werden. Gegeben ist folgendes System in Zustandsdarstellung

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad (12.15)$$

$$y(t) = [1 \ 1] \underline{x}(t) \quad (12.16)$$

mit dem in Abbildung 12.7 gezeigten Signalflussplan.



**Abb. 12.7:** Signalflussplan der Strecke (12.15) in Regelungsnormalform

Daran ist sofort zu erkennen, dass der Relativgrad  $\delta = 1$  ist, da  $x_2$  zum Ausgang geführt ist und der kürzeste Signalweg vom Eingang zum Ausgang über nur einen Integrator führt. Mittels elementarer Rechnung lässt sich die Übertragungsfunktion  $G(s)$  berechnen:

$$G(s) = \frac{(s+1)}{s(s+2)} \quad (12.17)$$

Wenn nun ein Steuersignal  $u(t) = e^{-t}$  bzw.  $u(s) = 1/(s+1)$  auf das System mit der Übertragungsfunktion  $G(s)$  wirkt, dann ergibt sich unter der Voraussetzung der Anfangsbedingung  $\underline{x}(0) = 0$

$$y(s) = G(s)u(s) = \frac{1}{s(s+2)} = \frac{1}{2} \cdot \frac{1}{\frac{s}{2} + 1} \cdot \frac{1}{s}. \quad (12.18)$$

Es zeigt sich somit das dynamische Verhalten eines PT<sub>1</sub>-Gliedes mit der Zeitkonstanten  $T = 1/2$ , welches von der Einheitssprungfunktion  $\sigma(t) \circ \bullet 1/s$  angeregt wird. Im Zeitbereich lautet die zugehörige Ausgangsfunktion:

$$y(t) = \frac{1}{2}(1 - e^{-2t}) \quad (12.19)$$

Dies bedeutet, das Eingangssignal  $u(t)$  regt lediglich die Eigendynamik des Systems an. Im Ausgangsignal sind zwei Anteile enthalten. Einerseits der konstante Teil  $1/2$ , der vom global proportionalen Verhalten des PT<sub>1</sub>-Gliedes herröhrt. Andererseits der Term  $e^{-2t}$ , der durch die Polstelle bei  $s = -2$  generiert wird. Nicht sichtbar ist im Ausgang ein Teil mit  $e^{-t}$ , obwohl eine solche Exponentialfunktion im Eingang enthalten ist. Damit kann das Eingangssignal keine Exponentialfunktion mit der Zeitkonstanten  $1$  im Ausgangssignal  $y(t)$  erregen, da sich das Zählerpolynom  $Z(s) = (s+1)$  und das Eingangssignal  $u(s) = 1/(s+1)$  kompensieren.

Wenn statt der Anfangsbedingung  $\underline{x}(0) = \underline{0}$  nun  $x_1(0) = -1$  und  $x_2(0) = 1$  angenommen wird, dann ist aus der Abbildung 12.7 zu erkennen, dass  $y(0) = 0$  sein wird. Wenn dieses System mit den obigen Anfangsbedingungen  $\underline{x}(0) = [-1, 1]^T$  wiederum mit dem Eingangssignal

$$u(t) = e^{-t} \quad \xrightarrow{\mathcal{L}} \quad u(s) = \frac{1}{s+1} \quad (12.20)$$

angeregt wird, dann gilt für  $x_2(t)$

$$\dot{x}_2(t) = -2x_2(t) + e^{-t}, \quad (12.21)$$

bzw. im Laplace-Bereich:

$$sx_2(s) - x_2(0) = -2x_2(s) + \frac{1}{s+1} \quad (12.22)$$

$$x_2(s) = \frac{1}{s+2}x_2(0) + \frac{1}{s+2} \underbrace{\frac{1}{s+1}}_{u(s)} \quad (12.23)$$

Durch Rücktransformation in den Zeitbereich erhält man für  $x_2(t)$  den Verlauf

$$x_2(t) = -e^{-2t} + e^{-t} + e^{-2t}x_2(0) \quad (12.24)$$

woraus sich mit dem Anfangszustand  $x_2(0) = 1$  das Ergebnis

$$x_2(t) = e^{-t} \quad (12.25)$$

errechnet. Aus Abbildung 12.7 ist darüberhinaus zu erkennen, dass sich  $x_1(t)$  durch eine Integration des Zustandes  $x_2(t)$  ergibt. Somit folgt für  $x_1(t)$  unter der Nebenbedingung  $x_1(0) = -1$ :

$$\begin{aligned} x_1(t) &= \int_0^t x_2(\tau) d\tau + x_1(0) = \\ &= [-e^{-\tau}]_0^t - 1 = -e^{-t} + e^0 - 1 = -e^{-t} \end{aligned} \quad (12.26)$$

Der Ausgang ist die Summe beider Zustände und errechnet sich somit zu

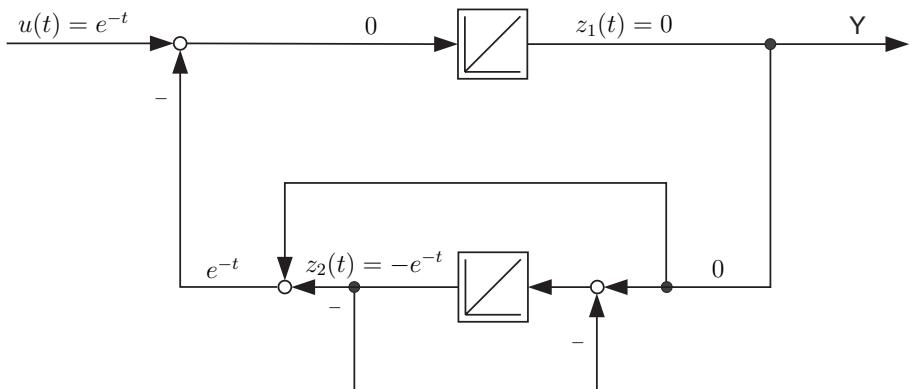
$$y(t) = x_1(t) + x_2(t) = -e^{-t} + e^{-t} \equiv 0 \quad \text{für alle } t. \quad (12.27)$$

Dies bedeutet, dass aufgrund der gewählten Anfangsbedingungen  $\underline{x}(0) \neq 0$  und der Filterung durch das Zählerpolynom das Eingangssignal nicht zum Ausgang abgebildet wird. Durch die Nullstelle geschieht eine Auslöschung (Filterung) des Eingangssignales.

Wird das System, das in Abbildung 12.7 in Regelungsnormalform abgebildet ist, in Byrnes-Isidori Form gebracht, zeigt sich wiederum der Grund für die Auslöschung des Eingangssignales. Die Transformation erfolgt mit der Abbildung:

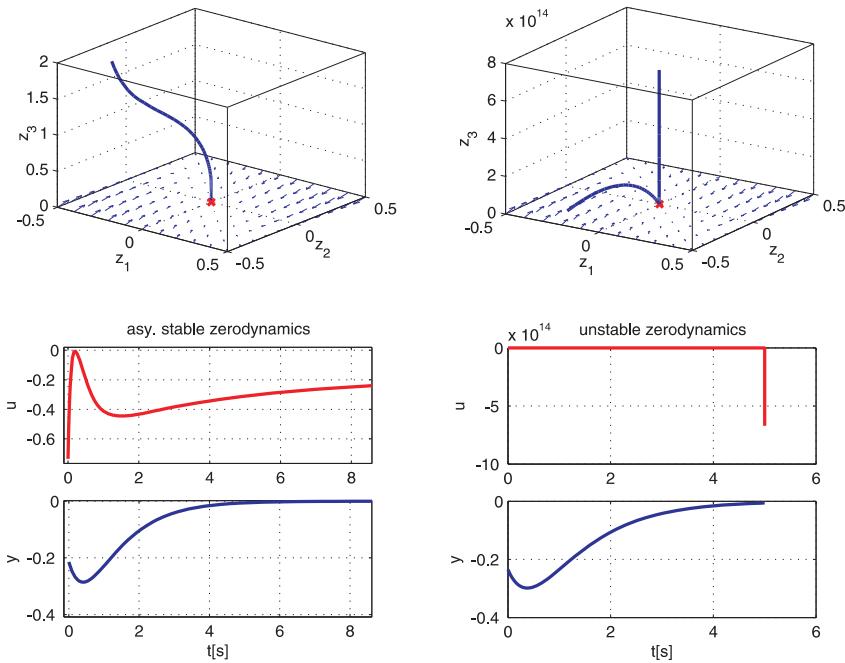
$$\underline{z}(t) = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \underline{x}(t) \quad \Leftrightarrow \quad \underline{x}(t) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \underline{z}(t) \quad (12.28)$$

Die Anfangszustände  $\underline{x}(0) = [-1, 1]^T$  übersetzen sich daher in  $\underline{z}(0) = [0, -1]^T$ . Das System ist durch die gewählten Anfangszustände derart initialisiert, dass der Integrator im Vorwärtszweig am Ausgang den Wert Null liefert. Dadurch erhält die Nulldynamik keine Erregung, die als homogenes System einzig ihrer Eigendynamik unterworfen ist. Weil der Zustand in der Nulldynamik auf den Anfangszustand  $z_2(0) = -1$  initialisiert wurde, wird das Signal  $-e^{-t}$  generiert und somit der Eingang ausgelöscht. Aus diesem Grund behält der Integrator seinen Anfangszustand bei, es gilt  $z_1(t) = 0$ .



**Abb. 12.8:** Signalflussplan der Strecke (12.15) in Byrnes-Isidori Form

Wenn durch den Verlauf der Nulldynamik das Eingangssignal unterdrückt werden kann, lässt sich im Umkehrschluss durch ein geeignetes Eingangssignal auch die Auswirkung der Nulldynamik kompensieren. Diesen Effekt macht sich zum Beispiel die exakte Ein-Ausgangslinearisierung (Kap. 12.4) zu Nutze.



**Abb. 12.9:** Stabilisierung der Nulllage  $z = 0$  durch das nichtlineare Regelgesetz (12.33) mit  $r_0 = 2, r_1 = 3$

### 12.2.5 Nulldynamik im nichtlinearen System

Für ein nichtlineares System wird das Problem der Ausgangsnullung betrachtet. Es stellt sich die Frage, unter welchen Anfangszuständen ein Eingangssignal  $u(t) \not\equiv 0$  gefunden werden kann, das den Ausgang nicht erreicht und, wie oben am linearen Beispiel beschrieben, geblendet wird, d.h. zu einem Ausgang  $y(t) \equiv 0$  führt. Die Definition der Nulldynamik und ihr Beitrag zur Stabilität des Gesamtsystems wird an einem Beispiel illustriert.

Das System sei bereits in der Normalform – siehe Kapitel 12.3 und 12.5.1 – gegeben:

$$\dot{z}_1 = z_2 \quad (12.29)$$

$$\dot{z}_2 = \underbrace{z_1 \cos z_2 + 2 z_2^2 + z_3}_{\beta(z)} + \underbrace{\frac{e^{z_2}}{1+z_1^2} u}_{\alpha(z)} = v \quad (12.30)$$

$$\dot{z}_3 = -z_3^3 \quad (12.31)$$

Man erkennt, dass für den Relativgrad  $\delta = 2$  gilt. Damit ist der Zustand  $z_3$  der Nulldynamik des Systems zugeordnet. Man beachte, dass die entsprechende Differentialgleichung aber nur dann die Nulldynamik beschreibt, wenn der Eingang  $u$  und die Anfangsbedingungen so gewählt wurden, dass  $z_1(t) \equiv z_2(t) \equiv 0$  gilt. Da jedoch die rechte Seite in (12.31) in diesem Fall nicht von  $z_1$  oder  $z_2$  abhängt, kann bereits jetzt festgestellt werden, dass der Ursprung in Gleichung (12.31) asymptotisch stabil ist. Wir bestimmen das Regelgesetz für beliebige Anfangsbedingungen  $z(0)$  entsprechend der Gleichungen

$$u = \frac{-\beta(z) + v}{\alpha(z)} \quad \text{und} \quad v = w - \underline{r}^T \underline{z} \quad (12.32)$$

zu

$$\begin{aligned} u &= -\frac{\beta(z) + r_0 z_1 + r_1 z_2}{\alpha(z)} \\ &= -\frac{1 + z_1^2}{e^{z_2}} [z_1 \cos z_2 + 2 z_2^2 + z_3 + r_0 z_1 + r_1 z_2] \end{aligned} \quad (12.33)$$

wobei  $w = 0$  gewählt wird. Man erkennt, dass  $z_3$  einen Beitrag zum Stellsignal  $u$  liefert, der jedoch abklingt, da  $z_3$  selbst Lösung eines asymptotisch stabilen Systems (12.31) ist. Der geschlossene Regelkreis ist teilweise linear und insgesamt asymptotisch stabil.

$$\dot{z}_1 = z_2 \quad (12.34)$$

$$\dot{z}_2 = -r_0 z_1 - r_1 z_2 \quad (12.35)$$

$$\dot{z}_3 = -z_3^3 \quad (12.36)$$

Ersetzt man (12.31) durch

$$\dot{z}_3 = z_3^2 \quad (12.37)$$

so entspricht dies offenbar einer instabilen Nulldynamik ( $z_3$  „explodiert“ in diesem Fall sogar, da die rechte Seite  $z_3^2$  keine globale Lipschitz-Bedingung erfüllt, siehe z.B. Khalil, 1996 [122]). Da  $z_3$  unbeschränkt anwächst, wächst auch das zur Stabilisierung notwendige Stellsignal  $u$ . Da die Stellenergie in jedem technischen System beschränkt ist, kann dieses System praktisch nicht stabilisiert werden. Die Simulation in Abbildung 12.9 stellt die beiden Fälle gegenüber.

### 12.2.6 Analogie zwischen Kompensation der Nulldynamik und Pol-Nullstellen-Kürzung

Die ausführlich diskutierte Auslöschung der Nulldynamik durch eine geeignete Stellgröße entspricht genau einer Pol-Nullstellen-Kompensation, wie am linearen Beispiel gezeigt werden kann. Im Prinzip handelt es sich bei der Ein-Ausgangslinearisierung um einen Zustandsregler, der die Nulldynamik (und damit auch die darin enthaltenen Nichtlinearitäten) unbeobachtbar macht, indem die Pole auf die Nullstellen geschoben werden.

Um diese Aussage am linearen Beispiel darzulegen, wird wiederum das System (12.15) aufgegriffen. Aus der Zustandsbeschreibung (12.15) ergibt sich die zugehörige Übertragungsfunktion zu:

$$G(s) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} s & -1 \\ 0 & s+2 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{s+1}{s(s+2)} = \frac{1}{2s} + \frac{1}{2(s+2)}$$
(12.38)

Deren Polstellen befinden sich bei

$$p_1 = 0 \quad \text{und} \quad p_2 = -2, \quad (12.39)$$

die Nullstelle dagegen liegt bei

$$n_1 = -1. \quad (12.40)$$

Mittels Partialbruchzerlegung errechnen sich die Residuen

$$r_1 = \frac{1}{2} \quad \text{und} \quad r_2 = \frac{1}{2}. \quad (12.41)$$

Mit deren Kenntnis lässt sich das System (12.15), das dort in Regelungsnormalf orm dargestellt ist, in modale Komponenten zerlegen und als Parallelsystem realisieren:

$$\dot{\underline{x}}(t) = \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix} \underline{x}(t) + \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} u(t) \quad (12.42)$$

$$y = (1 \ 1) \underline{x}(t) \quad (12.43)$$

Mit den oben errechneten Daten ergibt sich das Parallelsystem zahlenmäßig zu:

$$\dot{\underline{x}}(t) = \begin{pmatrix} 0 & 0 \\ 0 & -2 \end{pmatrix} \underline{x}(t) + \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(t) \quad (12.44)$$

$$y = (1 \ 1) \underline{x}(t) \quad (12.45)$$

Eine Darstellung als Signalfussplan ist in Abbildung 12.10 gezeigt.

Anhand der Darstellung in Abbildung 12.10 lässt sich sehr einfach ein Zustandsregelgesetz ableiten, das den Pol  $p_2 = -2$  auf die Nullstelle  $n_1 = -1$  schiebt und dadurch die Ordnung des Ein-Ausgangsverhaltens reduziert. Es ist unschwer zu erkennen, dass der entsprechende Zustandsregler

$$u(t) = -p_2 x_2 + 2w = 2x_2 + 2w \quad (12.46)$$

angesetzt werden muss, wobei eine Rückführung des Zustandes  $x_1$  nicht notwendig ist. Das Signal  $w$  bezeichne hierbei einen beliebigen Sollwert. Mit diesem Regler ergibt sich der geschlossene Regelkreis zu:

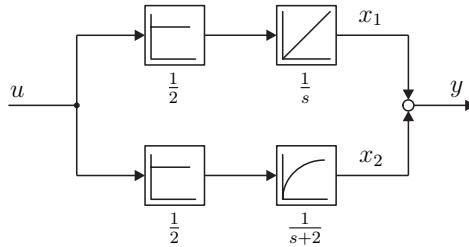


Abb. 12.10: Signalflussplan des Beispielsystem in Modalform (Parallelstruktur)

$$\dot{\underline{x}} = \begin{pmatrix} (1-r_1)p_1 & -r_1 p_2 \\ -r_2 p_1 & (1-r_2)p_2 \end{pmatrix} \underline{x} + \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} 2w \quad (12.47)$$

Werden die Zahlenwerte  $p_1 = 0$ ,  $p_2 = -2$ ,  $r_1 = r_2 = 1/2$  eingesetzt, folgt die Beschreibung:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}}_{=A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{=b} w \quad (12.48)$$

Weil die Systemmatrix  $A$  des geschlossenen Regelkreises eine Dreiecksmatrix ist, lassen sich die Pole über die Diagonalelemente ablesen und sind somit durch

$$p_1 = 0 \quad \text{und} \quad p_2 = -1 \quad (12.49)$$

gegeben. Bekanntlich werden die Positionen der Nullstellen einer linearen Regelstrecke durch einen Zustandsregler nicht beeinflusst. Deshalb bleibt die vorhandene Nullstelle  $n_1 = -1$  unverändert bestehen. Der Zustandsregler verschiebt jedoch eine der Polstellen nach  $-1$  und plaziert damit diese auf der Nullstelle. In der Folge resultiert eine Pol-Nullstellen-Kürzung, der Effekt der bestehenden Nullstelle wird kompensiert.

Grundsätzlich äußert sich eine Pol-Nullstellen-Kürzung im Verlust der Steuerbarkeit bzw. Beobachtbarkeit, abhängig von der vorliegenden Zustandsbeschreibung. Im Falle der Parallelstruktur geht die Beobachtbarkeit verloren, wie anhand einer Überprüfung der Kalman'schen Beobachtbarkeitsmatrix gezeigt werden kann. Weil die Beobachtbarkeitsmatrix

$$Q_B = \begin{bmatrix} c \\ cA \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1-1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad (12.50)$$

eine Nullzeile enthält, ist der Rangabfall sofort offensichtlich. Wegen

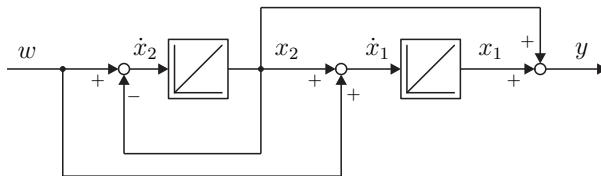
$$\det(Q_B) = 0 \quad (12.51)$$

gilt  $\text{Rang}(Q_B) \leq 1$  und ist damit kleiner als die Systemordnung  $n = 2$ . Dieser Sachverhalt zeugt nach dem Kalman'schen Beobachtbarkeitskriterium für

ein nicht vollständig beobachtbares System. Gleichzeitig äußert sich die Pol-Nullstellen-Kürzung in der Reduktion der Ordnung der Übertragungsfunktion. Das Ein-Ausgangsverhalten des geschlossenen Kreises wird beschrieben durch:

$$G_w(s) = [1 \ 1] \begin{bmatrix} s & -1 \\ 0 & s+1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{2s+2}{s(s+1)} = 2 \frac{s+1}{s(s+1)} = \frac{2}{s} \quad (12.52)$$

Daran lässt sich ablesen, dass der Teil mit  $1/(s+1)$ , welcher der Nulldynamik zuzuordnen ist, durch den Zustandsregler unbeobachtbar gemacht wird. Allerdings ist diese Tatsache am Signalflussplan des geregelten Systems in Abbildung 12.11 nicht direkt einsichtig.



**Abb. 12.11:** Signalflussplan des geregelten Systems

Um den Verlust der Beobachtbarkeit auch aus dem Signalflussplan entnehmen zu können, ist eine Ähnlichkeitstransformation notwendig, die aus Vollständigkeitsgründen aufgeführt wird.

Gegeben sei ein lineares System, dessen Zustandsbeschreibung in  $x$ -Koordinaten vorliegt, d.h. der Zustandsvektor ist durch den Vektor  $\underline{x}$  festgelegt. Ziel ist, die Systembeschreibung in  $\xi$ -Koordinaten zu überführen, ohne dabei das Ein-Ausgangsverhalten zu verändern. Diese Aufgabe ist lösbar, da zu einem gegebenen Ein-Ausgangsverhalten nicht eine eindeutige Zustandsbeschreibung existiert, sondern das gegebene Ein-Ausgangsverhalten durch unendlich viele verschiedene Zustandsbeschreibungen realisiert werden kann. Die Überführung (Transformation) der  $\underline{x}$ - in  $\xi$ -Koordinaten entspricht einer linearen Abbildung des Zustandsvektors  $\underline{x}$  in den Zustandsvektor  $\xi$  gemäß der Abbildungsvorschrift

$$\xi = T^{-1}x \quad \Leftrightarrow \quad x = T\xi, \quad (12.53)$$

wobei  $T \in \mathbb{C}^{n \times n}$  eine invertierbare Transformationsmatrix ist. Obgleich häufig reelle Transformationsmatrizen verwendet werden, sei angemerkt, dass  $T$  durchaus komplexe Einträge enthalten darf. So gelingt beispielsweise der Wechsel zwischen Modalform und Regelungsnormalform mit der VanderMonde-Matrix, die sämtliche Eigenwerte der Strecke in den Potenzen von 0 bis  $n-1$  enthält und daher im Allgemeinen komplexe Einträge besitzt.

Wird die Abbildungsvorschrift (12.53) nach der Zeit differenziert, ergibt sich

$$\dot{x} = T\dot{\xi}. \quad (12.54)$$

Nach Substitution in die Zustandsdifferentialgleichung

$$\dot{x} = Ax + bw \quad (12.55)$$

folgt zusammen mit (12.53)

$$T\dot{\xi} = AT\xi + bw. \quad (12.56)$$

Wird diese Gleichung von links mit der inversen Transformationsmatrix  $T^{-1}$  multipliziert, erhält man die Beschreibung des Systems in  $\xi$ -Koordinaten:

$$\dot{\xi} = T^{-1}AT\xi + T^{-1}bw \quad (12.57)$$

$$y = cT\xi \quad (12.58)$$

Angewandt auf das vorliegende Beispiel wird die Transformationsmatrix

$$T = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (12.59)$$

angesetzt, deren Inverse durch

$$T^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (12.60)$$

gegeben ist. Mit dieser Matrix ergibt sich:

$$T^{-1}AT = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \quad (12.61)$$

$$T^{-1}b = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad (12.62)$$

$$cT = [1 \ 1] \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = [1 \ 0] \quad (12.63)$$

Diese Berechnungen führen auf das transformierte System:

$$\begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} w \quad (12.64)$$

$$y = [1 \ 0] \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \xi_1 \quad (12.65)$$

Dessen Signalflussplan ist in Abbildung 12.12 dargestellt. Hieran ist deutlich zu erkennen, dass ein Subsystem der Ordnung eins nicht auf den Ausgang einwirkt. Der Zustand  $\xi_2$  hängt zwar vom Ausgang  $y$  ab (über das Signal  $y$  ist daher Steuerbarkeit gegeben), beeinflusst diesen jedoch nicht. Das unbeobachtbare Teilsystem war ursprünglich im ungeregelten System beobachtbar. Durch die spezielle Auslegung des Zustandsreglers bedingt, ist die Auswirkung der Nulldynamik auf den Ausgang eliminiert worden.

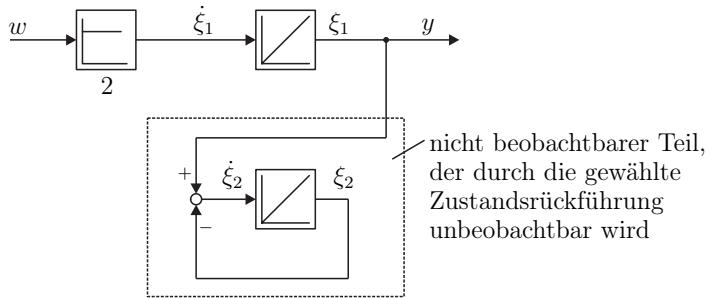


Abb. 12.12: Signalflussplan des transformierten Systems

### 12.2.7 Zusammenfassung

Die Nulldynamik beschreibt das Verhalten des Systems, wenn Eingang  $u$  und Anfangsbedingungen gleichzeitig so gewählt wurden, dass der Ausgang  $y$  identisch Null gehalten wird. Weil die dazu notwendige Stellgröße im allgemeinen ungleich Null ist, beschreibt die Nulldynamik denselben Effekt wie die Nullstellen in der Übertragungsfunktion eines linearen Systems: bestimmte Eingangssignale „verschwinden im System“ bei bestimmten Anfangswerten, sind also nicht am Ausgang sichtbar (dieser ist gleich Null). Um die systemtheoretische Bedeutung dieses Effekts zu verstehen, führen wir die nichtlineare Regelungsnormalform ein.

## 12.3 Nichtlineare Regelungsnormalform

Um zur nichtlinearen Regelungsnormalform (oder “Byrnes–Isidori” Normalform) zu kommen, nimmt man an, die nichtlineare Strecke sei in folgender Form gegeben:

$$\dot{\underline{x}} = \underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot u \quad y = h(\underline{x}) \quad (12.66)$$

Das Regelgesetz der exakten E/A-Linearisierung wird auf der Grundlage einer Normalform berechnet, bei der das System in eine Integratorkette im Vorwärtszweig und die Nulldynamik im Rückführzweig zerlegt wird. (Beispiele sind in den Kapiteln 12.4.1 und 12.4.2 für  $\delta = n$  (keine Nulldynamik) und Kapitel 12.5.1 für  $\delta \neq n$  (Nulldynamik) zu finden.) Zur Berechnung der Normalform geht man wie folgt vor: Man differenziert die Ausgangsgleichung so oft nach der Zeit, bis auf der rechten Seite erstmals die Steuergroße  $u$  erscheint.

$$\begin{aligned}
y &= h(\underline{x}) \\
\dot{y} &= \frac{\partial h(\underline{x})}{\partial \underline{x}} \frac{\partial \underline{x}}{\partial t} \quad (\text{Kettenregel der Diff.}) \\
&= \frac{\partial h(\underline{x})}{\partial \underline{x}} [\underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot u] \\
&= L_f h(\underline{x}) + \underbrace{L_g h(\underline{x}) \cdot u}_{= 0} \\
&\vdots \\
y^{(\delta)} &= L_f^\delta h(\underline{x}) + \underbrace{L_g L_f^{\delta-1} h(\underline{x}) \cdot u}_{\neq 0} = v
\end{aligned} \tag{12.67}$$

Die partielle Ableitung der skalaren Funktion  $h(x)$  nach ihrem Argument liefert den Zeilenvektor  $\partial h(x)/\partial x$ , der bekanntermaßen als Gradient der Funktion  $h(x)$  bezeichnet wird.  $L_f h(\underline{x})$  steht für die Richtungs-Ableitung von  $h(\underline{x})$  entlang  $\underline{f}(x)$  und  $L_g h(\underline{x})$  die Ableitung von  $h(\underline{x})$  entlang  $\underline{g}(x)$ :

$$\begin{aligned}
L_f h(\underline{x}) &= \frac{\partial h(\underline{x})}{\partial \underline{x}} \underline{f}(\underline{x}) \\
L_g h(\underline{x}) &= \frac{\partial h(\underline{x})}{\partial \underline{x}} \underline{g}(\underline{x})
\end{aligned} \tag{12.68}$$

Diese Ableitungen werden auch *Lie*-Ableitungen genannt. Bei wiederholter Anwendung gelten die folgenden Beziehungen:

$$\begin{aligned}
L_f L_f h(\underline{x}) &= \frac{\partial(L_f h(\underline{x}))}{\partial \underline{x}} \underline{f}(\underline{x}) \\
&\vdots \\
L_f^i h(\underline{x}) &= \frac{\partial(L_f^{i-1} h(\underline{x}))}{\partial \underline{x}} \underline{f}(\underline{x}) \\
\\
L_g L_f h(\underline{x}) &= \frac{\partial(L_f h(\underline{x}))}{\partial \underline{x}} \underline{g}(\underline{x}) \\
L_g L_f^2 h(\underline{x}) &= \frac{\partial(L_f^2 h(\underline{x}))}{\partial \underline{x}} \underline{g}(\underline{x}) \\
&\vdots \\
L_g L_f^i h(\underline{x}) &= \frac{\partial(L_f^{i-1} h(\underline{x}))}{\partial \underline{x}} \underline{g}(\underline{x})
\end{aligned} \tag{12.69}$$

Die letzte Gleichung in (12.67) ist dadurch gekennzeichnet, dass der Ausdruck  $L_g L_f^i h(\underline{x})$  zum ersten Mal nicht verschwindet, d.h.  $L_g L_f^i h(\underline{x}) = 0$  für  $i = 0, \dots, \delta - 2$  und  $L_g L_f^{\delta-1} h(\underline{x}) \neq 0$  für  $i = \delta - 1$ . Zur Vereinfachung schreibt man oft

$$\begin{aligned}
y^{(\delta)} &= L_f^\delta h(\underline{x}) + \underbrace{L_g L_f^{\delta-1} h(\underline{x}) \cdot u}_{\neq 0} \\
&= \beta(\underline{x}) + \alpha(\underline{x}) \cdot u = v
\end{aligned} \tag{12.70}$$

mit

$$\begin{aligned}\beta(\underline{x}) &= L_f^\delta h(\underline{x}) \\ \alpha(\underline{x}) &= L_g L_f^{\delta-1} h(\underline{x})\end{aligned}\quad (12.71)$$

Der Wert  $\delta$  liefert den *Relativgrad* und somit die niedrigste Ableitung der Ausgangsgröße  $y$ , auf welche die Steuergröße  $u$  direkt einwirkt. Der Relativgrad  $\delta$  kann nur dann sinnvoll definiert werden, wenn die Ungleichung  $L_g L_f^i h(\underline{x}) \neq 0$  unabhängig von  $x$ , also unabhängig von dem Punkt des Zustandsraumes ist, an dem die Richtungsableitung bestimmt wurde. Gilt beispielsweise  $\underline{x} = [x_1, x_2, x_3]^T$  und  $L_g L_f^{\delta-1} h(\underline{x}) = x_1 + x_3$ , so kann in der Umgebung des Ursprungs  $\underline{x} = \underline{0}$  kein Relativgrad definiert werden, da  $L_g L_f^{\delta-1} h(\underline{0}) = 0$  während  $L_g L_f^{\delta-1} h(\underline{x}) \neq 0$  für alle  $\underline{x}$  in der Umgebung von  $\underline{0}$  gilt. Ist dagegen der Ausdruck  $L_g L_f^{\delta-1} h(\underline{x})$  unabhängig von  $\underline{x}$  (also konstant) so kann der Relativgrad im gesamten Zustandsraum definiert werden. Unter dieser Voraussetzung lässt sich das nichtlineare System durch Koordinatentransformation auf eine besonders einfache Form bringen.

Für die Transformation gilt allgemein:  $\underline{z} = \underline{\gamma}(\underline{x})$  wobei  $\underline{x}$  ein Spaltenvektor von der Dimension  $n$  sei und  $\underline{z}$  den (Spalten-)Vektor der neuen Zustandskoordinaten bezeichne, der ebenfalls von der Dimension  $n$  ist. Komponentenweise gilt  $z_i = \gamma_i(\underline{x})$ ,  $i = 1, \dots, n$  wobei  $\gamma_1, \dots, \gamma_n$  jeweils nichtlineare Funktionen des gesamten Zustandsvektors  $\underline{x}$  sind. Die Gleichung  $\underline{z} = \underline{\gamma}(\underline{x})$  lässt sich im allgemeinen nicht nach  $\underline{x}$  auflösen, da die Inverse der nichtlinearen Funktion  $\underline{\gamma}(\cdot)$  bestimmt werden müsste. Man weiß jedoch, dass die Inverse *lokal* zumindest existiert, und zwar in der Umgebung eines Punktes  $\underline{x}_0$  (z.B. ein Gleichgewichtspunkt), an dem die Jacobi-Matrix  $\frac{\partial \underline{\gamma}(\underline{x})}{\partial \underline{x}}|_{\underline{x}_0}$  vollen Rang besitzt. Dies ist der entscheidende Schritt bei der Wahl der Transformationsvorschrift:  $\gamma_i$ ,  $i = 1, \dots, n$  muss so gewählt werden, dass

$$\frac{\partial \underline{\gamma}(\underline{x})}{\partial \underline{x}}|_{\underline{x}_0} = \begin{bmatrix} \frac{\partial \gamma_1}{\partial \underline{x}} \\ \vdots \\ \frac{\partial \gamma_n}{\partial \underline{x}} \end{bmatrix}|_{\underline{x}_0} \quad (12.72)$$

vollen Rang besitzt. Man kann zeigen, dass die Gradienten der in Gleichung (12.67) bestimmten Funktionen  $h(\underline{x}), L_f h(\underline{x}), \dots, L_f^{\delta-1} h(\underline{x})$  einen  $\delta$ -dimensionalen Raum aufspannen. In anderen Worten, die Vektoren

$$\frac{\partial h(\underline{x})}{\partial \underline{x}}, \frac{\partial L_f h(\underline{x})}{\partial \underline{x}}, \dots, \frac{\partial L_f^{\delta-1} h(\underline{x})}{\partial \underline{x}}$$

sind linear unabhängig. Wählt man also die Transformationsvorschrift gemäß

$$\begin{aligned}
z_1 &= \gamma_1(\underline{x}) = h(\underline{x}) \\
z_2 &= \gamma_2(\underline{x}) = L_f h(\underline{x}) \\
&\vdots \\
z_\delta &= \gamma_\delta(\underline{x}) = L_f^{\delta-1} h(\underline{x}) \\
z_{\delta+1} &= \gamma_{\delta+1}(\underline{x}) = \lambda_1(\underline{x}) \\
&\vdots \\
z_n &= \gamma_n = \lambda_{n-\delta}(\underline{x})
\end{aligned} \tag{12.73}$$

dann sind die ersten  $\delta$  Zeilen der Jacobimatrix, Gl. (12.72) linear unabhängig. Die noch unbekannten Funktionen  $\lambda_1(\underline{x}), \dots, \lambda_{n-\delta}(\underline{x})$  müssen so bestimmt werden, dass ihre Gradienten wiederum unabhängig sind, so dass also alle Zeilen der Jacobimatrix linear unabhängig sind. Dies erfordert etwas Kreativität, ist jedoch stets möglich. Man kann sogar zeigen, dass die Funktionen immer so gewählt werden können, dass  $L_g \lambda_i(\underline{x}) = 0$  für  $i = 1, \dots, n - \delta$  gilt. Führt man die Transformation aus, so erhält man,

$$\frac{dz_1}{dt} = \frac{\partial \gamma_1}{\partial \underline{x}} \frac{d\underline{x}}{dt} = \frac{\partial h}{\partial \underline{x}} \frac{d\underline{x}}{dt} = L_f h(\underline{x}) = \gamma_2(x) = z_2 \tag{12.74}$$

⋮

$$\frac{dz_{\delta-1}}{dt} = \frac{\partial \gamma_{\delta-1}}{\partial \underline{x}} \frac{d\underline{x}}{dt} = \frac{\partial (L_f^{\delta-2} h)}{\partial \underline{x}} \frac{d\underline{x}}{dt} = L_f^{\delta-1} h(\underline{x}) = \gamma_\delta(\underline{x}) = z_\delta \tag{12.75}$$

$$\frac{dz_\delta}{dt} = L_f^\delta h(\underline{x}) + L_g L_f^{\delta-1} h(\underline{x}) u(t) \tag{12.76}$$

Wie bereits angedeutet, muss  $\underline{x}$  in Gleichung (12.76) als Funktion von  $\underline{z}$  ausgedrückt werden. Man schreibt  $\underline{x} = \underline{\gamma}^{-1}(\underline{z})$  und beruft sich dabei auf die lokale Existenz der inversen Transformation, die ja durch die Tatsache sichergestellt wurde, dass die Jacobimatrix  $\partial \underline{\gamma}(\underline{x}) / \partial \underline{x}$  nichtsingulär ist. Noch kompakter lautet Gleichung (12.76):

$$\frac{dz_\delta}{dt} = \beta(\underline{z}) + \alpha(\underline{z}) u(t) = v \tag{12.77}$$

wobei

$$\alpha(\underline{z}) = L_g L_f^{\delta-1} h(\underline{\gamma}^{-1}(\underline{z})) \tag{12.78}$$

$$\beta(\underline{z}) = L_f^\delta h(\underline{\gamma}^{-1}(\underline{z})) \tag{12.79}$$

Für die letzten  $n - \delta$  Komponenten von  $\underline{z}$  gilt schließlich:

$$\begin{aligned}
\frac{dz_{\delta+i}}{dt} &= \frac{\partial \gamma_{\delta+i}}{\partial \underline{x}} [f(\underline{x}) + g(\underline{x}) u] = \frac{\partial \lambda_i}{\partial \underline{x}} [f(\underline{x}) + g(\underline{x}) u] \\
&= L_f \lambda_i(\underline{x}) + \underbrace{L_g \lambda_i(\underline{x})}_{=0} u(t) = L_f \lambda_i(x) \quad i = 1, \dots, n - \delta
\end{aligned} \tag{12.80}$$

Wie oben definiert man kompakt,

$$q_{\delta+i}(z) = L_f \gamma_{\delta+i}(\underline{\gamma}^{-1}(z)) \quad i = 1, \dots, n - \delta \quad (12.81)$$

und erhält abschließend die Normalform in  $\underline{z}$ -Koordinaten:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{\delta-1} &= z_\delta \\ \dot{z}_\delta &= \beta(z) + \alpha(z)u = v \\ \dot{z}_{\delta+1} &= q_{\delta+1}(z) \\ &\vdots \\ \dot{z}_n &= q_n(z) \end{aligned} \quad (12.82)$$

Für den Systemausgang gilt  $y = z_1$ .

Der Gleichungsteil von  $\dot{z}_1$  bis  $\dot{z}_\delta = v$  beschreibt die Integratorkette im Vorwärtszweig in der Darstellung der Normalform und der restliche Gleichungsteil ab  $\dot{z}_{\delta+1}$  die Nulldynamik im Rückwärtszweig.

### 12.3.1 Beispiel zur NRNF

Das Modell einer Gleichstromnebenenschlussmaschine mit variablem Feld ist in seiner einfachsten Form (linearisierte Magnetisierungskennlinie) gegeben durch:

$$\begin{aligned} \dot{x}_1 &= -\frac{1}{T_E} x_1 + u_E; \quad u_E = u \\ \dot{x}_2 &= -\frac{1}{T_A} x_2 + u_A - x_1 x_3 \\ \dot{x}_3 &= \frac{1}{T_{\theta N}} x_1 x_2 \end{aligned} \quad (12.83)$$

wobei  $x_1$  den Erregerfluss,  $x_2$  den Ankerstrom und  $x_3$  die Drehzahl des Motors darstellen.  $T_A$  und  $T_E$  stehen für die Anker- bzw. Erregerzeitkonstanten und  $T_{\theta N}$  bezeichnet die Trägheitszeitkonstante der Maschine. Die Drehzahl wird über die Erregerspannung  $u_E$  geregelt, während die Ankerspannung  $u_A$  konstant gehalten wird. Als Ausgangsgröße erhalten wir  $y = x_3$ . Bestimmt man die Lie-Ableitungen des Ausgangs, so ergibt sich

$$\begin{aligned}
L_g h &= [0 \ 0 \ 1] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0 \\
L_f h &= [0 \ 0 \ 1] \begin{bmatrix} -\frac{1}{T_E} x_1 \\ -\frac{1}{T_A} x_2 + u_A - x_1 x_3 \\ \frac{1}{T_{\theta N}} x_1 x_2 \end{bmatrix} = \frac{1}{T_{\theta N}} x_1 x_2 \\
L_g L_f h &= \left[ \frac{1}{T_{\theta N}} x_2 \quad \frac{1}{T_{\theta N}} x_1 \quad 0 \right] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{T_{\theta N}} x_2 \\
L_f^2 h &= \left[ \frac{1}{T_{\theta N}} x_2 \quad \frac{1}{T_{\theta N}} x_1 \quad 0 \right] \begin{bmatrix} -\frac{1}{T_E} x_1 \\ -\frac{1}{T_A} x_2 + u_A - x_1 x_3 \\ \frac{1}{T_{\theta N}} x_1 x_2 \end{bmatrix} \\
&= -\frac{1}{T_{\theta N}} x_1 x_2 \left( \frac{1}{T_E} + \frac{1}{T_A} \right) + \frac{1}{T_{\theta N}} x_1 (u_A - x_1 x_3)
\end{aligned}$$

Das System besitzt also den Relativgrad  $\delta = 2$ , der in der Region  $U_0 = \{x \in \mathbb{R}^3 \mid x_2 \neq 0\}$  definiert werden kann. Wir setzen also an:

$$\begin{aligned}
z_1 &= \gamma_1(x) = h(x) = x_3 \\
z_2 &= \gamma_2(x) = L_f h(x) = \frac{1}{T_{\theta N}} x_1 x_2.
\end{aligned}$$

Um die Transformation zu vervollständigen, muss eine skalarwertige Funktion  $\lambda_1$  gefunden werden mit der Eigenschaft, dass  $d\lambda_1 g = \frac{\partial \lambda_1}{\partial x} g = 0$  gilt. Beispielsweise kann man ansetzen,  $\lambda_1 = x_2$  dann gilt:

$$z_3 = \gamma_3(x) = \lambda_1 = x_2.$$

Aus (12.82) folgt

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= L_f^2 h(\underline{\gamma}^{-1}(z)) + L_g L_f h(\underline{\gamma}^{-1}(z)) u = v \\
\dot{z}_3 &= L_f \gamma_3(\underline{\gamma}^{-1}(z)).
\end{aligned}$$

Mit  $x_1 = T_{\theta N} \frac{z_2}{z_3}$  erhält man schließlich

$$\begin{aligned}
\dot{z}_2 &= -z_2 \left( \frac{1}{T_E} + \frac{1}{T_A} \right) + \frac{z_2}{z_3} \left( u_A - T_{\theta N} \frac{z_1 z_2}{z_3} \right) u \\
\dot{z}_3 &= \frac{\partial \gamma_3}{\partial x} \Big|_{\underline{\gamma}^{-1}(z)} f(\underline{\gamma}^{-1}(z)) = -\frac{1}{T_A} z_3 + u_A - T_{\theta N} \frac{z_1 z_2}{z_3}
\end{aligned}$$

## 12.4 Exakte Ein-/Ausgangslinearisierung

Mit Hilfe der Normalform (12.82) lassen sich auf transparente Weise eine Vielzahl nichtlinearer Zustandsregelgesetze entwerfen. Insbesondere kann man ansetzen,

$$u = \frac{-\beta(z) + v}{\alpha(z)} \quad (12.84)$$

Hierbei ist  $v$  die neue Steuergröße. Setzt man das Regelgesetz in Gleichung (12.82) ein, so erhält man

$$\begin{aligned} \dot{z}_i &= z_{i+1} & i = 1 \dots \delta - 1 \\ \dot{z}_\delta &= v \\ \dot{z}_{\delta+1} &= q_{\delta+1}(z) \\ &\vdots \\ \dot{z}_n &= q_n(z) \end{aligned} \quad (12.85)$$

Dies entspricht einer partiellen Linearisierung des Systems mit Eingang  $v$  und Ausgang  $y = z_1$ . Man erkennt eine Integratorkette mit den Zuständen  $z_1, \dots, z_\delta$  und ein  $n - \delta$  dimensionales Subsystem mit den Zuständen  $z_{\delta+1}, \dots, z_n$ . Zu beachten ist, dass dieses (nichtlineare) Subsystem keinen Einfluss auf das Ein-/Ausgangsverhalten des Systems (12.85) hat, da die rechten Seiten der Differentialgleichungen  $\dot{z}_i = \dots$  in (12.85) für  $i = 1, \dots, \delta$  nicht von  $z_{\delta+1}, \dots, z_n$  abhängen (wohl aber umgekehrt! – d.h. die Integratorkette hat Einfluss auf den Verlauf der Zustände  $z_{\delta+1}, \dots, z_n$  des Subsystems). Das E/A-Verhalten des zustandsgegebenen Systems (12.85) entspricht also dem einer linearen Integratorkette der Ordnung  $\delta$ :

$$y^{(\delta)} = v \quad (12.86)$$

Daher wird (12.84) auch als exakt E/A-linearisierendes Zustandsregelgesetz bezeichnet. Für das neue System kann jetzt ein linearer Zustandsregler der Form

$$v = w - r^T z \quad (12.87)$$

entworfen werden. Abbildung 12.13 verdeutlicht die gesamte Regelkreisstruktur.

Nun soll ein Regelgesetz für die Originalkoordinaten  $\underline{x}$  gefunden werden, so dass auf eine explizite Transformation verzichtet werden kann. Aus Gleichung (12.84) und den Definitionen (12.78) und (12.79) ergibt sich zunächst

$$u = \frac{-L_f^\delta h(\underline{x}) + v}{L_g L_f^{\delta-1} h(\underline{x})} \quad (12.88)$$

Löst man das lineare Regelgesetz (12.87) komponentenweise auf, so erhält man

$$v = w - r_0 z_1 - r_1 z_2 - \dots - r_{\delta-2} z_{\delta-1} - r_{\delta-1} z_\delta \quad (12.89)$$

Dies ist in Abbildung 12.14 veranschaulicht. Ersetzt man  $z_i$ ,  $i = 1, \dots, \delta$  gemäß (12.73) durch entsprechende Ausdrücke in  $\underline{x}$  so erhält man

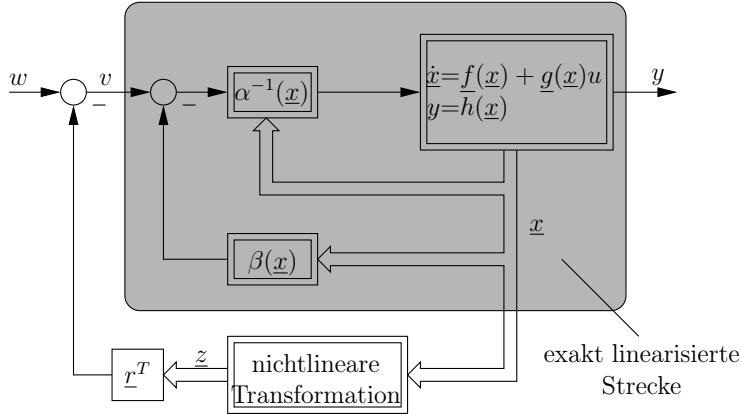


Abb. 12.13: Signalflußplan des exakt linearisierten Regelkreises

$$\begin{aligned} v &= w - r_0 h(\underline{x}) - r_1 L_f h(\underline{x}) - \dots - r_{\delta-1} L_f^{\delta-1} h(\underline{x}) \\ &= w - \sum_{\nu=0}^{\delta-1} r_\nu L_f^\nu h(\underline{x}) \end{aligned} \quad (12.90)$$

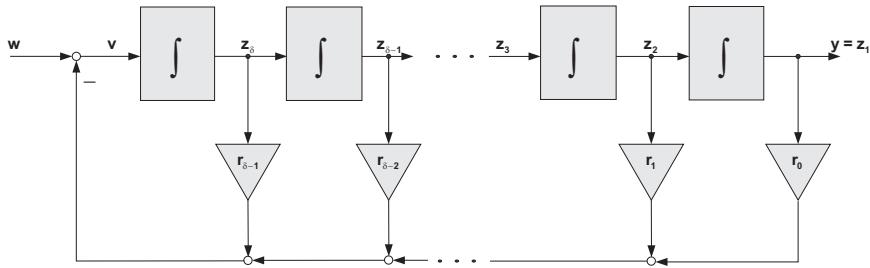
was mit Gleichung (12.88) zum exakt linearisierenden Regelgesetz in Orginalkoordinaten führt:

$$\begin{aligned} u &= \frac{-L_f^\delta h(\underline{x}) + v}{L_g L_f^{\delta-1} h(\underline{x})} \\ &= \left[ -L_f^\delta h(\underline{x}) + w - \sum_{\nu=0}^{\delta-1} r_\nu \cdot L_f^\nu h(\underline{x}) \right] \cdot \frac{1}{L_g L_f^{\delta-1} h(\underline{x})} \\ &= \frac{1}{L_g L_f^{\delta-1} h(\underline{x})} \cdot w - \frac{\sum_{\nu=0}^{\delta-1} r_\nu \cdot L_f^\nu h(\underline{x})}{L_g L_f^{\delta-1} h(\underline{x})} \end{aligned} \quad (12.91)$$

Das Regelgesetz nach Abbildung 12.14 muss durch ein Vorfilter  $K_v$  ergänzt werden, um stationäre Genauigkeit zu erreichen. Das um  $K_v$  erweiterte Regelgesetz hat die Übertragungsfunktion:

$$\begin{aligned} G_{reg} = \frac{y}{w} &= \frac{K_v \cdot \frac{1}{s^\delta}}{1 + \frac{1}{s^\delta} \cdot (r_0 + r_1 \cdot s + \dots + r_{\delta-1} \cdot s^{\delta-1})} \\ &= \frac{K_v}{s^\delta + r_{\delta-1} \cdot s^{\delta-1} + \dots + r_1 \cdot s + r_0} \end{aligned} \quad (12.92)$$

Für stationäre Genauigkeit ist eine Übertragungsfunktion  $G_{reg} = 1$  für  $\lim_{s \rightarrow 0}$  erforderlich, d.h. die Terme im Nenner und im Zähler ohne  $s$  müssen gleich sein.



**Abb. 12.14:** Regelgesetz mit Integratorkette

Damit gilt  $K_v = r_0$ . Das exakt linearisierende Regelgesetz in Orginalkoordinaten mit stationärer Genauigkeit lautet somit:

$$u = \frac{r_0}{L_g L_f^{\delta-1} h(\underline{x})} \cdot w - \frac{L_f^\delta h(\underline{x}) + \sum_{\nu=0}^{\delta-1} r_\nu \cdot L_f^\nu h(\underline{x})}{L_g L_f^{\delta-1} h(\underline{x})} \quad (12.93)$$

Der Vektor mit den Reglerparametern  $r$  kann durch Koeffizientenvergleich mit dem Wunschk Polynom  $N_{soll}(s) = s^\delta + q_{\delta-1}s^{\delta-1} \dots + q_1s + q_0$  bestimmt werden.

Für die in diesem Kapitel betrachtete Regelungsmethode ist es im Wesentlichen notwendig, das vorliegende System in die Byrnes-Isidori-Form zu transformieren, um ein einfaches Kompensationsgesetz entwerfen zu können. Im Folgenden schließen sich zwei Beispiele an, die im Detail die Transformation in Byrnes-Isidori Form zeigen. Die dabei betrachteten Regelstrecken besitzen vollen Relativgrad ( $\delta = n$ ), so dass der Rückführzweig in der Byrnes-Isidori-Form keine dynamischen Anteile enthält.

### 12.4.1 Beispiel zur exakten Ein-Ausgangslinearisierung

Es sei folgendes System gegeben:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{T_\theta} (-\mathcal{N}(x_1) + x_2) \\ -\frac{1}{T_2} x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{T_2} \end{bmatrix} u \quad (12.94)$$

$$-u_{max} < u < u_{max} \quad (12.95)$$

$$y = x_1 \quad (12.96)$$

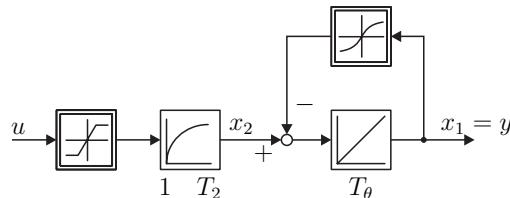
$$\text{mit } \mathcal{N}\mathcal{L}(x_1) = 0.2 \arctan(100x_1) \quad (12.97)$$

$$T_\theta = 0.2 \text{ s} \quad (12.98)$$

$$T_2 = 0.05 \text{ s} \quad (12.99)$$

$$|u|_{max} = 1.5 \quad (12.100)$$

Der zugehörige Signalflussplan ist in Abbildung 12.15 abgebildet.



**Abb. 12.15:** Signalflussplan des Systems in Beispiel 1

Die Systemgleichungen (12.94) und (12.96) können in der Form entsprechend Gleichung (12.101) und (12.102) oder in der Form entsprechend der Gleichungen (12.103) und (12.104) beschrieben werden. Die erste Darstellung repräsentiert ein lineares System mit isolierter Nichtlinearität; die Zweite entspricht der allgemeinen Darstellung eines nichtlinearen Systems:

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{b}u + \underline{k}\mathcal{N}\mathcal{L}(x_e) \quad (12.101)$$

$$y = \underline{c}^T \underline{x} \quad (12.102)$$

oder

$$\dot{\underline{x}} = \underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot u \quad (12.103)$$

$$y = h(\underline{x}) \quad (12.104)$$

Hiermit gilt im vorliegenden Fall für die allgemeine Darstellung:

$$\underline{f}(\underline{x}) = \begin{bmatrix} \frac{1}{T_\theta}(-\mathcal{N}\mathcal{L}(x_1) + x_2) \\ -\frac{1}{T_2}x_2 \end{bmatrix} \quad (12.105)$$

$$\underline{g}(\underline{x}) = \begin{bmatrix} 0 \\ \frac{1}{T_2} \end{bmatrix} \quad (12.106)$$

$$h(\underline{x}) = x_1 \quad (12.107)$$

Für die Bestimmung des Relativgrades  $\delta$  und der Durchführung der Byrnes-Isidori-Transformation ist es nun notwendig, das Ausgangssignal  $y$  mehrmals nach der Zeit abzuleiten, bis ein Durchgriff der Stellgröße  $u$  vorliegt:

$$\dot{y} = \frac{\partial h(\underline{x})}{\partial \underline{x}} \frac{d\underline{x}}{dt} = \frac{\partial h(\underline{x})}{\partial \underline{x}} \dot{\underline{x}} = \frac{\partial h(\underline{x})}{\partial \underline{x}} (\underline{f}(\underline{x}) + \underline{g}(\underline{x}) \cdot u) \quad (12.108)$$

$$= \underbrace{\frac{\partial h(\underline{x})}{\partial \underline{x}} \underline{f}(\underline{x})}_{L_f h(\underline{x})} + \underbrace{\frac{\partial h(\underline{x})}{\partial \underline{x}} \underline{g}(\underline{x}) \cdot u}_{L_g h(\underline{x})} \quad (12.109)$$

$$= [1 \ 0] \begin{bmatrix} \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \\ -\frac{1}{T_2} x_2 \end{bmatrix} + [1 \ 0] \begin{bmatrix} 0 \\ \frac{1}{T_2} \end{bmatrix} u \quad (12.110)$$

$$= \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) + 0 \cdot u \quad (12.111)$$

Wird  $\dot{y}$  über die Richtungsableitungen von  $h(\underline{x})$  entlang  $f(\underline{x})$  bzw.  $g(\underline{x})$  ausgedrückt, den sogenannten Lie-Ableitungen  $L_f h(\underline{x})$  und  $L_g h(\underline{x})$ , so ergibt sich

$$\dot{y} = L_f h(\underline{x}) + L_g h(\underline{x}) \cdot u \quad (12.112)$$

mit:

$$L_f h(\underline{x}) = \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \quad (12.113)$$

$$L_g h(\underline{x}) = 0 \quad (12.114)$$

Nach der ersten Ableitung zeigt sich, dass das Eingangssignal  $u$  wegen (12.114) noch keinen Durchgriff auf  $\dot{y}$  besitzt, so dass eine weitere Ableitung notwendig ist. Für den Relativgrad  $\delta$ , welcher besagt, über welche minimale Anzahl von Integratoren der Eingang verzögert auf den Ausgang wirkt, kann bereits die Aussage  $\delta > 1$  getroffen werden. Die zweite Ableitung, welche den Relativgrad  $\delta = 2$  untersucht, berechnet sich nach der Vorschrift

$$y^{(\delta)} = L_f^\delta h(\underline{x}) + L_g L_f^{\delta-1} h(\underline{x}) \cdot u \quad (12.115)$$

wie folgt:

$$\ddot{y} = L_f^2 h(\underline{x}) + L_g L_f h(\underline{x}) \cdot u \quad (12.116)$$

Für den ersten und zweiten Term in Gleichung (12.116) gilt:

$$L_f^2 h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} f(\underline{x}) \quad (12.117)$$

$$= \frac{\partial \left( \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \right)}{\partial \underline{x}} \begin{bmatrix} \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \\ -\frac{1}{T_2} x_2 \end{bmatrix} \quad (12.118)$$

$$= \begin{bmatrix} -\frac{1}{T_\theta} \mathcal{NL}'(x_1) & \frac{1}{T_\theta} \end{bmatrix} \begin{bmatrix} \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \\ -\frac{1}{T_2} x_2 \end{bmatrix} \quad (12.119)$$

$$= -\frac{1}{T_\theta^2} \mathcal{NL}'(x_1) [x_2 - \mathcal{NL}(x_1)] - \frac{1}{T_\theta T_2} x_2 \quad (12.120)$$

$$L_g L_f h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \quad (12.121)$$

$$= \frac{\partial \left( \frac{1}{T_\theta} (-\mathcal{NL}(x_1) + x_2) \right)}{\partial \underline{x}} \begin{bmatrix} 0 \\ \frac{1}{T_2} \end{bmatrix} \quad (12.122)$$

$$= \begin{bmatrix} -\frac{1}{T_\theta} \mathcal{NL}'(x_1) & \frac{1}{T_\theta} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{T_2} \end{bmatrix} \quad (12.123)$$

$$= \frac{1}{T_\theta T_2} \neq 0 \quad (12.124)$$

Hiermit ergibt sich für die zweite Ableitung:

$$\ddot{y} = L_f^2 h(\underline{x}) + L_g L_f h(\underline{x}) u \quad (12.125)$$

$$= \underbrace{-\frac{1}{T_\theta^2} \mathcal{NL}'(x_1) [x_2 - \mathcal{NL}(x_1)]}_{\beta(\underline{x})} - \frac{1}{T_\theta T_2} x_2 + \underbrace{\frac{1}{T_\theta T_2} \cdot u}_{\alpha(\underline{x})} = v \quad (12.126)$$

Es ist nun zu erkennen, dass der Eingang  $u$  einen direkten Durchgriff auf  $\ddot{y}$  besitzt, womit feststeht, dass das betrachtete nichtlineare System einen Relativgrad  $\delta = 2$  besitzt. Dies ist bei Betrachten des Signalfussplanes 12.15 leicht verständlich, da das Eingangssignal über nicht weniger als zwei Integratoren den Ausgang erreicht. Da das System aus insgesamt 2 Integratoren besteht, besitzt das System die Ordnung  $n = 2$  und entsprechend obiger Betrachtung Relativgrad  $\delta = 2$ .

Für den Fall eines vollen Relativgrades  $n = \delta$  ist es leicht zu verstehen, dass die vorgenommene Differentiation einer Koordinatentransformation entspricht,

wenn das System nicht mehr durch die Zustände  $x_1$  und  $x_2$ , sondern durch die neuen Zustände  $z_1 = y$  und  $z_2 = \dot{y}$  mit der Zustandsdarstellung

$$\dot{z}_1 = \dot{y} \quad (12.127)$$

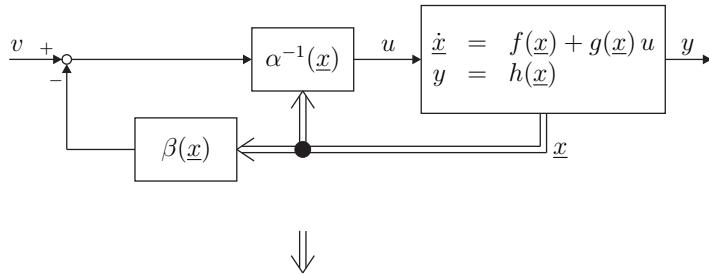
$$\dot{z}_2 = \ddot{z}_1 = \ddot{y} = \beta(\underline{x}) + \alpha(\underline{x}) \cdot u = \beta(\underline{\gamma}^{-1}(\underline{z})) + \alpha(\underline{\gamma}^{-1}(\underline{z})) \cdot u \quad (12.128)$$

beschrieben wird. Die Umrechnung zwischen den Koordinaten ist gemäß  $\underline{x} = \underline{\gamma}^{-1}(\underline{z})$  über die zumindest lokal existierende inverse Transformationsmatrix möglich, wird für das weitere Vorgehen jedoch nicht benötigt und hat an dieser Stelle rein formalen Charakter, da die Aufschaltung in  $\underline{x}$ -Koordinaten auf Basis des transformierten Systems realisiert werden soll. Da das Ein-Ausgangsverhalten eines Systems durch eine Zustandstransformation nicht verändert wird, kann mit Hilfe des transformierten Systems in  $\underline{z}$ -Koordinaten eine Aufschaltung für das ursprüngliche System in  $\underline{x}$ -Koordinaten gefunden werden, welche dazu führt, dass sich das resultierende Gesamtsystem gleich einer Integratorkette verhält. Im transformierten System (12.127) / (12.128) liegt nun eine Integratorkette vor, wo bei der Eingang des ersten Integrators  $\dot{z}_2 = \ddot{y}$  auch linear und nichtlinear von den Zuständen abhängt. Dieser Einfluss soll nun durch eine Aufschaltung kompensiert werden. Hierzu wird Gleichung (12.128) bzw. (12.126) nach  $u$  aufgelöst und ein neuer Eingang  $v = \ddot{y}$  deklariert. Durch diese Stellgröße

$$u = \frac{1}{\alpha(\underline{x})} (v - \beta(\underline{x})) = T_\theta T_2 \left[ v + \frac{1}{T_\theta^2} \mathcal{N}'(x_1) [x_2 - \mathcal{N}'(x_1)] + \frac{1}{T_\theta T_2} x_2 \right] \quad (12.129)$$

wird erreicht, dass sich der Eingang  $v$  des Gesamtsystems und die zweite Ableitung  $\ddot{y}$  des Systemausganges identisch verhalten. Das nichtlineare System wird somit durch die Stellgröße (12.129) linearisiert und das neue Gesamtsystem besteht lediglich aus zwei Integratoren, siehe Abbildung 12.16. Das resultierende System zeigt demnach zweifach integrierendes Verhalten – es wird von einer exakten Ein-Ausgangslinearisierung gesprochen.

Durch diese Linearisierung wird versucht, maximale Unbeobachtbarkeit der Dynamik des ursprünglichen Systems im Ausgang zu erreichen. Besitzt beispielsweise ein System die Ordnung  $n = 4$  und den Relativgrad  $\delta = 1$ , so wird das Teilsystem der Ordnung  $n - \delta = 3$ , welches der Nulldynamik entspricht, durch die E-A-Linearisierung keine Auswirkung auf den Ausgang zeigen. Es sei an dieser Stelle erwähnt, dass es eine erfolgreiche Aufschaltung der Nulldynamik und somit exakte Kompensation nur geben kann, wenn diese stabil ist. Das Ein-Ausgangsverhalten des neuen Gesamtsystems besitzt generell  $\delta$  Integratoren. Wie bereits erwähnt, stellt der Relativgrad ein Maß dar, wie schnell sich Änderungen am Eingang am Ausgang auswirken. Diese systembedingte Eigenschaft kann aus Sicht der Realisierbarkeit durch einen Regler nur beschleunigt, nicht jedoch kompensiert werden. Zur Kompensation der Integratoren wäre eine Differentiation, d.h. Blick in die Zukunft notwendig, was jedoch bekanntlich nicht umsetzbar ist.



**Abb. 12.16:** Aufschaltung zur Ein-Ausgangslinearisierung und resultierendes Gesamtsystem

Bezogen auf den linearen Fall bedeutete dies, dass ein System mit mehr Pol- als Nullstellen aus Realisierbarkeitsgründen nicht invertiert werden kann. Insofern ist es verständlich, dass das Gesamtsystem der E-A-Linearisierung minimal die Ordnung entsprechend des Relativgrades besitzen muss.

Aus diesem Beispiel ergibt sich, dass bei der Ein-Ausgangslinearisierung die Struktur und die Parameter des nichtlinearen Systems bekannt sein müssen. Außerdem muss die singuläre Nichtlinearität bzw. die singulären Nichtlinearitäten ableitbar sein. Dies bedeutet, dass unstetige Nichtlinearitäten wie beispielsweise die Reibung und die Lose nicht in der realen Form angesetzt werden dürfen, sondern wie bei der Reibung beispielhaft die *arctan*-Funktion genutzt wird. Noch schwieriger wird die Situation bei Nichtlinearitäten wie der mehrdeutigen Hysterese-Nichtlinearität in Kap. 12.8.

#### 12.4.2 Beispiel zur exakten Ein- Ausgangslinearisierung mit Reglerentwurf

Im Folgenden wird ein zweites Beispiel betrachtet, welches eine nichtlineare System-Dynamik  $f(\underline{x})$  als auch eine nichtlineare Einkopplung des Einganges mit  $g(\underline{x})$  zeigt und somit nicht auf den Fall einer isolierten Nichtlinearität gemäß (12.101) und (12.102) zurückzuführen ist:

$$\dot{\underline{x}} = \underbrace{\begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix}}_{f(\underline{x})} + \underbrace{\begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix}}_{g(\underline{x})} u \quad (12.130)$$

$$y = \underbrace{x_3}_{h(\underline{x})} \quad (12.131)$$

Es muss die allgemeine Form (12.103) mit (12.104) herangezogen werden:

$$\dot{\underline{x}} = f(\underline{x}) + g(\underline{x}) \cdot u \quad (12.132)$$

$$y = h(\underline{x}) \quad (12.133)$$

$$f(\underline{x}) = \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix} \quad (12.134)$$

$$g(\underline{x}) = \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} \quad (12.135)$$

$$h(\underline{x}) = x_3 \quad (12.136)$$

Zur Bestimmung des Relativgrades  $\delta$  wird nun im Folgenden der Ausgang  $y$  mehrmals nach der Zeit abgeleitet, bis bei der  $\delta$ -ten Ableitung ein Durchgriff der Stellgröße  $u$  auf den  $\delta$ -mal abgeleiteten Ausgang  $y^{(\delta)}$  vorliegt:

$$\dot{y} = \frac{\partial h(\underline{x})}{\partial \underline{x}} \frac{d\underline{x}}{dt} = \frac{\partial h(\underline{x})}{\partial \underline{x}} \dot{\underline{x}} = \frac{\partial h(\underline{x})}{\partial \underline{x}} (f(\underline{x}) + g(\underline{x}) \cdot u) \quad (12.137)$$

$$= \underbrace{\frac{\partial h(\underline{x})}{\partial \underline{x}} f(\underline{x})}_{L_f h(\underline{x})} + \underbrace{\frac{\partial h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \cdot u}_{L_g h(\underline{x})} \quad (12.138)$$

Mit

$$\frac{\partial h(\underline{x})}{\partial \underline{x}} = \frac{\partial x_3}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial x_3}{\partial x_1} & \frac{\partial x_3}{\partial x_2} & \frac{\partial x_3}{\partial x_3} \end{bmatrix} = [0 \ 0 \ 1] \quad (12.139)$$

folgt für die erste Ableitung des Ausgangs:

$$\dot{y} = L_f h(\underline{x}) + L_g h(\underline{x}) \cdot u \quad (12.140)$$

$$= [0 \ 0 \ 1] f(\underline{x}) + [0 \ 0 \ 1] g(\underline{x}) \cdot u \quad (12.141)$$

$$= \underbrace{[0 \ 0 \ 1] \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix}}_{L_f h(\underline{x})} + \underbrace{[0 \ 0 \ 1] \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} u}_{L_g h(\underline{x}) = 0} \quad (12.142)$$

$$= x_1 - x_2 + 0 \cdot u \quad (12.143)$$

Da der Eingang bzgl. der ersten Ableitung des Ausgangs ( $y^{(\delta)}$  mit  $\delta = 1$ ) mit

$$\dot{y} = L_f h(\underline{x}) = x_1 - x_2 \quad (12.144)$$

keinen Durchgriff besitzt, muss das System einen Relativgrad  $\delta > 1$  aufweisen. Somit ist die zweite Ableitung zu bestimmen. Hierfür seien nochmals die berechneten Lie-Ableitungen aufgeführt:

$$L_g h(\underline{x}) = \frac{\partial h(\underline{x})}{\partial \underline{x}} g(\underline{x}) = [0 \ 0 \ 1] \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} = 0 \quad (12.145)$$

$$L_f h(\underline{x}) = \frac{\partial h(\underline{x})}{\partial \underline{x}} f(\underline{x}) = [0 \ 0 \ 1] \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix} = x_1 - x_2 \quad (12.146)$$

Für die Bestimmung der zweiten Ableitung muss Gleichung (12.138) bzw. (12.140) erneut nach der Zeit abgeleitet werden:

$$\ddot{y} = \frac{d}{dt} L_f h(\underline{x}) + \underbrace{\frac{d}{dt} (L_g h(\underline{x}) \cdot u(t))}_0 \quad (12.147)$$

Für den zweiten Summanden wäre die Kettenregel anzuwenden, wobei auf Grund von Gleichung (12.145) dieser jedoch zu Null wird, so dass sich die Berechnung vereinfacht:

$$\ddot{y} = \frac{d}{dt} L_f h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} \dot{x} = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} (f(\underline{x}) + g(\underline{x}) \cdot u) \quad (12.148)$$

$$= \underbrace{\frac{\partial L_f h(\underline{x})}{\partial \underline{x}} f(\underline{x})}_{L_f^2 h(\underline{x})} + \underbrace{\frac{\partial L_f h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \cdot u}_{L_g L_f h(\underline{x})} \quad (12.149)$$

Entsprechend der Definition der Lie-Ableitungen (12.138) wird folgender Formalismus verwendet:

$$L_f^2 h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} f(\underline{x}) \quad (12.150)$$

$$L_g L_f h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \quad (12.151)$$

Gemäß dieser Zusammenhänge berechnet sich die zweite Ableitung des Ausgangs mit der entsprechenden Lie-Ableitung aus (12.146) wie folgt:

$$\ddot{y} = L_f^2 h(\underline{x}) + L_g L_f h(\underline{x}) \cdot u \quad (12.152)$$

$$= \frac{\partial(x_1 - x_2)}{\partial \underline{x}} f(\underline{x}) + \frac{\partial(x_1 - x_2)}{\partial \underline{x}} g(\underline{x}) \cdot u \quad (12.153)$$

$$= [1 \ -1 \ 0] f(\underline{x}) + [1 \ -1 \ 0] g(\underline{x}) \cdot u \quad (12.154)$$

$$= \underbrace{[1 \ -1 \ 0] \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix}}_{L_f^2 h(\underline{x})} + \underbrace{[1 \ -1 \ 0] \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} u}_{L_g L_f h(\underline{x})} = 0 \quad (12.155)$$

$$= -x_1 - x_2^2 + 0 \cdot u \quad (12.156)$$

Da ebenfalls der Eingang bzgl. der zweiten Ableitung des Ausgangs ( $y^{(\delta)}$  mit  $\delta = 2$ ) mit

$$\ddot{y} = L_f^2 h(\underline{x}) = -x_1 - x_2^2 \quad (12.157)$$

keinen Durchgriff besitzt, muss die dritte Ableitung zur Bestimmung des Relativgrades berechnet werden. Für das iterative Vorgehen sind nun die im zweiten Schritt bestimmten Lie-Ableitungen von Interesse, für die zusammenfassend gilt:

$$L_g L_f h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} g(\underline{x}) = [1 \ -1 \ 0] \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} = 0 \quad (12.158)$$

$$L_f^2 h(\underline{x}) = \frac{\partial L_f h(\underline{x})}{\partial \underline{x}} f(\underline{x}) = [1 \ -1 \ 0] \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix} = -x_1 - x_2^2 \quad (12.159)$$

Die dritte Ableitung ergibt sich durch erneutes Differenzieren der Gleichung (12.152) nach der Zeit:

$$\ddot{y} = \frac{d}{dt} L_f^2 h(\underline{x}) + \underbrace{\frac{d}{dt} (L_g L_f h(\underline{x}) \cdot u(t))}_0 \quad (12.160)$$

Der zweite Summand wird wegen Gleichung (12.158) erneut zu Null, so dass gilt:

$$\ddot{y} = \frac{d}{dt} L_f^2 h(\underline{x}) = \frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} \dot{x} = \frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} (f(\underline{x}) + g(\underline{x}) \cdot u) \quad (12.161)$$

$$= \underbrace{\frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} f(\underline{x})}_{L_f^3 h(\underline{x})} + \underbrace{\frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \cdot u}_{L_g L_f^2 h(\underline{x})} \quad (12.162)$$

Der Formalismus der Lie-Ableitungen wird folgendermaßen fortgesetzt:

$$L_f^3 h(\underline{x}) = \frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} f(\underline{x}) \quad (12.163)$$

$$L_g L_f^2 h(\underline{x}) = \frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \quad (12.164)$$

Zur Berechnung der dritten Ableitung des Ausgangs wird für das iterativen Vorgehen die Lie-Ableitung (12.159) verwendet:

$$\ddot{y} = L_f^3 h(\underline{x}) + L_g L_f^2 h(\underline{x}) \cdot u \quad (12.165)$$

$$= \frac{\partial(-x_1 - x_2^2)}{\partial \underline{x}} f(\underline{x}) + \frac{\partial(-x_1 - x_2^2)}{\partial \underline{x}} g(\underline{x}) \cdot u \quad (12.166)$$

$$= [-1 \quad -2x_2 \quad 0] f(\underline{x}) + [-1 \quad -2x_2 \quad 0] g(\underline{x}) \cdot u \quad (12.167)$$

$$= \underbrace{[-1 \quad -2x_2 \quad 0]}_{L_f^3 h(\underline{x})} \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix} + \underbrace{[-1 \quad -2x_2 \quad 0]}_{L_g L_f^2 h(\underline{x})} \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} u \quad (12.168)$$

$$= -2x_2(x_1 + x_2^2) - (1 + 2x_2)e^{x_2} \cdot u \quad (12.169)$$

Hiermit steht der Relativgrad des Systems fest, da der Eingang bzgl. der dritten Ableitung des Ausgangs ( $y^{(3)}$  mit  $\delta = 3$ ) mit

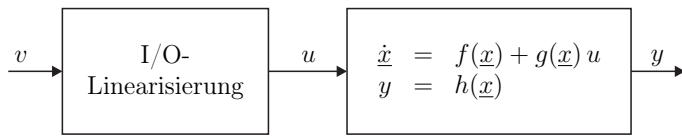
$$\ddot{y} = L_f^3 h(\underline{x}) + L_g L_f^2 h(\underline{x}) u = \underbrace{-2x_2(x_1 + x_2^2)}_{\beta(\underline{x})} + \underbrace{(-1 - 2x_2)e^{x_2} \cdot u}_{\alpha(\underline{x})} \quad (12.170)$$

einen Durchgriff besitzt, solange  $x_2 \neq -0.5$  gilt. Der Relativgrad des nichtlinearen Systems dritter Ordnung ( $n = 3$ ) beträgt  $\delta = 3$ , der in der Region

$$U_0 = \{x \in \mathbb{R}^3 | x_2 \neq -0.5, x_2 > -\infty\} \quad (12.171)$$

definiert ist. Der Vollständigkeit halber seien nochmals die Lie-Ableitungen des dritten Schrittes aufgeführt, welche für die Ein-Ausgangslinearisierung benötigt werden:

$$\begin{aligned} \alpha(\underline{x}) &= L_g L_f^2 h(\underline{x}) = \frac{\partial L_f^2 h(\underline{x})}{\partial \underline{x}} g(\underline{x}) = [-1 \ -2x_2 \ 0] \begin{bmatrix} e^{x_2} \\ e^{x_2} \\ 0 \end{bmatrix} = -(1+2x_2) e^{x_2} \\ \beta(\underline{x}) &= L_f^3 h(\underline{x}) = \frac{\partial L_f^3 h(\underline{x})}{\partial \underline{x}} f(\underline{x}) = [-1 \ -2x_2 \ 0] \begin{bmatrix} 0 \\ x_1 + x_2^2 \\ x_1 - x_2 \end{bmatrix} = -2x_2(x_1 + x_2^2) \end{aligned} \quad (12.172)$$



**Abb. 12.17:** Prinzipschaltbild der Ein-Ausgangslinearisierung

Wie bereits im Beispiel in Kapitel 12.4.1 diskutiert, dient die Bildung der ersten bis  $\delta$ -ten Ableitung einer Koordinatentransformation, mit deren Hilfe eine Aufschaltung gemäß Abbildung 12.17, die sog. Ein-Ausgangslinearisierung, realisiert werden kann, so dass das resultierende Gesamtsystem das Verhalten einer Integratorkette ohne Nichtlinearitäten zeigt. Da im Falle vollen Relativgrades ( $\delta = n$ ) die transformierten Systemzustände mit der Transformationsvorschrift nach (12.133), (12.144) und (12.157)

$$z_1 = y = h(\underline{x}) = x_3 \quad (12.173)$$

$$z_2 = \dot{y} = L_f h(\underline{x}) = x_1 - x_2 = \dot{z}_1 \quad (12.174)$$

$$z_3 = \ddot{y} = L_f^2 h(\underline{x}) = -(x_1 + x_2^2) = \dot{z}_2 \quad (12.175)$$

einen  $n$ -dimensionalen Raum aufspannen und somit alle Gleichungen linear unabhängig sind, müssen keine weiteren unbekannten Funktionen  $\lambda(\underline{x})$  bestimmt werden und die Transformationsvorschrift ist eindeutig. Für die transformierte Zustandsdarstellung mit demselben Eingang  $u$  und Ausgang  $y$  wie im ursprünglichen System gilt unter Beachtung von Gleichung (12.170):

$$\dot{z}_1 = z_2 = \dot{y} = L_f h(\underline{x}) = x_1 - x_2 \quad (12.176)$$

$$\dot{z}_2 = z_3 = \ddot{y} = L_f^2 h(\underline{x}) = -(x_1 + x_2^2) \quad (12.177)$$

$$\begin{aligned} \dot{z}_3 = \ddot{y} = v &= L_f^3 h(\underline{x}) + L_g L_f^2 h(\underline{x}) u = \underbrace{-2x_2(x_1 + x_2^2)}_{\beta(\underline{x})} + \underbrace{-(1 + 2x_2)e^{x_2} \cdot u}_{\alpha(\underline{x})} \\ &\quad (12.178) \end{aligned}$$

$$y = z_1 = x_3 \quad (12.179)$$

Hiermit liegt eine Integratorkette in  $\underline{z}$ -Koordinaten vor, wobei jedoch der Integrator mit dem Eingang  $\dot{z}_3$  neben dem Systemeingang  $u$  auch eine nichtlineare Abhängigkeit der Zustände mit  $\underline{x} = \gamma^{-1}(\underline{z})$  besitzt; es sei an dieser Stelle erwähnt, dass nach Definition die inverse Transformationsmatrix zumindest lokal existiert. Diese lineare und nichtlineare Abhängigkeit von den Systemzuständen soll mit Hilfe einer Aufschaltung kompensiert werden mit dem Ziel, dass sich das Gesamtsystem entsprechend einer Integratorkette mit dem neuen Eingang  $v$  und dem Ausgang  $y$  verhält und somit keine Abhängigkeit mehr von den Zuständen zeigt. Mit dem Ziel  $\ddot{y} = v$  erhält man über die Gleichung (12.178) die notwendige Stellgröße für die Aufschaltung:

$$u = \frac{1}{\alpha(\underline{x})} (v - \beta(\underline{x})) = \frac{1}{-(1 + 2x_2)e^{x_2}} (v + 2x_2(x_1 + x_2^2)) \quad (12.180)$$

Mit dieser Stellgröße, eingesetzt in die obige Zustandsdarstellung, wird klar ersichtlich, dass es sich bei der transformierten Zustandsdarstellung um eine Integratorkette mit dreifach integrierendem Verhalten mit dem Eingang  $v$  handelt:

$$\dot{z}_1 = \dot{y} = z_2 \quad (12.181)$$

$$\dot{z}_2 = \ddot{y} = z_3 \quad (12.182)$$

$$\dot{z}_3 = \ddot{y} = v \quad (12.183)$$

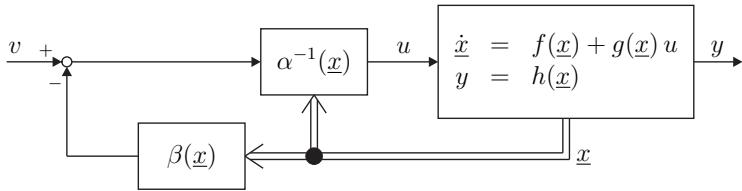
$$y = z_1 \quad (12.184)$$

bzw. in Vektorschreibweise:

$$\dot{\underline{z}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \underline{z} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \quad (12.185)$$

$$y = [1 \ 0 \ 0] \underline{z} \quad (12.186)$$

Da es nun für das Ein-Ausgangsverhalten irrelevant ist, in welcher Form das System vorliegt, führt die Stellgröße (12.180) auch im ursprünglichen System in  $\underline{x}$ -Koordinaten gemäß Abbildung 12.18



**Abb. 12.18:** Notwendige Aufschaltung des nichtlinearen Systems zur Ein-Ausgangslinearisierung

zu dem gewünschten Ein-Ausgangsverhalten – die Eigenschaft der Nichtlinearität wird kompensiert und es resultiert das lineare Verhalten einer dreifachen Integratorkette (vgl. Abbildung 12.19):

$$\frac{y(s)}{v(s)} = \frac{1}{s^3} \quad (12.187)$$



**Abb. 12.19:** Signalflussplan des resultierenden Gesamtsystems durch Ein-Ausgangslinearisierung

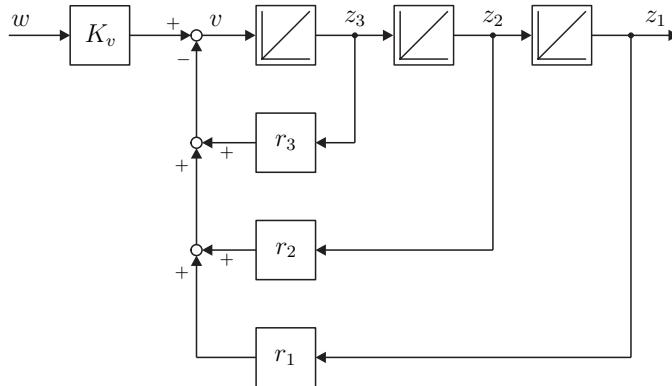
Nachdem jetzt ein sehr einfaches lineares System in Form einer Integratorkette vorliegt, kann das ursprüngliche nichtlineare System nun nach Anwenden der Ein-Ausgangslinearisierung mit Hilfe der linearen Regelungstechnik beherrscht werden. Im Folgenden wird beispielsweise der Zustandsregler herangezogen.

Der Signalflussplan in Abbildung 12.20 zeigt schematisch die Regelung der Integratorkette aus Abbildung 12.19 mit Hilfe eines Zustandsreglers. Diese einfache Realisierung ist jedoch nicht möglich, da das reale System in  $\underline{x}$ -Koordinaten und nicht in  $\underline{z}$ -Koordinaten vorliegt. Zur Bestimmung der Zustände müsste entsprechend der Transformationsvorschrift (12.174) und (12.175) der Ausgang differenziert werden, was jedoch in der Praxis nicht durchführbar ist. Es kann jedoch auch die Transformationsvorschrift  $\underline{z} = \gamma(\underline{x})$  aus den Gleichungen (12.173), (12.174) und (12.175) verwendet werden,

$$z_1 = y \quad (12.188)$$

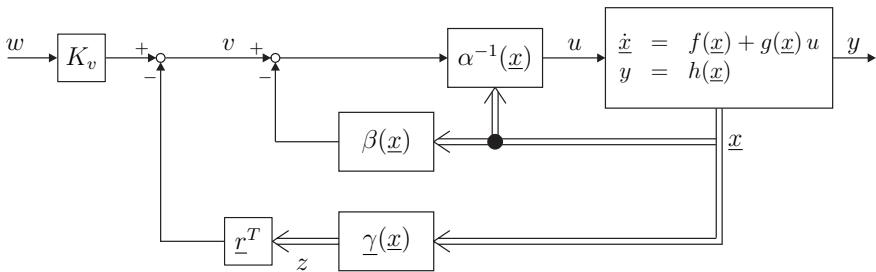
$$z_2 = x_1 - x_2 \quad (12.189)$$

$$z_3 = -(x_1 + x_2^2) \quad (12.190)$$



**Abb. 12.20:** Zustandsregelung des ein-ausgangslinearisierten Systems in  $\underline{z}$ -Koordinaten

um aus den gemessenen Zuständen  $x_1$ ,  $x_2$  und  $x_3$  die gesuchten Zustände  $z_1$ ,  $z_2$  und  $z_3$  des transformierten Systems zu berechnen. Das nichtlineare System kann somit sehr einfach über einen Zustandsregler entsprechend des Signalflussplanes 12.21 geregelt werden. Die nichtlineare Transformationsmatrix  $\underline{\gamma}(\underline{x})$  be-



**Abb. 12.21:** Zustandsregelung des ein-ausgangslinearisierten Systems in  $\underline{x}$ -Koordinaten

stimmt den  $z$ -Vektor aus dem Zustandsvektor  $\underline{x}$  des Systems; durch Multiplikation mit der vektoriellen Verstärkung  $r^T = [r_1 \ r_2 \ r_3]$  des Zustandsreglers kann nun bei gegebenem Sollwert  $w$  und Berücksichtigung der Ein-Ausgangslinearisierung mit  $\alpha(\underline{x})$  und  $\beta(\underline{x})$  die notwendige Stellgröße  $u$  berechnet werden, welche ein Stellglied auf das nichtlineare System aufschalten muss, um das Regelziel zu erreichen.

## 12.5 Regelung auf ein Referenzsignal (Tracking)

Ein sehr ähnliches Vorgehen wie bei der exakten E/A–Linearisierung ergibt sich beim sogenannten *Problem der Referenzsignal–Nachbildung*. Ausgehend von der Normalform

$$\begin{aligned}\dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{\delta-1} &= z_\delta \\ \dot{z}_\delta &= \beta(z) + \alpha(z)u = v \\ \dot{z}_{\delta+1} &= q_{\delta+1}(z) \\ &\vdots \\ \dot{z}_n &= q_n(z)\end{aligned}\tag{12.191}$$

lässt sich wieder ein Regelgesetz der Form

$$u = \frac{-\beta(z) + v}{\alpha(z)}\tag{12.192}$$

entwerfen. Außerdem müssen die Ableitungen des Referenzsignals  $y_R(t)$  bis zur Ordnung  $\delta - 1$  bekannt sein. Man erhält:

$$u = \frac{-\beta([y_R \dot{y}_R \dots {}^{(\delta-1)}y_R \ z_{r+1} \dots z_n]) + {}^{(\delta)}y_R}{\alpha([y_R \dot{y}_R \dots {}^{(\delta-1)}y_R \ z_{\delta+1} \dots z_n])}\tag{12.193}$$

Man erkennt leicht, wie dieses Regelgesetz zustande kommt: Wenn für alle Zeitpunkte  $t > 0$  gelten soll, dass  $y(t) \equiv y_R(t)$ , dann muss notwendigerweise

$$z_1 = y_R, \quad z_2 = \dot{y} = \dot{y}_R, \quad \dots, \quad z_\delta = {}^{(\delta-1)}y = {}^{(\delta-1)}y_R$$

gelten. Die verbleibenden  $n - \delta$  Zustände genügen der Differentialgleichung

$$\dot{z}_i = q_i([y_R \dot{y}_R \dots {}^{(\delta-1)}y_R \ z_{\delta+1} \dots z_n]), \quad i = \delta + 1 \dots n\tag{12.194}$$

Man beachte, dass der Anfangszustand  $x(0)$  des Systems mit  $y_R(0)$  kompatibel sein muss, das heißt in Normalkoordinaten muss gelten,

$$z_1(0) = y_R, \quad \dots, \quad z_\delta(0) = {}^{(\delta-1)}y_R(0)$$

Ansonsten differieren die beiden Trajektorien  $y(t)$  und  $y_R(t)$  um den konstanten Betrag  $|y(0) - y_R(0)|$ . Die verbleibenden Anfangswerte  $z_{\delta+1}(0), \dots, z_n(0)$  können beliebig gewählt werden, da sie ohne Wirkung auf den Ausgang  $y = z_1$  des Systems bleiben. Der konstante Regelfehler im Fall nicht-kompatibler Anfangswerte kann übrigens leicht behoben werden, indem man das Regelgesetz (12.193) im

Zähler um einen Term  $\sum_{\nu=0}^{\delta-1} c_\nu \left( z_{\nu+1} - y_R^{(\nu)} \right)$  (mit geeignet zu wählenden Konstanten  $c_\nu$ ) ergänzt.

Schließt man den Regelkreis mit Hilfe von Gleichung (12.193) und setzt  $y_R(t) \equiv 0$  so ergibt sich:

$$z_1 = 0, z_2 = 0, \dots, z_\delta = 0$$

Für die restlichen  $n - \delta$  Zustände ergibt sich analog zu Gleichung (12.194),

$$\dot{z}_i = q_i([0 \dots 0 \ z_{r+1} \ \dots \ z_n]). \quad (12.195)$$

Das so definierte Subsystem entspricht der in Kapitel 12.2 ausführlich behandelten **Nulldynamik**.

### 12.5.1 Beispiel zur Regelung auf ein Referenzsignal

Das System ist definiert wie folgt:

$$\dot{\underline{x}} = \underbrace{\begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix}}_{f(\underline{x})} + \underbrace{\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}}_{g(\underline{x})} u \quad (12.196)$$

$$y = \underbrace{x_1}_{h(\underline{x})} \quad (12.197)$$

Für die allgemeine Form (12.103) mit (12.104) folgt:

$$\dot{\underline{x}} = f(\underline{x}) + g(\underline{x}) \cdot u \quad (12.198)$$

$$y = h(\underline{x}) \quad (12.199)$$

$$f(\underline{x}) = \begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix} \quad (12.200)$$

$$g(\underline{x}) = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \quad (12.201)$$

$$h(\underline{x}) = x_1 \quad (12.202)$$

Es wird zunächst wie gehabt zur Relativgradbestimmung der Ausgang  $y$  nach der Zeit differenziert, bis die Stellgröße  $u$  einen Durchgriff auf den  $\delta$ -mal abgeleiteten Ausgang  $y^\delta$  zeigt:

$$\dot{y} = L_f h(\underline{x}) + L_g h(\underline{x}) \cdot u \quad (12.203)$$

$$\dot{y} = \frac{\partial h(\underline{x})}{\partial \underline{x}} f(\underline{x}) + \frac{\partial h(\underline{x})}{\partial \underline{x}} g(\underline{x}) \cdot u \quad (12.204)$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix}}_{L_f h(x)} + \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}}_{L_g h(x)} \cdot u \quad (12.205)$$

$$= x_3 - x_2^3 + 0 \cdot u \quad (12.206)$$

Die erste Ableitung des Ausgangs zeigt gemäß

$$\dot{y} = L_f h(\underline{x}) = x_3 - x_2^3 \quad (12.207)$$

keinen Durchgriff, weshalb mit der zweiten Ableitung des Ausgangs fortgesetzt wird:

$$\ddot{y} = L_f^2 h(\underline{x}) + L_g L_f h(\underline{x}) \cdot u \quad (12.208)$$

$$= \frac{\partial L_f h(\underline{x})}{\partial (\underline{x})} f(\underline{x}) + \frac{\partial L_f h(\underline{x})}{\partial (\underline{x})} g(\underline{x}) \cdot u \quad (12.209)$$

$$= \frac{\partial(x_3 - x_2^3)}{\partial (\underline{x})} f(\underline{x}) + \frac{\partial(x_3 - x_2^3)}{\partial (\underline{x})} g(\underline{x}) \cdot u \quad (12.210)$$

$$= \begin{bmatrix} 0 & -3x_2^2 & 1 \end{bmatrix} \begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix} + \begin{bmatrix} 0 & -3x_2^2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \cdot u$$

$$= \underbrace{3x_2^3 + x_1^2 - x_3}_{L_f^2 h(x)} + \underbrace{(1 + 3x_2^2)}_{L_g L_f h(x) \neq 0} \cdot u \quad (12.211)$$

Da in der zweiten Ableitung

$$\ddot{y} = \underbrace{x_1^2 + 3x_2^3 - x_3}_{\beta(\underline{x})} + \underbrace{(1 + 3x_2^2)}_{\alpha(\underline{x})} \cdot u = v \quad (12.212)$$

der Eingang  $u$  einen Durchgriff besitzt, steht fest, dass das System einen Relativgrad  $\delta = 2$  besitzt, welcher für die gesamte Region  $U_0 = \{x \in \mathbb{R}^3\}$  definiert ist. Das System der Ordnung  $n = 3$  hat somit eine Nulldynamik der Ordnung  $n - \delta = 1$ .

Im nächsten Schritt folgt die Transformation in die nichtlineare Regelungsnormalf orm (NRNF), der Byrnes-Isidori-Form, mit dem Ziel der Aufteilung in zwei

Subsysteme: eine Integratorkette der Ordnung  $\delta = 2$  mit eventuellen Rückkopplungen auf den Eingang  $\dot{y}$  des ersten Integrators der Kette und einer Nulldynamik der Ordnung  $m = n - \delta = 1$ . In den vorigen Beispielen 1 und 2 war die Koordinatentransformation leicht durchführbar, da das transformierte System durch Bilden der  $\delta = n$  Ableitungen mit  $\dot{z}_1 = \dot{y}, \dots, \dot{z}_n = y^{(\delta=n)}$  bereits vorlag. Eine Transformation  $\underline{z} = \underline{\gamma}(\underline{x})$  war somit mit Hilfe dieser  $\delta = n$  Ableitungen als  $n$  linear unabhängige Transformationsgleichungen eindeutig.

In diesem Beispiel sind jedoch nur  $\delta = 2$  Ableitung bzw. Integratoren notwendig, um den Vorwärtzweig des transformierten Systems zwischen Eingang  $u$  und Ausgang  $y$  zu bilden, womit für die transformierte Zustandsdarstellung in  $\underline{z}$ -Koordinaten nur  $\delta = 2$  von  $n = 3$  linear unabhängigen Gleichungen mit (12.207), (12.212) vorliegen:

$$\dot{z}_1 = z_2 = \dot{y} = L_f h(\underline{x}) = x_3 - x_2^3 \quad (12.213)$$

$$\dot{z}_2 = \ddot{y} = L_f^2 h(\underline{x}) + L_g L_f h(\underline{x}) u = \underbrace{x_1^2 + 3x_2^3 - x_3}_{\beta(\underline{x})} + \underbrace{(1 + 3x_2^2)}_{\alpha(\underline{x})} \cdot u = v$$

Unter der Beachtung von Gleichung (12.213) und (12.199) ist folglich die Transformationsvorschrift für  $z_1$  und  $z_2$  bekannt:

$$z_1 = y = h(\underline{x}) = x_1 \quad (12.214)$$

$$z_2 = \dot{y} = L_f h(\underline{x}) = x_3 - x_2^3 \quad (12.215)$$

Nachdem das transformierte System der Ordnung 3 im Vorwärtzweig aus nur 2 Integratoren besteht, muss sich folglich 1 Integrator im Rückwärtzweig befinden. Wie dieses Subsystem der Nulldynamik auszusehen hat, ist jedoch frei wählbar und nicht eindeutig. Demnach ist für die Transformation  $\underline{z} = \underline{\gamma}(\underline{x})$  noch eine linear unabhängige Gleichung

$$z_3 = \lambda_1(\underline{x}) \quad (12.216)$$

zu finden, die nun frei wählbar ist. Für die dritte Gleichung in der Zustandsdarstellung des transformierten Systems gilt hiermit:

$$\dot{z}_3 = \frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} \dot{\underline{x}} = \frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} (f(\underline{x}) + g(\underline{x}) \cdot u) \quad (12.217)$$

$$= \underbrace{\frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} f(\underline{x})}_{L_f \lambda_1(\underline{x})} + \underbrace{\frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} g(\underline{x}) \cdot u}_{L_g \lambda_1(\underline{x})} \quad (12.218)$$

Da es wünschenswert ist, eine vom Eingang  $u$  unabhängige Nulldynamik zu erhalten, die nur von den Zuständen abhängt, wird die frei wählbare Funktion  $\lambda_1(\underline{x})$  unter der Bedingung  $L_g \lambda_1(\underline{x}) = 0$  gesucht, womit die Wahl

$$z_3 = \lambda_1(\underline{x}) = x_2 + x_3 \quad (12.219)$$

ein geeigneter Kandidat ist:

$$\frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} g(\underline{x}) = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = 0 \quad (12.220)$$

Der Eingang  $u$  koppelt nicht in die Nulldynamik ein. Für die dritte Zustandsgleichung im transformierten System, der Nulldynamik, folgt hiermit:

$$\dot{z}_3 = L_f \lambda_1(\underline{x}) = \frac{\partial \lambda_1(\underline{x})}{\partial \underline{x}} f(\underline{x}) = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_3 - x_2^3 \\ -x_2 \\ x_1^2 - x_3 \end{bmatrix} = x_1^2 - x_2 - x_3$$

Im Anschluss sind nochmals alle  $n = 3$  Transformationsgleichungen aufgeführt:

$$z_1 = \gamma_1(\underline{x}) = x_1 \quad (12.221)$$

$$z_2 = \gamma_2(\underline{x}) = x_3 - x_2^3 \quad (12.222)$$

$$z_3 = \gamma_3(\underline{x}) = x_2 + x_3 \quad (12.223)$$

Auf Grund der Nichtlinearitäten lässt sich das Gleichungssystem  $\underline{z} = \underline{\gamma}(\underline{x})$  nicht nach  $\underline{x}$  auflösen bzw. nicht die Inverse der nichtlinearen Funktion  $\gamma(\cdot)$  bestimmen. Die Rücktransformation kann somit nur lokal bestimmt werden, was jedoch stets garantiert ist. Es soll hierfür beispielsweise die Jacobi-Matrix an der Stelle  $\underline{x}_0 = [0 \ 0 \ 0]^T$  berechnet werden:

$$\left. \frac{\partial \underline{\gamma}(\underline{x})}{\partial \underline{x}} \right|_{\underline{x}_0} = \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & -3x_2^2 & 1 \\ 0 & 1 & 1 \end{array} \right] \Bigg|_{\underline{x}_0} = \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{array} \right] \quad (12.224)$$

Für die inverse Jacobi-Matrix ergibt sich

$$\left( \left. \frac{\partial \underline{\gamma}(\underline{x})}{\partial \underline{x}} \right|_{\underline{x}_0} \right)^{-1} = \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \end{array} \right] \quad (12.225)$$

und hiermit die lokale Rücktransformation

$$\underline{x} = \underline{\gamma}^{-1}(\underline{z})|_{\underline{x}_0} = \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \end{array} \right] \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (12.226)$$

mit:

$$x_1 = \gamma_1^{-1}(\underline{z})|_{\underline{x}_0} = z_1 \quad (12.227)$$

$$x_2 = \gamma_2^{-1}(\underline{z})|_{\underline{x}_0} = z_3 - z_2 \quad (12.228)$$

$$x_3 = \gamma_3^{-1}(\underline{z})|_{\underline{x}_0} = z_2 \quad (12.229)$$

Mit Hilfe der Transformationsvorschriften (12.221), (12.222) und (12.223) können nun die Zustandsdifferentialgleichungen in gemischten Koordinaten

$$\dot{z}_1 = x_3 - x_2^3 \quad (12.230)$$

$$\dot{z}_2 = \underbrace{x_1^2 + 3x_2^3 - x_3}_{\beta(\underline{x})} + \underbrace{(1 + 3x_2^2)}_{\alpha(\underline{x})} \cdot u \quad (12.231)$$

$$\dot{z}_3 = x_1^2 - x_2 - x_3 \quad (12.232)$$

allgemein in  $\underline{z}$ -Koordinaten ausgedrückt werden,

$$\dot{z}_1 = z_2 \quad (12.233)$$

$$\dot{z}_2 = \beta(\underline{\gamma}^{-1}(\underline{z})) + \alpha(\underline{\gamma}^{-1}(\underline{z})) \cdot u \quad (12.234)$$

$$\dot{z}_3 = z_1^2 - z_3 \quad (12.235)$$

wobei für  $\alpha(\underline{x})$  und  $\beta(\underline{x})$  kein geschlossener Zusammenhang gefunden werden kann. Über die lokalen Rücktransformationsvorschriften (12.227), (12.228) und (12.229) ist eine vollständige Zustandsdarstellung in  $\underline{z}$ -Koordinaten möglich:

$$\dot{z}_1 = z_2 \quad (12.236)$$

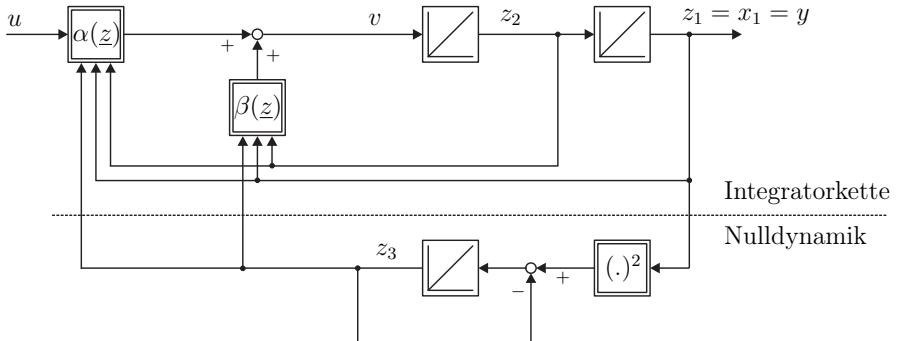
$$\dot{z}_2 = \underbrace{z_1^2 + 3(z_3 - z_2)^3 - z_2}_{\beta(\underline{z})} + \underbrace{(1 + 3(z_3 - z_2)^2)}_{\alpha(\underline{z})} \cdot u = v \quad (12.237)$$

$$\dot{z}_3 = z_1^2 - z_3 \quad (12.238)$$

$$y = z_1 \quad (12.239)$$

Der Signalflussplan für den allgemeinen Fall ist in Abbildung 12.22 zu finden.

Mit diesem Signalflussplan ist sehr gut zu erkennen, dass durch die Byrnes-Isidori-Transformation das System aufgespaltet wird in eine Integratorkette der Ordnung  $\delta$ , dessen Zustände auf den ersten Integrator der Kette wirken, und in eine Nulldynamik der Ordnung  $m = n - \delta$ . Alle Zustände, auch die der Nulldynamik, wirken wie beim Originalsystem auf den Ausgang. Verglichen mit dem linearen Fall repräsentiert die Nulldynamik die Dynamik der Nullstelle, weshalb deren Eigenschaft am Ausgang zu messen sein muss. Um nun zu erreichen, dass das nichtlineare System die Eigenschaft einer linearen Integratorkette der Ordnung  $\delta = 2$  erhält, muss die Wirkung von  $\alpha$  und  $\beta$  auf den Eingang  $\ddot{y} = \dot{z}_2$



**Abb. 12.22:** Transformiertes System in Byrnes-Isidori-Form

des ersten Integrators der Kette kompensiert werden. Wie in den ersten beiden Beispielen dargestellt, erreicht man dies durch Aufschalten der Stellgröße  $u$ :

$$u = \frac{1}{\alpha(z)} (v - \beta(z)) \quad (12.240)$$

Mit dieser Stellgröße, eingesetzt in die obige Zustandsdarstellung, wird klar ersichtlich, dass es sich bei der transformierten Zustandsdarstellung um eine Integratorkette mit zweifach integrierendem Verhalten mit dem Eingang  $v$  handelt:

$$\dot{z}_1 = z_2 \quad (12.241)$$

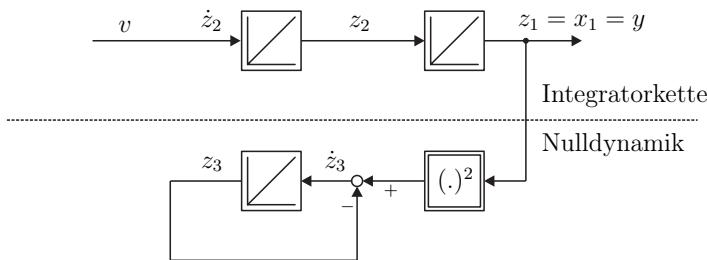
$$\dot{z}_2 = v \quad (12.242)$$

$$\dot{z}_3 = z_1^2 - z_3 \quad (12.243)$$

$$y = z_1 \quad (12.244)$$

Die Stellgröße  $v$  kompensiert den Einfluss der Nulldynamik auf den Eingang  $\ddot{y} = \dot{z}_2$  der Integratorkette und somit auf den Ausgang  $y$ . Der Zustand der Nulldynamik hat nun keinen Einfluss mehr auf die Zustände der Integratorkette und wirkt somit nicht mehr auf den Ausgang  $y$ . Der Signalflussplan des resultierenden Gesamtsystems ist in Abbildung 12.23 zu sehen.

Durch die vorgenommene Ein-Ausgangslinearisierung wird der Zustand  $z_3$  der Nulldynamik am Ausgang  $y$  unbeobachtbar. Bezüglich des linearen Falles wird die Eigenschaft der Nullstelle am Ausgang nicht mehr sichtbar – es findet durch die Kompensation der Nulldynamik eine Pol-Nullstellen-Kompensation statt. Insofern ist es leicht verständlich, dass die Nulldynamik bei Anwendung der Ein-Ausgangslinearisierung stabil sein muss, zumal der Ausgang bzw. die Zustände des Vorwärtszweiges weiter auf die Nulldynamik wirken. Im vorliegenden Beispiel



**Abb. 12.23:** Durch die Ein-Ausgangslinearisierung wird die Nulldynamik des Systems in Byrnes-Isidori-Form unbeobachtbar, so dass das Ein-Ausgangsverhalten einer Integratorkette entspricht

ist gewährt, dass die nicht beobachtbare Nulldynamik des Systems stabil ist, was leicht in Abbildung 12.23 zu erkennen ist. In diesem einfachen Fall lässt sich die Nulldynamik als lineares System mit dem Eingang  $z_1^2 = y^2$  darstellen. Das lineare System entspricht einem stabilen  $PT_1$ , wie dies die Differentialgleichung der Nulldynamik (12.243) mit

$$\dot{z}_3 = -1 \cdot z_3 + y^2 \quad (12.245)$$

zeigt. Der Eigenwert der Differentialgleichung bzw. Pol der Übertragungsfunktion

$$\frac{z_3(s)}{y^2(s)} = \frac{1}{1+s} \quad (12.246)$$

beträgt  $p_3 = -1$ , womit die Stabilität des Systems im Rückführzweig gezeigt ist und eine Ein-Ausgangslinearisierung bedenkenlos durchgeführt werden kann, ohne der Gefahr, dass die Stellgröße unkontrolliert anwächst. Es sei an dieser Stelle erwähnt, dass im Allgemeinen die Zustände der Nulldynamik eines nichtlinearen Systems nichtlinear verkoppelt sein können, weshalb dann beispielsweise eine Stabilitätsanalyse der Nulldynamik nach Lyapunov vorgenommen werden muss.

Im Grunde entspricht die Ein-Ausgangslinearisierung im Linearen einem Zustandsregler, der die Pole des Systems derart verschiebt, dass  $m = n - \delta$  Pole auf die Nullstellen geschoben werden, um diese zu kompensieren, und die restlichen  $\delta$  Pole werden auf  $s = 0$  verschoben, d.h. zeigen die Eigenschaft eines Integrators, woraus die Integratorkette mit  $\delta$  Integratoren resultiert. Liegt ein nichtlineares System vor, so wird die Nichtlinearität kompensiert und auf die Pol-Nullstellenkompensation des linearen Falles zurückgeführt.

Nachdem nun ein lineares System vorliegt, können übliche Regelungsmethoden herangezogen werden, um die Integratorkette der Ordnung  $\delta = 2$  aus Abbildung 12.23 zu regeln. Wie in den vorigen Beispielen kann ein Zustandsregler gemäß Abbildung 12.24 verwendet werden, um die Pole der Integratorkette nach Anwendungsfall zu verschieben.

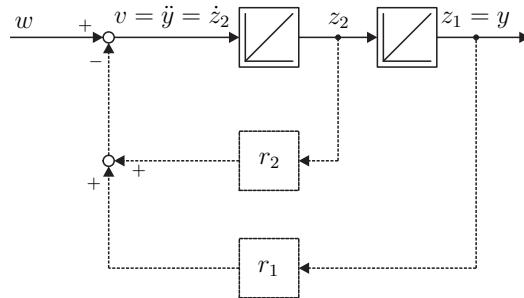
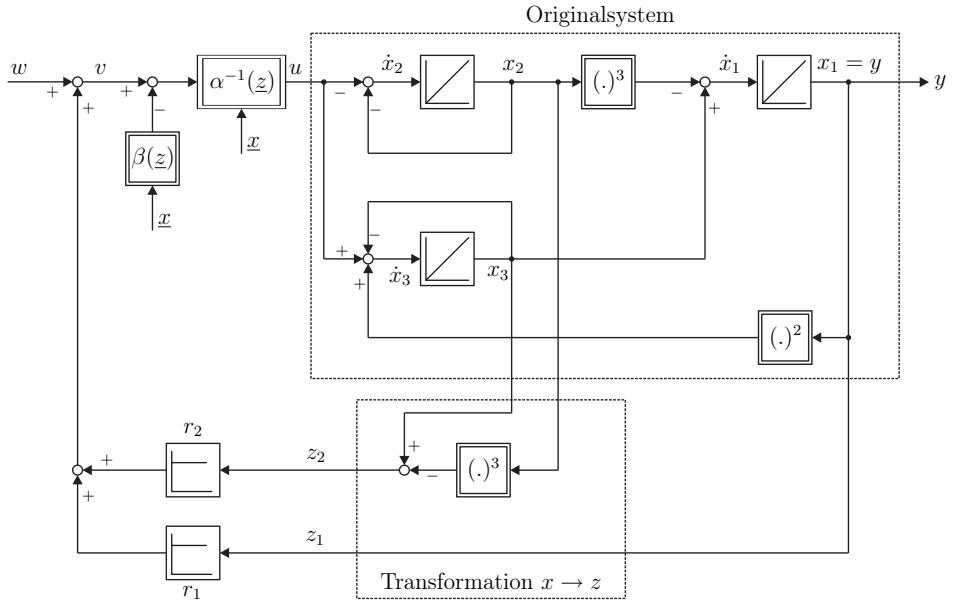


Abb. 12.24: Zustandsregler für eine Integratorkette

Für die Umsetzung des oben beschriebenen Konzeptes zur Regelung eines nichtlinearen realen Systems muss eine Koordinatentransformation vorgenommen werden, da die reale Anlage nicht in Byrnes-Isidori-Form, d.h.  $\underline{z}$ -Koordinaten vorliegt – es können nur die realen Zustände  $\underline{x}$  gemessen werden. Da ein Zustandsregler für die Integratorkette in  $\underline{z}$ -Koordinaten entworfen wurde, ist es notwendig, die gemessenen Zustände  $\underline{x}$  in das  $\underline{z}$ -Koordinatensystem zu transformieren, um die Zustände der Integratorkette zu bestimmen. Hierzu ist auf die Gleichungen (12.221) und (12.222) zurückzugreifen. Das mit Hilfe der Ein-Ausgangslinearisierung Zustands-geregelte System ist in Abbildung 12.25 dargestellt. Es ist zu erkennen, dass die Zustände des Originalsystems  $x_1$ ,  $x_2$  und  $x_3$  verwendet werden, um die Zustände  $z_1$  und  $z_2$  der Integratorkette des transformierten Systems zu berechnen. Über die Verstärkungen  $r_1$  und  $r_2$  des Zustandsreglers wird der Regelkreis geschlossen. Damit sich das nichtlineare System, wie angenommen, entsprechend einer Integratorkette verhält, ist die Aufschaltung mit  $\alpha(\underline{z})$  und  $\beta(\underline{z})$ , d.h. die Ein-Ausgangslinearisierung durchzuführen. Es können hierfür die berechneten Zustände  $z_1$ ,  $z_2$  und  $z_3$  benutzt werden. Wie jedoch mit Gleichung (12.234) und (12.237) gezeigt, ist die allgemeine Darstellung von  $\alpha(\underline{z})$  und  $\beta(\underline{z})$  auf Grund der nicht durchführbaren Inversion  $\gamma^{-1}(\underline{z})$  nur lokal möglich. Da die Vorschrift zur Ein-Ausgangslinearisierung jedoch auch in gemischten Koordinaten gemäß der Gleichung (12.231) vorliegt, bedarf es keiner Transformation und die Ein-Ausgangslienarisierung lässt sich stets für den allgemeinen Fall und nicht nur lokal durchführen. Dieser Hintergrund ist in Abbildung 12.25 angedeutet. Über die Sollgröße  $w$  kann nun das Wunschverhalten bzgl. des nichtlinearen Systems vorgegeben werden, welches sich auf Grund der Ein-Ausgangslinearisierung wie eine Zustands-geregelte Integratorkette verhält.

Um nun dem zustandsgeregelten System das gewünschte Folgeverhalten aufzuprägen, wird ein Referenzsystem eingeführt, dessen Ausgangsgröße  $y(t)$  das **Referenzsignal** liefert.

$$y(t) \equiv y_R(t) \quad (12.247)$$



**Abb. 12.25:** Ein-Ausgangslinearisiertes System, welches mit Hilfe eines Zustandsreglers geregelt wird

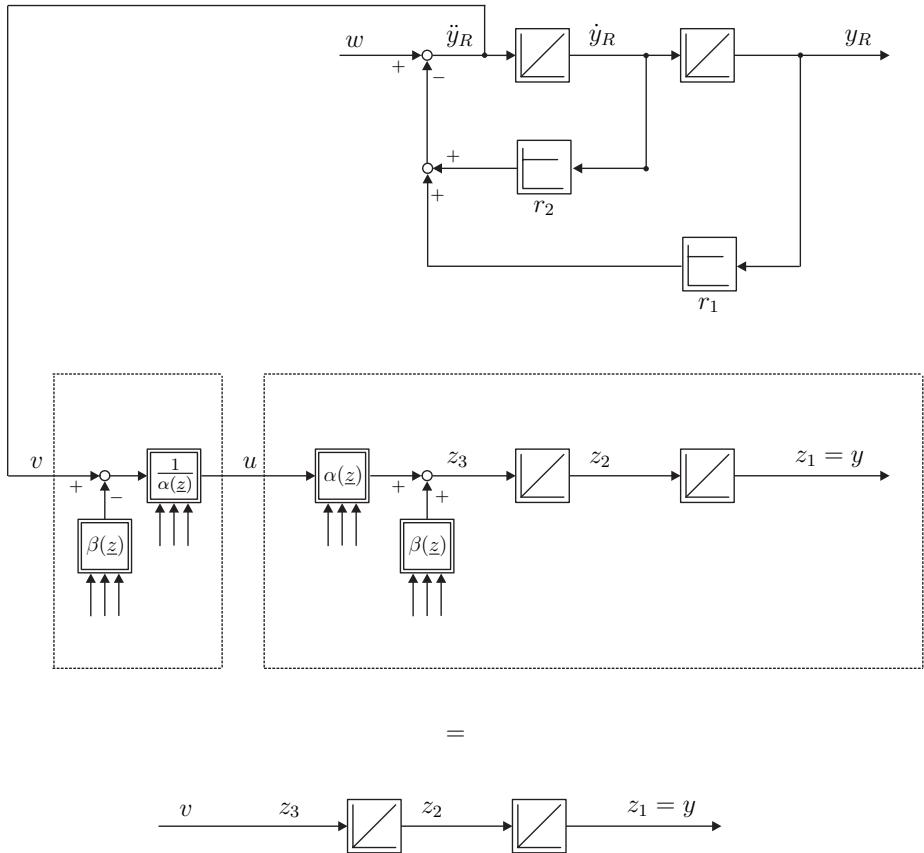
Das Referenzsystem muss mindestens mit der Ordnung  $\delta$  angesetzt werden, weil die nichtlineare Strecke in Byrnes-Isidori Form eine Integratorkette der Länge  $\delta$  im Vorwärtszweig enthält, die, wie bereits diskutiert, mit kausalen Mitteln nicht zu kompensieren ist. Weil die exakte Ein-Ausgangslinearisierung durch die Stellgröße sowohl Nulldynamik, als auch Nichtlinearitäten aufhebt und deren Auswirkung auf das System unterdrückt, ist nach außen hin lediglich die Integratorkette der Ordnung  $\delta$  sichtbar. Daher wird im Folgenden für das Referenzsystem die kleinstmögliche Ordnung  $\delta$  angesetzt.

Um das Ziel  $y(t) \equiv y_R(t)$  zu erreichen, muss für das vorliegende Beispielsystem mit einer Integratorkette der Länge  $\delta = 2$  die Stellgröße

$$u(t) = \frac{\dot{y}_R(t) - \beta(z(t))}{\alpha(z(t))} \quad (12.248)$$

aufgebracht werden. Die zweifache Ableitung erklärt sich aus  $\delta = 2$ : weil die Stellgröße zwei Integratoren überwinden muss, ist die zweite Ableitung des Referenzsignals in die Kette einzuspeisen, damit deren Ausgang mit  $y_R(t)$  übereinstimmt.

Allerdings ist dabei zu berücksichtigen, dass die Anfangszustände der Kette mit denen des Referenzmodells identisch sein müssen. Andernfalls kommt es zu einer Abweichung zwischen beiden Ausgängen, wie bei Betrachtung der Abbildung 12.26 eingesehen werden kann. Der Grund hierfür liegt darin, dass das



**Abb. 12.26:** Asymptotische Modellfolge ohne Fehlerrückführung

beschriebene Vorgehen lediglich eine Steuerung darstellt.

In der praktischen Anwendung ist die Forderung  $y(t) \equiv y_R(t)$  durch die Steuerung (12.248) nicht zu erfüllen. Selbst wenn die Anfangszustände der Kette exakt bekannt sind, führen Störungen zu anwachsenden Abweichungen. Deshalb ist eine Regelung vorzuziehen, die solche Abweichungen beseitigt. Es sind nun folgende Abweichungen anzusetzen:

$$e(t) = y(t) - y_R(t) \quad (12.249)$$

$$\dot{e}(t) = \dot{y}(t) - \dot{y}_R(t) \quad (12.250)$$

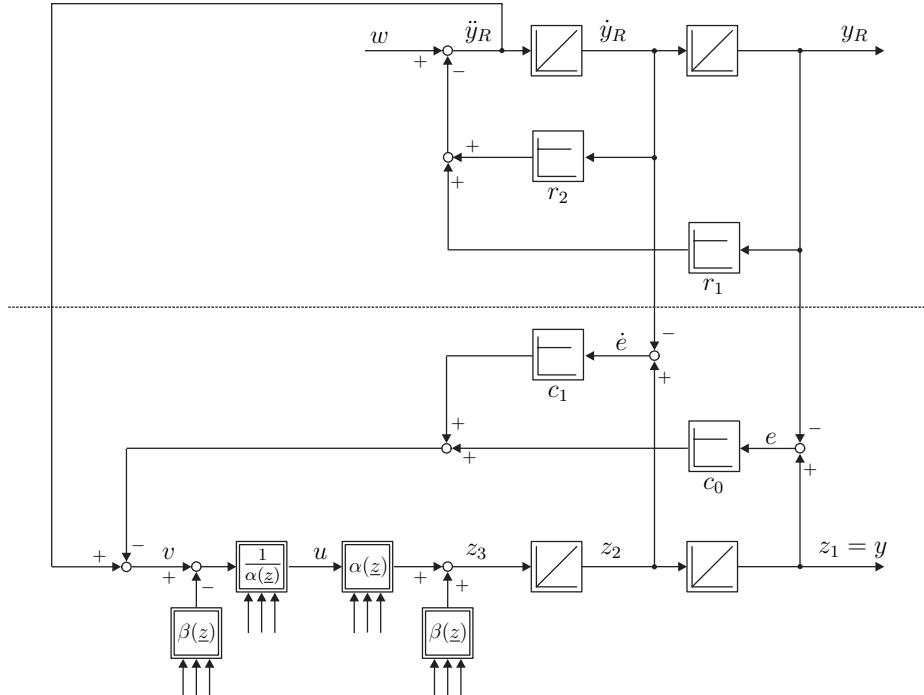
$$\ddot{e}(t) = \ddot{y}(t) - \ddot{y}_R(t) \quad (12.251)$$

Eine zusätzliche Aufschaltung der Fehlersignale auf den Eingang der Kette stellt die benötigte Rückführung dar, welche die bisherige Steuerung zur Regelung

erweitert. Dadurch ändert sich die Stellgröße aus Gleichung (12.248) zu:

$$u(t) = \frac{\ddot{y}_R(t) - \beta(\underline{z}(t)) - c_0 e(t) - c_1 \dot{e}(t)}{\alpha(\underline{z}(t))} \quad (12.252)$$

Die Struktur, die bei Verwendung des Regelgesetzes (12.252) entsteht, ist in Abbildung 12.27 dargestellt.



**Abb. 12.27:** Asymptotische Modellfolgeregelung mit Fehlerrückführung

Deutlich zu erkennen sind die Rückführschleifen, die über  $c_0$  und  $c_1$  den Ausgang  $y$  und dessen Ableitung  $\dot{y}$  auf den Eingang der Kette aufschalten und damit einen Regelkreis schließen. Darüberhinaus kann aus Abbildung 12.27 der Zusammenhang  $\ddot{y}(t) = v(t)$  entnommen werden, der durch die Ein-Ausgangslinearisierung durch

$$u(t) = \frac{v(t) - \beta(t)}{\alpha(t)} \quad (12.253)$$

garantiert ist. Daher führt die Verwendung der Stellgröße (12.252) zu einem Stellsignal

$$v(t) = \ddot{y}_R(t) - c_1 \dot{e}(t) - c_0 e(t). \quad (12.254)$$

am Eingang der resultierenden Integratorkette. Sofern keine Abweichung besteht ( $e = \dot{e} = 0$ ), entspricht die Stellgröße der Regelung (12.254) bzw. (12.252) der Steuerung (12.248). Tritt jedoch ein Fehlersignal auf, hat der zusätzliche Term im Zähler von Gleichung (12.254) bzw. (12.252) die Aufgabe, dieses zu beseitigen. Ob dieser Mechanismus greift, hängt entscheidend von der Wahl der konstanten Gewichtungsfaktoren  $c_0$  und  $c_1$  ab. Es muss daher geklärt werden, wie diese Werte gewählt werden müssen, damit der Fehler  $e(t)$  verschwindet. Hierzu soll nun die Fehlerdifferentialgleichung analysiert werden. Da die Ein-Ausgangslinearisierung den Zusammenhang  $\ddot{y}(t) = v(t)$  garantiert, folgt mit  $\ddot{y}_R(t)$  aus Gleichung (12.251) aus der Stellgröße (12.254) die homogene Differentialgleichung

$$\ddot{e}(t) + c_1\dot{e}(t) + c_0e(t) = 0. \quad (12.255)$$

Um das zeitliche Verhalten des Fehlers  $e(t)$  zu analysieren, bietet sich eine Laplace-Transformation der Fehlerdifferentialgleichung an:

$$\ddot{e}(t) + c_1\dot{e}(t) + c_0e(t) = 0 \quad (12.256)$$



$$s^2e(s) + sc_1e(s) + c_0e(s) = 0 \quad (12.257)$$

Die Differentialgleichung (12.255) ist stabil, wenn das Polynom in (12.257) ein Hurwitzpolynom ist, d.h. wenn sämtliche Nullstellen des Polynoms negative Realeile haben und damit in der linken Halbebene liegen. Der Hurwitztest ergibt, dass ein Polynom zweiter Ordnung mit ausschließlich positiven Koeffizienten ein solches Hurwitzpolynom ist. Dementsprechend müssen die beiden Werte  $c_0$  und  $c_1$  positiv gewählt werden. In der Folge ist die Fehlerdifferentialgleichung asymptotisch stabil und der Fehler  $e(t)$  klingt exponentiell schnell ab: die Regelung erreicht asymptotisch das Ziel  $y(t) = y_R(t)$  trotz abweichenden Anfangszuständen, beispielsweise aufgrund von Störungen. Entsprechend des asymptotisch verschwindenden Fehlerverlaufes erklärt sich die Bezeichnung asymptotische Modellfolgeregelung.

### 12.5.2 Wahl des Referenzsystems

Grundsätzlich kann zur Erzeugung des Referenzsignals ein beliebiges System (linear oder nichtlinear) ausreichender Ordnung herangezogen werden, vorausgesetzt, das System ist stabil. In der Praxis greift man häufig auf ein lineares System zurück, dessen Eigenschaften wohlbekannt sind. Im folgenden wird ein solcher **linearer Vergleichsregelkreis** vorgestellt.

Als Vergleichsregelstrecke wird – bis auf das Proportionalglied mit der Verstärkung 0,1 – ein geregelter Motor verwendet. Das System besteht aus einem Umrichtermodell, welches hier zur Vereinfachung nur aus einem Stromregelkreis besteht, sowie dem Mechanikteil des Motors. Das Gesamtsystem ist in Abb. 12.28

dargestellt. Die Zustandsgleichungen sind in (12.258) aufgeführt. Es handelt sich hierbei um ein System 2. Ordnung. Weiterhin wird angenommen, dass alle Systemzustände messbar sind. Als Regler wird ein Zustandsregler verwendet

$$\begin{aligned}\dot{\underline{x}} &= \underbrace{\begin{bmatrix} -100 & 0 \\ 180 & 0 \end{bmatrix}}_A \cdot \underline{x} + \underbrace{\begin{bmatrix} 1 \\ 0.2 \end{bmatrix}}_b \cdot u \\ y &= \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{c^T} \cdot \underline{x}\end{aligned}\quad (12.258)$$

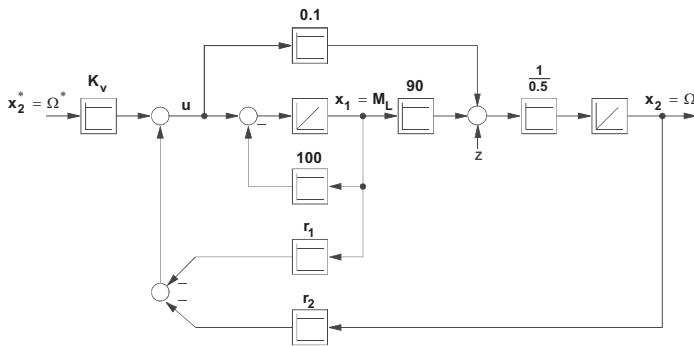


Abb. 12.28: Signalflußplan des linearen Vergleichregelkreises

mit  $\underline{x} = [M_L \quad \Omega]$ . Hierbei ist  $M_L$  das antreibende Luftspaltmoment des Motors und  $\Omega$  die Drehzahl der Maschine.

Für das gegebene System wird zunächst ein Zustandsregler entworfen. Hierbei wird nach der Methode der Polvorgabe vorgegangen [204]. Zunächst wird die Systemmatrix  $\mathbf{A}_{reg}$  des geregelten Systems aufgestellt.

$$\mathbf{A}_{reg} = \mathbf{A} - b \cdot \underline{r}^T = \begin{bmatrix} -100 - r_1 & -r_2 \\ 180 - 0.2 \cdot r_1 & -0.2 \cdot r_2 \end{bmatrix} \quad (12.259)$$

Hierbei sind  $\mathbf{A}$  die Systemmatrix des linearen Systems,  $b$  der Eingangsvektor und  $\underline{r} \in \mathbb{R}^2$  der Vektor mit den Reglerparametern des linearen Zustandsreglers.

Mit Hilfe der charakteristischen Gleichung zweiter Ordnung  $N_{Reg}(s)$ , welche sich aus der Determinante von  $A_{Reg}$  ergibt, kann man die Polvorgabe vornehmen.

$$N_{reg}(s) = \det(s\mathbf{E} - \mathbf{A}_{reg}) = 200 \cdot r_2 + (100 + r_1 + 0.2 \cdot r_2) \cdot s + s^2 \quad (12.260)$$

Nach einem Koeffizientenvergleich mit dem Wunschpolynom  $N_{soll}(s)$  entsprechend dem Dämpfungsoptimum (wobei  $T$  die Ersatzzeitkonstante des Systems beschreibt)

$$N_{soll}(s) = s^2 + q_1 \cdot s + q_0 = s^2 + \frac{2}{T} \cdot s + \frac{2}{T^2} \quad (12.261)$$

ergeben sich die Reglerparameter  $r_1$  und  $r_2$  zu:

$$r_1 = -100 - 0.001 \cdot q_0 + q_1 \quad (12.262)$$

$$r_2 = 0.005 \cdot q_0 \quad (12.263)$$

Um stationäre Genauigkeit zu erhalten, muss nun noch der Vorfilter  $K_v$  berechnet werden.

$$K_v = \frac{1}{\underline{c}^T \cdot (\underline{b} \cdot \underline{r}^T - A)^{-1} \cdot \underline{b}} \quad (12.264)$$

Mit  $T = 0.1$  ergeben sich folgende Werte:

$$r_1 = -80.2$$

$$r_2 = 1$$

$$K_v = 1$$

Der Drehzahlverlauf<sup>1)</sup> ( $\Omega_{lin}$ ) aufgrund eines Sollwertsprunges und einer anschließenden Störung  $z$  auf den Ausgang ist in Abbildung 12.29 dargestellt. Man erkennt erwartungsgemäß, dass die stationäre Genauigkeit nach dem Sollwertprung erreicht, die Störung  $z$  aber nicht voll ausgeregelt wird. Um diesen unerwünschten Effekt zu vermeiden, erweitert man den Zustandsregler um einen I-Anteil. Dieser I-Anteil kann als zusätzlicher Zustand der Regelstrecke aufgefasst werden. Die Streckenmatrix  $\mathbf{A}$  und der Eingangsvektor  $\underline{b}$  ergeben sich nun zu:

$$\mathbf{A} = \begin{pmatrix} -100 & 0 & 0 \\ 180 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 1 \\ 0.2 \\ 0 \end{pmatrix} \quad (12.265)$$

Somit ergeben sich die Systemmatrix  $\mathbf{A}_{Reg} = \mathbf{A} - \underline{b} \cdot \underline{r}^T$  und der Eingangsvektor  $\underline{b}_{Reg}$  des geschlossenen Regelkreises zu:

$$\mathbf{A}_{Reg} = \begin{pmatrix} -100 - r_1 & -r_2 & -r_3 \\ 180 - 0.2 \cdot r_1 & -0.2 \cdot r_2 & -0.2 \cdot r_3 \\ 0 & -1 & 0 \end{pmatrix} \quad \underline{b}_{Reg} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (12.266)$$

In Abbildung 12.39 ist dies veranschaulicht. Mit den bekannten Parametern, dem DO Wunschkoeffizienten 3. Ordnung  $N_{soll}(s) = s^3 + \frac{4}{T}s^2 + \frac{8}{T^2}s + \frac{8}{T^3}$  und der DO-Zeitkonstante  $T = 0.1s$  berechnen sich die Reglerparameter zu:

$$\begin{aligned} r_1 &= -100 + 0.1 \cdot 10^{-5} \cdot q_0 - 0.001 \cdot q_1 + q_2 = -60.792 \\ r_2 &= -0.5 \cdot 10^{-5} \cdot q_0 + 0.005 \cdot q_1 = 3.96 \\ r_3 &= -0.005 \cdot q_0 = -40 \end{aligned} \quad (12.267)$$

---

<sup>1)</sup> Der Index *lin* der Drehzahl bezeichnet die lineare Vergleichsstrecke mit einem linearen Zustandsregler.

In Abbildung 12.30 erkennt man, dass nun auch die Störung  $z$  voll ausgeregelt wird. Weiterhin ist festzustellen, dass durch den zusätzlichen Zustand sich die Dynamik des geschlossenen Regelkreises verändert hat.

Regelung einer linearen Strecke 2.Ordnung nach DO

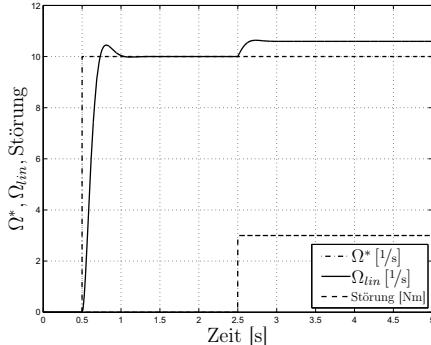


Abb. 12.29:  $\Omega_{lin}$  bei Zustandsregelung ohne I-Anteil

Regelung einer linearen Strecke 2.Ordnung nach DO

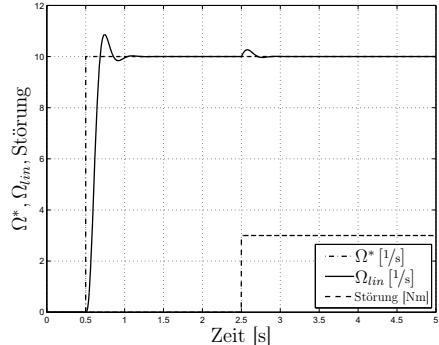


Abb. 12.30:  $\Omega_{lin}$  bei Zustandsregelung mit I-Anteil

Die lineare Strecke aus (12.258) wird nun um eine Nichtlinearität erweitert.

$$\underline{k}_{\mathcal{NL}} \cdot \mathcal{NL}(x_E) = \begin{pmatrix} 0 \\ -\frac{1}{0.5} \end{pmatrix} \cdot \kappa \cdot \arctan(2 \cdot n) \quad (12.268)$$

Man erhält

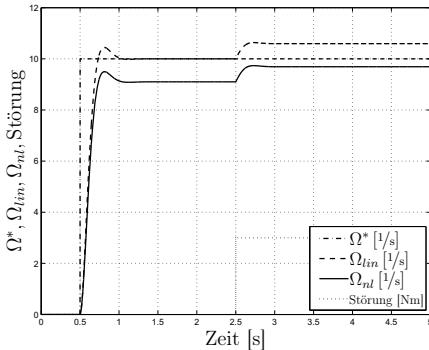
$$\begin{aligned} \dot{x} &= \underbrace{\begin{bmatrix} -100 & 0 \\ 180 & 0 \end{bmatrix}}_{\underline{A}} \cdot \underline{x} + \underbrace{\begin{bmatrix} 1 \\ 0.2 \end{bmatrix}}_{\underline{b}} \cdot u + \begin{bmatrix} 0 \\ -\frac{1}{0.5} \end{bmatrix} \cdot \kappa \cdot \arctan(2 \cdot n) \\ y &= \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{\underline{c}^T} \cdot \underline{x} \end{aligned} \quad (12.269)$$

Diese Nichtlinearität wirkt wie ein zusätzliches, geschwindigkeitsabhängiges Gegenmoment, ähnlich einer Reibung. Die Auswirkung der Reibung auf das System kann mit der Konstanten  $\kappa$  variiert werden. Um die Effekte dieser Nichtlinearität zu veranschaulichen, wird die nichtlineare Regelstrecke mit dem Zustandsregler ohne I-Anteil geregelt. In den Abbildungen 12.31 und 12.32 sind die Drehzahlverläufe des linearen Vergleichssystems ( $\Omega_{lin}$ ) und des nichtlinearen Regelkreises<sup>2)</sup> ( $\Omega_{nl}$ ) für  $\kappa = 3$  und  $\kappa = 15$  zu sehen. Man erkennt sehr deutlich die bleibende stationäre Regelabweichung. Diese Abweichung ist noch einmal in

<sup>2)</sup> Der Index  $nl$  charakterisiert die nichtlineare Strecke, welche mit einem linearen Zustandsregler geregelt wird.

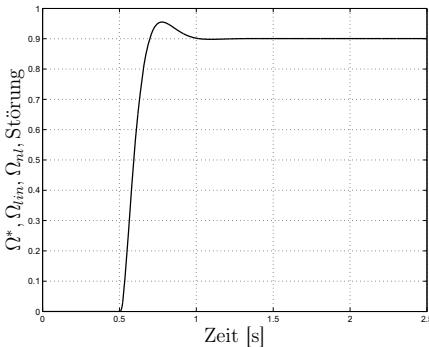
den beiden Abbildungen 12.33 und 12.34 zu sehen. Hier ist jeweils die Differenz zwischen den Drehzahlen  $\Omega_{lin}$  und  $\Omega_{nl}$  dargestellt.

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler ohne I-Anteil



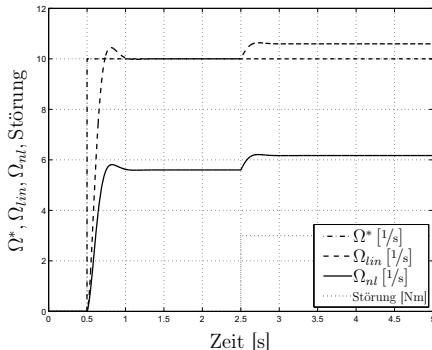
**Abb. 12.31:**  $\Omega_{lin}$  und  $\Omega_{nl}$  bei Regelung ohne I-Anteil und  $\kappa = 3$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler ohne I-Anteil



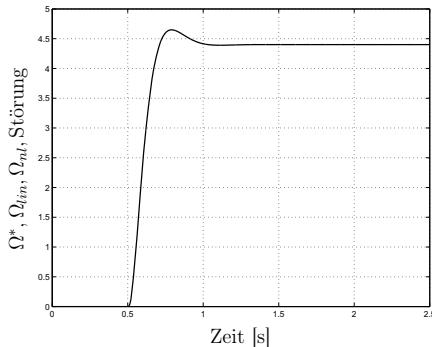
**Abb. 12.33:**  $e = \Omega_{lin} - \Omega_{nl}$  bei einer Regelung ohne I-Anteil und  $\kappa = 3$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler ohne I-Anteil



**Abb. 12.32:**  $\Omega_{lin}$  und  $\Omega_{nl}$  bei Regelung ohne I-Anteil und  $\kappa = 15$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler ohne I-Anteil

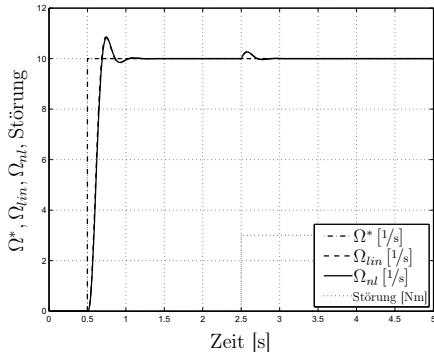


**Abb. 12.34:**  $e = \Omega_{lin} - \Omega_{nl}$  bei einer Regelung ohne I-Anteil und  $\kappa = 15$

Sehr viel besser sind die Ergebnisse, wenn man die nichtlineare Regelstrecke mit einem Zustandsregler mit zusätzlichem I-Anteil betreibt (siehe Abbildungen 12.35 bis 12.38).

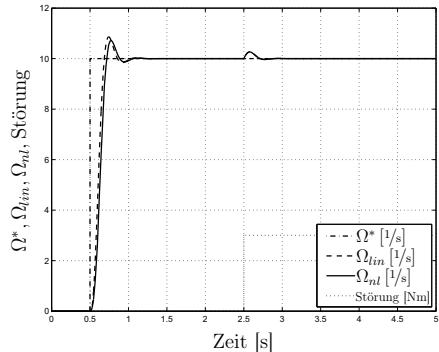
Hierbei wird, wie erwartet, die bleibende Regelabweichung zu Null und auch Störungen werden ausgeregelt. Abbildung 12.39 veranschaulicht den Signalfussplan für diesen Fall. In diesem Beispiel gelingt es bereits, mit Hilfe des zusätzlichen I-Anteiles die Nichtlinearität vollständig auszuregeln. Es verbleibt nur noch eine geringe Abweichung in der Dynamik (siehe Abbildungen 12.37 und 12.38).

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler mit I-Anteil



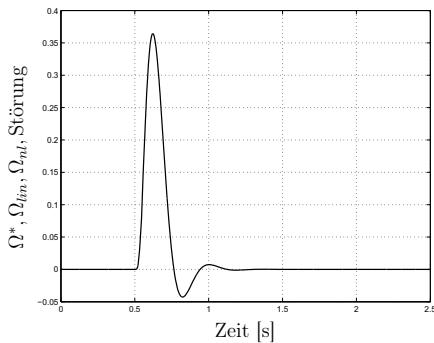
**Abb. 12.35:**  $\Omega_{lin}$  und  $\Omega_{nl}$  bei Regelung mit I-Anteil und  $\kappa = 3$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler mit I-Anteil



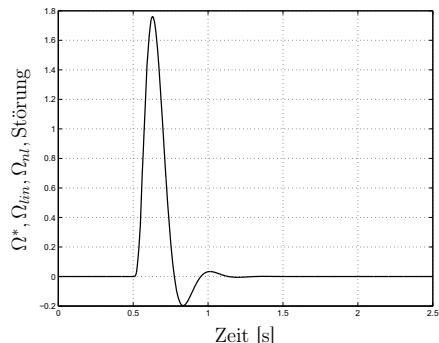
**Abb. 12.36:**  $\Omega_{lin}$  und  $\Omega_{nl}$  bei Regelung mit I-Anteil und  $\kappa = 15$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler mit I-Anteil



**Abb. 12.37:**  $e = \Omega_{lin} - \Omega_{nl}$  bei einer Regelung mit I-Anteil und  $\kappa = 3$

Vergleich zwischen nichtlinearer und linearer Strecke mit einem Regler mit I-Anteil

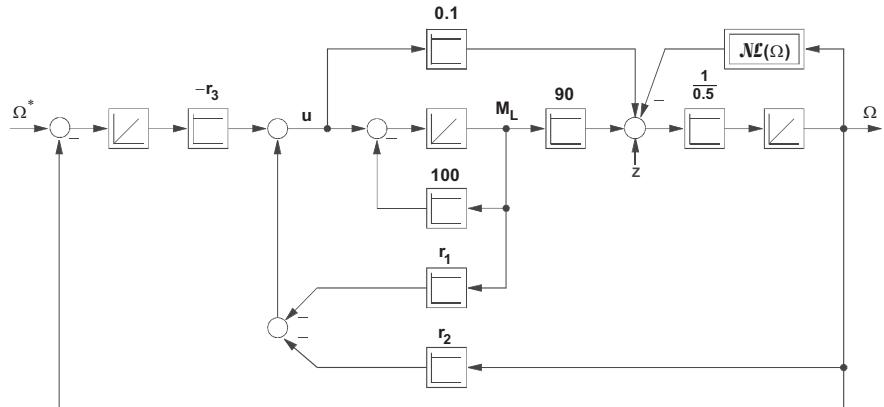


**Abb. 12.38:**  $e = \Omega_{lin} - \Omega_{nl}$  bei einer Regelung mit I-Anteil und  $\kappa = 15$

In anderen Fällen gelingt dies jedoch nicht unbedingt. Es soll darauf hingewiesen werden, dass der Anwender zunächst einmal einen Zustandsregler mit I-Anteil in Erwägung ziehen sollte, bevor er versucht mit nichtlinearen Regelungsmethoden den Regelkreis zu betreiben.

Eine exakte Aufprägung des linearen Referenzverhaltens gelingt allerdings nur mit einem nichtlinearen Regler. Dazu wird das System in Abb. 12.39 in NNF transformiert. Dabei ist

$$z_1 = h(\underline{x}) = x_2 \quad (12.270)$$

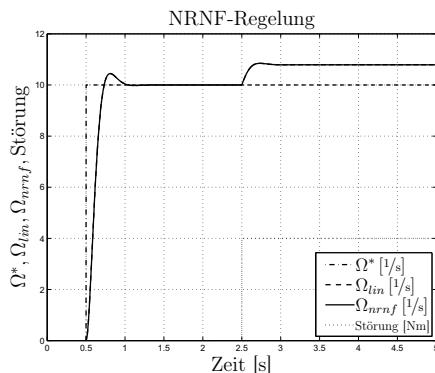


**Abb. 12.39:** Signalflussplan der nichtlinearen Strecke 2. Ordnung mit Zustandsregler

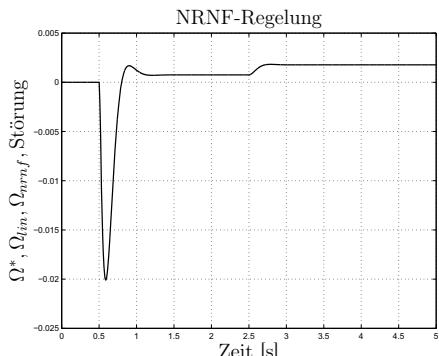
Das System hat Relativgrad  $\delta = 1$ . Daher muss  $z_2 = \lambda(x)$  direkt so bestimmt werden, dass  $\frac{d\lambda}{dx}g = 0$ . Man erhält,

$$z_2 = x_1 - 5x_2 \quad (12.271)$$

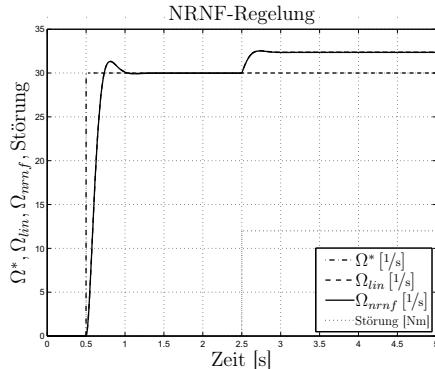
Hieraus kann dann das nichtlineare Regelgesetz gemäß Gleichung (12.84) und entsprechend Abb. 12.13 (proportionaler Zustandsregler) bestimmt werden. Eine Simulation des nichtlinearen Regelkreises schliessen diesen Abschnitt ab.



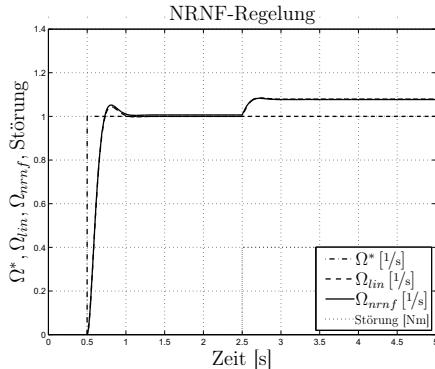
**Abb. 12.40:** Drehzahlverläufe bei Reglung nach NRNF



**Abb. 12.41:**  $\Omega_{lin} - \Omega_{nrrf}$  bei Regelung nach NRNF



**Abb. 12.42:** Drehzahlverläufe bei Regelung nach NRNF



**Abb. 12.43:** Drehzahlverläufe bei Regelung nach NRNF

In den Abbildungen 12.40, 12.42 und 12.43 kann man keine Abweichung zwischen den Drehzahlverläufen des linearen Vergleichsregelkreises ( $\Omega_{lin}$ ) und den Drehzahlverläufen des nichtlinearen Systems ( $\Omega_{nrf}$ ), welches mittels der NRNF geregelt wurde, erkennen. Dies gilt sowohl für die Sprungantworten bzgl. verschiedener Sollwertsprünge, als auch für die Störantworten. Der Verlauf des Differenzsignals zwischen  $\Omega_{lin}$  und  $\Omega_{nrf}$  in Abbildung 12.41 zeigt dies noch einmal sehr deutlich.

## 12.6 Der Einsatz von neuronalen Beobachtern

Bisher wurde bei allen Regelungsverfahren angenommen, dass die Nichtlinearität bekannt ist, sowie dass alle Zustände ( $\underline{x}$ ) messbar sind.

In diesem Abschnitt werden Wege aufgezeigt, um das Problem der vollständigen Zustandsmeßbarkeit für eine Regelung nichtlinearer Strecken zu beseitigen. Dies wird mit Hilfe lernfähiger Beobachter gelöst, die sowohl geschätzte Zustandsgrößen ( $\hat{\underline{x}}$ ), wie auch eine approximierte Nichtlinearität ( $\hat{\mathcal{N}}\mathcal{L}$ ) liefern.

Anschließend werden diese Beobachter in einem Verfahren zur adaptiven Regelung von nichtlinearen Strecken eingesetzt. Hierbei soll mit Hilfe des in Abschnitt 12.5 gefundenen nichtlinearen Regelgesetzes wieder das Ziel erreicht werden, dass sich der nichtlineare Regelkreis so verhält wie das lineare Referenzsystem.

### 12.6.1 Anwendung auf das nichtlineare System 2. Ordnung

Für das nichtlineare System in Abb. 12.39 aus Kap. 12.5.2 sei vorausgesetzt, dass  $\mathbf{A}, \mathbf{b}, \underline{c}, d$  sowie der Angriffspunkt der Nichtlinearität bekannt sind. Unbekannt sei

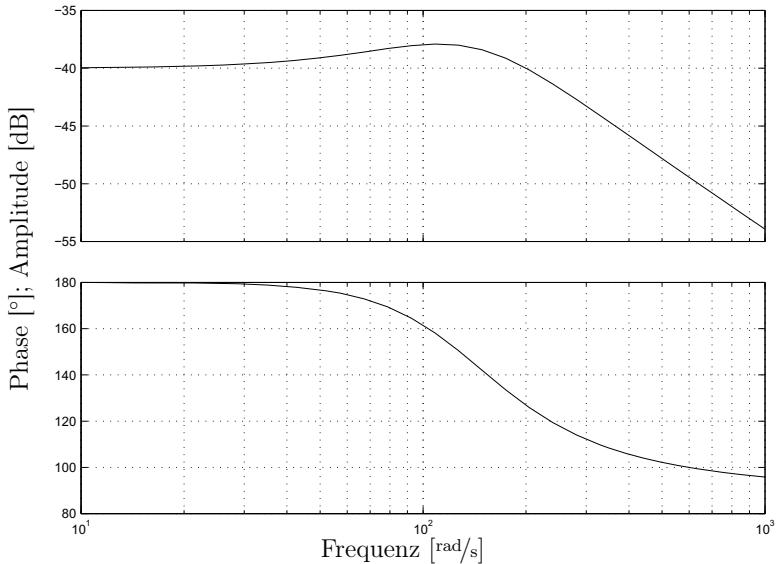
nur die zu identifizierende Nichtlinearität  $\mathcal{NL}$ . Zunächst wird überprüft, ob  $H(s)$  ungleich Null ist

$$H(s) = \underline{c}^T (s\mathbf{E} - \mathbf{A})^{-1} \underline{k}_{\mathcal{NL}} = -\frac{2}{s} \neq 0 \quad (12.272)$$

Nun wird gemäß [142] per Polvorgabe für den linearen Streckenanteil der Beobachterrückführvektor  $\underline{l}$  festgelegt. Die Fehlerübertragungsfunktion  $H(s)$  lautet nun

$$H(s) = -2 \cdot \frac{100 + s}{100l_1 + 180l_2 + (100 + l_2)s + s^2} \quad (12.273)$$

Am Bodediagramm in Abbildung 12.44 sieht man deutlich, dass für diese Fehlerübertragungsfunktion die SPR-Bedingung nicht erfüllt ist.



**Abb. 12.44:** Bode Diagramm zu  $H(s)$  aus Gleichung (12.273) mit  $\underline{l}$  aus Gleichung (12.274)

## 12.6.2 Simulationsergebnis

Die Beobachterkoeffizienten werden mittels Polvorgabe nach DO ( $T=0.01s$ ) festgelegt. Es ergeben sich folgende Werte:

$$l_1 = 55.55 \quad l_2 = 200 \quad (12.274)$$

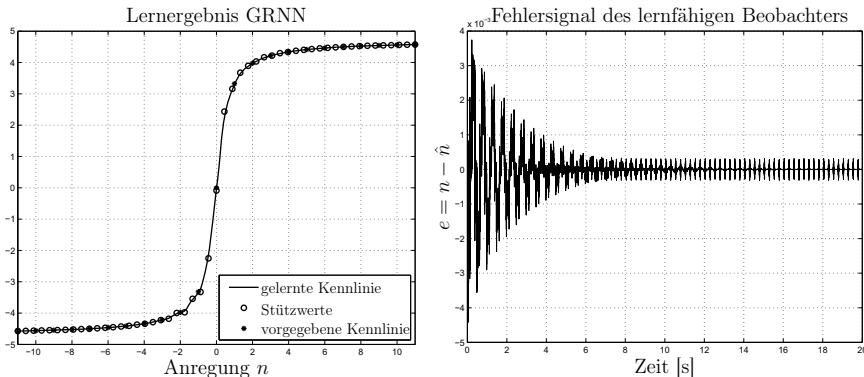
Als Nichtlinearität wird

$$\mathcal{NL}(x_2) = \kappa \cdot \arctan(2 \cdot x_2) \quad (12.275)$$

mit  $\kappa = 3$  gewählt. Für das GRNN werden folgende Einstellungen gewählt:

- Stützwertezahl  $p = 51$ ;
- Glättungsfaktor  $\sigma_{norm} = 0.5$
- Lernschrittweite  $\eta = 1000000$

Man sieht in Abbildung 12.45, dass der Beobachter die Nichtlinearität sehr gut identifiziert. Auch das Fehlersignal wird sehr schnell sehr klein. (Abbildung 12.46)



**Abb. 12.45:** Lernergebnis des lernfähigen Beobachters

**Abb. 12.46:** Fehlersignal

### 12.6.3 Anwendung auf Regelung mit NRNF

Auch bei einer Anwendung des lernfähigen Beobachters auf die Regelung nach der NRNF schätzt der Beobachter die Systemzustände und identifiziert die Nichtlinearität „online“. Da bei der Regelung nach NRNF für das Regelgesetz die erste Ableitung nach  $x_E$  benötigt wird, muss zusätzlich noch die erste Ableitung gelernt werden. Es gilt

$$\frac{d\mathcal{NL}(x_E)}{dt} = \frac{d\mathcal{NL}(x_E)}{dx_E} \cdot \dot{x}_E \quad (12.276)$$

Der Beobachter liefert  $\widehat{\mathcal{NL}}(x_E)$ . Dieses Signal wird nach der Zeit differenziert und dient zum Fehlervergleich eines GRNNs, dessen Ausgangssignal mit  $\dot{x}_E$  multipliziert wird. Abbildung 12.47 verdeutlicht diese Lernstruktur.

Mit Hilfe dieser Lernarchitektur können beliebig viele weitere Ableitungen von  $\mathcal{NL}(x_E)$  identifiziert werden.

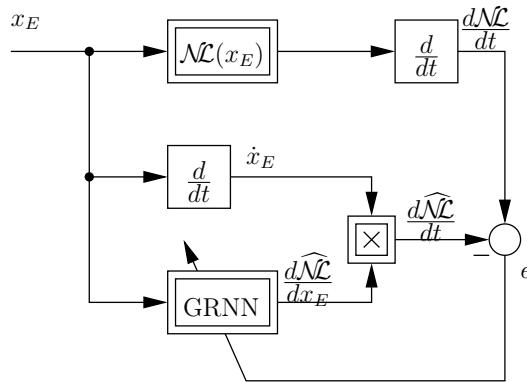


Abb. 12.47: Blockschaltbild für das Lernprinzip der Ableitung

Für die gewählte Nichtlinearität  $\mathcal{NL}(x_2) = 3 \cdot \arctan(2 \cdot x_2)$  ergibt sich

$$\frac{d\mathcal{NL}(x_2)}{dx_2} = \frac{6}{1 + 4 \cdot x_2^2} \quad (12.277)$$

In der Abbildung 12.48 ist das Lernergebnis für diese erste Ableitung zu sehen.

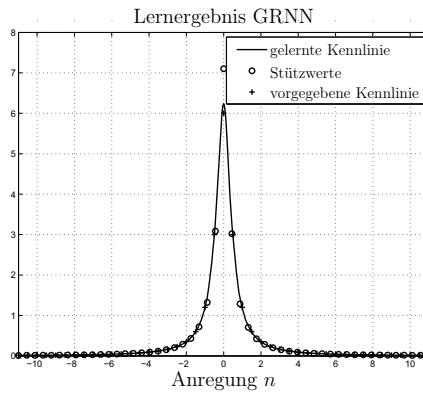


Abb. 12.48: erste Ableitung von  $\mathcal{NL}(x_E)$

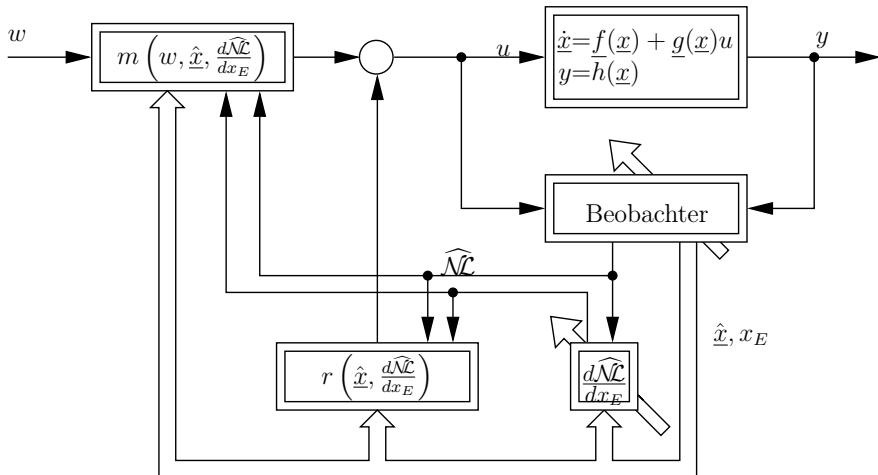
Das GRNN zur Identifizierung der Ableitung hat folgende Einstellungen:

- Stützwertzahl  $p = 51$
- Glättungsfaktor  $\sigma_{norm} = 0.5$
- Lernschrittweite  $\eta = 1$

Wie man bereits in Abbildung 12.47 sieht, braucht man hier keine Fehlerübertragungsfunktion, da der Ausgang des GRNN direkt auf das Fehlersignal wirkt. Da aber durch die Multiplikationsstelle das Vorzeichen des GRNN-Ausgangs durch  $\dot{x}_E$  wechseln kann, ist dafür zu sorgen, dass bei negativem  $\dot{x}_E$  die Lernschrittweite auch negativ wird. Dies entspricht einer vorzeichenvariablen Fehlerübertragungsfunktion. Es ergibt sich das folgende Adoptionsgesetz

$$\dot{\Phi} = -\eta \cdot \text{sign}(\dot{x}_E) \cdot \underline{\mathcal{A}} \cdot e \quad (12.278)$$

Abbildung 12.49 zeigt den Signalflussplan für die Regelung des nichtlinearen Systems 2. Ordnung mit Hilfe eines lernfähigen Beobachters.

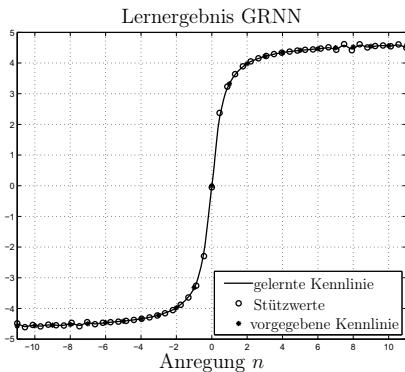


**Abb. 12.49:** Signalflussplan der NNF Regelung mit Hilfe eines neuronalen Beobachters

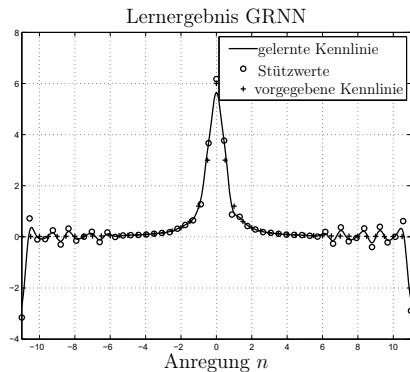
#### 12.6.4 Simulationsergebnisse

Die folgenden Simulationsergebnisse wurden während einer „online“ Identifizierung erzielt. Dies bedeutet, dass es möglich war, während eines Regelsvorganges die Nichtlinearität und ihre erste Ableitung zu identifizieren. In den Abbildungen 12.50 und 12.51 ist zu erkennen, dass der nichtlineare Beobachter die Nichtlinearität einschließlich der ersten Ableitung richtig identifiziert hat. In Abbildung 12.52 ist zu sehen, dass mit zunehmendem Lernfortschritt die Abweichungen zwischen den Drehzahlen des linearen Vergleichssystems ( $\Omega_{lin}$ ) und der adaptiven NNF-Regelung ( $\Omega_{NNF}$ ) immer geringer werden. Den Fehlersignalen in Abbildung 12.53 bzw. 12.54 ist zu entnehmen, dass der Identifikationsvorgang nach ca.

20 Sekunden noch nicht abgeschlossen ist, da das Fehlersignal noch weiter abnimmt. Nach ca. 30 Sekunden stellt sich allerdings kein weiterer Lernfortschritt ein.

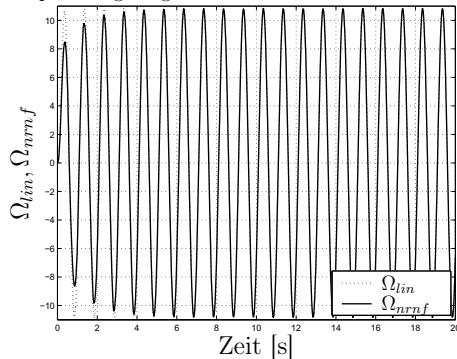


**Abb. 12.50:** identifizierte Nichtlinearität  $\mathcal{NL}(x_E)$



**Abb. 12.51:** erste Ableitung von  $\mathcal{NL}(x_E)$

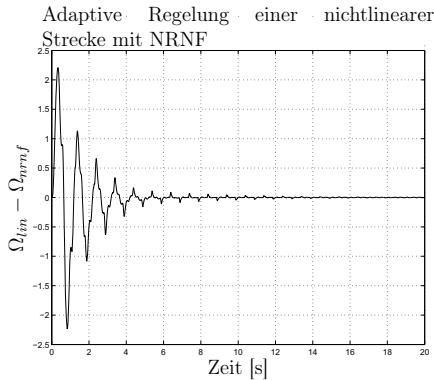
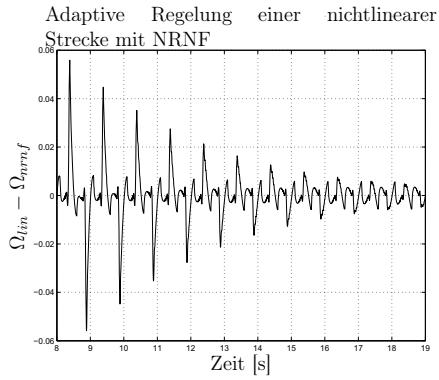
Adaptive Regelung einer nichtlinearen Strecke mit NRNF



**Abb. 12.52:** Drehzahlverläufe während des Lernens

## 12.6.5 Kurzzusammenfassung: NRNF und lernfähiger Beobachter

In diesem Abschnitt wurde gezeigt, wie man mit Hilfe lernfähiger Beobachter adaptive Regelungen realisieren kann. Hierbei identifizierte der neuronale Beobachter die vorhandene Nichtlinearität. Diese zusätzliche Information wurde in

Abb. 12.53:  $\Omega_{lin} - \Omega_{nrrnf}$ Abb. 12.54:  $\Omega_{lin} - \Omega_{nrrnf}$ 

dem nichtlinearen Regelgesetz der NRRNF-Regelung verarbeitet, um ein möglichst lineares Verhalten des gesamten Regelkreises zu erzielen. Auch ein „online“ Lernen ist durchaus möglich. Da sich die NRRNF-Regelung auf einen linearen Zustandsregler zurückführen lässt, muss bei einer Online-Identifikation sicher gestellt sein, dass bereits der lineare Zustandsregler für die nichtlineare Strecke Stabilität gewährleistet. Andernfalls muss während des Lernvorganges ein einfacher Regler, welcher auf jeden Fall Stabilität garantiert, die Regelung so lange übernehmen, bis die Nichtlinearität und ihre Ableitungen ungefähr gelernt worden sind. Dann kann eine Umschaltung auf die dann stabile NRRNF-Regelung erfolgen und das Regelergebnis entscheidend verbessern.

## 12.7 Ein–Ausgangslinearisierung zur neuronalen Regelung

In den vorhergehenden Kapiteln wurden verschiedene Verfahren zur Identifikation nichtlinearer Strecken vorgestellt. Die verschiedenen Verfahren unterscheiden sich im Wesentlichen hinsichtlich der Vorkenntnisse bei den Parametern und der Struktur. Es wurde in den Kapiteln 12.3 bis 12.6 als folgender Schritt nach der Identifikation der unbekannten Strecke die Regelung dieser nun bekannten Strecke vorgestellt. Auch hier gibt es unterschiedliche Verfahren, die Vor- und Nachteile dieser Verfahren wurden herausgearbeitet. Es stellt sich nun die Frage, ob ein getrenntes Vorgehen, d.h. zuerst Identifikation und danach Reglerentwurf, unbedingt notwendig ist.

In diesem Abschnitt soll ein erster Ansatz vorgestellt werden, bei dem eine unbekannte nichtlineare Strecke direkt geregelt wird. Wenn nun die Strecke unbekannt ist und ein Regler entworfen werden soll, dann muss für den Reglerentwurf ein gewünschtes Modell des geregelten Systems vorliegen. Die hier dargestellte Vorgehensweise nutzt ein Verfahren, welches in [228] vorgestellt wurde, allerdings werden für die Funktionsapproximatoren nun GRNN verwendet [193]. Die Grundidee ist, einen MRAC-Ansatz (Model Reference Adaptiv Control) zu wählen, aber gleichzeitig den Grundgedanken der exakten Ein-

Ausgangslinearisierung aus Kapitel 12.4 zu verwenden. Dies bedeutet, das gewünschte Modell des MRAC enthält eine Integratorkette entsprechend des Relativgrads der unbekannten nichtlinearen Strecke. Die Integratorkette im MRAC repräsentiert somit die linearisierte unbekannte nichtlineare Strecke. Ausgehend von dieser Integratorkette wird nun der Regler für die Integratorkette entworfen. Der Reglerentwurf kann die linearen Entwurfsverfahren nutzen. Ein kurzer Blick auf das Regelbeispiel in Abbildung 12.56 (Kapitel 12.7.2) macht die Vorgehensweise beim Wunschpolynom deutlich. Da die unbekannte nichtlineare Strecke in diesem Beispiel zweiter Ordnung ( $PT_1$ - und  $I$ -Glied) ist, muss eine Integratorkette zweiter Ordnung angesetzt werden. Die wesentlichen Punkte bei diesem Vorgehen sind, dass erstens der Relativgrad der nichtlinearen Strecke bekannt ist und zweitens bei der Realisierung des Wunschverhaltens des geregelten Systems Stellgliedbeschränkungen berücksichtigt werden müssen, um wind up zu vermeiden. Wenn nun die unbekannte nichtlineare Strecke bekannt wäre, dann wäre die input-output-Linearisierung möglich (Voraussetzung: die Nichtlinearität(en) ist (sind) differenzierbar, hier gegeben). In diesem Fall würde somit die Steuergröße sowohl die Integratorkette des MRAC-Modells als auch die resultierende Integratorkette der realen linearisierten Strecke steuern. Damit ist sichergestellt, dass die reale Strecke dem Wunschmodell folgt. Die Problematik ist aber nun, dass die nichtlineare Strecke unbekannt ist und somit die Lie-Ableitungen nicht bekannt sind. Der Ansatz in [228, 193] ist, diese Lie-Ableitungen zu lernen. Dieser Lernvorgang ist im unteren Teil der Abbildung 12.56 dargestellt.

### 12.7.1 Erlernen der Input-Output Linearisierung mit Neuronalen Netzen

Da die *Lie*-Ableitungen (vergleiche Kap. 12.3) nichtlineare statische Funktionen sind, können sie durch neuronale Netze nachgebildet werden. Dazu sollen die in Kap. 12.3 diskutierten GRNNs verwendet werden. Die *Lie*-Ableitung  $L_f^n h(\underline{x})$  wird durch das GRNN  $\underline{\Theta}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x})$  nachgebildet, während der Ausdruck  $L_g L_f^{n-1} h(\underline{x})$  durch ein zweites GRNN  $\hat{\underline{\Theta}}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x})$  übernommen wird. Eingesetzt in Gleichung (12.84) erhält man:

$$u = \left[ -\hat{\underline{\Theta}}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) + v \right] \cdot \left[ \hat{\underline{\Theta}}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \right]^{-1} \quad (12.279)$$

Mit dem in Gleichung (3.5) eingeführten Adoptionsfehler  $e$  und dem Parameterfehler  $\Phi$  nach Gleichung (3.6) lassen sich die unbekannten *Lie*-Ableitungen darstellbar als:

$$\begin{aligned} L_f^\delta h(\underline{x}) &= \underline{\Theta}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) + e_f(\underline{x}) \\ L_g L_f^{\delta-1} h(\underline{x}) &= \hat{\underline{\Theta}}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) + e_g(\underline{x}) \end{aligned} \quad (12.280)$$

Die beiden Vektoren  $\underline{\Theta}_f^T$  und  $\underline{\Theta}_g^T$  sind die unbekannten optimalen Parametervektoren. Setzt man die Gleichungen (12.280) in Gleichung (12.67) ein, so erhält man zunächst

$$y^{(n)} = \underline{\Theta}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) + e_f(\underline{x}) + \underline{\Theta}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \cdot u + e_g(\underline{x}) \cdot u \quad (12.281)$$

Eliminiert man nun gemäß der bekannten Definition für einen Parameterfehlervektor (Gl. 3.6) die optimalen Parametervektoren mittels  $\underline{\Theta}_f^T = \hat{\underline{\Theta}}_f^T - \underline{\Phi}_f^T$  und  $\underline{\Theta}_g^T = \hat{\underline{\Theta}}_g^T - \underline{\Phi}_g^T$  aus Gleichung (12.281), dann ergibt sich:

$$y^{(n)} = -\underline{\Phi}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) - \underline{\Phi}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \cdot u + e_f(\underline{x}) + e_g(\underline{x}) \cdot u + \underbrace{\hat{\underline{\Theta}}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) + \hat{\underline{\Theta}}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \cdot u}_{=v} \quad (12.282)$$

bzw. mit Gleichung (12.279)

$$y^{(n)} = -\underline{\Phi}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) - \underline{\Phi}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \cdot u + e_f(\underline{x}) + e_g(\underline{x}) \cdot u + v \quad (12.283)$$

Um dem Ausgang  $y_m$  eines Wunschmodells folgen zu können, wird das Fehlersignal

$$e = y - y_m \quad (12.284)$$

definiert und ein lineares Rückführgesetz mit den Koeffizienten  $\alpha_k (k = 1, 2, \dots, n)$  wie folgt verwendet:

$$v = y_m^{(n)} - \alpha_n \cdot e^{(n-1)} - \dots - \alpha_2 \cdot \dot{e} - \alpha_1 \cdot e \quad (12.285)$$

Durch Einsetzen dieser Gleichung in (12.283) erhält man die Differentialgleichung für den Fehler  $e$ :

$$e^{(n)} + \alpha_n \cdot e^{(n-1)} + \dots + \alpha_2 \cdot \dot{e} + \alpha_1 \cdot e = -\underline{\Phi}_f^T \cdot \underline{\mathcal{A}}_f(\underline{x}) - \underline{\Phi}_g^T \cdot \underline{\mathcal{A}}_g(\underline{x}) \cdot u + e_f(\underline{x}) + e_g(\underline{x}) \cdot u \quad (12.286)$$

$s^n + \alpha_n \cdot s^{n-1} + \dots + \alpha_2 \cdot s + \alpha_1$  muss aus Stabilitätsgründen ein Hurwitzpolynom sein. Um Fehlerkonvergenz zu erzielen, muss die Fehlerübertragungsfunktion (vergleiche Kapitel 5.2.2) die SPR-Bedingung erfüllen. Dies ist bei der Erzeugung des modifizierten Fehlers  $\tilde{e}$ , welcher für die Gewichtsadaption verwendet wird, zu beachten.

$$\tilde{e} = \tilde{\alpha}_n \cdot e^{(n-1)} + \dots + \tilde{\alpha}_2 \cdot \dot{e} + \tilde{\alpha}_1 \cdot e \quad (12.287)$$

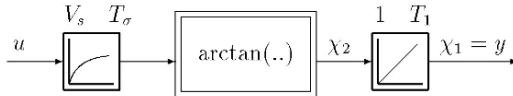
Mit dem modifizierten Fehlersignal  $\tilde{e}$  gelten nach (4.12) die folgenden Lernregeln:

$$\begin{aligned} \dot{\underline{\Theta}}_f &= \dot{\underline{\Phi}}_f = -\eta_f \cdot \tilde{e} \cdot \underline{\mathcal{A}}_f(\underline{x}) \\ \dot{\underline{\Theta}}_g &= \dot{\underline{\Phi}}_g = -\eta_g \cdot \tilde{e} \cdot \underline{\mathcal{A}}_g(\underline{x}) \end{aligned} \quad (12.288)$$

$\eta_f$  und  $\eta_g$  sind die Lernschrittweiten. Sind die verwendeten GRNNs zu guter Approximation in der Lage, d.h.  $e_f(\underline{x}) \approx 0$  und  $e_g(\underline{x}) \approx 0$ , dann führen die obigen Lernregeln, falls  $u$  in (12.279) beschränkt bleibt, zur Fehlerkonvergenz im Sinne von  $\tilde{e} \rightarrow 0$ .

### 12.7.2 Regelung einer nichtlinearen Strecke zweiter Ordnung

Das vorgestellte Verfahren zum Erlernen der Input-Output Linearisierung mit GRNNs wird nun exemplarisch auf ein nichtlineares System zweiter Ordnung gemäß Abb. 12.55 angewendet.



**Abb. 12.55:** Nichtlineare Beispielstrecke zweiter Ordnung

Es sei kein Vorwissen über die  $PT_1$ -Struktur und die Nichtlinearität vorhanden, lediglich der Integrator mit seinem Parameter  $T_1$  sei als bekannt angenommen.

Die Realisierung des Verfahrens zeigt der Signalflussplan in Abb. 12.56, wobei zur Vereinfachung und Rechenzeitsparnis  $\mathcal{A}_f = \mathcal{A}_g = \mathcal{A}$  gesetzt wurde. Das Wunschmodell besteht aus der erwähnten linearen Integratorkette zweiter Ordnung, die hier zeitoptimal geregelt ist. Es könnte jedoch auch jedes andere auf eine lineare Integratorkette anwendbare Regelverfahren verwendet werden.

Die Gewichte der GRNNs wurden mit beliebigen Werten initialisiert. Als Sollwertverlauf für das Wunschmodell  $\varpi(t)$  wurden Sprungfunktionen gewählt, deren Sprunghöhen stochastisch verteilt waren. Nach etwa 250 Sekunden Lernzeit folgt der Streckenausgang  $y(t)$  dem des Wunschmodells  $y_m(t)$  bereits sehr gut. Abbildung 12.57 zeigt das dynamische Verhalten des Wunschmodells  $y_m(t)$  und der geregelten nichtlinearen Strecke  $y(t)$ . Die beiden Signale sind praktisch nicht unterscheidbar. Das bedeutet, es ist mittels MRAC-Ansatz, exakter Ein-Ausgangs-Linearisierung und lernender Lie-Ableitungen möglich, diese unbekannte nichtlineare Strecke (ohne vorgesetzter Identifizierung) on-line zu lernen.

Allerdings sind einige wesentliche Randbedingungen zu beachten:

- die Nichtlinearitäten müssen differenzierbar sein
- die unbekannte nichtlineare Strecke darf nicht zu hohe Ordnung haben, um bei den abgeleiteten Signalen noch ein nutzbares Signal-zu Rauschverhältnis zu haben
- es dürfen keine Störsignale auf die unbekannte Strecke wirken
- alle Zustandsvariablen der unbekannten Strecke müssen verfügbar sein
- das Anregesignal während der Lernphase muss der persistent excitation-Bedingung genügen, d.h. es müssen alle Zustände ausreichend angeregt werden. In Abbildung 12.57 erfolgte eine stochastisch verteilte Anregung

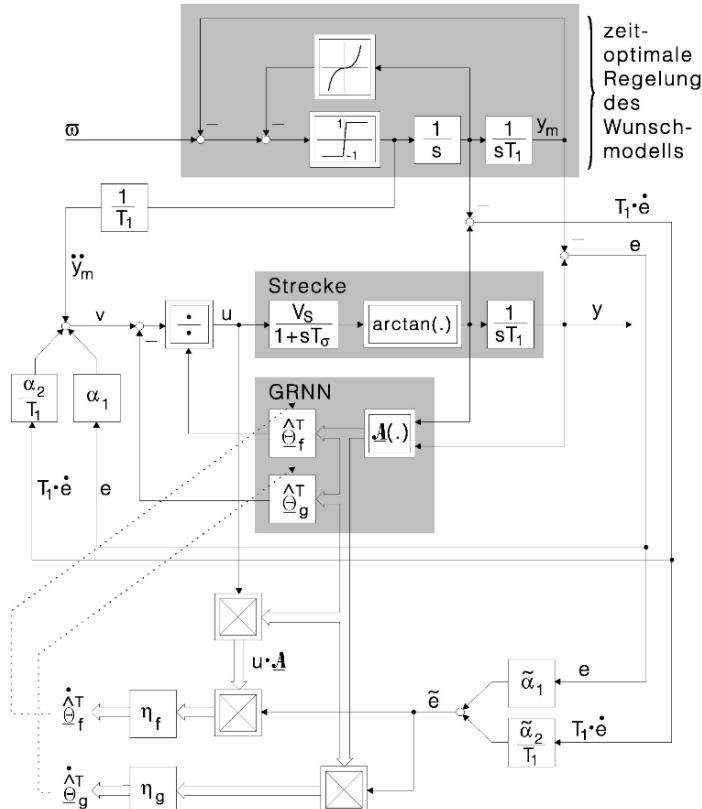
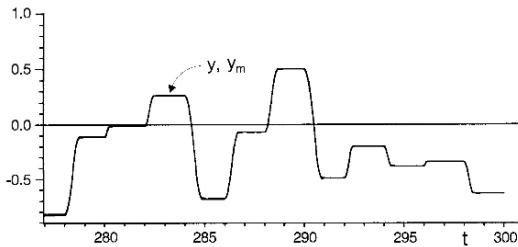


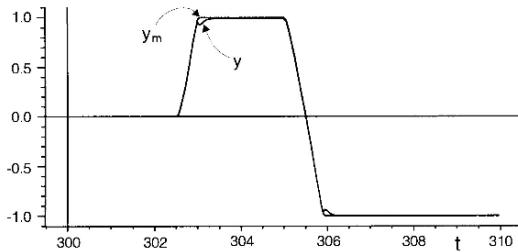
Abb. 12.56: Regelungsstruktur zum Erlernen der Input-Output Linearisierung mit GRNNs

der sprungförmigen Sollwertänderungen. In wie weit derartige Anregungen in dem realen Prozessen zulässig sind, ist abzuklären.

- In Abbildung 12.58 wird dargestellt, wie sich das System in Abbildung 12.56 verhält, wenn der Eingangsraum des Sollwertes den gelernten Bereich verlässt. Aufgrund der besonderen Extrapolationseigenschaft der GRNN's kann das System mit einem dynamischen Fehler folgen. Dieser Fehler wird aber genutzt, um im erweiterten Eingangsraum die Gewichte des GRNN anzupassen.
- Es verbleiben die Lernzeiten beim Gradientenabstieg (siehe auch Kap. 10). Eine weitere Frage in diesem Zusammenhang ist auch, ob ein lokales oder das globale Minimum der Fehlerfläche erreicht wird.



**Abb. 12.57:** Simulationsergebnisse.  $y(t)$  und  $y_m(t)$  sind nicht unterscheidbar. Der Sollwertverlauf  $\varpi$  besteht aus Sprungfunktionen, deren Sprunghöhen stochastisch verteilt sind.



**Abb. 12.58:** Simulationsergebnisse.  $y(t)$  und  $y_m(t)$  stimmen nicht mehr exakt überein. Der Eingangsraum verlässt den gelernten Bereich.

Wäre die nichtlineare Strecke (Abbildung 12.55) bekannt, so könnte man eine exakte Ein-Ausgangs-Linearisierung wie in Kapitel 12.4 durchführen:

$$\underline{f}(\underline{x}) = \begin{bmatrix} \frac{1}{T_1} \cdot x_2 \\ -\frac{1}{T_\sigma} \cdot \frac{\tan(x_2)}{1 + \tan^2(x_2)} \end{bmatrix} \quad \underline{g}(\underline{x}) = \begin{bmatrix} 0 \\ \frac{V_s}{T_\sigma} \cdot \frac{1}{1 + \tan^2(x_2)} \end{bmatrix} \quad (12.289)$$

$$h(\underline{x}) = x_1 \quad (12.290)$$

$$\frac{\partial h(\underline{x})}{\partial \underline{x}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (12.291)$$

$$L_f h(\underline{x}) = \frac{\partial h(\underline{x})}{\partial \underline{x}} \cdot \underline{f}(\underline{x}) = \frac{1}{T_1} \cdot x_2$$

$$L_g h(\underline{x}) = \frac{\partial h(\underline{x})}{\partial \underline{x}} \cdot \underline{g}(\underline{x}) = 0 \quad (12.292)$$

$$\Rightarrow \dot{y} = L_f h(\underline{x}) + \underbrace{L_g h(\underline{x})}_{=0} \cdot u = \frac{1}{T_1} \cdot x_2$$

$$\begin{aligned}
\frac{\partial(L_f h(\underline{x}))}{\partial \underline{x}} &= \begin{bmatrix} 0 & \frac{1}{T_1} \end{bmatrix} \\
L_f^2 h(\underline{x}) &= \frac{\partial(L_f h(\underline{x}))}{\partial \underline{x}} \cdot f(\underline{x}) = -\frac{1}{T_1 \cdot T_\sigma} \cdot \frac{\tan(x_2)}{1 + \tan^2(x_2)} = -\frac{1}{2 \cdot T_1 \cdot T_\sigma} \cdot \sin(2x_2) \\
L_g L_f h(\underline{x}) &= \frac{\partial(L_f h(\underline{x}))}{\partial \underline{x}} \cdot g(\underline{x}) = \frac{V_s}{T_1 \cdot T_\sigma} \cdot \frac{1}{1 + \tan^2(x_2)} = \frac{V_s}{2 \cdot T_1 \cdot T_\sigma} + \frac{V_s}{2 \cdot T_1 \cdot T_\sigma} \cdot \cos(2x_2) \\
\Rightarrow \ddot{y} &= L_f^2 h(\underline{x}) + \underbrace{L_g L_f h(\underline{x}) \cdot u}_{\neq 0} \\
&= -\frac{1}{2 \cdot T_1 \cdot T_\sigma} \cdot ((1 + \cos(2x_2)) \cdot V_s \cdot u - \sin(2x_2))
\end{aligned} \tag{12.293}$$

Die nichtlineare Strecke 2. Ordnung hat wie erwartet einen Relativgrad von  $\delta = n = 2$ . Somit würde sich eine lineare Regelstrecke 2. Ordnung ergeben. Die hier berechneten Lie-Ableitungen zeigen, dass die beiden GRNNs in diesem Beispiel nur sehr einfache Funktionen lernen mussten (das erste GRNN musste lediglich eine Sinusfunktion nachbilden, während das zweite GRNN einen nach oben verschobenen Cosinus lernen musste). Zudem treten in der Simulation die oben beschriebenen Probleme nicht auf.

Im folgenden Abschnitt soll der MRAC-Ansatz zur direkten Regelung nichtlinearer unbekannter Strecken beibehalten werden. Allerdings sollen einige der obigen Einschränkungen, wie beispielsweise die n-fache Differentiation des Steckenausgangssignals, vermieden werden. Mit der Aufhebung derartiger Einschränkungen ist eine praktische Verwendbarkeit wesentlich realistischer.

## 12.8 Stabile referenzmodellbasierte Neuroregelung (SRNR)

Im vorigen Kapitel war ein MRAC-Ansatz dargestellt worden, bei dem die unbekannte nichtlineare Strecke mittels Erlernen der Lie-Ableitungen geregelt wurde, so dass das Ausgangssignal der unbekannten Strecke dem Modellausgang folgt. Allerdings waren bei diesem Vorgehen einige wichtige Voraussetzungen wie die mehrfache Ableitbarkeit des Ausgangssignals der unbekannten Strecke zu beachten. Diese Voraussetzung beschränkt beispielsweise durch die Rauschanteile im Ausgangssignal der unbekannten Strecke die Zahl der Ableitungen. In der Arbeit von Fischle [51] werden einige der Voraussetzungen, die die allgemeine Anwendbarkeit bei realen Systemen einschränken, aufgehoben. Im vorigen Kapitel war angegeben worden, dass mit dem Steuergesetz

$$u = \frac{v - L_f^n h(x)}{L_g L_f^{n-1} h(x)} \tag{12.294}$$

bzw.

$$v = u L_g L_f^{n-1} h(x) + L_f^n h(x) \tag{12.295}$$

die unbekannte nichtlineare Strecke

$$\begin{aligned}
\dot{y} &= L_f h(x) \\
\ddot{y} &= L_f^2 h(x) \\
&\vdots \\
y^{(n)} &= L_f^n h(x) + L_g L_f^{n-1} h(x) \cdot u
\end{aligned} \tag{12.296}$$

mit dem Eingangssignal  $u$  in eine Integratorkette der Ordnung  $n$  (entspricht der Ordnung der unbekannten nichtlinearen Strecke) transformiert werden kann, was der Technik der Ein-Ausgangslinearisierung folgt. Das Eingangssignal der transformierten Strecke ist nun  $v$  und das Ausgangssignal  $y$ , damit ist bei der resultierenden Integratorkette  $v = y^{(n)}$ . Aufgrund der Ordnung  $n$  der unbekannten Strecke und der Ordnung der Integratorkette wird ein geregeltes System ebenfalls der Ordnung  $n$  angesetzt. Das Regelgesetz führt auf ein Signal  $u$

$$u = \frac{y_m^{(n)} - k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)} - L_f^n h(x)}{L_g L_f^{n-1} h(x)} \tag{12.297}$$

mit  $e = y - y_m$ , so dass sich folgende Fehler-Differentialgleichung ergibt:

$$e^{(n)} = -k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)}. \tag{12.298}$$

Wenn die Fehler  $e, \dot{e}, \dots, e^{(n-1)}$  abgeklungen sind, d.h.  $e = \dot{e} = \dots = e^{(n-1)} = 0$ , dann ist  $y \equiv y_m$ . Die Koeffizienten  $k_i$  beschreiben das gewünschte Einschwingverhalten des Fehlers  $e$  und sind frei vorgebar – sie sollten allerdings zu einem stabilen Abklingen führen.

Aus der obigen Gleichung (12.297) zur Berechnung der Stellgröße ist zu erkennen, dass die gemessenen Ausgangssignale  $(n-1)$ -fach ableitbar sein müssen, dies ist eine der oben genannten einschränkenden Voraussetzungen. Die zur Beschreibung des Prozessmodells verwendeten Lie-Ableitungen bestimmen sich aus

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) \tag{12.299}$$

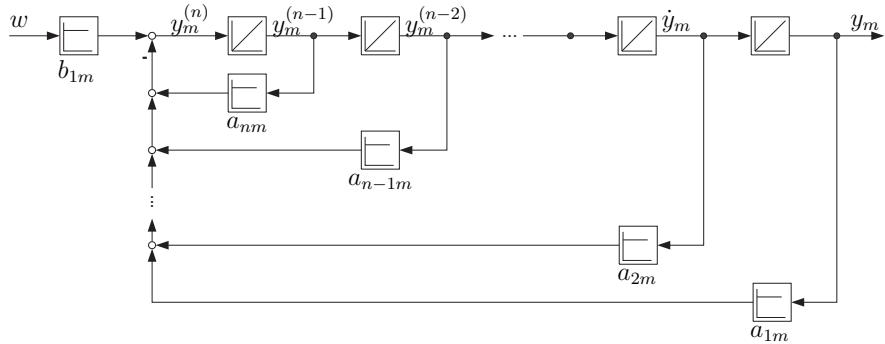
und somit

$$L_f^n h(x) = L_f [L_f^{n-1} h(x)] \tag{12.300}$$

Eine nähere Erläuterung hierzu findet sich in Kapitel 12.4. Die erforderliche  $n$ -fache Ableitbarkeit des Ausgangssignals kann vermieden werden, wenn der Zustandsvektor des Systems als Zusatzwissen verwendet wird, und wenn das gewünschte Regelverhalten (Modellverhalten) linear ist und mittels Nennerpolynom-Ansatz realisiert wird (kein Zählerpolynom). Mit dieser Bedingung ist das Referenzmodell auf folgende Form festgelegt:

$$y_m^{(n)} = -a_{1m} y_m - a_{2m} \dot{y}_m - \dots - a_{nm} y^{(n-1)} + b_{1m} w \tag{12.301}$$

Der Signalflussplan zu diesem Referenzmodell ist in Abb. 12.59 dargestellt.



**Abb. 12.59:** Signalflussplan des Referenzmodells.

Die Parameter  $a_{im}$  mit  $i = 1..n$  können entsprechend den bekannten Optimierungskriterien – beispielsweise dem Dämpfungsoptimum – gewählt werden. Die Koeffizienten im Regelgesetz (12.297) sind daher:

$$k_\nu = a_{\nu m}. \quad (12.302)$$

Mit diesem Ansatz ergibt sich:

$$v = y_m^{(n)} - k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)} \quad (12.303)$$

Gleichung (12.301) in (12.303) eingesetzt:

$$v = b_{1m}w - a_{1m}y_m - a_{2m}\dot{y}_m - \dots - a_{nm}y_m^{(n-1)} - k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)} \quad (12.304)$$

bzw. mit  $k_n = a_{nm}$  und  $e = y - y_m$  etc.:

$$\begin{aligned} v &= b_{1m}w - a_{1m}y_m - a_{2m}\dot{y}_m - \dots - a_{nm}y_m^{(n-1)} - a_{1m}e - a_{2m}\dot{e} - \dots - a_{nm}e^{(n-1)} \\ &= b_{1m}w - a_{1m}y - a_{2m}\dot{y} - \dots - a_{nm}y^{(n-1)} \end{aligned} \quad (12.305)$$

mit  $y = h(x)$ ,  $\dot{y} = L_f h(x)$  etc. ergibt sich:

$$= b_{1m}w - a_{1m}h(x) - a_{2m}L_f h(x) - \dots - a_{nm}L_f^{(n-1)}h(x) \quad (12.306)$$

Damit ergibt sich ein umgeformtes Regelgesetz,

$$u = \frac{b_{1m}w - a_{1m}h(x) - a_{2m}L_f h(x) - \dots - a_{nm}L_f^{(n-1)}h(x) - L_f^n h(x)}{L_g L_f^{(n-1)} h(x)} \quad (12.307)$$

das nur noch von  $w$  und  $x$  abhängt. Die störende Differentiation ist damit durch die Lie-Ableitungen ersetzt worden. Diese werden nicht durch eine Differentiation

gewonnen, sondern von Neuronalen Netzen erlernt. Wie aus den verschiedenen Beispielen in Kapitel 12.4 zu ersehen ist, sind die Lie-Ableitungen Funktionen der Zustandsgrößen des realen Systems. Dies bedeutet, dass die  $n$ -fache Ableitung der Ausgangsgröße  $y$  durch eine Verfügbarkeit aller Zustandsgrößen ersetzt wird. Dies ist eine etwas geringerwertigere Voraussetzung. Es sei allerdings angemerkt, dass dadurch die Messbarkeit der Zustände nicht mehr ganz so unbekannt ist wie vorher angenommen und dass damit auch die vorher diskutierten Identifikationsverfahren für die Teilsysteme genutzt werden könnten.

In den Regelgesetzen (12.297) bzw. (12.307) werden neben dem Regelfehler  $e$  und dessen Ableitungen auch Lie-Ableitungen des unbekannten Prozesses verwendet. Für die Implementierung der Regelgesetze müssen daher die Lie-Ableitungen durch geeignete Schätzwerte ersetzt werden. Hierfür werden universelle nichtlineare Funktionsapproximatoren verwendet, welche durch ein Lernverfahren iterativ so eingestellt werden, dass der geschätzte Wert dem tatsächlichen Wert der Lie-Ableitungen möglichst nahe kommt. Als Funktionsapproximatoren eignen sich in diesem Zusammenhang RBF-Netze in besonderer Weise, da deren Gewichte linear in das Ausgangssignal eingehen. Dies ist von besonderer Wichtigkeit, um mit Hilfe von Lyapunovfunktionen ein stabiles Adaptionsgesetz zu entwickeln. Die im Regelgesetz (12.297) benötigten Lie-Ableitungen werden damit ersetzt durch:

$$L_f^n h(x) = \theta_I^T \xi_I(x) \quad (12.308)$$

$$L_g L_f^{n-1} h(x) = \theta_{II}^T \xi_{II}(x) \quad (12.309)$$

Damit ergibt sich die Grundform des SRNR-Basisverfahrens zu:

$$u = \frac{y_m^{(n)} - k_1 e - k_2 \dot{e} - \dots - k_n e^{(n-1)} - \theta_I^T \xi_I(x)}{\theta_{II}^T \xi_{II}(x)} \quad (12.310)$$

Die Adaption der Netze erfolgt über die Verstellung der Stützstellengewichte  $\theta_I^T$  und  $\theta_{II}^T$ . Dazu wird ein Fehlermaß  $\tilde{e}$  benötigt, das aus einer Filterung des Fehlersignals  $e$  resultiert. Dadurch wird gewissermaßen eine Vorhersage des zukünftigen Fehlers erreicht. Damit haben  $\phi^T \xi$  und  $\tilde{e}$  im Mittel das gleiche Vorzeichen, was sich mathematisch ausgedrückt in einer streng positiv reellen Übertragungsfunktion niederschlägt. Der zugehörige lineare Filter wird durch die Übertragungsfunktion

$$G_F(s) = \frac{\tilde{e}}{e} = p_n s^{n-1} + \dots + p_2 s + p_1 \quad (12.311)$$

beschrieben, weswegen sich das Fehlermaß im Zeitbereich aus

$$\tilde{e} = p_n e^{(n-1)} + \dots + p_2 \dot{e} + p_1 e \quad (12.312)$$

zusammensetzt. Daraus ergibt sich eine gesamte Übertragungsfunktion zwischen den Signalen  $\phi \xi$  und  $\tilde{e}$ :

$$G_{\tilde{e}}(s) = \frac{\tilde{e}}{\phi \xi} = \frac{p_n s^{n-1} + \dots + p_2 s + p_1}{s^n + k_n s^{n-1} + \dots + k_2 s + k_1} \quad (12.313)$$

Diese Übertragungsfunktion ist physikalisch zwar realisierbar. Da jedoch deren Eingangssignal  $\phi\xi$  nicht bekannt ist, muss das Fehlermaß  $\tilde{e}$  aus dem messbaren Fehler  $e$  gebildet werden. Dazu ist jedoch der nicht realisierbare Kompensator aus Gleichung (12.311) nötig. Genau hierin liegt das Problem bei der praktischen Umsetzung dieses Konzeptes, welches durch die Verwendung des messbaren Zustandsvektors umgangen wird. Die Netze werden gemäß

$$\dot{\theta}_I = \gamma\tilde{e}\xi_I \quad (12.314)$$

$$\dot{\theta}_{II} = \gamma\tilde{e}S_{II}\xi_{II}u \quad (12.315)$$

adaptiert. Die positive Adaptionsverstärkung  $\gamma > 0$  dient als frei wählbare Lernschrittweite, die Diagonalmatrix  $S_{II}$  wird zur Abschaltung der Adaption für einzelne Parameter benötigt, wenn diese eine bestimmte Untergrenze unterschreiten. Dies ist erforderlich, um eine Division durch Null in Gl. (12.310) zu vermeiden. Unter der Voraussetzung, dass die gewählten Funktionsapproximatoren die nachzubildenden Lie-Ableitungen exakt annähern können, d.h. dass es einen idealen Parametervektor  $\theta^*$  gibt, so dass

$$\theta_I^*\xi_I(x) = L_f^n h(x) \quad (12.316)$$

$$\theta_{II}^*\xi_{II}(x) = L_g L_f^{n-1} h(x) \quad (12.317)$$

gilt, ist der folgende Satz gültig, welcher die Stabilität des Gesamtsystems garantiert:

**Satz:** Gegeben sei das Differentialgleichungssystem aus der Fehlerdifferentialgleichung (12.298) und dem Adaptionsgesetz (12.314). Wenn die Übertragungsfunktion (12.313) streng positiv reell ist, dann sind für alle Lösungen  $(\phi(t), e(t))$  dieses Differentialgleichungssystems  $\phi(t)$ ,  $e(t)$  sowie  $\dot{e}(t)$ ,  $\ddot{e}(t)$ ,  $\dots$ ,  $e^{(n-1)}(t)$  beschränkt. Ist außerdem  $\xi(t)$  beschränkt, so ist  $\lim_{t \rightarrow \infty} e(t) = 0$ .

### 12.8.1 Parameteradaption

Genau wie im Regelgesetz erfordert die erfolgreiche Adaption der Parameter die Ableitungen des Fehlers. Daher muss auch hier das Differenzieren eliminiert werden, was durch das Konzept des "augmented errors" (bzw. des "vermehrten Fehlers") geschieht. Dabei wird das Fehlersignal  $e$  um einen additiven Anteil  $e_{aux}$  erweitert, der auch als Hilfsfehler bezeichnet wird. Somit entsteht ein erweitertes Fehlersignal,

$$e^* = e + e_{aux} \quad (12.318)$$

wobei der Hilfsfehler

$$e_{aux} = \theta_I^T \xi_{If} - \theta_{II}^T (\xi_{II} u)_f - (\theta_I^T \xi_I - \theta_{II}^T \xi_{II} u)_f \quad (12.319)$$

ist. Der Index  $f$  steht hierbei für eine Filterung des Signales mit dem Referenzmodell. Für die Filterung eines beliebigen Zeitsignales  $z$  gilt daher beispielsweise:

$$z_f^{(n)} = -a_{1m}z_f - a_{2m}\dot{z}_f - \cdots - a_{nm}z_f^{(n-1)} + z \quad (12.320)$$

Die Adaption wird nun gemäß

$$\dot{\theta}_I = -\gamma_I e^* \xi_{If} \quad (12.321)$$

$$\dot{\theta}_{II} = \gamma_{II} e^* S_{II}(\xi_{II} u)_f \quad (12.322)$$

mit den gefilterten Signalen und dem vermehrten Fehler vorgenommen. Die Grundidee besteht darin, dass durch die Verwendung der gefilterten (und daher natürlich verzögerten) Signale  $\xi_{If}$  und  $(\xi_{II} u)_f$  der aktuelle Ausgangsfehler  $e$  gewissermaßen zur Adaption der vergangenen Parameterfehler  $\phi$  verwendet wird, so dass der dynamische Zusammenhang zwischen  $\phi$  und  $e$  damit berücksichtigt wird.

### 12.8.2 Stellgrößen-Beschränkung

In den bis jetzt beschriebenen Verfahren der exakten Eingangs-Ausgangslinearisierung und der Weiterentwicklung ist die Stellgrößenbeschränkung eine weitere Schwierigkeit. Grundsätzlich ist die Stellgrößenbeschränkung eine unstetige Nichtlinearität, die im strengen Sinne der Einschränkungen der Eingangs-Ausgangslinearisierung nicht zulässig ist. In vielen Anwendungen ist aber das Stellsignal  $u$  der Strecke beschränkt und es ist deshalb erwünscht, auch diesen Fall mit einzuschließen. Ein naheliegender Ansatz bei Stellgrößenbeschränkungen war, den integralen Anteil des Reglers anzuhalten, wenn die Stellgrößenbeschränkung wirksam ist. Dieser Ansatz ist logisch, denn bei Stellgrößenbeschränkung ist der Regelkreis geöffnet und der Integralanteil des Reglers kann die Regelabweichung nicht verringern. Im übertragenen Sinne kann diese Überlegung auch bei der Parameteradaption genutzt werden. Es ist einsichtig, dass eine unbekannte und wirksame Stellgrößenbeschränkung zu Adoptionsfehlern führen muss, da der Fehler  $e$  nicht mehr alleine durch den Parameterfehler bedingt ist. Die Übertragung der obigen Überlegung bedeutet somit, dass die Adaption der Parameter angehalten wird, wenn die Stellgrößenbeschränkung wirksam ist. Eine genauere Analyse zeigt aber, dass das Anhalten der Parameteradaption alleine nicht ausreicht. Wenn beispielsweise das reale System die Begrenzung verlässt, und die Adaption wieder aktiv wird, dann sind die Zustandsgrößen des Modells – welches keine Stellgrößenbeschränkung aufweist – unterschiedlich zu den Zustandsgrößen der realen Strecke. Dies gilt auch, wenn das Modell ebenso eine Stellgrößenbeschränkung hat und aufgrund der unterschiedlichen Parameter die Stellgrößenbeschränkungen unterschiedlich ansprechen. Damit ergibt sich ein Zusatzterm im Fehlersignal, welcher durch das Ansprechen der Stellgrößenbeschränkung hervorgerufen ist. Dieser Zusatzterm führt zu einer Fehladaption der Parameter bei der

Systemidentifikation und somit auch beim Regler. Um diesen Effekt zu vermeiden, müssen die Zustandsgrößen des Modells auf die bekannten Zustandsgrößen der realen Strecke gesetzt werden, wenn die Begrenzung verlassen wird; dies ist unproblematisch bei abgetasteten Systemen zu realisieren.

Eine weitere Fehlermöglichkeit ergibt sich bei hohen Adoptionsverstärkungen. In diesem Fall kann es vorkommen, dass durch die große Parameterverstellung das Stellsignal die Begrenzung überschreitet und diese wirksam wird. Im Grenzfall kann dadurch die Stellgröße dauerhaft begrenzt bleiben, die Adaption der Parameter ist somit abgeschaltet. Um diesem Sonderfall zu beherrschen, muss sowohl abgefragt werden, ob die Begrenzung wirksam ist und ein  $e_{Grenz}$  überschritten ist, als auch ob  $u \cdot \dot{\theta}_i < 0$  ist. Sind beide Bedingungen erfüllt, dann kann die Adaption weiter wirksam sein. Der Fehler  $e_{Grenz}$  sollte im Bereich  $(0, 1 - 1)\hat{y}_m(t)$  sein.

### 12.8.3 Aufteilung in Teilfunktionen

Im Allgemeinen ist die nichtlineare Strecke nicht völlig unbekannt, d.h. es bestehen Teilbereiche, die bekannt sind. In diesem Falle sind die Zusammenhänge zwischen dem Ein- und Ausgangssignal des bekannten Teilbereiches bekannt. In diesem Fall können somit die Lie-Ableitungen für diesen Teilbereich berechnet werden. Der wesentliche Vorteil der Berücksichtigung bekannter Teilbereiche ist, dass damit die Anzahl der zu erlernenden Parameter deutlich verringert werden kann, eine deutliche Verkürzung der Lernzeiten ist die Folge. Eine weitere Verringerung der Adoptionszeiten ist zu erreichen, wenn die Strecke symmetrisch ist, d.h. bei positiven und negativen Signalen symmetrisch zum Nullpunkt der Zustandsgrößen ist.

Mit den obigen Maßnahmen ist die Ein-Ausgangs-Linearisierung etwas mehr in den Bereich der praktischen Realisierbarkeit gerückt worden. Allerdings verbleibt weiterhin eine wesentliche Einschränkung, die Strecke darf nicht gestört sein.

# 13 Modellbasierte Adaptive Regelung

## Christian Westermaier

Der Inhalt dieses Kapitels ist Teil der Dissertation [240] und soll den Leser auf anschauliche Weise in das Gebiet der modellbasierten adaptiven Regelung einführen.

In der klassischen Regelungstechnik werden die Themen Regelung sowie Identifikation im Allgemeinen getrennt voneinander betrachtet. Zum einen wurden ausgereifte Regelungsmethoden entwickelt, die ein in einer mathematischen Form vorliegendes System erfolgreich regeln und zum anderen benötigt man Identifikations-Methoden, die eine reale Anlage auf ein mathematisches Modell abbilden, damit die Voraussetzungen für den Regelungsentwurf gegeben sind. Nur wenn das Modell die dominanten Eigenschaften der realen Anlage widerspiegelt, d.h. die Anlage richtig identifiziert wurde, garantiert die entworfene Regelung die Stabilität des geschlossenen Regelkreises sowie eine den Spezifikationen entsprechende stationäre Genauigkeit bzw. gutes Folgeverhalten. Man spricht von zwei getrennten Themengebieten, die im Grunde eng miteinander verkoppelt sind: wird eine ungeeignete Identifikationsmethode verwendet, bedarf es eines sehr guten robusten Reglers, für den Parameterschwankungen kaum Auswirkungen auf das Regelergebnis besitzen; wird ein einfacher Regler verwendet, so müssen die identifizierten Parameter das zu regelnde System eindeutig repräsentieren, um eine stabile Regelung des realen Systems gewähren zu können.

Im klassischen Sinne sieht nun eine Reglerauslegung für ein reales System wie folgt aus: Zunächst ist es für eine Identifikation notwendig, die Struktur des Systems zu kennen, um zu wissen, welche Systemparameter identifiziert werden müssen, so dass das dominante Verhalten des Systems widergespiegelt wird. Für den Identifikations-Vorgang wird ein geeignetes Eingangssignal (vgl. Abb. 8.24) mit sog. beständiger Anregung auf die Anlage geschaltet, um durch Anregen aller Zustände und möglichst vieler Eigenfrequenzen ausreichend Systeminformation zu sammeln. Dadurch wird das System über eine längere Identifikationszeit sehr belastet. Es ist ungewiss, wie lange diese Zeit sein muss, ob das System mit einer ausreichenden Zahl an Frequenzen angeregt wurde und ob folglich nach Beendigung der Identifikation die richtigen Parameter überhaupt vorliegen. Zudem verhindern stochastische Störungen sowie nicht modellierte deterministische Störungen eine Konvergenz des Identifikationsalgorithms, womit eine eindeutige Identifikation bei präsenter Störung nicht möglich ist. Legt man nun den Regler

mit diesen unsicheren Parametern aus, so stellt sich die Frage, ob der Gesamtregelkreis stets Stabilität zeigen wird; dies kann nicht garantiert werden. Zudem kann der klassische fest eingestellte Regler nicht auf spätere Parameteränderungen bzw. Schwankungen reagieren, was ebenfalls die Stabilität gefährden kann. Weiter muss nach jeder Modifikation des Systems erneut ein Identifikationslauf durchgeführt werden. Man sieht, mit einer seriellen Abfolge von Identifikation und Regelung sind durchaus Risiken und Nachteile verbunden, vor allem, wenn die Komplexität der realen Systeme und somit Modelle zunimmt.

Die modellbasierte adaptive Regelung ([10], [68], [148], [158]) ermöglicht es, Regelung und Identifikation stabil zu vereinen, um die oben beschriebenen Probleme zu umgehen. Hierbei finden Regelung und Identifikation nicht mehr seriell, sondern parallel statt, d.h. es wird im geschlossenen Regelkreis identifiziert bzw. es wird bereits während der Identifikation geregelt. Da von Beginn an versucht wird, das Regelziel zu erreichen, ist verständlich, dass nicht ausreichend beständige Anregung für die Identifikation zur Verfügung steht, d.h. es ist im Allgemeinen zu erwarten, dass die Parameter nicht gegen die wahren Systemparameter konvergieren werden. Für eine erfolgreiche Regelung ist es aber auch nicht notwendig, in einem bestimmten Betriebspunkt bei einem bestimmten Sollsignal, Wissen über das gesamte Systemverhalten zu besitzen. Der Grundgedanke des adaptiven Konzeptes besteht vielmehr darin, nur so viel Systeminformation zu sammeln bzw. zu identifizieren, wie momentan für das Erreichen der Solltrajektorie bzw. des Regelziels notwendig ist.

In diesem Kapitel wird exemplarisch obiges Prinzip an Hand eines diskreten adaptiven Referenzmodell-Reglers nach [68] vorgestellt. Dieser ist unter dem Begriff MRAC (Model Reference Adaptive Control) bekannt. Für die Erfüllung des beschriebenen Grundgedankens stützt sich das adaptive Regelungskonzept MRAC mit seiner enthaltenen Identifikation auf das einfachste Modell einer Anlage, die Übertragungsfunktion mit variablen Parametern. Nachdem eine Übertragungsfunktion nur das Ein- Ausgangsverhalten eines Systems beschreibt, muss folglich keine Kenntnis über die Struktur des Systems vorliegen; lediglich die Ordnung  $n$  sowie Relativgrad  $\delta$  des Systems müssen bekannt sein bzw.  $n$  darf nicht kleiner und  $\delta$  nicht größer angenommen werden als sie tatsächlich sind. Zur Identifikation der Parameter einer Übertragungsfunktion eignet sich u.a. der einfache Projektionsalgorithmus bzw. der Rauschsignal-optimierte RLS-Algorithmus, der bereits in Kapitel 4.2.2 vorgestellt wurde. Abhängig vom Betriebspunkt bzw. Sollsignal und dem bis dato angeeigneten Vorwissen muss dieser nun zu jedem Zeitpunkt den Parametersatz erneuern, so dass stets das notwendige Wissen für die momentane Regelung zur Verfügung steht. Die Anpassung der Parameter bezeichnet man als Adaption.

Wie in diesem Kapitel gezeigt wird, beruht das Konzept auf der Tatsache, dass für das Erreichen eines Regelziels nicht das vollständige Systemwissen notwendig ist, womit eine gleichzeitige Regelung und Identifikation möglich wird. Mit der Anregung des Systems durch das Sollsignal, welches für die Identifikation keine beständige Anregung gemäß Abb. 8.24 garantiert, kann dieses zwar

nicht vollständig identifiziert werden, es ist jedoch für das Erreichen des momentanen Sollsignals das Wissen über das angeregte Systemverhalten ausreichend - effektiv muss nur so viel Wissen über das System gelernt werden, wie für das Erreichen des momentanen Regelziels bezogen auf das Sollsignal notwendig ist. Sobald sich das Sollsignal ändert, kann neues Systemwissen identifiziert werden, mit dem das neue Regelziel zu erreichen ist. Mit jeder Änderung der Systemanregung in Frequenz und Amplitude auf noch nicht genutzte Signale nähert man sich dem Verhalten einer beständigen Anregung, so dass letztendlich ein vollständig identifiziertes System resultieren kann. Im Unterschied zum konventionellen Vorgehen kann jedoch bereits erfolgreich geregelt werden, auch wenn noch nicht das vollständige Systemwissen vorliegt. Mit Hilfe eines später aufgeführten Widerspruchsbeweises in Kapitel 13.3.2 lässt sich zeigen, dass trotz eines nicht vollständig identifizierten Systems stets ein stabiler Regelkreis vorliegt.

Zusammengefasst bedeutet das, dass ein unbekanntes System ohne vorhergehende Untersuchungen stabil geregelt werden kann, wenn ausreichend viele Modellparameter für die Identifikation vorliegen und somit das System in seiner Ordnung und seinem Relativgrad nachgebildet werden kann. Dabei ist es nicht notwendig, die richtigen Parameter zu finden, um ein gutes Regelergebnis zu erzielen. Dementsprechend stellen zeitlich begrenzte Parameterschwankungen sowie ständige nicht zu dynamische Parameterschwankungen kein Problem mehr dar.

Die Einfachheit des Regelkonzeptes erkauft man sich durch die Zeitvarianz der Parameter – obwohl von einer linearen Steckenstruktur ausgegangen wird, handelt es sich bei der Verknüpfung mit dem zeitvarianten Regler um ein hochgradig nichtlineares Gesamtsystem.

Der wesentliche Punkt in der Akzeptanz des adaptiven Konzeptes besteht zum einen darin, zu zeigen, dass das Regelziel trotz eines nicht korrekt geschätzten Parametersatzes erreicht wird und zum anderen darin, dass trotz falscher Parameter die Stabilität des Gesamtregelkreises bewiesen werden kann. Nachdem ein zeitvariantes Gesamtsystem vorliegt, führen LTI-Methoden hinsichtlich der Stabilitätsanalyse zu keiner Aussage. Für das MRAC-Konzept findet daher im Folgenden ein Widerspruchsbeweis Anwendung, mit dem zeitvariante / nicht-lineare Systeme auf Stabilität untersucht werden können.

Es wird im Folgenden das theoretische Grundkonzept der adaptiven Regelung (MRAC) vorgestellt und basierend auf dieser Theorie ein entsprechender stabiler Regler für das in Abschnitt 2.1 vorgestellte Zwei-Massen-System, die Grundkomponente eines jeden mechatronischen Systems, entwickelt.

Sowohl im Zeitkontinuierlichen [158] als auch im Zeitdiskreten [68] ist das Prinzip von MRAC umsetzbar. Da mechatronische Systeme heutzutage generell über digitale Rechner gesteuert bzw. geregelt werden, ist es aus Stabilitätsgründen sinnvoll, direkt einen diskreten Reglerentwurf durchzuführen. Dementsprechend wird hier das diskrete MRAC-Prinzip erläutert.

### 13.1 ARMA-Modell als Prädiktionsmodell

Da es sich bei dem adaptiven Referenzmodell-Regler um einen modellgestützten Regler handelt, bedarf es eines geeigneten Modells der Strecke, dessen Parameter an Hand von Systemsignalen parallel zum Regelvorgang identifiziert werden können. Nimmt man im ungünstigsten Fall an, dass weder die Systemstruktur bekannt ist noch die Systemzustände  $\underline{x}[k]$  messbar sind, kann die ein System vollständig beschreibende Zustandsdarstellung nicht als Modell angewandt werden. Liegen lediglich die Messwerte des Systemausgangs  $y[k]$  vor, ist das ARMA-Modell nach [68]

$$A[q^{-1}] y[k] = B[q^{-1}] u[k] \quad (13.1)$$

zu verwenden, welches im Zeitbereich einen direkten Zusammenhang zwischen Eingangssignal  $u[\cdot]$  und Ausgangssignal  $y[\cdot]$  bietet, d.h. das Ein- Ausgangsverhalten eines Systems eindeutig beschreibt ( $q^{-1}$  entspricht dem Schiebeoperator im Zeitbereich). Mit Hilfe des Ein- und Ausgangssignals wird durch das ARMA-Modell eine Identifikation des Systemverhaltens und dadurch eine adaptive Regelung ermöglicht.

Das ARMA-Modell nach [68] lautet im Frequenzbereich:

$$y(z) = \frac{B(z)}{A(z)} \cdot u(z) \quad (13.2)$$

Wird durch das Modell neben dem Eingangssignal  $u[\cdot]$  auch das Störsignal  $v[\cdot]$  berücksichtigt, ergibt sich das sog. ARMAX-Modell nach [68]:

$$y(z) = \frac{B(z)}{A(z)} \cdot u(z) + \frac{C(z)}{A(z)} \cdot v(z) \quad (13.3)$$

Die Gleichung (13.3) entspricht exakt der des ARMAX-Modells nach [158] in Tabelle 7.1. Jedoch zeigt sich in der Definition der Eingangssignale der Modelle zwischen [68] und [158] ein Unterschied, welcher in einem formell unterschiedlichen ARMA-Modell resultiert. Im Unterschied zu Gleichung (13.2) lautet das ARMA-Modell nach [158]:

$$y(z) = \frac{B(z)}{A(z)} \cdot v(z) \quad (13.4)$$

Je nach Anwendungsgebiet liegt bzgl. der Definition des ARMA-Modells (ARMA: Auto Regressive Moving Average) ein deterministisches Eingangssignal  $u[\cdot]$  oder ein stochastisches Störsignal  $v[\cdot]$  vor, welches den „Moving Average“-Anteil bildet. Wirken beide Signale auf das System, so bezeichnet man eines der beiden Signale als einen zusätzlichen Eingang, d.h. „Exogenous Input“ bzw. „Auxiliary Input“, womit das sog. ARMAX-Modell resultiert. Nach [158] wird in Tabelle 7.1 das stochastische Störsignal  $v[\cdot]$  als Hauptsignal und das deterministische Signal

$u[\cdot]$  als Zusatzsignal verwendet. Dies ist sinnvoll, wenn Identifikationsprozesse betrachtet werden, die für die Systemanregung stochastische Signale verwenden (vgl. Abbildung 8.24). Für Regelungsaufgaben hingegen greift man auf das deterministische Signal  $u[\cdot]$  als Hauptsignal zurück. Dementsprechend wird für den MRAC-Reglerentwurf das ARMA-Modell (13.2) bzw. (13.1) nach [68] mit dem Eingangssignal  $u[\cdot]$  und dem Ausgangssignal  $y[\cdot]$  angewandt.

Das ARMA-Modell (13.1) kann direkt im Zeitbereich an Hand der allgemeinen Zustandsbeschreibung eines Systems abgeleitet werden, was beweist, dass das ARMA-Modell eindeutig das Ein- Ausgangsverhalten eines Systems repräsentiert. Die Zustandsdarstellung beschreibt sämtliche die Dynamik des Systems bestimmende Zustände und spiegelt damit das vollständige Systemverhalten mit einer eindeutigen Systemstruktur wider, so dass bei gegebenen Eingangssignal und bekannten Anfangszuständen auch das Ausgangssignal zu berechnen ist. Beziiglich der adaptiven Regelung nimmt man jedoch den ungünstigsten Fall an, bei dem die Systemzustände nicht gemessen bzw. ausgewertet werden können, um die Identifikation des Zustandsmodells zu ermöglichen. Für eine modellgestützte Regelung ist jedoch Wissen über die Systemstruktur nicht notwendig, sondern die Modellierung des Ein- Ausgangsverhaltens ausreichend. Durch Ersetzen der Zustände in der Zustandsbeschreibung durch eine Linearkombination von vergangenen Ein- und Ausgangssignalen ergibt sich aus der Zustandsdarstellung ein Modell, welches das Ein- Ausgangsverhalten eindeutig beschreibt, die Zustände dafür aber nicht explizit verwendet und somit kein Wissen über die Systemstruktur benötigt. Dieses sog. ARMA-Modell wird im Folgenden abgeleitet:

Ausgehend von der diskreten Zustandsbeschreibung eines SISO-Systems n-ter Ordnung

$$\underline{x}[k+1] = \mathbf{A} \underline{x}[k] + \underline{b} u[k] \quad (13.5)$$

$$y[k] = \underline{c}^T \underline{x}[k] \quad (13.6)$$

mit  $k \in \mathbb{N}$ ,  $\{u, y\} \in \mathbb{R}$ ,  $\{\underline{x}, \underline{b}, \underline{c}\} \in \mathbb{R}^n$  und  $\mathbf{A} \in \mathbb{R}^{n \times n}$  bedarf es der Elimination der  $n$  Zustände. Wendet man die Differenzengleichung (13.5) rekursiv an, so ergibt sich für wachsende  $k$  folgende Sequenz,

$$\begin{aligned} \underline{x}[k] &\equiv \underline{x}_0[k] \\ \underline{x}[k+1] &= \mathbf{A} \underline{x}[k] + \underline{b} u[k] = \mathbf{A} \underline{x}_0 + \underline{b} u[k] \\ \underline{x}[k+2] &= \mathbf{A} \underline{x}[k+1] + \underline{b} u[k+1] = \mathbf{A}^2 \underline{x}_0 + \mathbf{A} \underline{b} u[k] + \underline{b} u[k+1] \\ &\vdots \end{aligned}$$

woraus sich eine allgemeine Lösungsformel der diskreten Zustandsbeschreibung bestimmen lässt, die ausgehend vom Zustandsvektor  $\underline{x}_0[k]$  den in  $m$  Zeitschritten folgenden Zustandsvektor  $\underline{x}[k+m]$

$$\underline{x}[k+m] = \mathbf{A}^m \underline{x}_0[k] + \sum_{i=0}^{m-1} \mathbf{A}^{m-1-i} \underline{b} u[k+i] \quad (13.7)$$

bzw. den gemäß Gleichung (13.6) zugehörigen Ausgangswert  $y[k+m]$  berechnet:

$$\begin{aligned} y[k+m] &= \\ &= \underline{c}^T \mathbf{A}^m \underline{x}_0[k] + \sum_{i=0}^{m-1} \underline{c}^T \mathbf{A}^{m-1-i} \underline{b} u[k+i] \quad (13.8) \\ &= \underline{c}^T \mathbf{A}^m \underline{x}_0[k] + \underbrace{\left( \begin{array}{cccc} \underline{c}^T \underline{b} & \cdots & \underline{c}^T \mathbf{A}^{m-2} \underline{b} & \underline{c}^T \mathbf{A}^{m-1} \underline{b} \end{array} \right)}_{\underline{q}_U^T(m)} \underbrace{\begin{pmatrix} u[k+m-1] \\ \vdots \\ u[k+1] \\ u[k] \end{pmatrix}}_{\underline{U}_S[k+m-1]} \end{aligned}$$

Ist der Anfangszustand  $\underline{x}_0[k]$  bekannt, so kann ohne Kenntnis weiterer Zustandssignale an Hand der Stellgrößenabfolge  $u[\cdot]$  jeder zukünftige Wert  $\hat{y}[k+m]$  des Ausgangssignals mit Hilfe des Modells (13.8) prädiziert werden. Würde nach einer bestimmten Zeit  $i$  ein aktuellerer Anfangszustand  $\underline{x}_0[k+i]$  vorliegen, könnte von diesem aus prädiziert werden, was die Ungenauigkeiten, bedingt durch Untermodellierungen, und wachsenden Rechenaufwand bei zunehmender Zeit verkleinert. Nachdem jedoch keine Zustandssignale vorliegen, ist die Frage zu beantworten, ob an Hand vergangener Ein- und Ausgangssignale ein aktuellerer Anfangszustandsvektor  $\underline{x}_0[k+i]$  zu bestimmen ist, der als neue Referenz für Gleichung (13.8) dienen könnte. Zunächst ist zu klären, wie viele vergangene Ein- und Ausgangssignale bekannt sein müssen, um auf einen Zustandsvektor schließen zu können. Hierzu ist im Umkehrschluss zu überlegen, wie viele Schritte  $m$  im Gleichungssystem (13.7) nötig sind, damit, ausgehend vom Zustandsvektor  $\underline{x}_0[k]$ , ein beliebiger Wert des Zustandsvektors über eine Steuersequenz  $u[k], \dots, u[k+m-1]$  erreicht werden kann, d.h. wie viele Gleichungen aus (13.8) sind für die Bildung des gesuchten Modells notwendig, damit dieses das Kriterium der Steuerbarkeit erfüllt. Wird beispielsweise der Nullzustand  $\underline{x}[k+m] = \underline{0}$  in Gleichung (13.7) gefordert, so erhält man

$$-\mathbf{A}^m \underline{x}_0[k] = \underbrace{\left( \begin{array}{cccc} \underline{b} & \mathbf{A}\underline{b} & \cdots & \mathbf{A}^{m-2}\underline{b} & \mathbf{A}^{m-1}\underline{b} \end{array} \right)}_{\mathbf{Q}_S} \underbrace{\begin{pmatrix} u[k+m-1] \\ u[k+m-2] \\ \vdots \\ u[k+1] \\ u[k] \end{pmatrix}}_{\underline{U}_S[k+m-1]} \quad (13.9)$$

Um die Steuersequenz  $\underline{U}_S[k+m-1]$  bestimmen zu können, muss zum einen die Steuerbarkeitsmatrix  $\mathbf{Q}_S$  invertierbar sein, d.h. das zeitdiskrete Modell (13.5) muss steuerbar sein. Zum anderen ist das Gleichungssystem (13.9) mit  $\mathbf{A} \in \mathbb{R}^{n \times n}$  nur für  $m = n$  eindeutig lösbar, wobei  $n$  die Ordnung des Systemmodells ist. Dementsprechend sind mindestens  $n$  Gleichungen zur Bestimmung eines Zu-

standsvektors aus vergangenen Ein- und Ausgangssignalen und somit zur Bestimmung eines Prädiktions-Modells nötig, was die Tatsache unterstreicht, dass die Systemdynamik gemäß der Zustandsdarstellung (13.5) durch genau  $n$  Zustände eindeutig beschrieben wird.

Mit diesen Überlegungen wurde deutlich, dass zum Aufstellen eines lösbarer Gleichungssystems hinsichtlich der Bestimmung eines Zustandsvektors  $\underline{x}_0[k]$  der Ordnung  $n$  die Gleichung (13.8) zu  $n$  unterschiedlichen Zeitpunkten herangezogen werden muss; mit  $m \in \{0, \dots, n-1\}$  resultiert eine Sequenz von  $n$  Ausgangswerten:

$$\begin{aligned} & \underbrace{\begin{pmatrix} y[k+n-1] \\ \vdots \\ y[k+2] \\ y[k+1] \\ y[k] \end{pmatrix}}_{\underline{Y}[k+n-1] \in \mathbb{R}^n} = \\ &= \begin{pmatrix} \underline{c}^T \mathbf{A}^{n-1} \\ \vdots \\ \underline{c}^T \mathbf{A}^2 \\ \underline{c}^T \mathbf{A} \\ \underline{c}^T \end{pmatrix} \underline{x}_0[k] + \begin{pmatrix} \underline{c}^T \mathbf{A}^{n-2} \underline{b} \cdot u[k] + \dots + \underline{c}^T \underline{b} \cdot u[k+n-2] \\ \vdots \\ c^T \mathbf{A} \underline{b} \cdot u[k] + c^T b \cdot u[k+1] \\ \underline{c}^T \underline{b} \cdot u[k] \\ 0 \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} \underline{c}^T \mathbf{A}^{n-1} \\ \vdots \\ \underline{c}^T \mathbf{A}^2 \\ \underline{c}^T \mathbf{A} \\ \underline{c}^T \end{pmatrix}}_{\mathbf{Q}_B(n) \in \mathbb{R}^{n \times n}} \underline{x}_0[k] + \underbrace{\begin{bmatrix} \underline{c}^T \underline{b} & \dots & \underline{c}^T \mathbf{A}^{n-3} \underline{b} & \underline{c}^T \mathbf{A}^{n-2} \underline{b} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \underline{c}^T \underline{b} & \underline{c}^T \mathbf{A} \underline{b} \\ 0 & 0 & 0 & \underline{c}^T \underline{b} \\ 0 & \dots & 0 & 0 \end{bmatrix}}_{\mathbf{Q}_U(n) \in \mathbb{R}^{n \times (n-1)}} \underbrace{\begin{pmatrix} u[k+n-2] \\ u[k+1] \\ \vdots \\ u[k] \end{pmatrix}}_{\underline{U}[k+n-2] \in \mathbb{R}^{(n-1)}} \end{aligned} \quad (13.10)$$

Unter der Bedingung, dass das Modell der realen Anlage beobachtbar und somit die Beobachtbarkeitsmatrix  $\mathbf{Q}_B$  invertierbar ist, kann nach den Zuständen  $\underline{x}_0[k]$  aufgelöst werden; hierbei impliziert Beobachtbarkeit, dass alle Zustände des Modells eine Wirkung auf den Ausgang  $y[\cdot]$  aufweisen:

$$\underline{x}_0[k] = \mathbf{Q}_B^{-1}(n) \begin{pmatrix} y[k+n-1] \\ \vdots \\ y[k] \end{pmatrix} - \mathbf{Q}_B^{-1}(n) \mathbf{Q}_U(n) \begin{pmatrix} u[k+n-2] \\ \vdots \\ u[k] \end{pmatrix} \quad (13.11)$$

Gemäß Gleichung (13.11) können nun an Hand der Sequenzen von Ein- und Ausgangswerten

$$\underline{Y}[k+n-1] = (y[k+n-1] \ \dots \ y[k])^T \quad (13.12)$$

$$\underline{U}[k+n-2] = (u[k+n-2] \ \dots \ u[k])^T \quad (13.13)$$

die Systemzustände  $\underline{x}_0[k]$  des Zeitpunkts  $k$  bestimmt werden, womit für eine Modellbildung die Systemzustände nicht mehr explizit notwendig sind. Durch Einsetzen des Zustandsvektors  $\underline{x}_0[k]$  aus Gleichung (13.11) in Gleichung (13.8) ergibt sich das gesuchte Modell, welches eindeutig das Ein- Ausgangsverhalten beschreibt – liegen Sequenzen vergangener Ein- und Ausgangswerte vor, so können Prädiktionen des Systemausgangs  $\hat{y}[k+m]$  mit  $m \geq n$  durchgeführt werden;  $y[k+n-1]$  gilt gemäß Gleichung (13.12) bereits als bekannt:

$$\begin{aligned}\hat{y}[k+m] &= \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \begin{pmatrix} y[k+n-1] \\ \vdots \\ y[k] \end{pmatrix} + \dots \quad (13.14) \\ &\quad - \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \mathbf{Q}_U(n) \begin{pmatrix} u[k+n-2] \\ \vdots \\ u[k] \end{pmatrix} + \underline{q}_U^T(m) \begin{pmatrix} u[k+m-1] \\ \vdots \\ u[k] \end{pmatrix}\end{aligned}$$

Durch eine Verschiebung der Zeitachse um  $n-1$  Schritte genügt man der Deklaration, dass der Zeitpunkt  $k$  als die Gegenwart bezeichnet wird und somit alle Daten von Ein- und Ausgängen bis zum Zeitpunkt  $k$  bekannt sind. Die Differenz  $p = m - n + 1$  gibt an, wie viele Schritte  $p \in \{1, 2, \dots\}$  in die Zukunft prädiziert werden:

$$\begin{aligned}\hat{y}[k+p] &= \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \underbrace{\begin{pmatrix} y[k] \\ \vdots \\ y[k-n+1] \end{pmatrix}}_{\underline{Y}[k] \in \mathbb{R}^n} + \dots \quad (13.15) \\ &\quad - \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \mathbf{Q}_U(n) \underbrace{\begin{pmatrix} u[k-1] \\ \vdots \\ u[k-n+1] \end{pmatrix}}_{\underline{U}[k-1] \in \mathbb{R}^{(n-1)}} + \underline{q}_U^T(n+p-1) \underbrace{\begin{pmatrix} u[k+p-1] \\ \vdots \\ u[k-1] \\ \vdots \\ u[k-n+1] \end{pmatrix}}_{\underline{U}_S[k+p-1] \in \mathbb{R}^{(n-1+p)}}\end{aligned}$$

Die ersten beiden Summanden bestimmen den für das Modell bzw. für die Prädiktion notwendigen Ausgangszustand  $\underline{x}_0[k-n+1]$  mit Hilfe der erforderlichen Anzahl an aktuell gemessenen Ein- und Ausgangssignalen, d.h. in  $\underline{Y}[k]$  bzw.  $\underline{U}[k-1]$  finden sich keine Werte für Zeiten größer  $k$ . Der dritte Summand dient der Prädiktion von diesem abgeleiteten Ausgangszustand aus, weshalb in der Steuersequenz  $\underline{U}_S[k+p-1]$  Werte der Stellgröße für Zeiten bis  $k+p-1$  enthalten sein können, d.h. auch zukünftige Stellgrößen, die einen Beitrag zum Ausgangswert  $\hat{y}[k+p]$  leisten. Ob bereits die Stellgröße  $u[k+p-1]$  nach einem Zeitschritt eine Auswirkung auf den Ausgang  $y[k+p]$  zeigen wird oder auf

Grund der Signallaufzeiten durch das System erst  $\delta$  Zeitschritte verzögert den Ausgang erreicht, wird durch den sog. Relativgrad  $\delta$  beschrieben. Sofern kein System mit Durchgriff vorliegt, dauert es mindestens  $\delta = 1$  Zeitschritt, bis sich eine Stellgröße aufschaltung  $u[k]$  auf das Ausgangssignal  $y[\cdot]$  auswirkt.

Im Gleichungssystem (13.10) ist zu erkennen, dass ein Eingangssignal  $u[k]$  zum Zeitpunkt  $k$  sofort im nächsten Zeitpunkt  $k + 1$  auf das Ausgangssignal  $y[k + 1]$  wirkt – es wird somit ein System mit Relativgrad  $\delta = 1$  beschrieben. Im Falle eines höheren Relativgrads  $\delta > 1$  würde es  $\delta$  Zeitschritte dauern, bis eine Erregung am Eingang am Ausgang gemessen werden kann. Im Gleichungssystem (13.10) hätte dies zur Konsequenz, dass  $\underline{c}^T \underline{b} = \underline{c}^T \mathbf{A} \underline{b} = \dots = \underline{c}^T \mathbf{A}^{\delta-2} \underline{b} \equiv 0$  und erst  $\underline{c}^T \mathbf{A}^{\delta-1} \underline{b} \neq 0$  wären. Die momentan anliegende Stellgröße  $u[k]$  wirkt sich demnach erst bei  $y[k + \delta]$  aus. Sobald auf Grund von  $\delta > 1$  die Einträge

$$\underline{c}^T \mathbf{A}^i \underline{b} = 0 \quad \text{für} \quad i \in \{0, \dots, \delta - 2\} \quad (13.16)$$

in der Matrix  $\mathbf{Q}_U(n)$  des Gleichungssystems (13.10) sowie Vektor  $\underline{q}_U(m)$  der Gleichung (13.8) zu Null gesetzt werden können, hat das Stellsignal  $u[\cdot]$  im Unterschied zu dem im Prädiktor-Modell (13.15) zu  $\delta - 1$  Zeitpunkten noch keine Auswirkung auf den betrachteten Ausgang  $\hat{y}[k + p]$ . Dementsprechend lässt sich die Ordnung der Vektoren  $\underline{U}[\cdot]$  und  $\underline{U}_S[\cdot]$  auf  $n - \delta$  bzw.  $n - \delta + p$  reduzieren. Die Ordnung der Matrix  $\mathbf{Q}_U(n)$  muss durch Streichen von  $\delta - 1$  Null-Spalten angepasst werden, womit sich eine Ordnung von  $n \times (n - \delta)$  ergibt. Ebenso muss die Ordnung des Vektors  $\underline{q}_U(m)$  durch Streichen der entsprechenden Nullen auf  $n - \delta + p$  reduziert werden. Für den allgemeinen Fall mit Relativgrad  $\delta \geq 1$  lässt sich das Prädiktor-Modell (13.15) dann wie folgt formulieren:

$$\begin{aligned} \hat{y}[k + p] &= \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \underbrace{\begin{pmatrix} y[k] \\ \vdots \\ y[k - n + 1] \end{pmatrix}}_{\underline{Y}[k] \in \mathbb{R}^n} + \dots \\ &\quad - \underline{c}^T \mathbf{A}^m \mathbf{Q}_B^{-1}(n) \underbrace{\mathbf{Q}_{U,red}(n)}_{\in \mathbb{R}^{(n \times (n-\delta))}} \underbrace{\begin{pmatrix} u[k - \delta] \\ \vdots \\ u[k - n + 1] \end{pmatrix}}_{\underline{U}_{red}[k-\delta] \in \mathbb{R}^{(n-\delta)}} + \dots \\ &\quad + \underbrace{\underline{q}_{U,red}^T(n + p - 1)}_{\in \mathbb{R}^{(n-\delta+p)}} \underbrace{\begin{pmatrix} u[k + p - \delta] \\ \vdots \\ u[k - \delta] \\ \vdots \\ u[k - n + 1] \end{pmatrix}}_{\underline{U}_{S,red}[k+p-\delta] \in \mathbb{R}^{(n-\delta+p)}} \end{aligned} \quad (13.17)$$

Hiermit wird deutlich, dass für die Bestimmung des zurückliegenden Anfangszustandes  $\underline{x}_0[k - n + 1]$  zum Zeitpunkt  $k$  die aktuellsten  $n$  Ausgangswerte herangezogen werden (1. Summand) sowie die letzten auf das System geschalteten  $n - \delta$  Stellsignale, die entsprechend des Relativgrades eine Auswirkung auf  $y[k]$  zeigen (2. Summand). Ausgehend von diesem Anfangszustand  $\underline{x}_0[k - n + 1]$  wird die gewünschte Prädiktion von  $\hat{y}[k + p]$  durchgeführt (3. Summand). Hierzu werden alle Stellsignale ausgehend vom Zeitpunkt  $k - n + 1$  des bestimmten Anfangszustandes  $\underline{x}_0[k - n + 1]$  benötigt, die entsprechend des Relativgrades  $\delta$  eine Auswirkung auf  $\hat{y}[k + p]$  zeigen. Dies können je nach Prädiktionshorizont und Relativgrad ( $p < \delta$ ) auch lediglich vergangene Stellgrößen sein, die sich auf Grund des Relativgrades noch nicht auf den aktuellen Ausgangswert  $y[k]$  ausgewirkt haben, sondern erst auf einen zukünftigen Ausgangswert wirken. Für  $p = \delta$  bzw.  $p > \delta$  werden neben vergangener Werte für das Stellsignal  $u[\cdot]$  auch die aktuelle Stellgröße  $u[k]$  bzw. zukünftige Stellgrößen  $u[k + i]$  mit  $i \in \mathbb{N}$  für die Bestimmung des zu prädizierenden Ausgangswertes benötigt.

Führt man die Matrizen- und Vektoroperationen in Gleichung (13.17) durch und fasst alle Ein- und Ausgänge in einem Regressionsvektor  $\underline{x}[k + 1]$  zusammen, ergibt sich das allgemeine ARMA-Modell in Vektordarstellung:

$$\hat{y}[k + p] = \underbrace{\begin{pmatrix} -\alpha'_{n-1} & \cdots & -\alpha'_0 & \beta'_{n-1-\delta+p} & \cdots & \beta'_0 \end{pmatrix}}_{\theta_0'^T} \underbrace{\begin{pmatrix} y[k] \\ \vdots \\ y[k - n + 1] \\ u[k + p - \delta] \\ \vdots \\ u[k - n + 1] \end{pmatrix}}_{\underline{x}[k + 1]} \quad (13.18)$$

Die berechneten Koeffizienten werden als die Parameter des ARMA-Modells bezeichnet und sind in dem Parametervektor  $\theta_0$  zusammengefasst. Die Bedeutung des Namens „ARMA-Modell“, „Auto-Regressive-Moving-Average-Model“ wird in der Summendarstellung deutlich:

$$\hat{y}[k + p] = \underbrace{- \sum_{i=0}^{n-1} \alpha'_{n-1-i} \cdot y[k - i]}_{\text{Auto-Regressive-Anteil}} + \underbrace{\sum_{j=\delta-p}^{n-1} \beta'_{n-1-j} \cdot u[k - j]}_{\text{Moving-Average-Anteil}} \quad (13.19)$$

Die Grundform des *ARMA-Modells* entsteht, wenn  $p = 1$  gewählt wird:

$$\hat{y}[k+1] = \underbrace{\begin{pmatrix} -\alpha''_{n-1} & \dots & -\alpha''_0 & \beta''_{n-\delta} & \dots & \beta''_0 \end{pmatrix}}_{\theta''_0^T} \underbrace{\begin{pmatrix} y[k] \\ \vdots \\ y[k-n+1] \\ u[k+1-\delta] \\ \vdots \\ u[k-n+1] \end{pmatrix}}_{\underline{x}[k+1]} \quad (13.20)$$

Hierbei werden nur vergangene Werte der Stellgröße bzw. bei  $\delta = 1$  der aktuelle Wert der Stellgröße verwendet und es wird lediglich einen Schritt in die Zukunft prädiziert. Das dynamische Systemverhalten wird damit eindeutig beschrieben – es kann zu jedem Zeitpunkt an Hand des Modells der nächste zu erwartende Ausgangswert mit Hilfe vergangener und aktueller Ein- und Ausgangswerte bestimmt werden.<sup>1)</sup> Der Vorteil der Modellbeschreibung (13.20) besteht darin, dass auf Grund des geringen Prädiktionshorizontes von  $p = 1$  keine zusätzliche Systemdynamik wegen der Prädiktion entsteht, d.h. in der entsprechenden Übertragungsfunktion der Ordnung  $n$

$$\frac{\hat{y}(z)}{u(z)} = \frac{B''(z)}{A''(z)} = \frac{\beta''_{n-\delta} \cdot z^{n-\delta} + \dots + \beta''_0}{z^n + \alpha''_{n-1} \cdot z^{n-1} + \dots + \alpha''_0} \quad (13.21)$$

wird es keine prädiktionsbedingte Pol- Nullstellenkürzungen geben. Der Nachteil dieser Modellform hingegen zeigt sich bei  $\delta > 1$ , wenn das Modell zur Stellgrößenberechnung, wie im Falle des adaptiven Referenzmodell-Reglers, verwendet werden soll. In Gleichung (13.20) ist zu erkennen, dass bei  $\delta > 1$  nach einer Verschiebung um  $\delta - 1$  zwar nach  $u[k]$  aufgelöst werden kann, jedoch dann auch zukünftige Werte des Ausgangs bis  $y[k+\delta-1]$  bekannt sein müssten. Die Stellgrößenberechnung stößt hiermit auf ein Realisierungsproblem.

Dieses kann gelöst werden, wenn in Gleichung (13.18)  $p = \delta$  gewählt wird; sobald  $p > 1$  gilt, bezeichnet man das ARMA-Modell als *Prädiktor-ARMA-Modell*:

---

<sup>1)</sup> Gleichung (13.20) entspricht nach einer zeitlichen Verschiebung um einen Schritt im Kontext der Identifikation der Gleichung (7.24), mit der eine rekursive Berechnung des momentanen Ausgangswertes  $y[k]$  aus der gewichteten Summe bekannter Vergangenheitswerte der Ein- und Ausgangsgrößen möglich ist. Es sei an dieser Stelle angemerkt, dass sich die Indizierungen der genannten Gleichungen unterscheiden - die Indizierung wurde in Gleichung (13.20) derart gewählt, dass die Übertragungsfunktion (13.21) die gewohnte Form hat. Ist dies nicht erwünscht, kann wie in Gleichung (7.24) indiziert werden. Zur Vermeidung von Verwechslungen werden in diesem Kapitel die Koeffizienten mit  $\alpha_i$  und  $\beta_j$  an Stelle von  $a_i$  und  $b_j$  benannt.

$$\hat{y}[k + \delta] = \underbrace{\begin{pmatrix} -\alpha_{n-1} & \dots & -\alpha_0 & \beta_{n-1} & \dots & \beta_0 \end{pmatrix}}_{\underline{\theta}_0^T} \begin{pmatrix} y[k] \\ \vdots \\ y[k-n+1] \\ u[k] \\ \vdots \\ u[k-n+1] \\ \underline{x}[k+1] \end{pmatrix} \quad (13.22)$$

Für diesen Spezialfall wird bestimmt, wie und wann eine zum Zeitpunkt  $k$  aufgeschaltete Stellgröße  $u[k]$  das Ausgangssignal beeinflusst - gemäß Gleichung (13.22) wird sich deren Wirkung zum Zeitpunkt  $k + \delta$  zeigen. Im Kontext der Stellgrößenberechnung kann nun mit dem zukünftigen Wert  $\hat{y}[k + \delta]$  ein Sollwert  $y^*[k + \delta]$  vorgegeben und nach  $u[k]$  aufgelöst werden. Im Unterschied zum zeitlich verschobenen Modell (13.20) liegen für die Berechnung nur bekannte Werte vor, womit die Stellgrößenberechnung realisierbar wird. Im Gegenzug wird jedoch durch den höheren Prädiktionshorizont mit  $p = \delta$  eine nicht beobachtbare zusätzliche Dynamik entstehen; da mehr vergangene Stellgrößenwerte benötigt werden, erhöht sich die Ordnung der entsprechenden Übertragungsfunktion im Vergleich zur Übertragungsfunktion (13.21) von  $n$  auf  $\delta + n - 1$ :

$$\frac{\hat{y}(z)}{u(z)} = \frac{B(z)}{A(z)} = \frac{\beta_{n-1} \cdot z^{n-1} + \dots + \beta_0}{z^{\delta+n-1} + \alpha_{n-1} \cdot z^{n-1} + \dots + \alpha_0} \quad (13.23)$$

Die zusätzliche Dynamik der Prädiktion der Ordnung  $\delta - 1$  ist jedoch auf Grund von Pol- Nullstellenkürzungen am Ausgang nicht beobachtbar, sofern diese stabil ist.

Der große Vorteil des ARMA-Modells besteht in der Tatsache, dass das Ausgangssignal linear in den Parametern darstellbar ist und daher, wie im folgenden Abschnitt dargestellt, sehr einfache lineare Identifikationsmethoden angewandt werden können. Im Weiteren bezieht man sich auf das Prädiktor-ARMA-Modell (13.22):

$$\hat{y}[k + \delta] = \underline{\theta}_0^T \underline{x}[k + 1] \quad (13.24)$$

## 13.2 Systemidentifikation

Das zentrale Element der adaptiven Regelung ist ein Parameterschätzer. Seine Aufgabe ist es, durch eine geeignete Adaption im geschlossenen Regelkreis dem Regler Systemparameter zur Verfügung zu stellen, mit denen eine sichere und gute Regelung des Systems zu jedem Zeitpunkt durchgeführt werden kann. Zum Erreichen dieses Ziels wird die Eigenschaft des Identifikationsalgorithmus ausgenutzt, zu jedem Zeitpunkt durch eine Adaption der Parameter einen Identifikationsfehler von Null zu erzwingen. Mit einer geeigneten Wahl des Regelgesetzes

sowie Kombination von Regler und Identifikation ist zu erreichen, dass der Regelfehler die Eigenschaft des Identifikationsfehlers erbt und somit gegen Null geht, weshalb ein erfolgreiches Regelverhalten trotz unbekannter Systemparameter zu erwarten ist.

Es werden zwei Identifikations-Algorithmen vorgestellt, die jeweils die notwendige Eigenschaft für die Eingliederung in den adaptiven Regler besitzen: der Projektionsalgorithmus [68] sowie der rekursive Least-Squares-Algorithmus (RLS: vgl. Kapitel 4.2.2 und [68]). Beide setzen die Trennung von Parametern und Systemsignalen (Ein- und Ausgänge) in Form von Vektoren entsprechend des ARMA-Modells (vgl. Kapitel 13.1) voraus.

Wie dargestellt wurde, beschreibt das ARMA-Modell das dynamische Verhalten eines linearen Systems. Sind die Systemparameter beispielsweise an Hand einer Zustandsdarstellung (13.5)/(13.6) oder einer Übertragungsfunktion (13.21) bekannt, so kann im ersten Fall mit Hilfe der Gleichung (13.17) bzw. im zweiten Fall an Hand rekursiver Umformung das ARMA-Modell (13.22) mit dem Parametervektor  $\underline{\theta}_0$  erzeugt werden. Dieser Parametervektor  $\underline{\theta}_0$  beschreibt dann exakt das Ein- Ausgangsverhalten des Systems, so dass mit Hilfe des bereits vergangenen Ausgangssignalverlaufes sowie Stellsignalverlaufes der aktuelle Wert  $y[k]$  des Ausgangssignals bestimmt werden kann. Hierzu dient das um  $\delta$  Schritte verschobene ARMA-Modell (13.22) bzw. (13.24), das sog. Strecken-Modell:

$$y[k] = \hat{y}[k] = \underline{\theta}_0^T \underline{x}[k - \delta + 1] \quad (13.25)$$

Liegt der genaue Parametervektor  $\underline{\theta}_0$  der Strecke vor, so entspricht der über das Modell berechnete Wert  $\hat{y}[k]$  exakt dem zum Zeitpunkt  $k$  gemessenen Wert  $y[k]$ . Nachdem ein unbekanntes System vorliegt und folglich der Parametervektor  $\underline{\theta}_0$  nicht bekannt ist, kann im Umkehrschluss zum Zeitpunkt  $k$  mit Hilfe des gemessenen Werts  $y[k]$  ein Parametervektor  $\hat{\underline{\theta}}[k]$  bestimmt werden, der in Kombination mit dem Regressionsvektor  $\underline{x}[k + 1 - \delta]$  den Ausgangswert  $y[k]$  ergibt, d.h. die Gleichung (13.25) zumindest für den Zeitpunkt  $k$  erfüllt.

Mit Hilfe der Identifikations-Algorithmen soll nun zu jedem Zeitschritt  $k$  ein Schätzwert  $\hat{\underline{\theta}}[k]$  von  $\underline{\theta}_0$  bestimmt werden. Wie gut dieser Schätzwert bezogen auf die Dynamik des Ein- Ausgangsverhaltens, d.h. die zeitliche Veränderungen des Regressionsvektors ist, lässt sich frühestens einen Abtastschritt später feststellen. Aus diesem Grund prädiziert man mit dem neuen Schätzwert  $\hat{\underline{\theta}}[k]$  einen Schritt ( $p = 1$ ) in die Zukunft; das dabei verwendete ARMA-Modell (13.24) bezeichnet man als Schätz-Modell:

$$\hat{y}[k + 1] = \hat{\underline{\theta}}[k]^T \underline{x}[k - \delta + 2] \quad (13.26)$$

Bildet man dann im nächsten Zeitschritt die Differenz zwischen dem nun messbaren Ausgangswert  $y[k+1]$  des Systems bzw. des exakten Strecken-Modells (13.25) und dem Ausgangswert  $\hat{y}[k+1]$  des Schätz-Modells (13.26), so kann der *Identifikationsfehler*

$$\varepsilon[k+1] = y[k+1] - \hat{y}[k+1] = y[k+1] - \underline{\hat{\theta}}[k]^T \underline{x}[k-\delta+2] \quad (13.27)$$

$$= (\underline{\theta}_0 - \underline{\hat{\theta}}[k])^T \underline{x}[k-\delta+2] = \tilde{\underline{\theta}}[k]^T \underline{x}[k-\delta+2] \quad (13.28)$$

definiert werden, welcher eine Aussage darüber liefert, wie gut ein geschätzter Parametervektor das Ein- Ausgangsverhalten des Systems in Abhängigkeit des Regressionsvektors nach Verstreichen einer Abtastzeit  $h$  repräsentiert. Gemäß Gleichung (13.28) gibt es zwei Möglichkeiten, dass der Identifikationsfehler  $\varepsilon[k+1]$  Null wird: dies tritt ein, wenn der geschätzte Parametervektor  $\hat{\underline{\theta}}[k]$  dem wahren System-Parametervektor  $\underline{\theta}_0$  entspricht und somit das System korrekt identifiziert wurde, d.h.  $\underline{\hat{\theta}}[k] = 0$  gilt, oder, wenn das Skalarprodukt  $\underline{\hat{\theta}}[k]^T \underline{x}[k-\delta+2]$  zu Null wird, womit der Parameterfehlervektor  $\tilde{\underline{\theta}}[k]^T$  und der Regressionsvektor  $\underline{x}[k-\delta+2]$  wie zum Zeitpunkt  $k-\delta+1$  der Schätzung weiterhin senkrecht aufeinander stehen. Der letzte Fall tritt auf, wenn ein bestimmtes dynamisches Verhalten des Regressionsvektors, d.h. Ein- Ausgangsverhaltens bereits gelernt wurde – dann bewegt sich zwar der Regressionsvektor, steht jedoch immer senkrecht auf dem Parameterfehlervektor. In anderen Worten wurde das momentan auftretende dynamische Verhalten bereits gelernt, womit  $\varepsilon[k+1] = 0$  resultiert; da jedoch noch nicht das vollständige dynamische Systemverhalten gelernt wurde, verbleibt trotzdem ein Parameterfehlervektor ungleich Null. Sobald noch nicht identifizierte Systemdynamik angeregt wird, stehen die Vektoren nicht mehr senkrecht aufeinander und es resultiert ein Identifikationsfehler  $\varepsilon[k+1] \neq 0$ . Ähnelt das Ein- Ausgangsverhalten bereits identifiziertem Verhalten, so stehen der Regressionsvektor und der Parameterfehlervektor weiterhin annähernd senkrecht zueinander, womit ein nur sehr kleiner Identifikationsfehler resultiert. Der geschätzte Parametervektor  $\hat{\underline{\theta}}[k]$  repräsentiert daher immer noch ausreichend das Systemverhalten, womit gleichzeitig geregelt ( $\hat{\underline{\theta}}[k]$  dient der Auslegung des Reglers) und auf Grund der neuen Anregung weiter identifiziert werden kann, so dass ein neuer Schätzwert  $\hat{\underline{\theta}}[k+1]$  zukünftig auch bei der aktuellen Anregung zu einem Identifikationsfehler Null führt. Tritt jedoch eine bisher noch nicht identifizierte Anregung auf, so stehen der Regressionsvektor und der Parameterfehlervektor weder annähernd senkrecht aufeinander noch ist letzter Null. In diesem Fall wird sich ein großer Identifikationsfehler zeigen; das Regelverhalten wird zwar in diesem Moment schlechter, jedoch folgt auf Grund der hohen Anregung ein rascher Identifikationsvorgang, so dass neben erneut senkrecht stehendem Regressionsvektor und Parameterfehlervektor sich der Betrag des Parameterfehlervektors  $\tilde{\underline{\theta}}[k+1]$  stark verkleinert und schnell wieder zu einem guten Regelergebnis führt. Sobald ein Identifikationsfehler ungleich Null vorliegt, führt der nächste Identifikationsschritt zu einer Verkleinerung des Parameterfehlervektors bis er schließlich zu Null wird, was bedeutet, dass das System inzwischen vollständig angeregt und somit vollständig identifiziert wurde. Der Vorteil dieses Konzeptes besteht darin, dass nur so viel Systemdynamik identifiziert werden muss, wie durch das Wunschregelverhalten durch das Sollsignal angeregt wird. In anderen Worten muss nur so viel Information identifiziert werden, wie zum momentanen Regeln des Systems notwendig ist.

Entsprechend dieser Überlegungen müsste stets ein stabiler Regler vorliegen, der trotz unbekannter Systemparameter ein gutes Regelverhalten zeigt und parallel zum Regelsvorgang das System entsprechend der Anregung des Systems ausreichend identifiziert – dies soll in den nächsten Kapiteln näher untersucht werden.

Auf der Grundlage dieser Erkenntnisse ist nun ein Algorithmus zu finden, der, wie oben angedeutet, zu jedem Zeitpunkt  $k$  für den aktuellen Regressionsvektor  $\underline{x}[k - \delta + 1]$  und Ausgangswert  $y[k]$  einen Parametervektor  $\hat{\underline{\theta}}[k]$  bestimmt, der bei gleichbleibender Anregung, d.h. der Regressionsvektor steht senkrecht auf dem Parameterfehlervektor ( $\underline{x}[\cdot] \perp \hat{\underline{\theta}}[k]$ ), zu einem Identifikationsfehler  $\varepsilon[\cdot] = 0$  führt.

### 13.2.1 Projektionsalgorithmus

Das Prinzip der Identifikation lässt sich sehr anschaulich an Hand des Projektionsalgorithmus erklären. Man bezieht sich im Folgenden auf das Strecken-Modell (13.25). Da zum Zeitpunkt  $k$  sowohl der Ausgang  $y[k]$  als auch der Regressionsvektor  $\underline{x}[k - \delta + 1]$  durch Messung bekannt sind, ist eine Aussage über den zu identifizierenden unbekannten Systemparametervektor  $\underline{\theta}_0$  möglich. Die Gleichung (13.25) entspricht mathematisch gesehen einer Hyperebenengleichung:

$$H = \{z \mid y[k] = z^T \underline{x}[k - \delta + 1]\} \quad (13.29)$$

Alle Punkte  $z$ , die die Gleichung (13.29) erfüllen, liegen in der Hyperebene, deren Abstand vom Ursprung durch  $y[k]$  bestimmt wird und dessen Normalenvektor  $\underline{x}[k - \delta + 1]$  ist. Ein einziger Punkt  $z = \underline{\theta}_0$  der Ebene entspricht dem wahren Parametervektor; jedoch führen alle anderen Parametervektoren  $\underline{\theta} = z$  für diese Konstellation von Ein- Ausgangswerten, beschrieben durch den Regressionsvektor  $\underline{x}[k - \delta + 1]$ , zu dem selben Ausgangswert  $y[k]$  wie das Strecken-Modell (13.25) des realen Systems. Es wird nun von all den möglichen Parametervektoren  $z$  ein Schätzwert  $\hat{\underline{\theta}}[k]$  ausgewählt. Geht man davon aus, dass sich der Regressionsvektor, wie einleitend beschrieben, innerhalb eines Abtastschrittes von  $\underline{x}[k - \delta + 1]$  auf  $\underline{x}[k - \delta + 2]$  nicht zu sehr ändert bzw. nur bereits identifizierte Änderungen vollzieht, bei denen gemäß Gleichung (13.28)  $\underline{x}[k - \delta + 2] \perp \hat{\underline{\theta}}[k]$  gilt, so wird der neue Parametervektor  $\hat{\underline{\theta}}[k]$  zu einem Identifikationsfehler  $\varepsilon \approx 0$  bzw.  $\varepsilon = 0$  führen.

Von den möglichen Vektoren  $z$  der Hyperebene  $H$  wird im Projektionsalgorithmus das  $\hat{\underline{\theta}}[k] \in H$  mit dem geringsten Abstand zum zuletzt geschätzten Parametervektor  $\hat{\underline{\theta}}[k - 1]$  ausgewählt. Es muss hierzu das Minimum von

$$J = \frac{1}{2} \left\| \hat{\underline{\theta}}[k] - \hat{\underline{\theta}}[k - 1] \right\|^2$$

unter der Nebenbedingung, dass  $\hat{\underline{\theta}}[k]$  die Ebenengleichung (13.29) erfüllt, berechnet werden. Dies kommt einer orthogonalen Projektion auf die Hyperebene  $H$  gleich, womit der Name des Algorithmus erklärt ist.

Extremwertuntersuchungen mit Nebenbedingungen können mit Hilfe des Lagrange-Multiplikators durchgeführt werden. Die Lagrange Hilfsfunktion lautet:

$$L(\hat{\underline{\theta}}[k], \lambda) = \frac{1}{2} \left\| \hat{\underline{\theta}}[k] - \hat{\underline{\theta}}[k-1] \right\|^2 + \lambda \underbrace{\left( y[k] - \hat{\underline{\theta}}[k]^T [k-\delta+1] \right)}_{\text{homogene NB}}$$

Für die Minimumssuche bildet man  $\nabla L(\hat{\underline{\theta}}[k], \lambda) = 0$ , womit zwei Gleichungssysteme resultieren:

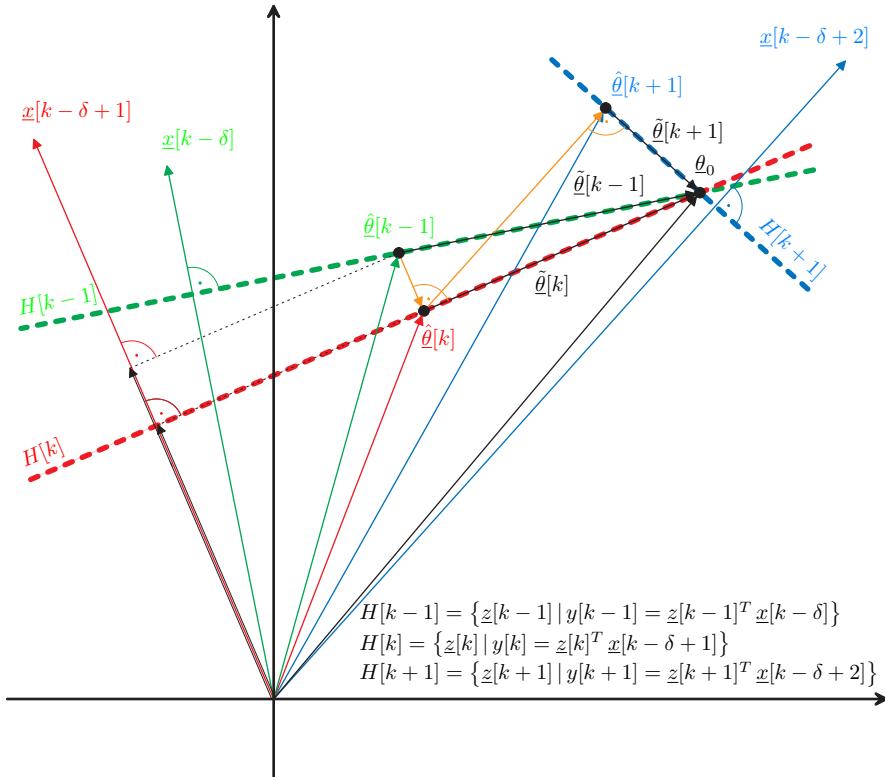
$$\hat{\underline{\theta}}[k] - \hat{\underline{\theta}}[k-1] - \lambda \underline{x}[k-\delta+1] = 0 \quad (13.30)$$

$$y[k] - \hat{\underline{\theta}}[k]^T \underline{x}[k-\delta+1] = 0 \quad (13.31)$$

Löst man nun Gleichung (13.30) nach  $\hat{\underline{\theta}}[k]$  auf und setzt den Ausdruck in (13.31) ein, so lässt sich mit der neu gewonnenen Gleichung das  $\lambda$  in (13.30) ersetzen. Das Ergebnis ist der Projektionsalgorithmus, mit dem zu jedem Zeitschritt eine Aktualisierung des Parametervektors durchgeführt werden kann, so dass ein Identifikationsfehler  $\varepsilon[\cdot] \approx 0$  durch ständige Adaption von Beginn an erreicht werden kann:

$$\hat{\underline{\theta}}[k] = \hat{\underline{\theta}}[k-1] + \frac{\underline{x}[k-\delta+1]}{\underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \underbrace{\left( y[k] - \hat{\underline{\theta}}[k-1]^T \underline{x}[k-\delta+1] \right)}_{\varepsilon[k]} \quad (13.32)$$

Die geometrische Interpretation des Projektionsalgorithmus ist in Abbildung 13.1 für einen zweidimensionalen Fall zu sehen. Es sind für drei Abtastzeitpunkte ( $k-1, k, k+1$ ) die Hyperebenen  $H[\cdot]$  mit ihren senkrecht stehenden Regressionsvektoren  $\underline{x}[\cdot]$  als Normalenvektoren gemäß Gleichung (13.29) eingezeichnet. Da der Parametervektor  $\underline{\theta}_0$  die wirklichen Systemparameter enthält und folglich  $\underline{\theta}_0$  die Gleichung (13.29) zu jedem beliebigen Zeitpunkt und bei jedem beliebigen Regressionsvektor  $\underline{x}[\cdot]$  erfüllt, müssen die Hyperebenen  $H[\cdot]$  stets durch den Punkt  $\underline{\theta}_0$  verlaufen und diesen somit enthalten. Entsprechend obiger Überlegungen erfüllen zum Zeitpunkt  $k$  alle Parametervektoren  $\hat{\underline{\theta}}[k]$  in der Hyperebene  $H[k]$  in Verbindung mit dem Regressionsvektor  $\underline{x}[k-\delta+1]$  die Modellgleichung (13.25), d.h. führen zu dem gleichen Ausgangswert  $y[k]$ . Geometrisch ist dieser Sachverhalt deutlich in Abbildung 13.1 zu erkennen. Betrachtet man den Parameterfehlervektor  $\hat{\underline{\theta}}[k] = \underline{\theta}_0 - \hat{\underline{\theta}}[k]$ , so steht dieser senkrecht auf dem Regressionsvektor  $\underline{x}[k-\delta+1]$ , womit das Skalarprodukt zwischen beiden Null wird; dies gilt für alle Vektoren  $\underline{z}$  der Ebene  $H[k]$ . Folglich hat der Parameterfehler zum Zeitpunkt  $k$  keine Auswirkung auf den Ausgang – sowohl  $\underline{\theta}_0$  als auch alle  $\hat{\underline{\theta}}[k] \in H$  ergeben in Verbindung mit  $\underline{x}[k-\delta+1]$  den Ausgangswert  $y[k]$ . Würde der Regressionsvektor zum nächsten Zeitpunkt  $k+1$  unverändert bleiben bzw. nur seinen Betrag und nicht die Richtung verändern, d.h. sich in keiner oder nur einer Dimension bewegen (gedrehtes kartesisches Koordinatensystem oder Polarkoordinatensystem), wäre das entsprechende stationäre bzw. beschränkt dynamische Systemverhalten durch den geschätzten Parametervektor



**Abb. 13.1:** Geometrische Interpretation des Projektionsalgorithmus an einem System erster Ordnung ( $n = 1$ ,  $\delta = 1$ ) mit zweidimensionalem Parametervektor

$\hat{\theta}[k]$  eindeutig beschrieben – es gelte dann weiter  $\underline{x}[k-\delta+2] \perp \hat{\theta}[k]$ , womit gemäß Gleichung (13.28) der Identifikationsfehler  $\varepsilon[k+1]$  zu Null wird.<sup>2)</sup>

<sup>2)</sup> Ein zweidimensionaler Regressionsvektor ist einem dynamischen System erster Ordnung zuzuordnen. Wird diesem ein Sinussignal aufgeschaltet, so zeigt sich eine ellipsenförmige Bewegung im zweidimensionalen Raum des Regressionsvektors. Bei hohen Frequenzen ist das Ein- und Ausgangssignal um  $90^\circ$  phasenverschoben und die Ellipse wird zum Kreis; durch die Anregung des Systems findet eine schnelle Konvergenz der geschätzten Parameter  $\hat{\theta}[k]$  zu den wahren Parametern  $\theta_0$  statt (siehe Anmerkung auf Seite 505). Bei niedrigen Frequenzen ergibt sich annähernd keine Phasenverschiebung und der Regressionsvektor erfährt lediglich eine sehr langsame Betrags- und Richtungsänderung, womit die geschätzten Parameter  $\hat{\theta}[k]$  sehr langsam zu den wahren Parametern  $\theta_0$  konvergieren. Im Grenzübergang (konstante Anregung) wird die Ellipse zur Geraden und es findet keine Richtungsänderung sondern nur noch eine Betragsänderung des Regressionsvektors statt und das dynamische System verhält sich wie ein statisches System, welches lediglich einen Verstärkungsfaktor zwischen Ein- und Ausgang

Es ist in der Grafik weiter gut zu erkennen, dass eine kleine Richtungsveränderung des Regressionsvektors zu einer vernachlässigbaren Komponente von  $\tilde{\theta}[k]$  parallel zu  $\underline{x}[k - \delta + 1]$  führt, d.h. das entsprechende Skalarprodukt zur Berechnung des Identifikationsfehlers ergibt nur einen sehr kleinen Wert:  $\varepsilon[k + 1] \approx 0$ .

Der Projektionsalgorithmus wählt als neuen Parametervektor  $\hat{\theta}[k]$  nach Definition innerhalb aller möglichen Vektoren  $z[k]$  der Hyperebene  $H[k]$  den mit dem kleinsten Abstand zum zuletzt gültigen Parametervektor  $\hat{\theta}[k - 1]$  aus: geometrisch bedeutet dies, dass ein Lot durch  $\hat{\theta}[k - 1]$  auf die Ebene  $H[k]$  gefällt wird, um den neuen Schätzwert  $\hat{\theta}[k]$  zu erhalten. Für die Richtung dieser orthogonalen Projektion kann  $\underline{x}[k - \delta + 1]$  verwendet werden, da dieser den Normalenvektor der Hyperebene darstellt und somit senkrecht auf dieser steht. Der gesuchte Projektionsvektor zwischen  $\tilde{\theta}[k - 1]$  und  $\hat{\theta}[k]$  ergibt sich demnach durch eine orthogonale Projektion des Parameterfehlervektors  $\tilde{\theta}[k - 1]$  auf den Regressionsvektor  $\underline{x}[k - \delta + 1]$ . Im rechtwinkligen Dreieck gilt

$$\cos \varphi = \frac{\|\hat{\theta}[k] - \hat{\theta}[k - 1]\|}{\|\hat{\theta}[k - 1]\|} \quad (13.33)$$

und das Skalarprodukt ist wie folgt definiert:

$$\tilde{\theta}[k - 1]^T \underline{x}[k - \delta + 1] = \cos \varphi \cdot \|\tilde{\theta}[k - 1]\| \cdot \|\underline{x}[k - \delta + 1]\| \quad (13.34)$$

Nachdem die Gleichungen (13.33) und (13.34) ineinander eingesetzt wurden, kann nach dem Betrag des gesuchten Projektionsvektors  $\|\hat{\theta}[k] - \hat{\theta}[k - 1]\|$  aufgelöst werden. Skaliert man mit diesem den normierten Regressionsvektor  $\underline{x}[k - \delta + 1]$ , so resultiert der Projektionsvektor:

$$\hat{\theta}[k] - \hat{\theta}[k - 1] = \frac{\tilde{\theta}[k - 1]^T \underline{x}[k - \delta + 1]}{\underline{x}[k - \delta + 1]^T \underline{x}[k - \delta + 1]} \cdot \underline{x}[k - \delta + 1]$$

Setzt man  $\tilde{\theta}[k - 1] = \underline{\theta}_0 - \hat{\theta}[k - 1]$  und löst nach  $\hat{\theta}[k]$  auf, so ergibt sich über die geometrische Betrachtung ebenfalls der Projektionsalgorithmus:

$$\begin{aligned} \hat{\theta}[k] &= \hat{\theta}[k - 1] + \frac{\underline{\theta}_0^T \underline{x}[k - \delta + 1]}{\underline{x}[k - \delta + 1]^T \underline{x}[k - \delta + 1]} \cdot \underline{x}[k - \delta + 1] \\ &\quad - \frac{\hat{\theta}[k - 1]^T \underline{x}[k - \delta + 1]}{\underline{x}[k - \delta + 1]^T \underline{x}[k - \delta + 1]} \cdot \underline{x}[k - \delta + 1] \end{aligned} \quad (13.35)$$

---

aufzeigt; in diesem Fall ließe sich der Regressionsvektor auf die Ordnung eins verringern. Es tritt keine Annäherung an den wahren Parametervektor  $\underline{\theta}_0$  auf.

Sobald der Regressionsvektor in einem gedrehten Koordinatensystem in mindestens einer Dimension unverändert bleibt, wird der entsprechende dynamische Teil des Systems nicht mehr angeregt, womit dieser durch den zuletzt geschätzten Parametervektor  $\hat{\theta}[k] \neq \underline{\theta}_0$  ausreichend beschrieben wird, bis erneut eine Anregung auftritt. Für eine Konvergenz zu den wahren Parametern ist daher für höherdimensionale Systeme der Ordnung  $n$  eine Anregung in  $n$  Dimensionen des Regressionsvektor, d.h. eine Anregung mit  $n$  verschiedenen Frequenzen notwendig.

Die orthogonale Projektion des zweiten und dritten Summanden sind in Abbildung 13.1 durch die gestrichelten Linien angedeutet. Nach Anwendung der Gleichung (13.25) ist die direkte Übereinstimmung von (13.35) mit (13.32) klar zu sehen.

Mit Hilfe der Abbildung 13.1 kann bereits an dieser Stelle eine Aussage über die Stabilität und Parameterkonvergenz des Projektionsalgorithmus getroffen werden. Auf Grund der orthogonalen Projektion von Ebene  $H[k - 1]$  auf  $H[k]$  entsteht generell ein rechtwinkliges Dreieck mit den Ecken  $\hat{\underline{\theta}}[k - 1]$ ,  $\hat{\underline{\theta}}[k]$  und  $\underline{\theta}_0$ . Nach Pythagoras ist die Hypotenuse des Dreiecks stets größer (bzw. im Grenzfall gleich) als die Katheten; es gilt:

$$\|\tilde{\underline{\theta}}[k]\| \leq \|\tilde{\underline{\theta}}[k - 1]\| \quad (13.36)$$

Man stellt fest, dass im Laufe des Regelungs- bzw. Identifikationsvorganges der Parameterfehler nie zunimmt – der Projektionsalgorithmus ist stabil.

Im Falle einer sehr langsamem oder fast konstanten Anregung des Systems, d.h. der Regressionsvektor erfährt kaum eine Richtungsänderung, wie es die Abbildung 13.1 beim Übergang von  $\underline{x}[k - \delta]$  auf  $\underline{x}[k - \delta + 1]$  darstellen soll, sind die Hyperebenen  $H[k - 1]$  und  $H[k]$  nur gering voneinander verdreht, wodurch sich der geschätzte Parametervektor mit  $\hat{\underline{\theta}}[k - 1] \approx \hat{\underline{\theta}}[k]$  nur unmerklich an den wahren Parametervektor  $\underline{\theta}_0$  annähert und der Parameterfehler  $\|\tilde{\underline{\theta}}[k - 1]\| \approx \|\tilde{\underline{\theta}}[k]\|$  daher als unverändert angesehen werden kann. Es findet hiermit zwar keine oder eine nur sehr langsame Identifikation im Sinne der Konvergenz zum wahren Parametervektor  $\underline{\theta}_0$  statt, jedoch ist der Identifikationsfehler  $\varepsilon[\cdot]$  bei einer nur geringen Änderung des Regressionsvektors zu jedem Zeitpunkt annähernd Null, was einen erfolgreichen Reglevorgang des adaptiven Reglers bei Abwesenheit einer beständigen Systemanregung ermöglichen wird.

Erfährt das System eine beständige Anregung, d.h. hochdynamische Erregung mit einer Vielzahl von Frequenzen (Sprünge, Rauschen), so ändert der Regressionsvektor in kurzer Zeit sehr stark seine Richtung; in Abbildung 13.1 soll dies mit dem Übergang von  $\underline{x}[k - \delta + 1]$  auf  $\underline{x}[k - \delta + 2]$  gezeigt werden. Betrachtet man das rechtwinklige Dreieck, so stellt man eine wesentlich stärkere Abnahme des Betrages des Parameterfehlervektors von  $\|\tilde{\underline{\theta}}[k]\|$  auf  $\|\tilde{\underline{\theta}}[k + 1]\|$  fest. Liegt demnach eine beständigen Anregung des Systems vor, wird der geschätzte Parametervektor  $\hat{\underline{\theta}}[k + i]$  mit zunehmenden  $i$  schnell zu dem wahren Parametervektor  $\underline{\theta}_0$  konvergieren.<sup>3)</sup> Die Identifikationsfehler  $\varepsilon[\cdot]$  sind zwar in einer kurzen transientes Phase groß, jedoch werden schnell die richtigen Parameter gefunden, was wiederum  $\varepsilon[\cdot] = 0$  bedeutet und folglich zu einem guten Regelverhalten führen wird.

---

<sup>3)</sup> Im Falle eines Systems erster Ordnung mit einem zweidimensionalen Regressionsvektor ist eine Anregung mit lediglich einem Sinussignal beliebiger Frequenz ausreichend, um dieses eindeutig zu identifizieren. Dies bestätigt die Abbildung 13.1 mit der Darstellung eines rotierenden Regressionsvektors zweiter Ordnung. Bei jeder noch so kleinen Richtungsänderung findet eine Annäherung an den richtigen Parameterwert statt. Je größer die Richtungsänderung des Regressionsvektors bzw. je schneller dieser rotiert, desto schneller konvergieren die Parameter.

Es wurden die beiden Extremfälle „annähernd keine Anregung“ und „beständige Anregung“ diskutiert. Im ersten Fall dauert es sehr lange, bis die angeregte Dynamik durch den geschätzten Parametervektor identifiziert ist, jedoch wird durch die ständige Adaption  $\varepsilon[\cdot] \approx 0$  gewährt, bis schließlich  $\varepsilon[\cdot] = 0$  erreicht ist. Im Beispiel des Systems erster Ordnung in Abbildung 13.1 liegt dann eine abgeschlossene Identifikation mit  $\hat{\theta}[\cdot] = \underline{\theta}_0$  vor. Bei Systemen höherer Ordnung ist es möglich, dass nur ein Teil der Systemdynamik angeregt wurde; dann gilt das Gesamtsystem mit  $\hat{\theta}[\cdot] \neq \underline{\theta}_0$  als nicht identifiziert, dennoch resultiert  $\varepsilon[\cdot] = 0$ . Dies besagt lediglich, dass die angeregte Dynamik als vollständig identifiziert gilt. Sobald sich die Anregung ändert, ergibt sich erneut  $\varepsilon[\cdot] \neq 0$ . Ist die Anregung wieder dem ersten Extremfall zuzuordnen, zeigt sich in der transientes Phase  $\varepsilon[\cdot] \approx 0$ , bis nach längerer Zeit wieder  $\varepsilon[\cdot] = 0$  gilt. Im zweiten Extremfall „beständige Anregung“ ist  $\varepsilon[\cdot] \neq 0$ , bis nach bereits kurzer Zeit  $\varepsilon[\cdot] = 0$  erreicht wird. In beiden Fällen gilt  $\varepsilon[\cdot] \rightarrow 0$  und in langen transientes Phasen  $\varepsilon[\cdot] \approx 0$ . Dies lässt darauf schließen, dass bis auf kurze Zeiten mit dem geschätzten Parametervektor  $\hat{\theta}[\cdot]$  das angeregte Systemverhalten stets ausreichend bekannt ist, womit eine erfolgreiche stabile Regelung möglich sein muss, sofern der Regler eine Parameteranpassung ermöglicht.

Im Bezug auf physikalisch sinnvolle Erregungen des Systems (Sinus, Rampe, gefilterter Sprung) im Zuge einer Regelung stellt man fest, dass eine Konvergenz zu den wahren Parameterwerten gar nicht oder nur sehr langsam stattfindet, d.h. die entsprechende Anregung mehr dem ersten Extremfall zuzuordnen ist. Aus diesem Grund muss während des Regelvorganges eines unbekannten Systems stets die Möglichkeit einer Adaption der Parameter zum Erreichen eines Identifikationsfehlers  $\varepsilon[\cdot] \rightarrow 0$  gewährt werden, was die Notwendigkeit eines adaptiven Reglers zeigt.

Um den vorgestellten Projektionsalgorithmus für die Realisierung einer Identifikation verwenden zu können, sind zwei Modifikationen notwendig. Betrachtet man den Nenner des zweiten Summanden in Gleichung (13.32), so fällt auf, dass bei einem Betrag Null des Regressionsvektors eine Division durch Null stattfindet. Dies kann verhindert werden, wenn eine kleine Konstante  $c > 0$  dem Nenner addiert wird – hiermit entsteht zwar in der Projektion ein Fehler, der jedoch gering ist und wie in der weiteren Beweisführung gezeigt wird, zu keiner negativen Auswirkung führt. Die zweite Modifikation betrifft die Dynamik des Projektionsalgorithmus. Durch Einführung einer Konstante  $\eta$  mit

$$0 < \eta < 2 \tag{13.37}$$

im Zähler des zweiten Summanden in Gleichung (13.32) wird die Projektionsweite in Richtung der neuen Hyperebene beeinflusst.

Für  $\eta = 1$  findet die Projektion entsprechend obiger Darstellung direkt auf die nächste Hyperebene statt, sobald eine Änderung des Regressionsvektors auftritt. Ändert sich beispielsweise der Regressionsvektor von  $\underline{x}[k - \delta]$  auf  $\underline{x}[k - \delta + 1]$ , so wird ausgehend vom alten Schätzwert  $\hat{\theta}[k - 1]$  direkt auf die Hyperebene  $H[k]$  projiziert. Würde im nächsten Schritt der Regressionsvektor unverändert bleiben,

ergäbe sich ein Identifikationsfehler  $\varepsilon[k+1] = 0$ . Mit  $\eta = 1$  wird der Projektionsalgorithmus daher im Fall „annähernd keine Anregung“ maximal schnell einen Identifikationsfehler Null erzeugen. Würde im beschriebenen Szenario  $\eta < 1$  bzw.  $\eta > 1$  gelten, so fände ausgehend vom alten Schätzwert  $\hat{\theta}[k-1]$  eine Projektion vor bzw. hinter die Hyperebene  $H[k]$  statt – würde in diesem Fall im nächsten Schritt der Regressionsvektor unverändert bleiben, ergäbe sich ein Identifikationsfehler  $\varepsilon[k+1] \neq 0$ . In jedem folgenden Schritt nähert sich der Schätzwert  $\hat{\theta}[\cdot]$  bei gleichbleibendem Regressionsvektor an die Hyperebene an, wodurch mit jedem Schritt der Identifikationsfehler kleiner wird (ein Schätzwert auf der Hyperebene führt zu einem Identifikationsfehler Null). Es ist nun vom Betrag von  $\eta$  abhängig, wie schnell eine Abnahme des Identifikationsfehlers stattfindet. Effektiv werden mit  $\eta < 1$  nur kleine Änderungen im Betrag des geschätzten Parametervektors  $\hat{\theta}[k-1]$  pro Zeitschritt erlaubt. Die Wahl von  $\eta \neq 1$  ist daher auf Grund längerer Konvergenzzeiten im Fall „annähernd keine Anregung“ bei Abwesenheit von Störungen ungeeignet.

Für den Fall „beständige Anregung“ ergibt sich hingegen mit  $\eta < 1$  ein Vorteil: wie oben dargestellt, findet bei einer beständigen Anregung sehr schnell eine Konvergenz zu den wahren Parametern statt, jedoch ergibt sich unter Umständen in dieser transienten Phase ein großer schwankender Identifikationsfehler und folglich großer Regelfehler mit unruhigem Regelverhalten. Nimmt man beispielsweise in Abbildung 13.1 an, dass durch eine entsprechende Anregung der Regressionsvektor ständig zwischen  $\underline{x}[k-\delta+1]$  und  $\underline{x}[k-\delta+2]$  wechselt, nimmt bei einer Projektion mit  $\eta = 1$  der Betrag des Parameterfehlervektors  $\hat{\theta}[\cdot]$  sehr schnell ab, jedoch ergibt sich dabei auch ein größerer Identifikationsfehler. Nach kurzer Zeit sind die wahren Parameter  $\theta_0$  erreicht und es stellt sich  $\varepsilon[\cdot] = 0$  ein. Würde in dem beschriebenen Fall der im Wechsel auftretenden Regressionsvektoren bzw. Hyperebenen beispielsweise  $\eta = 0.5$  gelten, so würde nur zur Hälfte in die Richtung der neuen Hyperebenen projiziert werden, womit der beruhigte neu geschätzte Parametervektor das Systemverhalten beim ständigen Wechsel zwischen beiden Regressionsvektoren besser beschreibt – in der transienten Phase ergäbe sich ein Identifikationsfehler, der nur halb so groß ist, jedoch verlängert sich die Konvergenzzeit zu den wahren Parametern. Durch die Anpassung von  $\eta$  wird das Regelverhalten in der transienten Phase im Fall „beständige Anregung“ beruhigt und dadurch verbessert, womit im Verhalten ein Übergang zum Fall „annähernd keine Anregung“ stattfindet und der Identifikationsfehler insgesamt nur noch kleinere Werte annimmt. Die Konvergenzzeiten verlängern sich jedoch in beiden Fällen.

Durch die Wahl  $\eta \ll 1$  wird verhindert, dass sich durch eine beständige Anregung ein bereits gut gewählter Schätzwert der Parameter zu sehr verändert. Liegt beispielsweise Untermodellierung oder verrauchte Messsignale vor, würde der sich dadurch verändernde Regressionsvektor nicht das korrekte Systemverhalten widerspiegeln und folglich zu einer falschen Änderung der Parameterschätzwerke führen. Das daraus resultierende schlechtere bzw. unruhigere Regelverhalten wird durch eine Schrittweite  $\eta \ll 1$  unterbunden, da nur wenig in die neue Richtung

projiziert wird. Diese kleine Änderung der Schätzwerte erlaubt jedoch im Mittel eine Annäherung an die wahren Parameter, welche zwar langsamer erreicht werden, das System sich jedoch beruhigt zeigt. Somit kommt die Wahl  $\eta < 1$  einer Filterung der Schätzwerte gleich, d.h. die Parameteradaption weist ein beruhigtes Verhalten auf. Damit, wie im Folgenden bewiesen wird, die Stabilität des Systems trotz der Anpassung der Schrittweite  $\eta$  gewährleistet bleibt, sind die Grenzen in (13.37) einzuhalten. Die modifizierte Form des *Projektionsalgorithmus* lautet:

$$\begin{aligned} \underline{\hat{\theta}}[k] &= \\ &= \underline{\hat{\theta}}[k-1] + \eta \cdot \frac{\underline{x}[k-\delta+1]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \left[ y[k] - \underline{\hat{\theta}}[k-1]^T \underline{x}[k-\delta+1] \right] \\ &= \underline{\hat{\theta}}[k-1] + \eta \cdot \frac{\underline{x}[k-\delta+1]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \cdot \varepsilon[k] \end{aligned} \quad (13.38)$$

Es folgen nun wichtige Beweise (vgl. [68]), deren Relevanz sich in der Stabilitätsuntersuchung des geschlossenen adaptiven Regelkreises zeigen wird. Für die Untersuchung des linearen Projektionsalgorithmus auf Stabilität muss auf eine Methode zurückgegriffen werden, die auch für zeitvariante bzw. nichtlineare Systeme verwendet werden kann. Damit dürfen die Stabilitätsaussagen der Identifikation auch für die gesamtheitliche Stabilitätsuntersuchung des zeitvarianten adaptiven Reglers in Kapitel 13.3.2 Anwendung finden. Aus diesem Grund wird die Stabilitätsanalyse nun mit der nichtlinearen Lyapunov-Methode durchgeführt, welche für tiefergehende Betrachtungen näher in Kapitel 14.4.4 beschrieben wird.

Der Projektionsalgorithmus (13.38) wird für das weitere Vorgehen umgeschrieben, so dass eine Untersuchung des Betrages des Parameterfehlervektors  $\underline{\hat{\theta}}[k] = \underline{\theta}_0 - \underline{\hat{\theta}}[k]$  möglich ist; dazu subtrahiert man auf jeder Seite  $\underline{\theta}_0$ , multipliziert die Gleichung mit  $-1$  und stellt die Lyapunov-Funktion

$$V[k] = \underline{\hat{\theta}}[k]^T \underline{\hat{\theta}}[k] = \left\| \underline{\hat{\theta}}[k] \right\|^2 \quad (13.39)$$

auf:

$$\begin{aligned} V[k] &= \left( \underline{\hat{\theta}}[k-1] - \eta \cdot \frac{\underline{x}[k-\delta+1] \cdot \varepsilon[k]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \right)^T \cdot \\ &\quad \cdot \left( \underline{\hat{\theta}}[k-1] - \eta \cdot \frac{\underline{x}[k-\delta+1] \cdot \varepsilon[k]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \right) \\ &= \underbrace{\underline{\hat{\theta}}[k-1]^T \underline{\hat{\theta}}[k-1]}_{V[k-1]} - 2\eta \cdot \overbrace{\frac{\underline{\hat{\theta}}[k-1]^T \underline{x}[k-\delta+1] \cdot \varepsilon[k]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]}}^{\varepsilon[k]} + \\ &\quad + \eta^2 \cdot \frac{\underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1] \cdot \varepsilon[k]^2}{(c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1])^2} \end{aligned}$$

Es resultiert die Differenzen-Gleichung der Lyapunov-Funktion:

$$\begin{aligned}
 V[k] - V[k-1] &= \|\tilde{\theta}[k]\|^2 - \|\tilde{\theta}[k-1]\|^2 \\
 &= \underbrace{\eta \left( -2 + \eta \cdot \frac{\underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \right)}_{\kappa[k-\delta+1]} \cdot \underbrace{\frac{\varepsilon[k]^2}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]}}_{\geq 0} \quad (13.40)
 \end{aligned}$$

$$\kappa[\cdot] < 0 \quad \text{für } 0 < \eta < 2 \quad \wedge \quad c > 0 \quad (13.41)$$

Die rechte Seite der Gleichung (13.40) ist unter der Bedingung (13.41) stets negativ oder Null. Daraus folgt, dass die Differenzen-Gleichung der Lyapunov-Funktion  $V[k] - V[k-1]$  unter der Bedingung (13.41) negativ semidefinit ist. Dies hat zur Konsequenz, dass der Parameterfehler  $\|\tilde{\theta}[\cdot]\|$  im Laufe der Identifikation stets gleich bleibt oder abnimmt – der Parameterfehler wächst unter keinen Umständen an, selbst wenn der Regressionsvektor, d.h. sämtliche Ein- und Ausgänge unbeschränkt anwachsen:

$$\|\tilde{\theta}[k]\| \leq \|\tilde{\theta}[k-1]\| \leq \|\tilde{\theta}[0]\| \quad \forall k \in \mathbb{N} \quad (13.42)$$

Damit ist die bereits geometrisch getroffene Aussage in Gleichung (13.36) bestätigt. Ebenso ist gezeigt, dass die Einführung der Schrittweite  $\eta$  und der Konstante  $c$  keine destabilisierende Eigenschaft mit sich führt.

Summiert man alle Beträge  $\|\tilde{\theta}[\cdot]\|$  des Parameterfehlervektors von Beginn an ( $k = 0$ ) bis zum Zeitpunkt  $k$  auf, so ergibt sich folgende Gleichung:

$$\|\tilde{\theta}[k]\|^2 = \|\tilde{\theta}[0]\|^2 + \sum_{i=1}^k \kappa[i-\delta+1] \cdot \frac{\varepsilon[i]^2}{c + \underline{x}[i-\delta+1]^T \underline{x}[i-\delta+1]} \quad (13.43)$$

Mit der Tatsache, dass die Norm eines Vektors ( $\|\tilde{\theta}[k]\|^2 = \tilde{\theta}^T[k] \tilde{\theta}[k] \geq 0$ ) nach Definition nie negativ sein kann, folgt mit der Bedingung (13.41), dass der zweite Summand der Gleichung (13.43) beschränkt sein muss:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \frac{\varepsilon[i]^2}{c + \underline{x}[i-\delta+1]^T \underline{x}[i-\delta+1]} < \infty \quad (13.44)$$

Das hat zur Konsequenz, dass der Identifikationsfehler  $\varepsilon[\cdot]$ , sollte er tatsächlich unbegrenzt anwachsen können, nie schneller anwächst, als der Betrag des Regressionsvektors  $\underline{x}[\cdot]$ :

$$|\varepsilon[k]| = O \left[ \sup_{\kappa \leq k-\delta+1} \|\underline{x}[\kappa]\| \right] \quad (13.45)$$

Diese Erkenntnis wird bezüglich der Stabilitätsuntersuchung des geschlossenen Regelkreises von tragender Bedeutung sein.

Aus Gleichung (13.44) folgt direkt:

$$\lim_{k \rightarrow \infty} \frac{\varepsilon[k]}{\sqrt{c + \underline{x}[k - \delta + 1]^T \underline{x}[k - \delta + 1]}} = 0 \quad (13.46)$$

Hiermit ist gezeigt, dass bei beschränktem Regressionsvektor der Identifikationsfehler  $\varepsilon[\cdot]$  stets nach einer beschränkten Zeit zu Null wird, unabhängig von der Art der Anregung des Systems über den Regressionsvektor – sowohl für den Fall „annähernd keine Anregung“ als auch den Fall „beständige Anregung“ stellt sich ein Identifikationsfehler Null ein. Dies bestätigt die vorangegangenen Überlegungen zum Identifikationsalgorithmus: sobald die angeregte Systemdynamik identifiziert ist, kommt der Identifikationsprozess zum Erliegen ( $\varepsilon[\cdot] = 0$ ). Wird ein weiterer noch nicht identifizierter Teil der Systemdynamik angeregt, ergibt sich erneut ein Identifikationsfehler ungleich Null, bis nach einer bestimmten Identifikationszeit dieser wieder zu Null wird.  $\varepsilon[\cdot] = 0$  besagt somit nicht, dass der wahre Parametervektor  $\underline{\theta}_0$  gefunden wurde. Dies ist erst der Fall, wenn die vollständige Systemdynamik angeregt wurde, d.h. effektiv eine beständige Anregung des Systems stattfand. An dieser Stelle ist weiter anzumerken, dass bei Untermodellierung des Systems oder bei verrauschten Messsignalen eine Konvergenz des geschätzten Parametervektors  $\hat{\underline{\theta}}[\cdot]$  zu konstanten Parametern verhindert wird und folglich für den Identifikationsfehler in Gleichung (13.46) nur  $\varepsilon[\cdot] \approx 0$  gilt. Wie zuvor beschrieben, kann über die Schrittweite mit  $\eta < 1$  der Identifikationsprozess jedoch beruhigt werden, sodass trotz Untermodellierung oder Rauschen  $\varepsilon[\cdot] \rightarrow 0$  mit  $\eta \rightarrow 0$  erreicht werden kann. Im Anschluss wird bewiesen, dass für  $\varepsilon[\cdot] \rightarrow 0$  eine Konvergenz des geschätzten Parametervektors  $\hat{\underline{\theta}}[\cdot]$  zu konstanten Parametern stattfindet:

Bildet man den Betrag des Abstandsvektors zweier zeitlich benachbarter Schätzwerte über die Gleichung (13.38)

$$\begin{aligned} \|\hat{\underline{\theta}}[k] - \hat{\underline{\theta}}[k-1]\| &= \left\| \eta \cdot \frac{\underline{x}[k-\delta+1] \cdot \varepsilon[k]}{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]} \right\| \\ &= \sqrt{\eta^2 \cdot \frac{\underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1] \cdot \varepsilon[k]^2}{(c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1])^2}} \end{aligned}$$

und stellt des Weiteren fest, dass

$$\begin{aligned} \sqrt{\frac{\underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1] \cdot \varepsilon[k]^2}{(c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1])^2}} &< \sqrt{\frac{(c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]) \cdot \varepsilon[k]^2}{(c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1])^2}} \\ &= \frac{\varepsilon[k]}{\sqrt{c + \underline{x}[k-\delta+1]^T \underline{x}[k-\delta+1]}} \end{aligned}$$

gilt, so kann mit Hilfe der Gleichung (13.44) folgende Aussage getroffen werden:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\| \hat{\theta}[i] - \hat{\theta}[i-1] \right\| < \infty$$

Wird die Cauchy-Schwarz-Ungleichung

$$\begin{aligned} & \left\| \hat{\theta}[i] - \hat{\theta}[i-\delta] \right\|^2 = \\ &= \left\| \hat{\theta}[i] - \hat{\theta}[i-1] + \hat{\theta}[i-1] - \hat{\theta}[i-2] + \cdots + \hat{\theta}[i-\delta+1] - \hat{\theta}[i-\delta] \right\|^2 \\ &\leq \left\| \hat{\theta}[i] - \hat{\theta}[i-1] \right\|^2 + \left\| \hat{\theta}[i-1] - \hat{\theta}[i-2] \right\|^2 + \cdots + \left\| \hat{\theta}[i-\delta+1] - \hat{\theta}[i-\delta] \right\|^2 \end{aligned}$$

darauf angewendet, resultiert

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \left\| \hat{\theta}[i] - \hat{\theta}[i-\delta] \right\| < \infty$$

und daraus die für die spätere Regelkreisstabilitätsuntersuchung wichtige Gleichung

$$\lim_{k \rightarrow \infty} \left\| \hat{\theta}[k] - \hat{\theta}[k-\delta] \right\| = 0 \quad (13.47)$$

Diese Gleichung besagt, dass der Identifikationsprozess nach einer beschränkten Zeit zum Erliegen kommen wird und sich ein konstanter geschätzter Parametervektor  $\hat{\theta}[\cdot]$  einstellt, der nicht dem wahren Parametervektor  $\underline{\theta}_0$  entsprechen muss. Gemäß der Erklärung zu Gleichung (13.46) wird keine Aussage über die Konvergenz zu den wahren Parameterwerten getroffen.

Der Grundgedanke der adaptiven Regelung konnte mit dem Projektionsalgorithmus als zentrales Element sehr gut und anschaulich erklärt werden; auch die für den Stabilitätsbeweis notwendigen Gleichungen waren sehr einfach aufzustellen und geometrisch zu interpretieren, womit der Projektionsalgorithmus durch seine Einfachheit und Anschaulichkeit überzeugt.

Allgemein gilt, dass eine *schnelle* Konvergenz zu einem konstanten Parametersatz bzw. zu den wahren Parametern für eine stabile adaptive Regelung nicht notwendig ist, jedoch ergeben sich nur Vorteile, wenn die Parameterwerte möglichst schnell konstant sind bzw. den wahren Werten entsprechen. Je schneller die Konvergenz zu einem konstanten Parametersatz, desto schneller wird der Identifikationsfehler und somit der Regelfehler zu Null, d.h. die transienten Phasen verkürzen sich. Auch ist zu bedenken, dass sich die Konvergenzzeiten weiter erhöhen, sobald ein System hoher Ordnung mit vielen zu schätzenden Parametern vorliegt. Sind zudem Störungen gegenwärtig, so verliert der Projektionsalgorithmus auf Grund seiner geringen Konvergenzgeschwindigkeit an Bedeutung. Durch die Wahl eines kleinen  $\eta$  kann Rauschen der Systemsignale zwar geglättet werden, jedoch wiederum auf Kosten der Konvergenzgeschwindigkeit.

Ein großer Vorteil der Anwendung des Projektionsalgorithmus besteht in der ständigen Reaktionsmöglichkeit auf zeitvariante Parameterschwankungen. Ent-

sprechend der Darstellung in diesem Kapitel ist es möglich, den Identifikationsfehler zu jedem Zeitschritt zu minimieren. Dies gelingt sehr gut für Systeme mit ständig langsam variierenden Parametern (annähernd keine Anregung durch langsame Parameterschwankungen) oder für Systeme mit kurzzeitig dynamischen Parameteränderungen gefolgt von stückweise konstanten Systemparametern (beständige Anregung), so dass sich eine erfolgreiche Regelung mit einer, je nach Konvergenzeigenschaft des Identifikationsalgorithmus, kurzen bzw. längeren transienten Phase ergibt. Für ständige dynamische Parameterschwankungen steigt jedoch mit zunehmender Zeitvarianz der Identifikationsfehler pro Zeitschritt an. Es zeigt sich zwar eine beständige Anregung durch die schnellen Parameterschwankungen, da sich die wahren Parameter jedoch ebenfalls dynamisch verändern, muss der Identifikationsalgorithmus eine höhere Dynamik als die Parameterschwankungen aufweisen, um ein Nachführen der Parameter noch gewährleisten zu können. Auf Grund der langsamen Konvergenzeigenschaft des Projektionsalgorithmus kann bei ständig dynamisch schwankenden Systemparametern daher keine stabile Regelung mehr gewährleistet werden.

Eine Lösung der dargestellten Probleme bietet der sehr effiziente rekursive Least-Squares-Algorithmus, der im Vergleich zum Projektionsalgorithmus eine sehr hohe Konvergenzgeschwindigkeit ermöglicht.

### 13.2.2 Rekursiver Least-Squares-Algorithmus (RLS)

Das Prinzip des RLS-Algorithmus kann mit dem sog. *orthogonalisierten Projektionsalgorithmus*

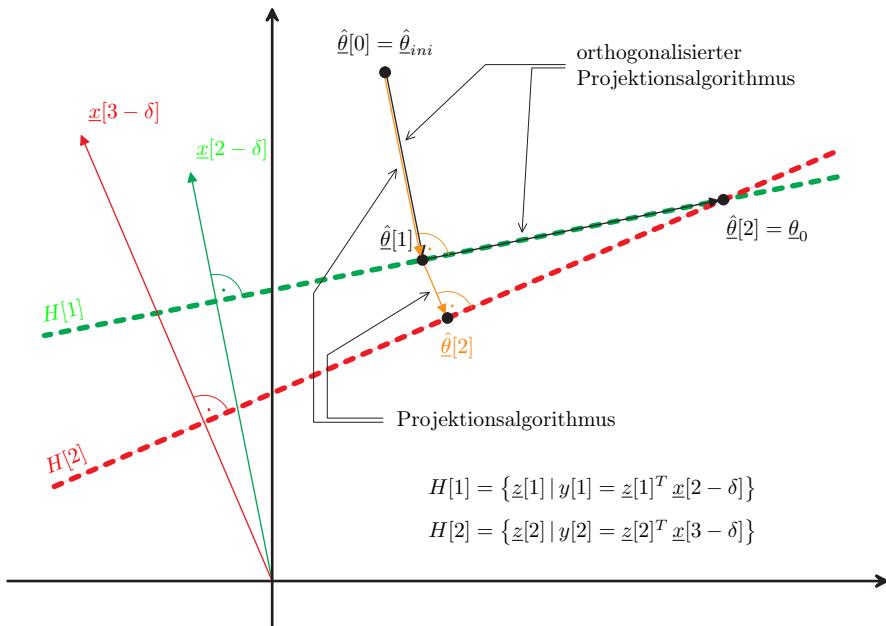
$$\hat{\theta}[k] = \hat{\theta}[k-1] + \frac{\mathbf{P}[k-\delta] \underline{x}[k-\delta+1]}{\underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \underline{x}[k-\delta+1]} \cdot \varepsilon[k] \quad (13.48)$$

$$\mathbf{P}[k-\delta+1] = \mathbf{P}[k-\delta] - \frac{\mathbf{P}[k-\delta] \underline{x}[k-\delta+1] \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta]}{\underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \underline{x}[k-\delta+1]} \quad (13.49)$$

erklärt werden. Es fällt die Ähnlichkeit der Form zum Projektionsalgorithmus (13.38) auf, wobei sich nun eine mehrdimensionale Matrix  $\mathbf{P}$  an der Stelle der Konstanten  $\eta$  befindet und der Spezialfall  $c = 0$  gilt. Da die Schrittweite  $\eta$  jetzt mehrdimensional ist, kann eine elementweise Gewichtung des Regressionsvektors durchgeführt werden, womit eine größere Freiheit einer Projektion interpretiert werden kann – dies macht eine schnellere Konvergenz plausibel. Mit  $c = 0$  findet, wie bei der in Abbildung 13.1 dargestellten Grundform (13.32) des Projektionsalgorithmus, eine exakte Projektion statt.

Bezieht man sich noch einmal auf die geometrische Darstellung des Projektionsalgorithmus in Abbildung 13.1, so ist vorstellbar, dass  $\hat{\theta}[k+1]$  mit  $\theta_0$  zusammenfällt, wenn  $\underline{x}[k-\delta+2]$  zufällig orthogonal zu  $\underline{x}[k-\delta+1]$  ist –  $\hat{\theta}_k$  wird dann direkt auf den Schnittpunkt der Ebenen, d.h.  $\theta_0$  projiziert. Dies würde im zweidimensionalen bzw. allgemeinen Fall eine Konvergenz nach 2 bzw.  $n$  Schritten bedeuten. Man könnte demnach eine beträchtliche Verbesserung des Projektionsalgorithmus erreichen, wenn in eine Richtung projiziert wird, die orthogonal zu den

letzten Regressionsvektoren ist. Letztendlich ist es die Aufgabe des orthogonalisierten Projektionsalgorithmus, sequentiell bzw. rekursiv ein Gleichungssystem zur Bestimmung dieser Projektionsrichtung mit Betrag zu lösen. Bei  $n$  Unbekannten des Parametervektors  $\hat{\theta}[\cdot]$  bedarf es  $n$  linear unabhängiger Gleichungen und somit Regressionsvektoren, um die wahren Werte von  $\theta_0$  durch eine Linearkombination von geschätzten Parametervektoren  $\hat{\theta}[\cdot]$  erreichen zu können, d.h. bezüglich der Konvergenzgeschwindigkeit zu den wahren Werten ist festzustellen, dass so viele Zeitschritte vergehen müssen, bis  $n$  linear unabhängige Regressionsvektoren genutzt werden konnten. Bezieht man sich auf die Gleichung (13.48), so repräsentiert der Vektor  $\mathbf{P}[k-\delta] \underline{x}[k-\delta+1]$  die Komponente des Regressionsvektors  $\underline{x}[k-\delta+1]$ , die orthogonal zu den letzten Regressionsvektoren steht, was in [68] bewiesen wird. Die Matrix  $\mathbf{P}[k-\delta]$  ist der dafür notwendige Projektionsoperator, der zur rekursiven Lösung des Gleichungssystems führt. Der Unterschied zwischen Projektionsalgorithmus und orthogonalisiertem Projektionsalgorithmus ist in Abbildung 13.2 veranschaulicht.



**Abb. 13.2:** Vergleich zwischen Projektionsalgorithmus und orthogonalisiertem Projektionsalgorithmus bezogen auf ein System erster Ordnung ( $n = 1$ ,  $\delta = 1$ ) mit zweidimensionalem Parametervektor

Damit der orthogonale Projektionsalgorithmus (13.48) entsprechend der Diskussion beim Projektionsalgorithmus (13.38) keine Division durch Null vollzieht, wird im Nenner eine Konstante  $c > 0$  hinzugefügt:

$$\hat{\theta}[k] = \hat{\theta}[k-1] + \frac{\mathbf{P}[k-\delta] \underline{x}[k-\delta+1]}{c + \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \underline{x}[k-\delta+1]} \cdot \varepsilon[k] \quad (13.50)$$

Hiermit wird, wie im Falle des Projektionsalgorithmus, eine exakte Projektion verhindert, was dazu führt, dass der wahre Parametervektor  $\hat{\theta}[\cdot] = \underline{\theta}_0$  bei einer beständigen Anregung noch nicht nach bereits  $n$  Schritten gefunden wird, d.h. durch eine Wahl  $c > 0$  wird der Identifikationsalgorithmus beruhigt, was besonders bei verrauschten Messsignalen sowie bei Untermodellierung von Interesse ist. Es hängt nun von der Höhe des Betrages der Initialisierungsmatrix  $\mathbf{P}_{ini}$  im Verhältnis zu der Konstanten  $c$  ab, wie lange der Identifikationsvorgang bzw. die Konvergenz zu den wahren Parametern dauert. Im Folgenden wird angenommen, dass  $c = 1$  gilt. Werden nun sehr große Beträge für die Elemente der Matrix  $\mathbf{P}_{ini}$  gewählt, ist die Addition der Konstanten  $c$  im Nenner des Algorithmus (13.50) vernachlässigbar, so dass sich ein Verhalten entsprechend des oben diskutierten orthogonalen Projektionsalgorithmus zeigen wird. Je kleiner jedoch der Betrag der Initialisierungsmatrix  $\mathbf{P}_{ini}$ , desto mehr ergibt sich eine Abweichung zum orthogonalen Projektionsalgorithmus bzgl. der Konvergenzgeschwindigkeit – es resultiert ein beruhigter, Rauschsignal-optimierter Identifikationsalgorithmus. Da die Matrix  $\mathbf{P}[\cdot]$  unabhängig von der Initialisierung  $\mathbf{P}_{ini}$  mit Verlauf des Identifikationsvorganges gemäß Gleichung (13.49) stetig abnimmt, findet stets ein Übergang von einem zunächst mehr oder weniger schnellem Identifikationsvorgang (beeinflusst durch  $\mathbf{P}_{ini}$ ) zu einem Rauschsignal-optimierten Identifikationsvorgang statt, sobald  $c > 0$  gewählt wird.

Durch diese Darstellung wird deutlich, dass die Matrix  $\mathbf{P}[\cdot]$  als mehrdimensionale Schrittweite interpretiert werden kann. Im Unterschied zu der eindimensionalen Schrittweite  $\eta$  des Projektionsalgorithmus nimmt die Schrittweite, beginnend bei der durch  $\mathbf{P}_{ini}$  vorgegebenen Schrittweite, mit zunehmender Zeit ab, womit eine stärker werdende Filterung einhergeht. Liegt bereits ein guter Schätzwert  $\hat{\theta}[0] = \hat{\theta}_{ini} \approx \underline{\theta}_0$  vor, so gewährleistet die Initialisierung von  $\mathbf{P}_{ini}$  mit kleinen Werten, d.h. mit einer kleinen Schrittweite, dass der Identifikationsvorgang dieses Vorwissen nutzt und den Initialisierungswert  $\hat{\theta}_{ini}$  nur langsam in Richtung des wahren Wertes adaptiert. Besteht hingegen kein Vorwissen, so sollte die Matrix  $\mathbf{P}_{ini}$  mit großen Beträgen initialisiert werden, so dass schnell durch einen zunächst aggressiven Identifikationsvorgang mit großer Schrittweite ein das System gut beschreibender Schätzwert vorliegt.

Entsprechend dieser Überlegungen entspricht der gesuchte schnelle Rauschsignal-optimierte RLS-Algorithmus einem orthogonalen Projektionsalgorithmus mit  $c = \text{konst.}$  Um dies zu zeigen, wird im Folgenden der RLS-Algorithmus direkt an Hand der oben dargestellten Eigenschaften abgeleitet. Es ist ein Algorithmus gesucht, der zum einen für alle vergangenen Identifikationsvorgänge nachträglich einen gemeinsamen Schätzwert bestimmt, so dass bei einer beständigen Anregung des Systems eine schnelle Konvergenz nach bereits  $n$  Schritten erreicht werden kann. Zum anderen muss der gesuchte Algorithmus auch eine Gewichtung der anfänglichen Parameterschätzwerte  $\hat{\theta}_{ini}$  berücksichtigen, so dass Vorwissen eingebracht werden kann und der Identifikationsvorgang dadurch zu

beruhigen ist. Die im ersten Punkt geforderte nachträgliche Bestimmung eines gemeinsamen Schätzwertes  $\underline{z}$  für die Parametervektoren  $\hat{\theta}[i-1]$  mit  $i \in \{1, \dots, k\}$  entspricht der Minimierung aller bisherigen Identifikationsfehler  $\varepsilon[i] \rightarrow 0$ , d.h. der neue Schätzwert  $\hat{\theta}[k] = \underline{z}$  wird wie gefordert das gesamte bisher aufgetretene Systemverhalten maximal repräsentiert. Sobald das System vollständig angeregt wurde, liegt durch dieses Vorgehen der wahre Parametervektor vor. Es ist somit eine Minimierung folgender quadratischen Kostenfunktion erforderlich:

$$J(\underline{z}) = \frac{1}{2} \sum_{i=1}^k \left( \underbrace{y[i] - \underline{z}^T \underline{x}[i-\delta+1]}_{\varepsilon[i]} \right)^2 + \frac{1}{2} (\underline{z} - \hat{\theta}_{ini})^T \mathbf{P}_{ini}^{-1} (\underline{z} - \hat{\theta}_{ini}) \quad (13.51)$$

Der erste Summand garantiert, dass bei einer beständigen Anregung des Systems bereits nach  $n$  Schritten der wahre Parameterwert  $\hat{\theta}[n] = \theta_0$  gefunden wird. Der zweite Summand wirkt dem durch die Wahl einer Matrix  $\mathbf{P}_{ini}$  mit kleinen Beträgen entgegen und führt zu einer stärkeren Gewichtung des Anfangswertes  $\hat{\theta}_{ini}$ , wodurch eine längere Konvergenzzeit und ein beruhigter Identifikationsvorgang resultiert – gut gewählte Anfangswerte  $\hat{\theta}_{ini} \approx \theta_0$  werden durch den Algorithmus nur angepasst und nicht verworfen, womit Systemwissen eingebracht werden kann. Durch die Wahl sehr großer Beträge für die Matrix  $\mathbf{P}_{ini}$  ist der zweite Summand zu vernachlässigen, weshalb die Parameteridentifikation maximal schnell zu den wahren Werten konvergieren wird – hierbei ist die Wahl geeigneter Anfangswerte  $\hat{\theta}_{ini}$  irrelevant, womit diese beliebig gesetzt werden können.

Die Durchführung der Minimierung (13.51) zeigt das Kapitel 4.2 und ergibt den sog. Recursive-Least-Squares-Algorithmus (RLS) für das ARMA-Modell (13.20); es sei diesbezüglich auch auf die Literatur [68], [223] und [10] verwiesen. Der RLS-Algorithmus zeichnet sich, wie erwartet, durch hohe Konvergenzgeschwindigkeit sowie Robustheit gegenüber Störgrößen aus. Dieser in Kapitel 4.2.2 ausführlich beschriebene *RLS-Algorithmus* ist für die Anwendung auf das Prädiktor-ARMA-Modell (13.24) bzw. (13.22) mit dem Relativgrad  $\delta$  wie folgt umzuschreiben:

$$\hat{\theta}[k] = \hat{\theta}[k-1] + \mathbf{P}[k-\delta+1] \underline{x}[k-\delta+1] \cdot \varepsilon[k] \quad (13.52)$$

$$= \hat{\theta}[k-1] + \underbrace{\frac{\mathbf{P}[k-\delta] \underline{x}[k-\delta+1]}{1 + \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \underline{x}[k-\delta+1]}}_{\gamma[k-\delta]} \cdot \varepsilon[k] \quad (13.53)$$

$$\begin{aligned} \mathbf{P}[k-\delta+1] &= \mathbf{P}[k-\delta] - \frac{\mathbf{P}[k-\delta] \underline{x}[k-\delta+1] \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta]}{1 + \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \underline{x}[k-\delta+1]} \\ &= \mathbf{P}[k-\delta] - \gamma[k-\delta] \underline{x}[k-\delta+1]^T \mathbf{P}[k-\delta] \end{aligned} \quad (13.54)$$

Zu Beginn der Identifikation ( $k = 0$ ) muss als Startwert des Parametervektors  $\hat{\theta}[0] = \hat{\theta}_{ini}$  gesetzt und die  $\mathbf{P}$ -Matrix mit einer beliebigen positiv definiten Matrix  $\mathbf{P}_{ini}$  (z.B. Diagonalmatrix) initialisiert werden. Der RLS-Algorithmus (13.53)

entspricht für  $\delta = 1$  nach Verschiebung um einen Zeitschritt exakt dem abgeleiteten Algorithmus (4.76), wobei der Identifikationsfehler  $\varepsilon[\cdot]$ , definiert in Gleichung (13.27), als der sog. Korrekturterm bezeichnet wird. Ebenso findet sich die Rekursionsformel (13.54) in Gleichung (4.77) wieder.

Mit der über Kapitel 4.2.2 abgeleiteten Gleichung (13.53) ist zu erkennen, dass der RLS-Algorithmus mit  $c = 1$  tatsächlich ein Spezialfall des oben beschriebenen orthogonalen Projektionsalgorithmus (13.50) ist. Durch den mit  $c \neq 0$  beruhigten orthogonalen Projektionsalgorithmus, dem sog. RLS-Algorithmus, wird sich schnell ein gutes und beruhigtes Regelergebnis des im folgenden Kapitel 13.3 abzuleitenden adaptiven Reglers zeigen. Für den Nachweis der Stabilität des adaptiven Reglers in Kombination mit dem RLS-Algorithmus im geschlossenen Regelkreis wird, wie beim Projektionsalgorithmus, auf dessen Eigenschaften zurückgegriffen.

Da es sich bei dem RLS-Algorithmus, wie gezeigt, um einen schnellen und Rauschsignal-optimierten Projektionsalgorithmus handelt, ist es nicht verwunderlich, dass die mathematische Untersuchung des RLS-Algorithmus (vgl. [68]) zu annähernd denselben für die Stabilitätsanalyse notwendigen Gleichungen führt, wie die des Projektionsalgorithmus. Als Basis der Untersuchung dient gemäß Gleichung (13.39) eine Lyapunov-Funktion der Form

$$V[k] = \tilde{\underline{\theta}}^T[k] \mathbf{P}[k]^{-1} \tilde{\underline{\theta}}[k] \quad (13.55)$$

Anstelle der Gleichung (13.42) ergibt sich

$$\left\| \tilde{\underline{\theta}}[k] \right\|^2 \leq \kappa_1 \left\| \tilde{\underline{\theta}}[0] \right\|^2 \quad \text{für } \forall k > 0 \quad (13.56)$$

$$\text{mit } \kappa_1 = \frac{\lambda_{\max}(\mathbf{P}_{\text{ini}})^{-1}}{\lambda_{\min}(\mathbf{P}_{\text{ini}})^{-1}}$$

$\lambda_{\min, \max}$  : kleinster, größter Eigenwert

was besagt, dass der Betrag des Parameterfehlervektors stets abnimmt oder gleich bleibt, d.h. stets beschränkt bleibt. Analog zu Gleichung (13.44) bzw. (13.46) resultiert

$$\lim_{k \rightarrow \infty} \sum_{i=1}^k \frac{\varepsilon[i]^2}{1 + \underline{x}[i - \delta + 1]^T \mathbf{P}[i - \delta] \underline{x}[i - \delta + 1]} < \infty \quad (13.57)$$

bzw.

$$\lim_{k \rightarrow \infty} \frac{\varepsilon[k]}{\sqrt{c + \underline{x}[k - \delta + 1]^T \mathbf{P}[i - \delta] \underline{x}[k - \delta + 1]}} = 0 \quad (13.58)$$

Dies hat, wie im Falle des Projektionsalgorithmus, zur Konsequenz, dass der Identifikationsfehler  $\varepsilon[\cdot]$  – sollte er tatsächlich unbegrenzt anwachsen können – nie schneller anwächst, als der Betrag des Regressionsvektors  $\underline{x}[\cdot]$ :

$$|\varepsilon[k]| = O \left[ \sup_{\kappa \leq k - \delta + 1} \|\underline{x}[\kappa]\| \right] \quad (13.59)$$

Und schließlich ist Gleichung (13.47) mit

$$\lim_{k \rightarrow \infty} \left\| \hat{\theta}[k] - \hat{\theta}[k - \delta] \right\| = 0 \quad (13.60)$$

identisch. Somit wird auch beim RLS-Algorithmus der Identifikationsprozess nach einer beschränkten Zeit zum Erliegen kommen und sich ein konstanter geschätzter Parametervektor  $\hat{\theta}[\cdot]$  einstellen, der nicht dem wahren Parametervektor  $\underline{\theta}_0$  entsprechen muss.

Die Konsequenz obiger Aussagen ist, dass beide Algorithmen für die Stabilitätsanalyse des adaptiven Reglers im Folgenden gleich behandelt werden können. Es wird sich zeigen, dass beide Schätzalgorithmen im adaptiven Regler Stabilität gewähren und somit als zentrales Element des adaptiven Konzeptes angewandt werden dürfen.

### 13.3 Entwurf des adaptiven Regelkreises

Um ein System klassisch regeln zu können, müssen die Parameter des Systems bekannt sein – nur dann kann eine Reglereinstellung erfolgen. Sind die Systemparameter jedoch unbekannt, wäre es naheliegend, die Parameter zu jedem Zeitschritt entsprechend der vorgestellten stabilen Algorithmen zu identifizieren, um mit den erhaltenen Werten einen Regler zu jedem Zeitschritt stabil auslegen zu können. Mit diesem Vorgehen besteht jedoch die Gefahr eines instabilen Verhaltens des Regelkreises – es ist nicht gewährleistet, dass das Gesamtsystem, bestehend aus einer stabilen Identifikation und einem zu jedem Zeitschritt stabil ausgelegten Regler, ebenfalls stabil ist. Da sich die Systemparameter durch die Identifikation zu jedem Zeitschritt ändern, entsteht durch die Zeitvarianz eine mögliche destabilisierende Dynamik, die mit linearen Stabilitätsuntersuchungen nicht bestimmt werden kann. Dies wird an Hand des Beispiels von Vinograd [122] deutlich, bei dem ein rein mathematisch begründetes System präsentiert wird, welches zeitvariant ist und stabile Eigenwerte liefert, aber dennoch Instabilität zeigt. Das entsprechende System

$$\dot{\underline{x}}(t) = \mathbf{A}(t) \underline{x}(t) \quad \text{mit}$$

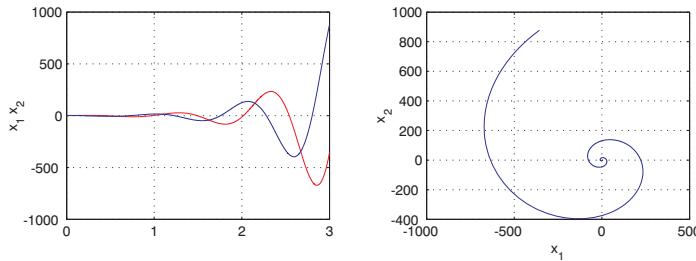
$$\mathbf{A}(t) = \begin{bmatrix} -1 - 9 \cos^2(6t) + 6 \sin(12t) & 12 \cos^2(6t) + 4.5 \sin(12t) \\ -12 \sin^2(6t) + 4.5 \sin(12t) & -1 - 9 \sin^2(6t) - 6 \sin(12t) \end{bmatrix}$$

besitzt eine zeitvariante Systemmatrix  $\mathbf{A}(\cdot)$ . Die Eigenwertuntersuchung

$$\det(\mathbf{A} - \lambda \mathbf{E}) = \lambda^2 + 11\lambda + 10$$

führt zu den beiden stabilen Eigenwerten

$$\lambda_1 = -1, \lambda_2 = -10$$



**Abb. 13.3:** Vinograd: Stabilitätsverhalten eines zeitvarianten Systems mit stabilen Eigenwerten

Die Simulation 13.3 zeigt jedoch ein instabiles System.

Man kommt zu dem Resultat, dass Regelkreis und Identifikation nicht getrennt voneinander betrachtet, d.h. keiner separaten Stabilitätsanalyse unterzogen werden dürfen. Es ist nun das Ziel, eine Kombination von Regler und Identifikation zu finden, die zu einem stabilen Gesamtregelkreis führt. Hierfür muss eine nichtlineare Stabilitätsuntersuchung Anwendung finden.

### 13.3.1 Inverser Regler mit integrierter Systemidentifikation

Anhand des letzten Kapitels wurde deutlich, dass durch den Identifikationsvorgang jede angeregte Systemdynamik stets vollständig identifiziert wird. Dies bestätigt die Gleichung (13.47) des Projektionsalgorithmus bzw. (13.60) des RLS-Algorithmus mit der Konvergenz des Parametervektors  $\hat{\theta}[\cdot]$  gegen einen konstanten Parametersatz, der das momentan angeregte Systemverhalten mit einem Identifikationsfehler  $\varepsilon[\cdot] = 0$  bei beschränktem Regressionsvektor gemäß der Gleichung (13.46) bzw. (13.58) eindeutig repräsentiert. Der konvergierte Parametersatz  $\hat{\theta}[\cdot] = \text{konst}$  muss jedoch nicht den wahren Systemparametern  $\theta_0$  entsprechen, denn sobald ein Teil der Systemdynamik nicht angeregt wird, kann diese auch nicht identifiziert werden. Es ist durch den Schätzvorgang nur so viel Systemverhalten zu lernen, wie an Hand des Ein- Ausgangsverhaltens durch die Systemanregung sichtbar wird. Da für eine Regelung nur das Systemverhalten bekannt sein muss, welches durch das Sollsignal angeregt wird, liegt die Vermutung nahe, dass für einen erfolgreichen stabilen Regelvorgang nicht der wahre Parametervektor  $\theta_0$  bekannt sein muss – vielmehr scheint ein Identifikationsfehler  $\varepsilon[\cdot] = 0$  hierfür ausreichend, was im Folgenden bewiesen wird.

Auf der Grundlage dieses sog. *Certainty-Equivalence Prinzips*, dass nach einem transienten Identifikationsvorgang stets ein Parametervektor vorliegt, der nicht den wahren Parameterwerten des Systems entsprechen muss, aber trotzdem das momentan angeregte Systemverhalten eindeutig beschreibt, kann ein Regelgesetz entworfen werden; das dafür notwendige Prädiktor-ARMA-Modell bzw. Streckenmodell lautet nach Gleichung (13.22) bzw. (13.25):

$$\begin{aligned}
y[k] &= \underline{\theta}_0^T \underline{x}[k - \delta + 1] = \\
&= -\alpha_{n-1} \cdot y[k - \delta] - \dots - \alpha_0 \cdot y[k - n + 1 - \delta] + \dots \\
&\quad \dots + \beta_{n-1} \cdot u[k - \delta] + \dots + \beta_0 \cdot u[k - n + 1 - \delta]
\end{aligned}$$

Zum Zeitpunkt  $k$  ist der Regressionsvektor  $\underline{x}[k - \delta + 1]$  und  $y[k]$  bekannt, womit die Schätzung des Parametervektors  $\underline{\theta}_0$  mit  $\hat{\underline{\theta}}[k]$  durchgeführt werden kann:

$$\begin{aligned}
y[k] &= \underline{\theta}_0^T \underline{x}[k - \delta + 1] = \hat{\underline{\theta}}[k]^T \underline{x}[k - \delta + 1] \quad (13.61) \\
\rightarrow \hat{\underline{\theta}}[k] &: \hat{\alpha}_{n-1}[k], \dots, \hat{\alpha}_0[k], \hat{\beta}_{n-1}[k], \dots, \hat{\beta}_0[k]
\end{aligned}$$

Nimmt man an, dass das oben beschriebene Certainty-Equivalence Prinzip mit  $\hat{\underline{\theta}}[k] \stackrel{!}{=} \hat{\underline{\theta}}[k + \delta]$  gemäß Gleichung (13.47) bzw. (13.60) nach einer transienten Phase anwendbar ist, so sind für den Schritt  $k + \delta$  nach entsprechender Verschiebung der Gleichung (13.61)

$$y[k + \delta] = \underline{\theta}_0^T \underline{x}[k + 1] = \hat{\underline{\theta}}[k + \delta]^T \underline{x}[k + 1] \stackrel{!}{=} \hat{\underline{\theta}}[k]^T \underline{x}[k + 1] \quad (13.62)$$

alle Parameter und Signalwerte bekannt, bis auf den zukünftigen Ausgangswert  $y[k + \delta]$  und den zum Zeitschritt  $k$  auszugebenden, gesuchten Stellwert  $u[k]$ :

$$\begin{aligned}
y[\color{red}{k + \delta}] &= -\alpha_{n-1} \cdot y[k] - \dots - \alpha_0 \cdot y[k - n + 1] + \dots \quad (13.63) \\
&\quad \dots + \beta_{n-1} \cdot \color{red}{u[k]} + \dots + \beta_0 \cdot u[k - n + 1] \\
&\stackrel{!}{=} -\hat{\alpha}_{n-1}[k] \cdot y[k] - \dots - \hat{\alpha}_0[k] \cdot y[k - n + 1] + \dots \\
&\quad \dots + \hat{\beta}_{n-1}[k] \cdot \color{red}{u[k]} + \dots + \hat{\beta}_0[k] \cdot u[k - n + 1]
\end{aligned}$$

Über  $y[k + \delta]$  und  $u[k]$  kann nun das Verhalten des geschlossenen Regelkreises vorgegeben werden.

Innerhalb des Regelbetriebs möchte man dem System das Verhalten des Ausgangssignals  $y[\cdot]$  über das Stellsignal  $u[\cdot]$  vorgeben. Wird im strengsten Fall gefordert, dass der gewünschte Ausgangswert  $y[k + i]^*$  maximal schnell, d.h. bereits nach der systembedingten Totzeit von  $i = \delta$  Zeitschritten, dem Sollwert  $r[k]$  entspricht, ergibt sich der sog. *Minimum-Varianz-Regler*:

$$y[k + \delta]^* = r[k] \quad \leftrightarrow \quad \frac{y^*(z)}{r(z)} = \mathbf{R}(z) = \frac{1}{z^\delta} \quad (13.64)$$

Das Ziel der adaptiven Regelung ist demnach, dass sich der geschlossene Regelkreis wie ein Referenzmodell  $\mathbf{R}(z)$  verhält – man spricht daher von einem *Referenzmodell-Regler*, engl. *Model Reference Adaptive Control (MRAC)*. Als Referenzmodell für den adaptiven Regler kann jede beliebige Übertragungsfunktion mit stabilen Pol- und Nullstellen dienen, wie dies in [68], [4] sowie [240] dargestellt wird; sofern die System-Nullstellen durch Vorgabe eines entsprechenden Referenzmodells des Reglers nur erhalten bleiben und nicht verändert werden,

darf das Referenzmodell gemäß der Erläuterung in Kapitel 13.3.2 auch die instabilen Nullstellen der Strecke besitzen. Fordert man ein maximal schnelles Einschwingen des Reglers, ergibt sich der Spezialfall (13.64) des adaptiven Minimum-Varianz-Reglers nach [68] bzw. [239], bei dem das Referenzmodell lediglich einem Verzögerer  $\delta$ -ten Ordnung entspricht.

Es ist nun zum Zeitschritt  $k$  ein Stellsignal  $u[\cdot]$  zu finden, welches die Forderung (13.64) am Beispiel des Minimum-Varianz-Reglers erfüllt. Auf Grund des beschriebenen Certainty-Equivalence Prinzips kann das Ein- Ausgangsverhalten zum Zeitschritt  $k + \delta$  als bekannt angenommen werden – damit lässt sich unter der Forderung (13.64) mit  $y[k + \delta] = y[k + \delta]^* = r[k]$  die Modellgleichung (13.63) nach  $u[k]$  auflösen, womit der für den gewünschten Ausgangswert  $y[k + \delta]^*$  bzw. Sollwert  $r[k]$  notwendigen Stellwert  $u[k]$  bestimmt werden kann; man erhält den Reglerausgang bzw. das Regelgesetz des Minimum-Varianz-Reglers, einem inversen Regler:

$$y[k + \delta]^* \stackrel{!}{=} \hat{\theta}[k]^T \underline{x}[k + 1] \quad (13.65)$$

$$\begin{aligned} r[k] &\stackrel{!}{=} -\hat{\alpha}_{n-1}[k] \cdot y[k] - \dots - \hat{\alpha}_0[k] \cdot y[k - n + 1] + \dots \\ &\quad \dots + \hat{\beta}_{n-1}[k] \cdot u[k] + \dots + \hat{\beta}_0[k] \cdot u[k - n + 1] \end{aligned}$$

↓

$$\begin{aligned} u[k] &= \frac{1}{\hat{\beta}_{n-1}} \left[ r[k] - \left( -\hat{\alpha}_{n-1}[k] \cdot y[k] - \dots - \hat{\alpha}_0[k] \cdot y[k - n + 1] + \dots \right. \right. \\ &\quad \left. \left. \dots + \hat{\beta}_{n-2}[k] \cdot u[k - 1] + \dots + \hat{\beta}_0[k] \cdot u[k - n + 1] \right) \right] \quad (13.66) \end{aligned}$$

In Summendarstellung lautet das Regelgesetz:

$$u[k] = \frac{1}{\hat{\beta}_{n-1}} \left[ r[k] + \sum_{i=0}^{n-1} \hat{\alpha}_{n-1-i} \cdot y[k - i] - \sum_{j=1}^{n-1} \hat{\beta}_{n-1-j} \cdot u[k - j] \right] \quad (13.67)$$

An dieser Stelle zeigt sich hinsichtlich der Realisierbarkeit des Reglers die Notwendigkeit der Verwendung des Prädiktor-ARMA-Modells (13.22) an Stelle des ARMA-Modells (13.20) in seiner Grundform. Mit letzterem stünden auf der rechten Seite der Gleichung (13.67) zukünftige, d.h. unbekannte Ausgangswerte  $y[k - i]$  mit  $i \in \mathbb{Z}^-$ , weshalb das Regelgesetz, wie auf Seite 497 beschrieben, nicht kausal wäre. Die zukünftigen Ausgangswerte müssten rekursiv mit der Gleichung (13.20) ersetzt werden, was wiederum zum Prädiktor-ARMA-Modell (13.22) führt.

Die Abbildung 13.4 zeigt schematisch die Kombination von inversem Regler und Systemidentifikation.

Der entscheidende Vorteil des vorgestellten adaptiven Regelkonzeptes ist, dass die Identifikation die Reglerparameter, welche die Parameter des inversen Modells

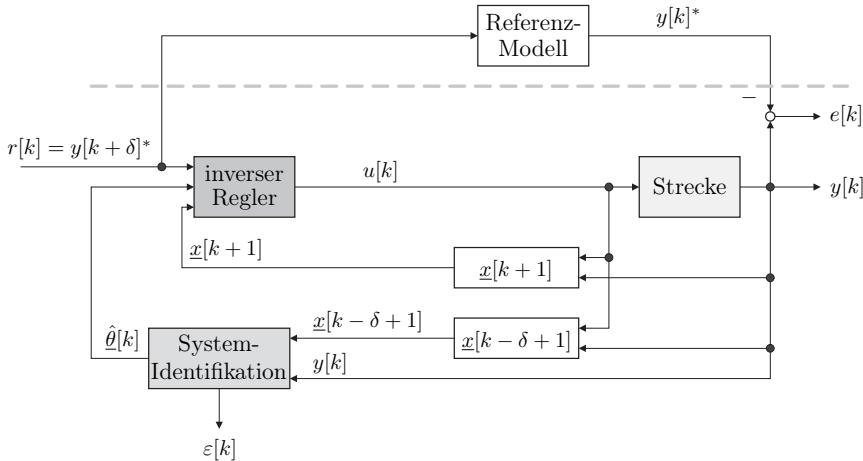


Abb. 13.4: Schema der adaptiven Regelung (MRAC)

sind, unmittelbar einstellt und aktualisiert, sobald neues Systemwissen vorliegt. Jedes durch den Regler angeregte Systemverhalten wird innerhalb eines kurzen transienten Vorgangs identifiziert. Sobald sich ein Identifikationsfehler  $\varepsilon[\cdot] = 0$  ergibt, gilt das auftretende Systemverhalten als bekannt, womit ein mit den geschätzten Parametern ausgelegter Regler ein erfolgreiches und stabiles Verhalten zeigen müsste:  $e[\cdot] = 0$ . Demnach stünde der Regelfehler  $e[\cdot]$  in direktem Zusammenhang mit dem Identifikationsfehler  $\varepsilon[\cdot]$  – der Regelfehler  $e[\cdot]$  erbt die Eigenschaft des Identifikationsfehlers  $\varepsilon[\cdot]$ , durch die Systemidentifikation zu jedem Zeitschritt minimiert zu werden.

Dieser Zusammenhang ist im Folgenden zu beweisen. Dabei ist darauf zu achten, dass durch den in den Regler integrierten Identifikationsprozess ein zeitvariantes System vorliegt und dementsprechend ein nichtlinearer Beweis, wie beispielsweise ein Widerspruchsbeweis, angesetzt werden muss. Konkret ist zu zeigen, dass sich mit dem Regelgesetz (13.67) ohne Kenntnis der wahren Parameter stets ein erfolgreiches Regelverhalten mit  $e[\cdot] = \varepsilon[\cdot] \rightarrow 0$  einstellen wird und folglich eine mögliche Instabilität auf Grund des mit falschen Parametern initialisierten Reglers innerhalb einer beschränkten Zahl an Abtastschritten zu unterdrücken ist.

### 13.3.2 Stabilitätsuntersuchung des geschlossenen Regelkreises

Der Stabilitätsbeweis basiert grundlegend auf den Eigenschaften der Identifikationsalgorithmen; deshalb greift man im Folgenden mehrmals auf die Ergebnisse des Projektions- bzw. RLS-Algorithmus in Kapitel 13.2.1 bzw. 13.2.2 zurück.

Das Regelziel wird erreicht, wenn sich das Ausgangssignal  $y[\cdot]$  des realen Systems (13.25) gleich dem Ausgangssignal  $y[\cdot]^*$  des Referenzmodells (13.64) verhält und damit der Regelfehler

$$e[k] = y[k] - y[k]^* \quad (13.68)$$

zu Null wird.

Es ist nun zu untersuchen, in welcher Beziehung der Regelfehler  $e[\cdot]$  und der Identifikationsfehler  $\varepsilon[\cdot]$  stehen. Da für den Reglerentwurf die Gleichheit (13.65) benutzt wurde, um das Stellsignal  $u[\cdot]$  zu berechnen, kann in der Formel des Regelfehlers (13.68)  $y[k]^*$  durch die um  $\delta$  zeitlich verschobene Gleichung (13.65) ersetzt werden. Mit zusätzlichem Einbringen des ebenfalls zeitlich verschobenen Schätz-Modells (13.26) sowie der Gleichung (13.27) des Identifikationsfehlers ergibt sich die gesuchte Verknüpfung zwischen Regel- und Identifikationsfehler:

$$e[k] = y[k] - \hat{y}[k] + \hat{y}[k] - y[k]^* \quad (13.69)$$

$$= \underbrace{y[k] - \hat{y}[k]}_{\varepsilon[k]} + \hat{\theta}[k-1]^T \underline{x}[k-\delta+1] - \hat{\theta}[k-\delta]^T \underline{x}[k-\delta+1]$$

$$= \varepsilon[k] + (\hat{\theta}[k-1]^T - \hat{\theta}[k-\delta]^T) \underline{x}[k-\delta+1] \quad (13.70)$$

Mit Gleichung (13.70) wird deutlich, dass das Verhalten des Regelfehlers  $e[\cdot]$  direkt von der Konvergenzeigenschaft der Identifikation abhängt und somit die Eigenschaft des Identifikationsfehlers  $\varepsilon[\cdot]$  erbt. Mit Hilfe der Gleichung (13.47) des Projektionsalgorithmus bzw. (13.60) des RLS-Algorithmus wurde gezeigt, dass die Parameteraktualisierung des Identifikationsalgorithmus in beschränkter Zeit zum Erliegen kommt, d.h. die Schätzparameter konvergieren zu einem festen Parametersatz, unabhängig davon, ob die richtigen Systemparameter erreicht wurden, oder nicht. Da die beiden Gleichungen (13.47) und (13.60) jeweils aus der Gleichung (13.46) des Projektionsalgorithmus bzw. der Gleichung (13.58) des RLS-Algorithmus abgeleitet wurden, muss sich zeitgleich entsprechend dieser Gleichungen (13.46) bzw. (13.58) ein Identifikationsfehler  $\varepsilon[\cdot] = 0$  einstellen, sofern die Signale des geschlossenen adaptiven Regelkreises beschränkt bleiben ( $\|x[\cdot]\| < \infty$ ). Hiermit werden die Summanden in Gleichung (13.70) und somit der Regelfehler entsprechend dem Konvergenzverhalten der Identifikation gegen Null gehen, d.h. wird der Identifikationsfehler zu Null, gilt dies auch für den Regelfehler – der Regelfehler erbt die Eigenschaft des Identifikationsfehlers:

$$\lim_{k \rightarrow \infty} e[k] = \lim_{k \rightarrow \infty} \varepsilon[k] + \underbrace{\left[ \lim_{k \rightarrow \infty} (\hat{\theta}[k-1]^T - \hat{\theta}[k-\delta]^T) \right]}_{0 \text{ wg. Gl. (13.47) bzw. (13.60)}} \underline{x}[k-\delta+1]$$

$$\lim_{k \rightarrow \infty} e[k] = \underbrace{\lim_{k \rightarrow \infty} \varepsilon[k] = 0}_{0 \text{ wg. Gl. (13.46) bzw. (13.58)}} \quad \text{für } \|x[k]\| < \infty \quad \forall k \quad (13.71)$$

Man kommt zu folgenden Ergebnissen: Der Minimum-Varianz-Regler erfüllt die harte Forderung, nach  $\delta$  Schritten einen Regelfehler Null zu erzwingen, sobald nach einer kurzen transienten Phase für den Identifikationsfehler  $\varepsilon[\cdot] = 0$

gilt und entsprechend die geschätzten Parameter zu einem konstanten Parametersatz konvergiert sind. Hierbei müssen nicht die wahren Systemparameter gefunden werden. In den transienten Phasen entsteht ein Regelfehler, der eine direkte Konsequenz der Systemidentifikation ist; dieser klingt jedoch entsprechend obiger Darstellung stets ab und geht gegen Null. Diese Aussage gilt jedoch nur, wenn der Beweis der Stabilität gelingt, d.h. alle Signale für  $k \rightarrow \infty$  beschränkt sind ( $\lim_{k \rightarrow \infty} \|\underline{x}[k]\| < \infty$ ):

Da Regelfehler und Identifikationsfehler in direktem Zusammenhang stehen, wachsen beide mit der selben Rate:

$$\sup_{\kappa \leq k} |e[\kappa]| \sim \sup_{\kappa \leq k} |\varepsilon[\kappa]| \quad (13.72)$$

Die Voraussetzung für die Stabilitätsanalyse des adaptiven Regelkonzeptes ist die Stabilität des Reglers bei bekannten Systemparametern, d.h. im nicht-adaptiven Fall bei konvergierten Parametern. Der geschlossene Regelkreis darf lediglich Polstellen bzw. Eigenwerte innerhalb des Einheitskreises besitzen. Diese Stabilitätseigenschaft zeigt sich in der Tatsache, dass der Eingang des Systems im geschlossenen Regelkreis nicht schneller anwachsen kann als der Ausgang (vgl. [68]):

$$|u[k - \delta]| = O \left[ \sup_{\kappa \leq k} |y[\kappa]| \right] \quad (13.73)$$

Wird das Eingangssignal in Form des Regressionsvektors ausgedrückt, so resultiert:

$$\|\underline{x}[k - \delta + 1]\| = O \left[ \sup_{\kappa \leq k} |y[\kappa]| \right] \quad (13.74)$$

Um die Gleichung (13.73) zu erfüllen, darf beispielsweise die Strecke bei einem inversen Reglerkonzept keine instabilen Nullstellen besitzen, d.h. das System muss minimalphasig sein. Es ist allgemein bekannt, dass nichtminimalphasige Systeme auf Grund von Stabilitätsproblemen im geschlossenen Regelkreis schwer regelbar sind (vgl. Kap. 12 sowie [113] und [240]). Für derartige Systeme ist u.a. eine Umkehrung des Wirkungssinns als Reaktion auf sprungartige Veränderungen des Stellsignals typisch, womit das Regelungsproblem bereits klar ersichtlich wird. Mathematisch kennzeichnet sich die Nichtminimalphasigkeit durch das Auftreten mindestens einer Nullstelle außerhalb des Einheitskreises. Liegt ein inverser Regler wie im Beispiel des Minimum-Varianz-Reglers vor (siehe Gleichung (13.64), (13.65) bzw. (13.67)), so werden instabile Nullstellen der Strecke zu instabilen Polstellen im geschlossenen Regelkreis. Hiermit wird deutlich, dass nichtminimalphasige Systeme bei einem Regelkonzept, welches die Elimination der Dynamik der Systemnullstellen zum Ziel hat, unabhängig von einer Adaption bereits zum instabilen Verhalten des Regelkreises führen. Nichtminimalphasige Systeme sind jedoch selten, womit die Annahme eines minimalphasigen realen Systems für die Anwendung des Minimum-Varianz-Reglers durchaus legitimiert ist. Liegt dennoch ein nichtminimalphasiges System vor, so muss beispielsweise an Stelle des

Minimum-Varianz-Reglers ein Dead-Beat-Regler Anwendung finden, bei dem die Systemnullstellen erhalten bleiben.

Für das weitere Vorgehen des Stabilitätsbeweises vergleicht man die Zuwachsrate von Ausgang und Regelfehler. Mit der Tatsache, dass das gewünschte Sollverhalten  $y[\cdot]^*$  stets als begrenzt angenommen werden kann, folgt über die Ungleichung

$$|y[k]| = |e[k] + y[k]^*| \leq |e[k]| + |y[k]^*|$$

dass das Ausgangssignal  $y[\cdot]$  und der Regelfehler  $e[\cdot]$  mit der selben Rate anwachsen:

$$\sup_{\kappa \leq k} |e[\kappa]| \sim \sup_{\kappa \leq k} |y[\kappa]| \quad (13.75)$$

Wendet man nun die Ausdrücke (13.75) und (13.72) auf die Gleichung (13.74) an, so kann mit

$$\|\underline{x}[k - \delta + 1]\| = O \left[ \sup_{\kappa \leq k} |\varepsilon[\kappa]| \right] \quad (13.76)$$

folgende Aussage getroffen werden: sollte im Falle einer instabilen adaptiven Regelung der Betrag des Regressionsvektors  $\|\underline{x}[k]\|$  ins Unendliche anwachsen, dann müsste gemäß der Gleichung (13.76) der Identifikationsfehler noch schneller als der Regressionsvektor unbeschränkt ansteigen.

Die Eigenschaft des Identifikationsalgorithmus besagt jedoch genau das Gegenteil: nach Gleichung (13.45) des Projektionsalgorithmus bzw. Gleichung (13.59) des RLS-Algorithmus wächst der Identifikationsfehler nie stärker als der Betrag des Regressionsvektors. Dieser Widerspruch erlaubt die Feststellung, dass der Regressionsvektor nicht unendlich groß werden kann, d.h. beschränkt ist (Widerspruchsbeweis). Folglich wird das adaptive System mit unbekannten Parametern durch die Systemidentifikation, wie zu erwarten war, stabilisiert. Über die Gleichung (13.46) des Projektionsalgorithmus bzw. Gleichung (13.58) des RLS-Algorithmus kann nun mit dem Wissen eines beschränkten Regressionsvektors  $\underline{x}[\cdot]$  gesichert werden, dass der Identifikationsfehler in beschränkter Zeit verschwindet. Man hält fest:

- Die Ein- und Ausgänge des Systems bleiben beschränkt, womit der adaptive Regler stabil ist:  $\lim_{k \rightarrow \infty} \|\underline{x}[k]\| < \infty$ .
- Das Regelziel wird erreicht:  $\lim_{k \rightarrow \infty} e[k] = 0$ .

Abschließend kommt man zu dem Resultat, dass die adaptive Regelung trotz unbekannter Systemparameter die geforderten Eigenschaften einer guten Regelung erfüllt: das Regelziel muss erreicht werden und der Regler muss eine stabilisierende Wirkung auf das System besitzen.

Für die Anwendung des MRAC-Konzeptes muss die Ordnung  $n$  des zu regelnden Systems bekannt sein oder höher angesetzt werden, so dass für die Identifikation des Systems ausreichend Parameter zur Verfügung stehen. Liegen nicht ausreichend viele Parameter vor, d.h. wird die Systemordnung zu niedrig angelegt, kann das Systemverhalten nicht eindeutig identifiziert werden. Ist hierbei

lediglich die schnelle Dynamik nicht zu beschreiben, so kommt es zwar öfters zu u.U. längeren transienten Phasen, der geschlossene Regelkreis bleibt jedoch stabil; kann hingegen die langsame Hauptdynamik des Systems nicht modelliert werden, ist instabiles Reglerverhalten möglich.

Der Relativgrad  $\delta$  muss nicht als bekannt angenommen werden, da mit der Annahme  $\delta = 1$  ausreichend Parameter zur Verfügung gestellt werden, die während des Identifikationsvorganges auf Grund eines tatsächlich größeren Relativgrades zu Null gesetzt werden könnten.

Des Weiteren dürfen keine Regler Anwendung finden, die im nicht-adaptiven Fall instabil werden. Wird beispielsweise ein Regler verwendet, der die Kürzung der Nullstellen eines Systems zum Ziel hat, muss bekannt sein, ob das zu regelnde System minimalphasig ist – andernfalls würden Zustände des geregelten Systems auf Grund instabiler Pol- Nullstellenkürzungen unbeschränkt anwachsen und das System somit instabiles Verhalten zeigen.

Da in der Beweisführung außer der Systemordnung keinerlei Kenntnis über das System vorausgesetzt wurde, lässt sich auch eine instabile Strecke mit dem adaptiven Konzept regeln.

Zusammengefasst muss für die Anwendung des MRAC-Konzeptes lediglich die Ordnung  $n_{red}$  der Hauptdynamik der zu regelnden linearisierten Strecke bekannt sein. Des Weiteren ist für die Reglerauswahl die Information notwendig, ob eine lineare Strecke nichtminimalphasig ist bzw. eine nichtlineare Strecke eine instabile Nulldynamik besitzt.

## 13.4 Anwendung des adaptiven Reglers auf ein reales Zwei-Massen-System

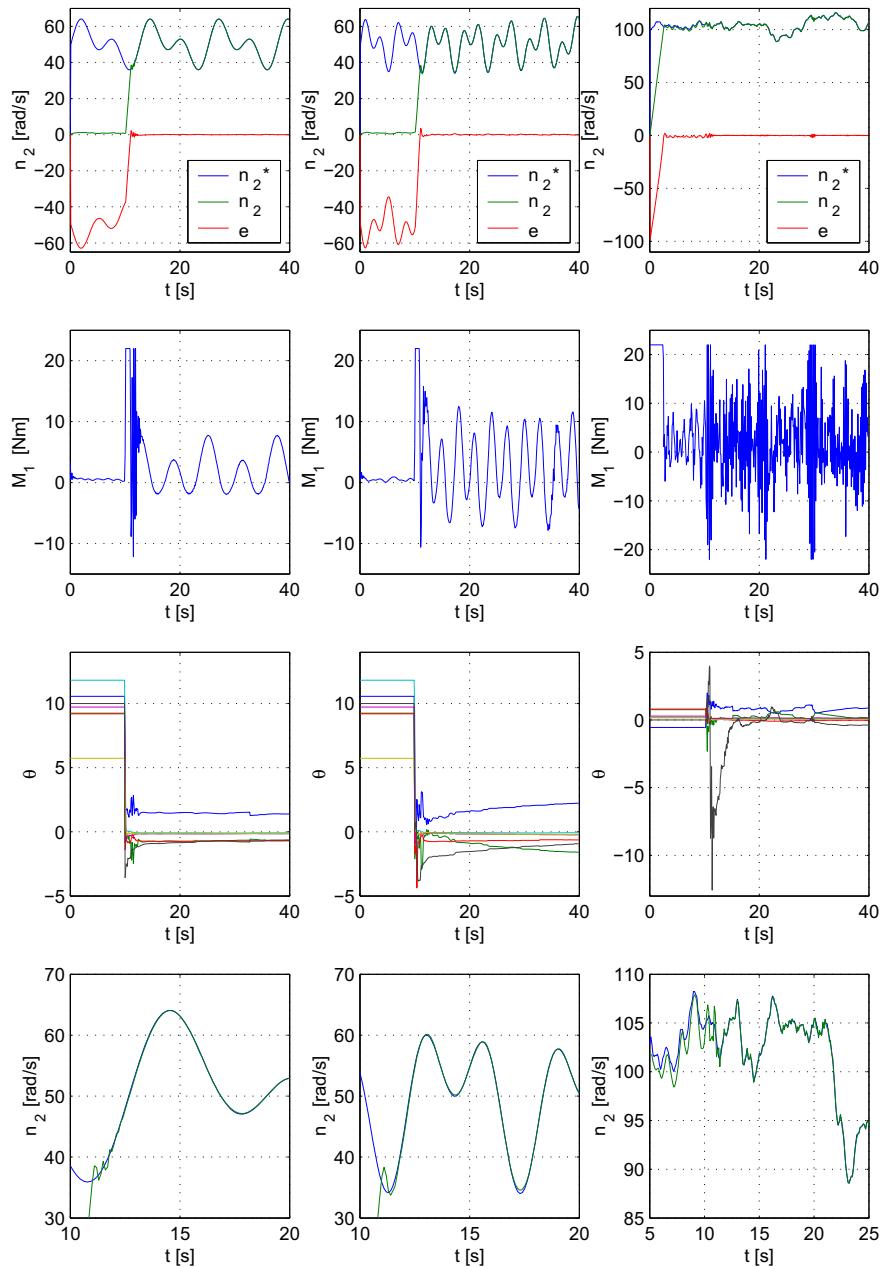
Im Folgenden wird der diskutierte und abgeleitete adaptive Minimum-Varianz-Regler (maximal schneller MRAC) auf ein reales Zwei-Massen-System angewandt, welches in Kapitel 2.1 ausführlich beschrieben wird. Es wird die Drehzahl  $n_2[\cdot]$  der Arbeitsmaschine geregelt. Die reale Anlage mit verrauschten Messsignalen zeigt neben dem theoretischen linearen Verhalten des Zwei-Massen-Systems mit zwei über eine weiche Welle gekoppelten Maschinen nichtlineares Verhalten auf Grund von Reibungseffekten sowie den Schaltvorgängen der Umrichter. Bei der Auslegung des adaptiven Reglers mit Identifikation wurde bewusst eine Untermodellierung vorgenommen, bei der nur minimal viele Parameter der Identifikation zur Verfügung gestellt wurden, sodass lediglich das dominante lineare Verhalten des Zwei-Massen-Systems beschrieben werden kann – das schnelle Umrichterverhalten sowie die Reibungseffekte können nicht berücksichtigt werden. Auf Grund der Möglichkeit einer ständigen Adaption ist zu erwarten, dass sich trotz Untermodellierung sowie Nichtlinearitäten ein zufriedenstellendes Regelergebnis zeigen wird.

In Abbildung 13.5 sind die Messungen des adaptiv geregelten Zwei-Massen-Systems für drei unterschiedliche Sollwertverläufe für die Arbeitsmaschine zu

sehen. Von links nach rechts steigert sich die Anforderung an die Regelung und an das System: die Signale werden immer hochfrequenter bei relativ großen Amplitudenänderungen.

Zunächst betrachtet man die linke und mittlere Messung. Für  $t < 10\text{ s}$  ist die Adaption abgeschaltet, es wirkt lediglich der inverse Regler (13.67) auf die Strecke. Da der Regler mit völlig falschen Werten initialisiert wird, d.h. die Schätzparameter entsprechen nicht den realen Parametern des Systems, und eine Adaption verhindert wird, ist das Regelergebnis äußerst schlecht: für den Regelfehler gilt  $e[\cdot] \gg 0$ . Hiermit wird deutlich, dass ohne Adaption und ohne jegliches Systemwissen nicht geregelt werden kann. Wenn zum Zeitpunkt  $t = 10\text{ s}$  die Adaption des RLS-Algorithmus (geringe Dynamik auf Grund der Initialisierung mit  $\mathbf{P}_{ini} = \mathbf{I}$ ) beginnt, besteht ein großer Regelfehler, womit die geschätzten Parameter durch den sich schnell verändernden Regressionsvektor in wenigen Abtastschritten in eine geeignete Region gelangen. Der Istverlauf entspricht sehr schnell der Solltrajektorie, wie in den gezoomten Ausschnitten zu sehen ist (Abbildung 13.5 unten); der anschließende Verlauf kommt einem sehr guten Folgeverhalten gleich. Dabei ändern sich die Parameter nur noch minimal und werden mit der Zeit gegen einen konstanten Parametersatz konvergieren. Vergleicht man beide Messungen, so ist zu erkennen, dass sich unterschiedliche Parametersätze einstellen, da die Systeme mit unterschiedlichen Frequenzen über die Solltrajektorie angeregt werden, d.h. eine Konvergenz zu den wahren Werten ist nicht gegeben und entsprechend der Theorie für eine erfolgreiche Regelung nicht zwingend.

Das Stellsignal  $M_1[\cdot]$  zeigt des Weiteren einen stets angemessenen Verlauf. Zum Zeitpunkt  $t \approx 11\text{ s}$  erfährt das Zwei-Massen-System beim Einregelvorgang in Abbildung 13.5 unten links und unten Mitte durch den Momentensprung eine starke Anregung der Eigenfrequenz der elastischen Welle, wodurch sich Schwingungen im Drehzahlsignal  $n_2[\cdot]$  der Arbeitsmaschine ergeben, die jedoch durch den adaptiven Regler schnell aktiv bedämpft werden. Es sei an dieser Stelle erwähnt, dass eine konventionelle Ausgangsregelung selbst bei exakt bekannten Parameter gemäß Kapitel 2.2 nicht auf die Drehzahl einer weich angekoppelten Arbeitsmaschine geregelt werden kann, ohne instabil zu werden. Folglich kann mit einem konventionellem Ausgangsregler auftretenden Schwingungen nicht entgegengewirkt werden. Dies ist darauf zurückzuführen, dass die Ordnung des Reglers, beispielsweise die eines *PI*-Reglers, nicht ausreicht, um die Dynamik des Zwei-Massen-Systems beliebig beeinflussen zu können. Der Regler ist somit bzgl. der Dynamik des Systems bei auftretenden Schwingungen zu langsam, was neben einem bleibenden Regelfehler bei ungünstigen Verzögerungen des Reglers im geschlossenen Regelkreis zur Instabilität führt. Ein Referenzmodell-Regler besitzt hingegen mindestens die Ordnung des Systems, womit dieser mindestens eine ebenso schnelle Dynamik wie das zu regelnde System zeigen kann. Ist das Verhalten der Strecke in Form eines Modells bekannt, kann dieses gezielt beeinflusst werden, so dass es sich im geschlossenen Regelkreis entsprechend eines beliebigen Referenzmodells verhält – auftretende Schwingungen sind daher mit einem



**Abb. 13.5:** Zwei-Massen-System mit adaptivem Regler (RLS-Algorithmus); es wird die Drehzahl  $n_2[\cdot]$  der Arbeitsmaschine geregelt;  $\mathbf{P}_{ini} = \mathbf{I}$ ; Beginn der Adaption bei  $t = 10$  s

Referenzmodell-Regler aktiv zu bedämpfen. In diesem Kontext ist zu erwähnen, dass ein Referenzmodell-Regler einem Zustandsregler mit Beobachter entspricht, was in [240] und [152] näher beleuchtet wird. Dies erklärt ebenfalls die Möglichkeit des Referenzmodell-Reglers, die Dynamik der Strecke beliebig beeinflussen zu können.

Auch der adaptive Referenzmodell-Regler, d.h. MRAC besitzt trotz unbekannter Systemparameter die Möglichkeit, die auftretenden Schwingungen bei starker Anregung des Zwei-Massen-Systems aktiv zu bedämpfen. Nachdem mit der im adaptiven Regler enthaltenen Identifikation jedes auftretende Systemverhalten gelernt wird und mit diesem Wissen die Parameter des modellbasierten Reglers entsprechend adaptiert werden, kann das neu aufgetretene Systemverhalten nach einer kurzen transienten Phase beliebig beeinflusst werden. Sobald demnach Schwingungen durch den Sollverlauf angeregt werden, ist die entsprechende Systemdynamik zu identifizieren, so dass nach kurzer Zeit ausreichend Wissen über das schwingungsfähige System vorliegt, um dieses aktiv zu bedämpfen – das ist möglich, ohne die exakten Parameter des Systems zu kennen.

Der Vorgang der aktiven Dämpfung durch den Minimum-Varianz-Regler ist in der Abbildung 13.5 unten links sowie unten Mitte gut zu erkennen. Beim schnellen Einregeln, d.h. starker Anregung der Welle, zeigt sich ein schnelles Abklingen der Schwingung. Je schneller das Schwingungsverhalten im transienten Vorgang der Identifikation gelernt wird, desto schneller findet das Einschwingen statt. Aus diesem Grund ist auch ein unterschiedliches Einschwingverhalten zwischen Abbildung 13.5 unten links und unten Mitte zu erkennen. Sobald sich ein konstanter Parametersatz mit einem Identifikationsfehler Null eingestellt hat, wird bei einem nächsten Momentensprung das Ausregeln der Schwingung gemäß des Minimum-Varianz-Reglers maximal schnell geschehen. Es ist jedoch auf Grund der untermodiellierten Dynamik nicht zu erwarten, dass dies in  $\delta$  Schritten geschieht.

In der nächsten Messung (Abbildung 13.5 rechts) fordert man vom adaptiv geregelten System, einem Zufallssignal als Solltrajektorie zu folgen. Die Identifikation wird mit den wahren Parameterwerten initialisiert, die zuvor in einem konventionellen Identifikationslauf ohne Regler bestimmt wurden. Mit diesen findet bis zum Zeitpunkt  $t = 10\text{ s}$  eine rein inverse Regelung statt. Ein Ausschnitt des Messergebnisses ist vergrößert unten rechts in Abbildung 13.5 dargestellt. Man erkennt, dass die angeblich „wahren Parameterwerte“ des Identifikationslaufs ohne Regler doch nicht exakt diesen entsprechen – die Maschine kann dem Random-Signal nicht folgen. Nach Hinzuschalten der Systemidentifikation der adaptiven Regelung zeigt sich perfektes Folgeverhalten. Dass dies ein hochfrequentes Stellsignal  $M_1[\cdot]$  mit großen Amplituden fordert, ist einzusehen und durch das zweite Bild von oben in der rechten Spalte bestätigt. Die Stellgröße gerät sogar mehrmals in die Begrenzung, d.h. das maximal stellbare Moment der Maschine bzw. des Umrichters ist erreicht – zu diesen Zeitpunkten wird es selbstverständlich zu einer Abweichung vom gewünschten Sollverlauf der Drehzahl kommen, die jedoch sofort korrigiert wird, sobald die Stellgröße die Begrenzung wieder verlässt. Da

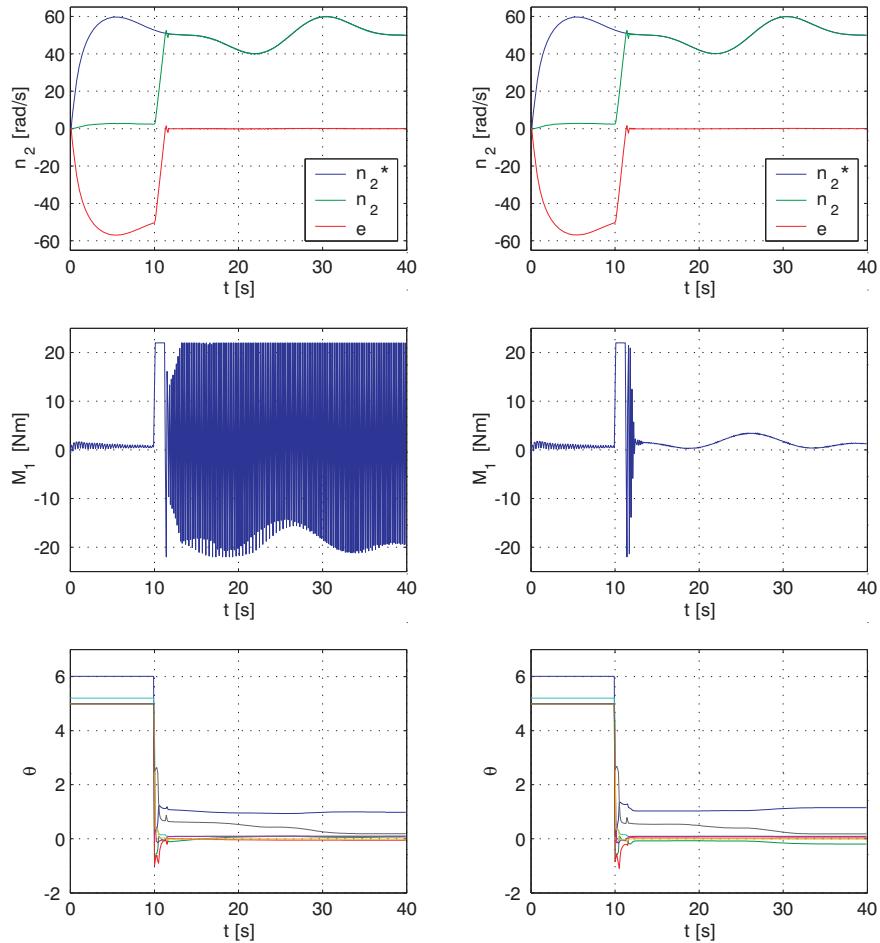
das System mit einem Random-Signal angeregt wird, erhält die integrierte Systemidentifikation des adaptiven Reglers sehr viel Information über das System. Dementsprechend zeigen sich starke Schwankungen im Schätzparameter-Verlauf. Für  $t > 30\text{ s}$  tritt eine Konvergenz ein, d.h. ab diesem Zeitpunkt bringt die Erregung keine neue Information – es ist anzunehmen, dass dies die wirklichen Parameter des Systems sind.

Man hält fest: Für eine gute Regelung müssen die wahren Parameter des Systems nicht bekannt sein, es muss nur so viel Systeminformation identifiziert werden, wie momentan für das Erreichen des Regelziels notwendig ist. Wird das System über die Solltrajektorie nur wenig erregt, wie es in der linken und mittleren Messung der Fall ist, so findet der Algorithmus einen falschen, aber zur erfolgreichen Regelung ausreichenden Parametersatz. Das Certainty-Equivalence Prinzip kommt hierbei zur Anwendung. Erfährt das System über die Zeit weitere neue Erregungsmuster, wird mehr Systeminformation gesammelt und die Schätzparameter entsprechen immer mehr den wahren Parametern, so dass in gewisser Weise ein Lernen vorliegt: wiederholen sich Erregungsmuster, so finden keine transienten Vorgänge mehr statt, bis das System erneut ein noch nicht aufgetretenes Erregungsmuster erfährt. Liegt von Beginn an eine starke Anregung vor, wie dies in der rechten Messung der Fall ist, so findet die Identifikation sehr schnell die wahren Parameter, wodurch die Strecke vollständig beschrieben ist. Es zeigt sich dann eine erfolgreiche Regelung basierend auf dem vollständig identifizierten System. Der Regelvorgang stützt sich nicht mehr auf das certainty-equivalence Prinzip. Es treten folglich keine transienten Vorgänge mehr auf. In allen Situation ermöglicht somit der adaptive Regler ein erfolgreiches Regeln des unbekannten Systems.

In diesem Kapitel wurde eine anschauliche Einführung in die adaptive Regelung gegeben. Dargestellt wurde das Grundprinzip und die Funktionsweise an Hand von MRAC. Da es sich auf Grund der Zeitvarianz des adaptiven Reglers um eine nichtlineare Regelungsmethode handelt, bei der bekannte lineare Stabilitätsanalysen nicht anwendbar sind, zeigte sich die Notwendigkeit nichtlinearer Stabilitätsanalysen. Ein mögliches Vorgehen wurde mit Hilfe der Lyapunov-Methode sowie dem Widerspruchsbeweis vorgestellt.

Es ist des Weiteren zu erwähnen, dass der in diesem Kapitel dargestellte Regler theoretisch gute Ergebnisse erzielt, für die Verwendung an realen Anlagen jedoch noch Defizite aufweist. Das beeindruckende Ergebnis in Abbildung 13.5 ist mit diesem Regler ohne Modifikation nicht möglich. Vielmehr würde das Ergebnis wie in Messung 13.6 linke Spalte, aussehen, bei der ein nicht vertretbares Verhalten des Stellsignals  $M_1[\cdot]$  zu erkennen ist; ohne Sättigung der Stellgröße wäre das geregelte System instabil. Die Dissertation [240] befasst sich aus diesem Grund mit der Robustifizierung und Erweiterung des adaptiven Konzeptes:

Mit dem vorgestellten MRAC können lineare Strecken erfolgreich geregelt werden. Besitzt die Strecke jedoch Nichtlinearitäten wie beispielsweise Sättigung oder Reibung, kann es je nach Dominanz der Nichtlinearität zu Instabilitäten kommen. Liegt Reibung vor, so wird zwar der Identifikationsfehler und folglich



**Abb. 13.6:** Zwei-Massen-System mit adaptivem Regler (RLS-Algorithmus),  $\hat{\theta}$  falsch initialisiert,  $\mathbf{P} = \mathbf{I}$ , Beginn der Adaption bei  $t = 10$  s, links: ohne Regleranpassung, rechts: mit Regleranpassung

der Regelfehler nicht zu Null, die Stabilität wird jedoch nicht gefährdet. Um das Regelverhalten dennoch zu verbessern, wird in [240] ein neuronales Netz stabil in das MRAC-Konzept integriert, um auch nichtlineares Verhalten parallel zum Regelvorgang identifizieren zu können. Sättigungen der Stellgröße führen hingegen bei schwingungsfähigen Systemen, wie dem Zwei-Massen-System, bei hochdynamischen Reglern sofort zur Instabilität. Um dem entgegenzuwirken und dennoch hochdynamische Regler einsetzen zu können, wird das MRAC-Konzept in [240] um einen prädiktiven Regler erweitert, wobei weiterhin Stabilität garantiert werden kann.

Ein weiteres Problem ergibt sich, wenn ein hochdynamischer inverser Regler, wie beispielsweise der Minimum-Varianz-Regler, Anwendung finden soll; hierbei wird sowohl die Dynamik des Nenners als auch die Dynamik des Zählers eliminiert, um maximal schnell das Regelziel zu erreichen. Wie in Kapitel 13.3.2 dargestellt wurde, darf dieser Regler auf Grund von Pol-Nullstellen-Kürzungen nur für minimalphasige Systeme verwendet werden. Da es sich z.B. beim Zwei-Massen-System um ein zeitkontinuierliches minimalphasiges System handelt, wäre zu erwarten, dass der Minimum-Varianz-Regler anwendbar ist. Es zeigt sich jedoch, dass durch die Diskretisierung instabile Nullstellen entstehen und somit ein instabiler Regelkreis resultiert – eine hochdynamische Regelung ist nicht möglich. Die entsprechende Messung zeigt Abbildung 13.6 links. Durch eine Beeinflussung des Diskretisierungsvorganges in [240] wird es jedoch möglich, inverse Regelungen auch für alle zeitkontinuierlichen minimalphasigen Systeme zu verwenden, die durch die Diskretisierung instabile Nullstellen erhalten. Die Anwendung des erweiterten Minimum-Varianz-Reglers auf das Zwei-Massen-System zeigt die Messung 13.6 rechte Spalte sowie die bereits diskutierten Messungen in Abbildung 13.5.

Durch diese genannten Erweiterungen der MRAC-Idee gewinnt man ein hochdynamisches adaptives Regelungskonzept, das für die Umsetzung an realen Anlagen geeignet ist und alle Vorteile des theoretischen MRAC-Konzeptes besitzt ([240], [241], [242], [49], [239]).

## 14 Disturbance Rejection

The previous chapter dealt with the problem of controlling a system with unknown, constant parameters. The material contained in this chapter was published previously as a doctoral thesis, see Feiler 2004 [48]. We consider the case where, in addition, the system is subject to unknown, external and time-varying disturbances. It is seen that under certain conditions, the problem can be translated to the previous one if a controller of extended order is used. The approach is based on the well-known fact from linear control theory that deterministic disturbances containing a finite number of frequencies can be completely rejected by placing appropriate poles in the feed-forward path of the control loop. According to the internal model principle (Francis and Wonham 1976 [58]), this can be regarded as a procedure of expanding the system by a “disturbance model” which generates an additional input that compensates the effect of the disturbance.

The main purpose of this chapter is to discuss how this linear design rule extends to the case where the parameters of the plant and the disturbance model are unknown. It is seen that both deterministic and stochastic disturbances can be rejected using the same method of augmenting the state space of the closed-loop system. In the stochastic case, the performance improvement is seen to depend upon the impulse response of the disturbance model. If the latter decays fast, the disturbance affecting the system is almost pure white noise which cannot be eliminated. Pseudo-linear regression algorithms can be employed in this case to minimize the variance of the resulting output error. If, on the other hand, the impulse response is slowly decaying, the disturbance is highly correlated with its past values and can be rejected almost completely, as in the deterministic case. In the nonlinear domain, a similar procedure can be adopted provided that certain properties of the nonlinear system hold. These properties, defined in terms of the linearized system, guarantee the existence of an input-output representation of the augmented system. The main theoretical questions that arise in this context are similar to the ones discussed in chapter 12. It is seen that nonlinear disturbance rejection is possible if the state of the system remains in a neighborhood of the origin. In summary, whenever an external disturbance can be described as the output of an unforced system of known order, it can be eliminated completely. The importance of this result lies in the fact that the proposed method not only guarantees stability under perturbations but also compensates for the perturbation, even as the latter is not known completely.

## 14.1 Linear Disturbance Rejection

### 14.1.1 Deterministic Disturbances

We start with a linear discrete-time system affected by an external, bounded, deterministic disturbance  $v(k)$ ,

$$\begin{aligned}\underline{x}(k+1) &= \mathbf{A} \underline{x}(k) + \underline{b} u(k) + \underline{b}_v v(k) \\ y(k) &= \underline{c}^T \underline{x}(k)\end{aligned}\quad (14.1)$$

where  $\underline{x}(k) \in \mathbb{R}^n$  is the state of the system,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  the system matrix,  $\underline{b} \in \mathbb{R}^{n \times 1}$  and  $\underline{b}_v \in \mathbb{R}^{n \times 1}$  are input vectors and  $y(k)$  a scalar output, i.e.  $\underline{c}^T \in \mathbb{R}^{n \times 1}$ . The signal  $v(k)$  is assumed to be the output of the following unforced system

$$\begin{aligned}\underline{x}_v(k+1) &= \mathbf{A}_v \underline{x}_v(k) \\ v(k) &= \underline{c}_v^T \underline{x}_v(k)\end{aligned}\quad (14.2)$$

where  $\underline{x}_v(k) \in \mathbb{R}^{n_v}$ ,  $\mathbf{A}_v \in \mathbb{R}^{n_v \times n_v}$  and  $\underline{c}_v^T \in \mathbb{R}^{n_v \times 1}$ . The equation may be regarded as a model which generates the external disturbance. If  $\mathbf{A}_v$  has its eigenvalues inside the unit circle,  $\underline{x}_v(k)$  and  $v(k)$  tend to zero asymptotically. If, on the other hand,  $\mathbf{A}_v$  is unstable,  $\underline{x}_v(k)$  will grow in an unbounded fashion and the control needed to reject the disturbance will also be unbounded. Since our interest is in bounded disturbances, we assume that  $\mathbf{A}_v$  is a stable matrix with simple eigenvalues on the unit circle. Hence  $v(k)$  can be expressed as a finite sum of sinusoidal signals (including a constant). Our objective in this case is to determine  $u(k)$  in the composite system such that  $\lim_{k \rightarrow \infty} |y(k) - y^*(k)| = 0$  where  $y^*(k)$  is the desired output and  $y^*(k + \delta)$  is known at instant of time  $k$  and  $\delta$  the relative degree.

The solution involves determining the input-output representation of both (14.1) and (14.2). The key step then is to use the homogeneous difference equation obtained from (14.2) in order to eliminate  $v(k)$  from (14.1). This is best illustrated by the following example where  $n = n_v = 1$ .

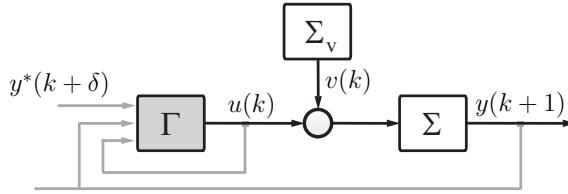
#### Example:

$$y(k+1) = ay(k) + v(k) + u(k) \quad (14.3)$$

$$v(k) = a_v v(k-1) \quad a_v = \pm 1 \quad (14.4)$$

If  $v(k)$  is known, the input  $u(k)$  can be computed from equation (14.3) to achieve exact tracking. From equations (14.3) and (14.4) we have  $v(k) = a_v v(k-1) = a_v [y(k) - ay(k-1) - u(k-1)]$ , i.e. the value of  $v(k)$  depends only on past values of  $y(\cdot)$  and  $u(\cdot)$ . By choosing

$$u(k) = y^*(k+1) - (a + a_v) y(k) + a_v a y(k-1) + a_v u(k-1) \quad (14.5)$$



**Abb. 14.1:** Disturbance as output of homogeneous difference equation

exact tracking is achieved in two steps. Note that  $v(k)$  is not known explicitly but is completely defined by its value at time  $(k - 1)$ , which, in turn can be expressed in terms of past values of both the input and the output.

The idea is readily generalized to the case where the system as well as the disturbance model are of higher order. The input–output representations obtained with the relative degree  $\delta$  in this case are:

$$y(k + \delta) = \sum_{i=0}^{n-1} a_i y(k - i) + \sum_{i=0}^{n-1} b_i u(k - i) + \sum_{i=0}^{n-1} c_i v(k - i) \quad (14.6)$$

$$v(k + 1) = \sum_{i=0}^{n_v-1} \alpha_i v(k - i) \quad (14.7)$$

The steps to eliminate the disturbance by means of the autonomous system are as follows: If  $n > n_v$ , (14.7) is used to express  $v(k - n + 1), \dots, v(k - n_v)$  in (14.6) in terms of  $v(k) \dots v(k - n_v + 1)$ , otherwise this step is skipped. Then, (14.6) is solved for  $v(k - n_v + 1)$  and the result used to express the right hand side of (14.7) in the more recent values of  $v(\cdot)$  and the inputs and outputs of the system. By shifting the time axis (14.7) backwards by one step we again obtain an expression for  $v(k - n_v + 1)$  which is used to eliminate  $v(k - n_v + 1)$  from (14.6). The procedure is repeated  $n_v$  times until the right hand side of the second equation is expressed completely in terms of the inputs and outputs of the plant. In the final stage, the first equation (14.6) contains only the current value  $v(k)$  which, in turn is eliminated using (14.7). The resulting equation represents the disturbance-free composite system. With  $n_v$  being the order of the disturbance generating system,  $n_v$  steps were needed to eliminate  $v(\cdot)$ . Consequently, the dimension of the system has increased by  $n_v$ :

$$y(k + \delta) = \sum_{i=0}^{n+n_v-1} a_i y(k - i) + \sum_{i=0}^{n+n_v-1} b_i u(k - i) \quad (14.8)$$

Clearly, the equation is of the form (13.3) and the same control law as in case without external disturbances can be applied, except that it now depends on the past  $(n + n_v - 1)$  values of the signals of the system:

$$u(k) = \frac{1}{b_0} \left[ y^*(k + \delta) - \sum_{i=0}^{n+n_v-1} a_i y(k-i) - \sum_{i=1}^{n+n_v-1} b_i u(k-i) \right] \quad (14.9)$$

The procedure is more transparent if we use an equivalent representation of the system (14.1) given by

$$A(z^{-1})y(k) = z^{-\delta}B(z^{-1})u(k) + z^{-\delta}G(z^{-1})v(k) \quad (14.10)$$

where

$$\begin{aligned} A(z^{-1}) &= 1 - a_1 z^{-1} - \cdots - a_{n_A} z^{-n_A} \\ B(z^{-1}) &= b_0 + \cdots + b_{n_B} z^{-n_B} \\ G(z^{-1}) &= g_0 + \cdots + g_{n_G} z^{-n_G} \end{aligned}$$

are polynomials in the delay operator  $z^{-1}$ . The relative degree  $\delta$  is assumed to be known and  $B(z^{-1})$  is a Hurwitz polynomial. The disturbance  $v(k)$  is generated by a homogenous system of the form

$$\begin{aligned} v(k) &= [\delta_1 z^{-1} + \cdots + \delta_{n_D} z^{-n_D}]v(k) \\ &:= [1 - D(z^{-1})]v(k) \end{aligned} \quad (14.11)$$

If we solve equation (14.10) for  $v(k)$  and use the result to substitute  $v(k)$  in (14.11) we obtain

$$DAy(k) = z^{-\delta}DBu(k) \quad (14.12)$$

which corresponds to a system model the order of which has been augmented in order to account for the presence of the disturbance. For ease of notation, the explicit dependence of the polynomials on  $q^{-1}$  has been omitted. Replacing  $y(k)$  in equation (14.12) by  $y^*(k)$  and solving for  $u(k)$  yields the equivalent of control law (14.9).

$$u(k - \delta) = \frac{1}{b_0} \left[ y^*(k) - [1 - DA]y(k) - [DB - b_0]u(k - \delta) \right] \quad (14.13)$$

The application of the control law results in the closed-loop system:

$$y(k) = y^*(k) + z^{-\delta}GDv(k) \quad (14.14)$$

The effect of using a controller of augmented order is immediately evident, since, according to (14.11),  $Dv(k) = 0$ . If the parameters of the system are unknown, the control law is determined on the basis of an underlying identification model. This model is of augmented order:

$$\hat{y}(k) = \underline{\phi}(k - \delta)^T \underline{\theta}(k - 1) \quad (14.15)$$

where  $\underline{\phi}(\cdot) \in \mathbb{R}^{n+n_v}$  and  $\underline{\theta}(\cdot) \in \mathbb{R}^{n+n_v}$ . Since (14.8) has been shown to be a valid representation of the composite system, we conclude that there exists a constant parameter vector  $\underline{\theta}^*$  for which  $\hat{y}(k) = y^*(k)$ . It is seen that the zeros

of the augmented<sup>1)</sup> system are given by the original zeros plus the poles of the disturbance model. Since the latter is stable, the zeros of the polynomial  $DB$  in equation (14.12) lie inside or on the closed unit circle and the zeros of the transfer function  $z^{-\delta}B/A$  obtained from (14.12) after pole-zero cancellation, i.e. its controllable modes, lie strictly inside the unit circle. The same arguments as in the disturbance-free case can be used to proof stability of the adaptive controller based on the augmented system.

As seen in equation (14.10), the relative degree  $\delta$  is the same as in the original system. The orders  $n$  and  $n_v$  must be known. In most practical situations, prior information about the plant as well as the disturbance is available. As an example, if the disturbances are harmonic,  $n_v$  is equal to the (expected) number of nonzero points in the two-sided spectrum of the disturbance signal. Using the parameter estimates  $\hat{\theta}(\cdot) = [\hat{a}_0, \dots, \hat{a}_{n+n_v-1}, \hat{b}_0, \dots, \hat{b}_{n+n_v-1}]^T$  obtained from (14.15) the control law reads:

$$u(k) = \frac{1}{\hat{b}_0(k)} \left[ y^*(k + \delta) - \sum_{i=0}^{n+n_v-1} \hat{a}_i(k) y(k-i) - \sum_{i=1}^{n+n_v-1} \hat{b}_i(k) u(k-i) \right] \quad (14.16)$$

Since the controller is based on the augmented system, exact cancellation of the disturbance is achieved. It is clear that the method asymptotically introduces poles in the feed-forward path of the control-loop which correspond to the poles of the disturbance model. As an example, the effect of a sinusoidal disturbance with frequency  $\omega$  can be nulled by introducing a pair of complex conjugate poles at  $z = e^{\pm j\omega h}$ . The resulting disturbance transfer function has a notch at  $\omega$ . These facts are well-known from linear systems theory. It is also known that the linear method fails if the frequency of the disturbance is not known exactly. The benefit of the adaptive version is that the frequency need not be known but the parameters are tuned automatically to generate a notch at the appropriate frequency such that, in any case,  $\lim_{k \rightarrow \infty} |y(k) - y^*(k)| = 0$ . In addition, slow drifts of the disturbance frequency can be tracked.

### Example:

Assume that the two-mass system of chapter 13 is subject to a sinusoidal disturbance of unknown frequency  $\omega$ . The unforced system generating  $v(\cdot)$  is given by

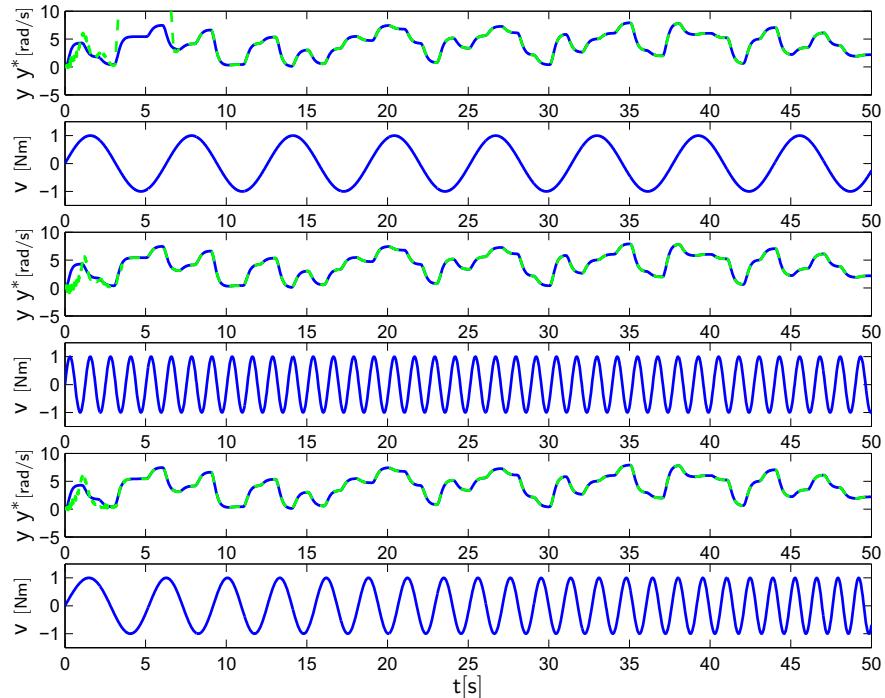
$$v(k+1) = 2 \cos(\omega h) v(k) - v(k-1) \quad (14.17)$$

where  $h$  is the sampling time. This is already more than we need to know. In fact, all that is needed to design an adaptive controller eliminating  $v(\cdot)$  is the order  $n_v = 2$  of the disturbance generating system. Figure 14.2 displays the

---

<sup>1)</sup> The word „augmented“ refers to the augmentation of the state-space of the undisturbed system by additional states from the disturbance model — not to be confused with the augmented error system of chapter 5.6.4

performance of the adaptive system, as the output is required to track a piecewise constant signal. The frequency of the disturbance is constant in the first 2 plots and (slowly) time-varying in the last row of figure 14.2.



**Abb. 14.2:** Adaptive rejection of disturbances with unknown frequency  $\omega = 1 \dots 5$  rad/s

### 14.1.2 Stochastic Disturbances

In the following, we will assume that the linear plant described in equation (14.1) is affected at the input by correlated noise  $v(k)$ . As in the previous section,  $v(k)$  is the output of a disturbance generating system. In contrast to equation (14.2), the disturbance model is driven by a white noise sequence  $\{w(k)\}$ .

$$\begin{aligned} \underline{x}_v(k+1) &= \mathbf{A}_v \underline{x}_v(k) + \underline{b}_w w(k) \\ v(k) &= \underline{c}_v^T \underline{x}_v(k) \end{aligned} \quad (14.18)$$

where  $\underline{b}_w \in \mathbb{R}^{n_v \times 1}$  and  $w(k)$  has the following properties:

$$E\{w(k)|k-1\} = 0 \quad (14.19)$$

$$E\{w(k)^2|k-1\} = \sigma^2 \quad \sigma^2 < \infty \quad (14.20)$$

$$\limsup_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N w^2(k) < \infty \quad \text{with probability 1} \quad (14.21)$$

In other words, the white noise has zero conditional mean, finite variance  $\sigma^2$  and is mean square bounded. As in the previous section, we assume that the system and disturbance model have an input-output representation. The problem is to design a control law  $u(k)$  such that  $y(k)$  follows a desired reference signal  $y^*(k)$  as closely as possible. Due to the presence of the noise the control error  $e(k) = [y(k) - y^*(k)]$  cannot be made zero, but its expected value can be minimized.

The effect of  $w(k)$  on  $v(k)$  obviously depends upon the matrix  $\mathbf{A}_v$  and the vectors  $\underline{b}_w$  and  $\underline{c}_v^T$ . We shall successively consider the control strategies when all the parameters are known, and when the parameters are unknown and have to be estimated on-line. In the former case we have a linear stochastic control problem, and in the latter case we have a linear stochastic adaptive control problem. Further, before proceeding to solve the two problems analytically, we shall discuss qualitatively the nature of the disturbance, and the conditions under which significant improvement in performance can be expected.

#### 14.1.2.1 A Qualitative Analysis

Proceeding as in section 14.1.1, it is possible to obtain an augmented input-output model of order  $n + n_v$ :

$$y(k+1) = \sum_{i=0}^{n+n_v-1} \bar{a}_i y(k-i) + \sum_{i=d-1}^{n+n_v-1} \bar{b}_i u(k-i) + \sum_{i=d-1}^{n+n_v-1} \bar{c}_i w(k-i) \quad (14.22)$$

In order to obtain the  $\delta$ -step ahead predictor form, the time axis is shifted and the outputs  $y(k+\delta-1) \dots y(k+1)$  are expressed in terms of past values.

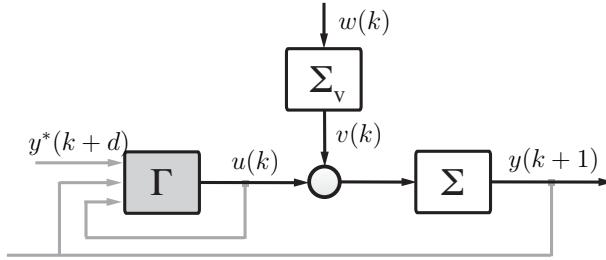
$$y(k+\delta) = \sum_{i=0}^{n+n_v-1} a_i y(k-i) + \sum_{i=0}^{n+n_v-1} b_i u(k-i) + \sum_{i=0}^{n+n_v-1} c_i w(k-i) \quad (14.23)$$

From equation (14.23), it is clear that all past values of the white noise (i.e.  $w(k-i), i = 0, \dots, n + n_v - 1$ ) affect the output at time  $k + \delta$ . The question that has to be addressed is the extent to which the control input at instant  $k$  can compensate for these values of the noise. This is discussed by means of the following simple examples:

*Order of the disturbance model  $n_v = 0$*

First, let the order of the disturbance generating system be  $n_v = 0$ :

$$y(k+1) = ay(k) + u(k) + w(k) \quad (14.24)$$



**Abb. 14.3:** Correlated noise affecting a linear system

Note that  $w(k)$  is an unknown random variable. Since  $w(k)$  is not known at instant of time  $k$ , no control input  $u(k)$  can be defined that will compensate for  $w(k)$ . However, solving the difference equation we have

$$y(k) = a^k y_0 + \sum_{i=0}^{k-1} a^{k-1-i} [u(i) + w(i)] \quad (14.25)$$

with the initial value  $y(0) = y_0$ . We see that  $y(k)$  is affected by all previous values  $w(i)$ ,  $i = 0 \dots (k-1)$  of the white noise, and therefore contains information about the disturbance. By choosing the control input  $u(k) = y^*(k+\delta) - f_b y(k)$  a tracking controller is realized that cancels all past disturbance values but not the present one.  $f_b$  is the feedback gain. The control error becomes:

$$e(k+1) = y(k+1) - y^*(k+\delta) = (a - f_b) y(k) + w(k) \quad (14.26)$$

The variance of the control error,  $E\{e^2(k+1)\} = (a - f_b)^2 E\{y^2(k)\} + \sigma^2$ , has a minimal value for  $f_b = a$ . Hence, the best choice for the control law in the case  $n_v = 0$  is obtained by simply ignoring the presence of the white noise  $w(k)$ .

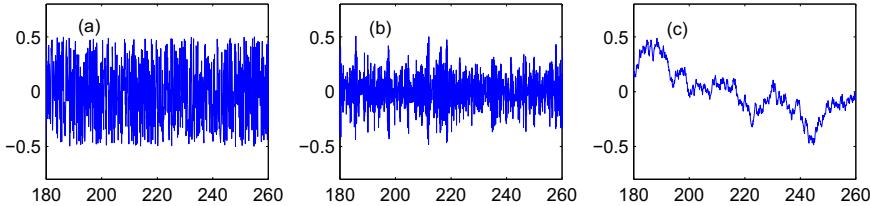
#### Order of the disturbance model $n_v > 0$

Proceeding to a more interesting case, we now assume that the order of the disturbance generating system is greater than zero. The signal  $v(k)$  is correlated with  $v(k-1)$ , i.e.  $E\{v(k)v(k-1)\} \neq 0$ . We have for example,

$$\begin{aligned} y(k+1) &= ay(k) + u(k) + v(k) \\ v(k+1) &= a_v v(k) + b_w w(k) \end{aligned} \quad (14.27)$$

The transfer function of the disturbance model is  $F_v(z) = \frac{v(z)}{w(z)} = \frac{b_w}{z - a_v}$ , with  $|a_v| < 1$ . It is clear that the model may be thought of as an IIR-filter with white-noise input. For different values of the eigenvalue  $a_v$  the nature of the output and the maximum values it assumes are different. For purposes of comparison, we would require the range of values assumed by the disturbance to be the same in

all cases considered. We assume that the high frequency gain  $b_w = b_w(a_v)$  can be determined experimentally for this purpose. Three values of the pair  $[b_w, a_v]$  and the corresponding evolution of  $v(k)$  for a specified white noise sequence  $\{w(k)\}$  are displayed in figure 14.4.



**Abb. 14.4:** Correlated noise  $v(k)$  for different values of the pair  $[b_w, a_v]$  in equation (14.27): (a)  $[1, 0]$ , (b)  $[0.35, -0.8]$ , (c)  $[0.09, 0.99]$

The impulse response of the filter (14.27) is given by

$$h(k) = b_w a_v^{k-1} \quad k \geq 1 \quad (14.28)$$

Assuming that  $v(0) = 0$ , the output  $v(k)$  can be determined as

$$v(k) = \sum_{i=0}^{k-1} h(k-i) w(i) = b_w \sum_{i=0}^{k-1} a_v^{k-1-i} w(i) \quad (14.29)$$

Hence,  $|v(k)| = |b_w| \left| \sum_{i=0}^{k-1} a_v^{k-1-i} w(i) \right|$ . If the eigenvalue  $a_v$  is close to  $\pm 1$ , the impulse response  $h(k)$  decays slowly and the contribution of the last sum to the magnitude of  $v(k)$  is large. Consequently,  $|b_w|$  has to be reduced in order to keep  $v(k)$  within the same bounds for all eigenvalues  $a_v \in (-1 \dots 1)$ . If  $|a_v|$  increases the correlation of  $v(k)$  with its past values also increases:

$$\begin{aligned} E\{v(k)v(k-1)\} &= a_v E\{v^2(k-1)\} + b_w E\{w(k-1)\} \cdot E\{v(k-1)\} \\ &= a_v E\{v^2(k-1)\} \quad \text{because of (14.19)} \end{aligned} \quad (14.30)$$

At the same time,  $|b_w|$  is reduced. Hence, the effect of the noise in the determination of disturbance signal  $v(k)$  becomes negligible. In summary we have,

- $|a_v| \approx 0$ :  $v(k)$  is (delayed) pure white noise
- $|a_v| < 1$ :  $v(k)$  is colored noise
- $|a_v| \approx 1$ :  $v(k)$  depends mainly on its past values

If  $|a_v| \approx 0$  and  $|b_w| = 1$ , the disturbance affecting the system is delayed pure white noise. The response cannot be improved by a controller of augmented order. On the other hand, if  $|a_v| \approx 1$ , the gain  $|b_w|$  is small and the resulting output highly correlated with its past values. From equation (14.29) we know that  $v(k)$  is affected by all previous values of the white noise, i.e.  $E\{v(k)w(k-i)\} \neq 0, i = 1 \dots k$ . Thus, the effect of past values of  $w(\cdot)$  can be indirectly observed through the autoregressive part of the disturbance model. Since  $|b_w| \approx 0$ , this part dominates the effect of the current white noise input. In this case, a controller based on the augmented system rejects disturbances almost completely. In the example, the elimination of  $v(k)$  yields:

$$y(k+1) = (a_v + a)y(k) - a a_v y(k-1) - a_v u(k-1) + u(k) + b_w w(k-1) \quad (14.31)$$

Since  $|b_w| \approx 0$ , the term due to the white noise  $w(k-1)$  is neglected in the control law:

$$u(k) = y^*(k+1) - (a_v + a)y(k) + a a_v y(k-1) + a_v u(k-1) \quad (14.32)$$

If  $v(k)$  is highly correlated with its past values the impact of the sum of the past values of the noise is large with respect to present ones. In such a case a substantial improvement of the performance is obtained, if the order of the controller is augmented. The same qualitative behavior is observed if the disturbance system is of order  $n_v$  as defined in (14.18). Its impulse response is given by

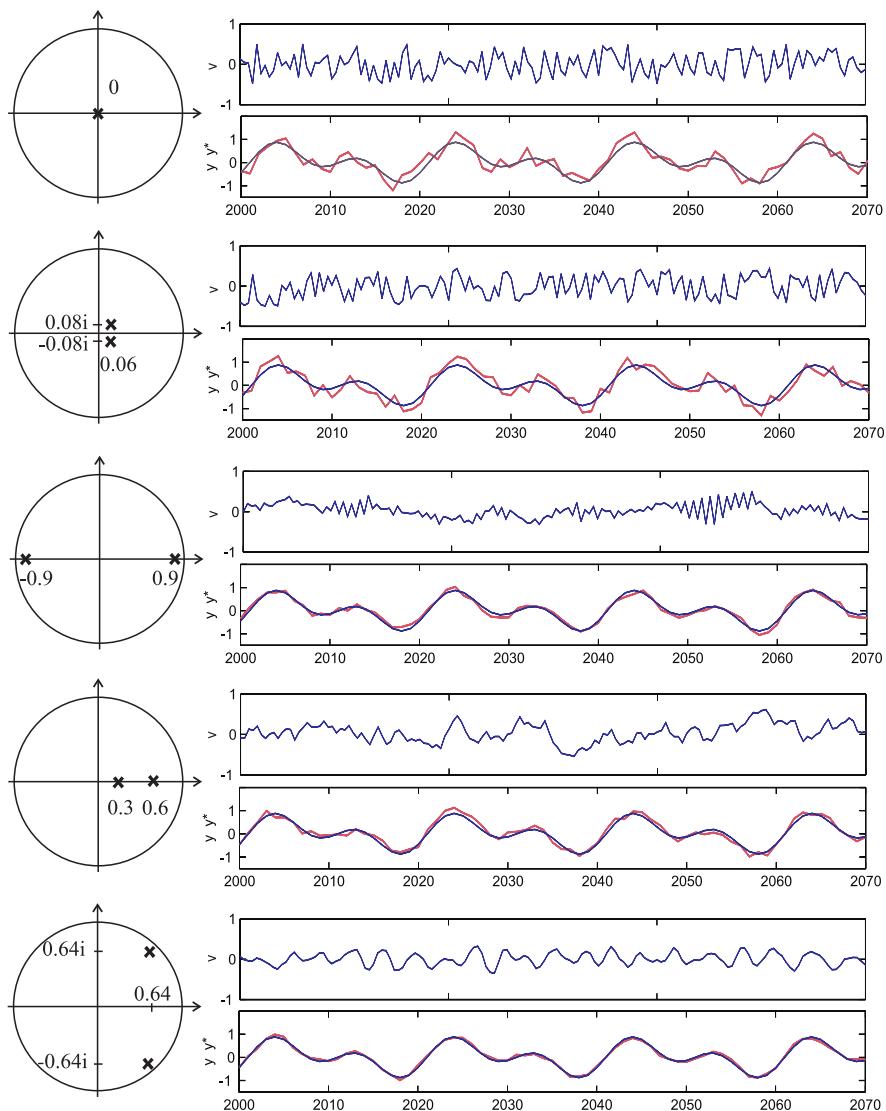
$$h(k) = \underline{c}_v^T \mathbf{A}_v^{k-1} \underline{b}_w \quad k \geq 1$$

If the eigenvalues of  $\mathbf{A}_v$  are close to the unit circle, the impulse response decays slowly and the contribution of past values of the noise is large. At the same time, the elements of the vectors  $\underline{c}_v^T$  and  $\underline{b}_w$  have to be made small such that the maximum amplitude of the disturbance is the same for all stable eigenvalues of  $\mathbf{A}_v$ . This is best illustrated by considering a disturbance generating system of order  $n_v = 2$ . Supposing that we have a pair of complex conjugate eigenvalues which are close to the unit circle, the output  $v(k)$  is a damped sinusoid, which is the natural response of the homogeneous part of the filter, while the contribution of  $w(k)$  to the output  $v(k)$  is small. Therefore, the disturbance can be treated as the output of a homogeneous system and expansion of the state space can be used to reject them.

### Example:

Let the system

$$\begin{aligned} \underline{x}(k+1) &= \begin{bmatrix} 0 & 0.2 \\ 1 & 0.1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} v(k) \\ y(k) &= [1 \ 2] \underline{x}(k) \end{aligned} \quad (14.33)$$



**Abb. 14.5:** Plant output  $y$  versus reference output  $y^*$  for various locations of the poles of the disturbance model

and disturbance model

$$\begin{aligned}\underline{x}_v(k+1) &= \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} \underline{x}_v(k) + \begin{bmatrix} b_{v1} \\ b_{v2} \end{bmatrix} w(k) \\ v(k) &= \begin{bmatrix} c_{v1} & c_{v2} \end{bmatrix} \underline{x}_v(k)\end{aligned}\quad (14.34)$$

be of second order. The eigenvalues are  $z_1 = 0.5$ ,  $z_2 = -0.4$  and  $z_{1v,2v} = \alpha \pm i\beta$  respectively. In the simulation,  $z_{v1}, z_{v2}$  are placed arbitrarily within the unit circle corresponding to different degrees of correlation of  $v(k)$  with its past values. The objective is to investigate if a control law based only on the autoregressive part of the disturbance model would yield acceptable results. Since the gain  $|v|/|w|$  also depends on the location of the eigenvalues, the correction factors  $c_{v1}$  and  $c_{v2}$  were chosen such that the peak-to-peak variation of  $v(k)$  remained the same throughout the experiment in order to obtain comparable results. Since both systems are observable the corresponding input-output representations exist. After the elimination of  $v(k)$  we obtain the augmented 4<sup>th</sup>-order system:

$$\begin{aligned}y(k+1) = & a_0y(k) + a_1y(k-1) + a_2y(k-2) + a_3y(k-3) + b_0u(k) + \\ & + b_1u(k-1) + b_2u(k-2) + b_3u(k-3) + c_2w(k-2) + c_3w(k-3)\end{aligned}\quad (14.35)$$

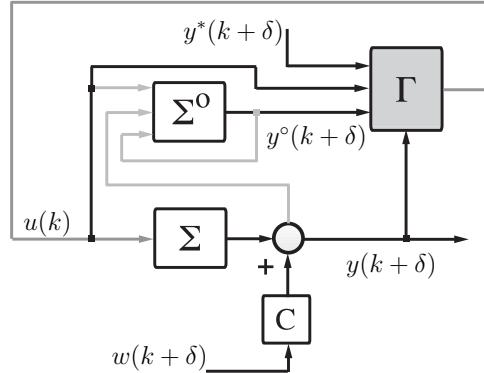
where  $a_i, b_i, c_i$  are constants depending only on the location of the eigenvalues  $z_{1v,2v}$  of the disturbance model. With  $y^*(k+1) = 0.5 \sin(\frac{2\pi k}{10}) + 0.5 \sin(\frac{2\pi k}{20})$  being the reference signal, the control law for asymptotic tracking of the reference output is given by:

$$u(k) = [y^*(k+1) - a_0y(k) - a_1y(k-1) - a_2y(k-2) - a_3y(k-3) - \\ - b_1u(k-1) - b_2u(k-2) - b_3u(k-3)] / b_0 \quad (14.36)$$

Again, the contribution of the white noise to the current output is neglected. The simulation results are displayed in figure 14.5. In the first case, the eigenvalues of the disturbance model are  $z_{1v} = z_{2v} = 0$  and hence the disturbance  $v(k)$  affecting the system is pure white noise  $v(k) = z^{-2}w(k)$ . Since there is no correlation of the noise  $v(k)$  with its past values, the performance of the controller is poor. A similar result is obtained in the second case, where  $z_{1v}, z_{2v}$  are close to zero. In the subsequent cases, the eigenvalues are placed closer to the unit circle corresponding to a large time constant of the impulse response of the filter (14.34). Since previous values of the output  $v(k)$  are now the dominant part of the disturbance, almost complete rejection is achieved using a controller (14.36) based on the augmented system. This becomes particularly evident in the last case, where a pair of complex conjugate eigenvalues with  $|z_{v1,2}| \approx 1$  is considered and  $v(k)$  is almost harmonic.

#### 14.1.2.2 Stochastic Adaptive Control

In principle, an adaptive version of the above control laws can be obtained by simply replacing the coefficients in (14.32) or (14.36) by their estimated values.



**Abb. 14.6:** Minimum variance control

A question arises, as to how the estimation model should be chosen. If the noise terms are neglected in the definition of the regression vector, the resulting output error will have zero mean since  $E\{w(k)\} = 0$  for all  $k > 0$ , but its variance may be large. In this paragraph we briefly discuss the stochastic adaptive control problem and present one of its solutions.

In studying this problem it is standard to assume that the white noise sequence  $\{w(k)\}$  not only drives the model generating the stochastic disturbance input  $v(k)$  but also affects the output of the system  $y(k)$  directly. Let us start with a simple example:

$$y(k+1) = ay(k) + cw(k) + w(k+1) + u(k) \quad (14.37)$$

with  $|a|, |c| < 1$ .  $w(k)$  cannot be used in the definition of the estimation model, since it is not measurable. The idea is to use the model error  $\varepsilon(k) = y(k) - \hat{y}(k)$  instead of  $w(k)$ :

$$\hat{y}(k+1) = \hat{a}y(k) + \hat{c}\varepsilon(k) + u(k) \quad (14.38)$$

The motivation for using  $\varepsilon(k)$  to replace  $w(k)$  is that, if the parameters converge to their true values, the residual identification error  $\varepsilon(k) = y(k) - \hat{y}(k)$  is white noise  $w(k)$ . By virtue of the certainty equivalence principle, the control law is obtained by replacing  $\hat{y}(k+1)$  in (14.38) by  $y^*(k+1)$ :

$$u(k) = y^*(k+1) - \hat{a}y(k) - \hat{c}\varepsilon(k) \quad (14.39)$$

It is obvious that no  $u(k)$  at time  $k$  can compensate for the noise  $w(k+1)$  at time  $k+1$ . However, the controller asymptotically cancels out the effect of the noise  $w(k)$  at instant of time  $k$ . The expected value of the squared error satisfies  $E\{|y(k+1) - y^*(k+1)|^2 | k\} = E\{w^2(k+1) | k\} = \sigma^2$  as  $k \rightarrow \infty$ , and equation (14.39) is referred to as a minimum variance controller.

The concept can be generalized as follows. The plant is usually given as an ARMAX model:

$$A(z^{-1})y(k) = z^{-\delta}B(z^{-1})u(k) + C(z^{-1})w(k) \quad (14.40)$$

where  $A$ ,  $B$  and  $C$  are polynomials in the delay operator  $z^{-\delta}$ , of degrees  $n_A$ ,  $n_B$  and  $n_C$  respectively:

$$\begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_A}z^{-n_A} \\ B(z^{-1}) &= b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{n_B}z^{-n_B} \\ C(z^{-1}) &= 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{n_C}z^{-n_C} \end{aligned} \quad (14.41)$$

It is assumed that the roots of  $C(z^{-1})$  lie strictly inside the unit circle.  $\{w(k)\}$  is a white noise sequence with the properties (14.19) to (14.21). Since  $C(z^{-1})$  is monic,  $w(k)$  directly affects the output  $y(k)$  at a given instant  $k$ . The control input is obtained by minimizing the mean-square tracking error  $E\{[y(k+d) - y^*(k+d)]^2\}$ . It can be shown (see Goodwin, 1984 [68]) that this is equivalent to

$$\min_{u(k)} E\{[y^o(k+\delta|k) - y^*(k+\delta)]^2\} \quad (14.42)$$

where  $y^o(k+\delta|k)$  is the optimal  $d$ -step ahead prediction of  $y(k)$ :

$$y^o(k+\delta|k) = E\{y(k+\delta)|k\} \quad (14.43)$$

In certainty equivalence control, the estimated parameters are used to determine  $u(k)$ . From the discussion in chapter 13 it is clear that the control input is based upon an underlying estimation process. More precisely,  $u(k)$  is defined implicitly by an equation of the form  $y^*(k+\delta) = \underline{\phi}(\cdot)^T \hat{\theta}(\cdot)$  where the regression vector  $\underline{\phi}(\cdot)$  is yet to be defined. If the parameter estimates are determined such that  $\underline{\phi}(\cdot)^T \hat{\theta}(\cdot)$  is equal to  $y^o(k+\delta|k)$ , then the corresponding  $u(k)$  clearly minimizes the expression in (14.42). In other words, the optimum is attained whenever  $\underline{\phi}(\cdot)^T \hat{\theta}(\cdot) \rightarrow y^o(k+\delta|k)$ .

Numerous methods have been proposed in the identification literature to deal with the problem of parameter estimation in a stochastic environment (see e.g. Landau, 1998 [133]). As an example, we present the **extended least squares** (ELS) algorithm which belongs to the class of pseudo linear regression algorithms. The distinctive feature of this class of algorithms is that the components of the regression vector depend upon previous values of the estimated parameters. As described above, the idea is to design an estimator producing an error  $\varepsilon(\cdot)$  that becomes white noise asymptotically.

Equation (14.40) can be rewritten as

$$y(k) = [1 - A(z^{-1})]y(k) + z^{-\delta}B(z^{-1})u(k) + C(z^{-1})w(k) \quad (14.44)$$

Adding and subtracting  $[C(z^{-1}) - 1]\varepsilon(k)$  we obtain

$$\begin{aligned} y(k) &= [1 - A]y(k) + z^{-\delta}Bu(k) + Cw(k) + [C - 1]\varepsilon(k) - [C - 1]\varepsilon(k) \\ &= \underline{\phi}(k-1)^T\underline{\theta}_0 + Cw(k) - [C - 1]\varepsilon(k) \end{aligned} \quad (14.45)$$

using

$$\begin{aligned} \underline{\phi}(k-1) &= [y(k-1), \dots, y(k-n_A), u(k-\delta), \dots, u(k-\delta-n_B), \\ &\quad \varepsilon(k-1), \dots, \varepsilon(k-n_C)]^T \\ \underline{\theta}_0 &= [-a_1, \dots, -a_{n_A}, b_0, \dots, b_{n_B}, c_1, \dots, c_{n_C}]^T \end{aligned}$$

The elements of  $\underline{\theta}_0$  are the coefficients of the polynomials in equation (14.41). The identification model reads:

$$\hat{y}(k) = \underline{\phi}(k-1)^T \hat{\underline{\theta}}(k-1) \quad (14.46)$$

Subtracting (14.46) from (14.45) we obtain:

$$\varepsilon(k) = \frac{1}{C(z^{-1})} \underline{\phi}(k-1)^T [\underline{\theta}_0 - \hat{\underline{\theta}}(k-1)] + w(k) \quad (14.47)$$

Using  $\varepsilon(k)$ , the parameters are updated according to

$$\hat{\underline{\theta}}(k) = \hat{\underline{\theta}}(k-1) + \mathbf{P}(k-1) \underline{\phi}(k-1) \varepsilon(k) \quad (14.48)$$

where  $\mathbf{P}(k-1)$  is a matrix satisfying

$\mathbf{P}(k)^{-1} = \mathbf{P}(k-1)^{-1} + \underline{\phi}(k)\underline{\phi}(k)^T$ ,  $\mathbf{P}(-1)^{-1} > 0$ . Note that  $\hat{y}(k)$  in equation (14.46) is called the *a priori* prediction, obtained *before* the parameters have been updated. The algorithm can be improved by making use of the most recent parameter estimate. We obtain the *a posteriori* prediction:

$$\bar{y}(k) = \underline{\phi}(k-1)^T \hat{\underline{\theta}}(k) \quad (14.49)$$

A posteriori prediction errors are sometimes used in the definition of the regression vector to simplify the convergence analysis of the algorithm. However, the distinction is of secondary importance. In any case, the entries of the regression vector depend upon previous values of the estimated parameters. It is seen that  $\varepsilon(k)$  will become white noise asymptotically if the parameter error  $\tilde{\underline{\theta}}(k-1) = [\underline{\theta}_0 - \hat{\underline{\theta}}(k-1)] \rightarrow 0$  or, alternatively, the first term in equation (14.47) tends to zero.

The a posteriori estimate  $\bar{y}(k)$  obtained by the ELS algorithm is the optimal one-step-ahead prediction of  $y(k)$ , i.e.  $\bar{y}(k) = E\{y(k)|k-1\}$ . Equation (14.49) allows us to determine the effect of the past values of the noise on the future response of the plant. With the justification provided above, the adaptive controller is designed on the basis of the optimal prediction of the system output. The control input is obtained by replacing the predicted output  $\bar{y}(k)$  by the desired output  $y^*(k)$ .

$$y^*(k) = \underline{\phi}(k-1)^T \hat{\underline{\theta}}(k) \quad (14.50)$$

When solving this equation for  $u(k)$  it follows that all past noise terms are cancelled out. Hence, the control law is the stochastic equivalent of the deadbeat-control law in the deterministic case. The residual error has minimum variance. To ensure convergence to the minimum variance controller the following conditions have to be satisfied:

1. The orders  $n_A$ ,  $n_B$  and  $n_C$  of the polynomials in equation (14.41) are known.
2. The relative degree  $\delta$  is known and equal to 1 (in this case).
3.  $B(z^{-1})$  has all roots inside the unit circle.
4.  $[1/C(z^{-1}) - \frac{1}{2}]$  is positive real.

While conditions 1 to 3 are the same as in the deterministic case, condition 4 is characteristic for estimation techniques based on pseudo linear regressions. It ensures that the extended least-squares algorithm has convergence properties similar to those of its deterministic counterpart. For further insight, see e.g. [140]. Condition 2 can be relaxed to allow for  $\delta > 1$ . Subject to the above assumptions regarding the system and assumptions (14.19) to (14.20) regarding the noise we obtain that both  $u(k)$  and  $y(k)$  are mean square bounded and that the output  $y(k)$  of the plant tracks  $y^*(k)$  with minimum variance:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N E[y(k) - y^*(k) | k-1]^2 = \sigma^2 \quad (14.51)$$

where  $\sigma^2$  is the variance of the white noise  $w(k)$  affecting the system.

**Comment:** The augmented system (14.23) can be written in the form (14.40), except that  $C(z^{-1})$  is not monic, since the  $v(\cdot)$  in equation (14.1) does not directly affect the output  $y(\cdot)$ . The derivation of the ELS algorithm was based on the assumption that the a posteriori estimation error becomes white noise asymptotically. Hence, the fact that  $C(z^{-1})$  is monic is critical and we have the assume that  $\{w(k)\}$  directly affects the output.

### Example:

Given a second-order plant subject to a stochastic input disturbance  $v$  and affected at the output by white noise  $w$ ,

$$[1 + a_1 z^{-1} + a_2 z^{-2}] y(k) = z^{-1}[b_0 + b_1 z^{-1}] u(k) + z^{-1} g_0 v(k) + w(k) \quad (14.52)$$

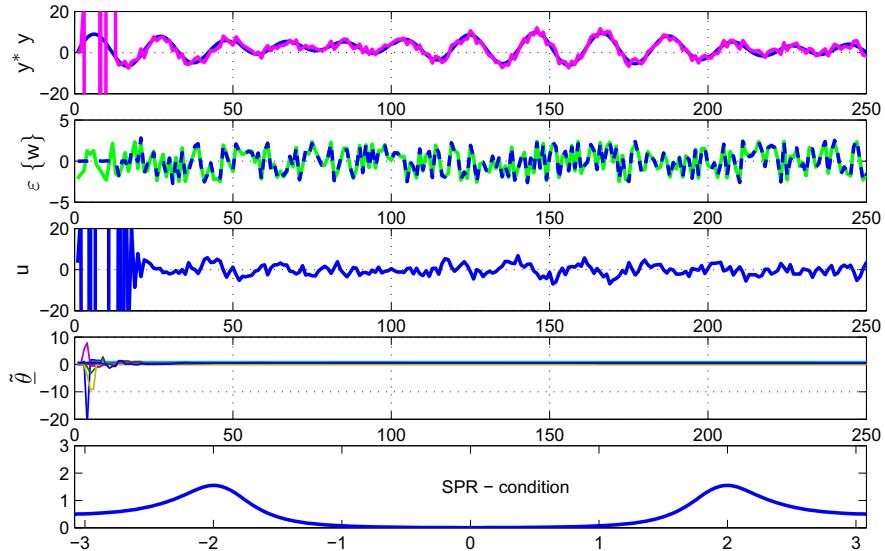
where  $v(\cdot)$  is the output of a first-order system driven by white noise:

$$[1 + \delta_1 z^{-1}] v(k) = z^{-1} w(k) \quad (14.53)$$

The augmented system obtained by eliminating  $v(k)$  in equation (14.52) reads

$$\begin{aligned} & [1 + (\delta_1 + a_1) z^{-1} + (a_2 + \delta_1 a_1) z^{-2} + \delta_1 a_2 z^{-3}] y(k) \\ & = z^{-1}[b_0 + (b_1 + b_0 \delta_1) z^{-1} + \delta_1 b_1 z^{-2}] u(k) + [1 + \delta_1 z^{-1} + g_0 z^{-2}] w(k) \end{aligned} \quad (14.54)$$

and is of the form (14.40). All coefficients are assumed to be unknown, but  $C(z^{-1}) = 1 + \delta_1 z^{-1} + g_0 z^{-2}$  satisfies the conditions stated in point 3 of the above list. The  $d = 1$  step ahead identification model is of the form  $\hat{y}(k+1) = \underline{\phi}(k)^T \hat{\theta}(k)$  where



**Abb. 14.7:** Adaptive minimum-variance control

$$\underline{\phi}(k) = [y(k) \ y(k-1) \ y(k-2) \ u(k) \ u(k-1) \ u(k-2) \ \varepsilon(k) \ \varepsilon(k-1) \ \varepsilon(k-2)]^T \quad (14.55)$$

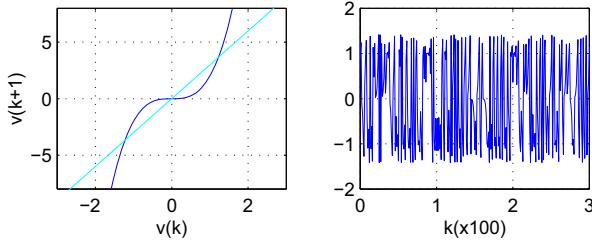
$\hat{\theta}(k) \in \mathbb{R}^9$  is the vector of parameter estimates and  $\varepsilon(k) = \hat{y}(k) - y(k)$ .  $\hat{\theta}(k)$  contains the parameters upon which the adaptive minimum variance control law is based:

$$u(k) = 1/\hat{\theta}_4(k) [\hat{y}^*(k+1) - \hat{\theta}_1(k)y(k) - \hat{\theta}_2(k)y(k-1) - \hat{\theta}_3(k)y(k-2) - \hat{\theta}_5(k)u(k-1) - \hat{\theta}_6(k)u(k-2) - \hat{\theta}_7(k)\varepsilon(k) - \hat{\theta}_8(k)\varepsilon(k-1) - \hat{\theta}_9(k)\varepsilon(k-2)] \quad (14.56)$$

The simulation displayed in figure 14.7 reveals, that the closed loop system tracks an arbitrary reference input and that the control error becomes white asymptotically. Due to the presence of noise, the degree of excitation is superior to the one obtained when no disturbances are present. As a consequence, the parameter error  $\hat{\theta}$  converges to zero. In the last row, the SPR condition has been verified, i.e.  $[C(z^{-1}) - 1/2]$  has been evaluated for all  $z^{-1} = e^{j\omega h}$ ,  $\omega h = -\pi \dots \pi$ .

## 14.2 Nonlinear Disturbance Rejection

In this section, we consider the case where the plant as well as the disturbance generating system are nonlinear:



**Abb. 14.8:** Nonlinear oscillation generated by a first-order discrete-time system

$$\begin{aligned}\underline{x}(k+1) &= \underline{f}[\underline{x}(k), u(k), v(k)] \\ y(k) &= h[\underline{x}(k)] \\ \underline{x}_v(k+1) &= \underline{f}_v[\underline{x}_v(k)] \\ v(k) &= h_v[\underline{x}_v(k)]\end{aligned}\tag{14.57}$$

where  $\underline{f} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^n$ ,  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $\underline{f}_v : \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_v}$ ,  $h_v : \mathbb{R}^{n_v} \rightarrow \mathbb{R}$  are unknown  $C^1$ -functions of their arguments. As in the linear case, the objective is to determine a control input  $u(k)$  such as to make  $\underline{x}_v(k)$  unobservable through the output  $y(k)$ . Our interest is in bounded nonvanishing disturbances  $v(k)$ . In the nonlinear domain, the class of unforced systems which generate a disturbance signal having the required properties is not immediately evident. We assume that the disturbance model is of the following form:

$$v(k+1) = \alpha v(k) - \beta f_v[v(k)]\tag{14.58}$$

where  $\alpha > 1$  and  $\beta$  are constant parameters. Since the linear part of the system (14.58) is unstable,  $\beta$  has to be chosen such that the nonlinearity  $f_v[v(k)]$  prevents  $v(k)$  from growing in an unbounded fashion. If so,  $v(k)$  lies in a compact set  $S \subset \mathbb{R}$ . The initial value  $v(0)$  has to be chosen from this set.

### Example:

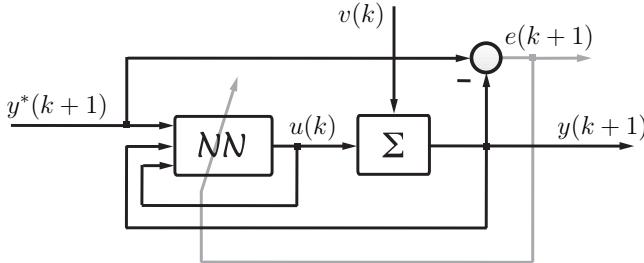
$$v(k+1) = 3v(k) - 2v(k)^3\tag{14.59}$$

For this particular choice of the parameters and for the initial condition  $v(0) \in (-\sqrt{2}; \sqrt{2}) \setminus \{-1, 0, 1\}$  (14.59) generates a bounded nonvanishing output, shown in figure 14.8. The figure also displays the linear and nonlinear parts of equation (14.59) in a one-dimensional map which illustrates the mechanism: Starting from an initial point  $v(0)$  near the origin,  $v(k)$  at first grows linearly until the nonlinear function becomes dominant for larger values of  $v(k)$ . The parameter  $\beta$  is chosen such that  $v(k)$  is mapped back to a point  $v^*(0)$  within the linear, unstable region. Since  $v^*(0)$  does not necessarily coincide with the original point  $v(0)$ , the oscillation may not be periodic. In fact, systems of the form (14.58) with  $\alpha > 1$

may give rise to quasiperiodic and chaotic motion. A classical example is the logistic map

$$v(k+1) = 4v(k) - 4v^2(k)$$

for values  $v(0) \in [0, 1]$ . Hence, in order to represent e.g. a periodic disturbance, the function  $f_v(\cdot)$  in equation (14.58) has to be appropriately chosen.



**Abb. 14.9:** Neurocontrol in the presence of disturbances

The principle of nonlinear disturbance rejection is the same as in the linear case and relies upon a system representation of augmented order. This representation is obtained by eliminating  $v(k)$  from the input–output model of the plant. The existence of an input–output model of the nonlinear state–vector representation is seen to depend on the observability of the linearized system. A similar condition is needed in order to derive an input–output model of the composite system . The linearized equations are given by:

$$\begin{aligned} \underline{x}(k+1) &= \mathbf{A} \underline{x}(k) + \underline{b} u(k) + \underline{b}_v v(k) \\ \underline{x}_v(k+1) &= \mathbf{A}_v \underline{x}_v(k) \\ y(k) &= \underline{c}^T \underline{x}(k) \\ v(k) &= \underline{c}_v^T \underline{x}_v(k) \end{aligned} \quad (14.60)$$

where

$$\begin{aligned} \mathbf{A} &= \frac{\partial f(x, u, v)}{\partial x} \Big|_0 \in \mathbb{R}^{n \times n} \\ \underline{b} &= \frac{\partial f(x, u, v)}{\partial u} \Big|_0 \in \mathbb{R}^n \\ \underline{b}_v &= \frac{\partial f(x, u, v)}{\partial v} \Big|_0 \in \mathbb{R}^n \\ \underline{c}^T &= \frac{\partial h(x)}{\partial x} \Big|_0 \in \mathbb{R}^n \\ \text{and } \mathbf{A}_v &= \frac{\partial f_v(x_v)}{\partial x_v} \Big|_0 \in \mathbb{R}^{n_v \times n_v} \\ \underline{c}_v^T &= \frac{\partial h_v(x_v)}{\partial x_v} \Big|_0 \in \mathbb{R}^{n_v} \end{aligned}$$

The Jacobians are evaluated at the origin of the composite system. In the linear case, it was seen that if the pairs  $[\underline{c}^T, \mathbf{A}]$  and  $[\underline{c}_v^T, \mathbf{A}_v]$  are observable, the system

has a linear input-output representation of dimension  $(n + n_v)$ , given by equation (14.8). Under the same conditions, a nonlinear input-output map of the form

$$y(k + \delta) = \mathcal{F}[Y_{n+n_v}(k), U_{n+n_v}(k)] \quad (14.61)$$

exists **locally** in the neighborhood of the origin in extended state-space  $\mathbb{R}^{n+n_v}$ . The map describes a system of augmented order where  $Y_{n+n_v}(k) = [y(k), \dots, y(k - \langle n + n_v - 1 \rangle)]$ ,  $U_{n+n_v}(k) = [u(k), \dots, u(k - \langle n + n_v - 1 \rangle)]$  and  $\delta \geq 1$ . Given the augmented system, the problem of disturbance rejection consists in determining a tracking control law such that the closed-loop system is asymptotically stable and tracks any bounded reference trajectory  $\{y^*(k)\}$ . Since the explicit dependence on  $v(k)$  has been removed, the design of the controller proceeds as in the disturbance-free case. A solution to the problem exists provided that the relative degree  $\delta$  is well defined and the zero-dynamics are stable, see (Cabrera and Narendra, 1999 [30]). Since the disturbance rejection problem is solved by designing a controller based on the NARX-model<sup>2)</sup> of the system of the augmented system we have to assume that the values of inputs  $\{u(k)\}$  and outputs  $\{y(k)\}$  do not leave the neighborhood  $\mathcal{U} \times \mathcal{Y}$  in which the NARX-model is valid. This implies that the reference trajectory  $\{y^*(k + \delta)\}$  lies in that neighborhood.

### Example:

The following example reveals some practical aspects while designing a neuro-controller for disturbance rejection. We consider the second-order system,

$$y(k + 1) = \frac{3y^2(k)[1 - v^2(k)] \tanh y(k - 1) + u(k)}{1 + y^2(k) + y^2(k - 1)} \quad (14.62)$$

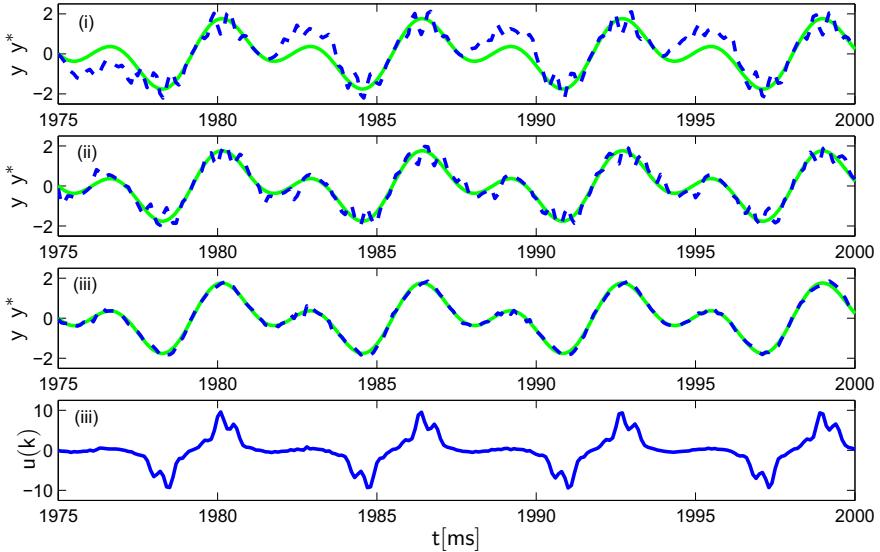
In the simulation, the bounded disturbance  $v(k)$  was chosen to be the filtered output of equation (14.59), described by the following homogenous input-output map:

$$v(k + 1) = 0.2v(k) + 0.2v(k - 1) + [v^2(k - 1) - 1][v(k) - v(k - 1)] \quad (14.63)$$

From the above discussion it is clear that there exists a local NARX representation of the composite system which is of augmented order  $n + n_v = 4$ . However, control based this model was found to result in poor performance. Hence the order of the NARX-model was further increased. This can be thought of as a way of increasing the domain in which the augmented input-output representation is valid. If the composite system is of 5th-order, almost complete disturbance rejection was obtained:

---

<sup>2)</sup> In reference [30] this is referred to as a NARMA-model where the MA-part corresponds to the moving-average of the (deterministic) input in contrast to the notation used here.



**Abb. 14.10:** Tracking performance using a neurocontroller; (i): no disturbance rejection, (ii): disturbance rejection with insufficient number of input-output measurements, (iii): application of the control law defined in equation (14.65).

$$y(k+1) = \mathcal{F}(Y_5(k), U_5(k)) \quad (14.64)$$

with  $Y_5(k) = [y(k), \dots, y(k-4)]^T$  and  $U_5(k) = [u(k), \dots, u(k-4)]^T$ . A multilayer neural network (MNN) of the form  $\mathcal{N}_{10,25,35,1}^3$  (i.e. a three-layered NN with a single output, 35 neurons in the first hidden layer, 25 neurons in the second hidden layer, and a 10-dimensional input vector) is used to identify the augmented system (14.64). After 350000 iterations with a step size of  $\eta = 0.1$ , perfect matching of the model and plant is achieved. Once the identification process is complete, a second neural net is trained to approximate the nonlinear map representing the controller:

$$u(k) = \mathcal{N}_c(y^*(k+1), y(k), \dots, y(k-4), u(k-1), \dots, u(k-4)) \quad (14.65)$$

The network is of the form  $\mathcal{N}_{10,15,25,1}^3$  and is trained for 100000 steps with a step size of  $\eta = 0.01$ . Note that, since the controller is in the feedback-loop of a dynamical system, the parameter adjustments in the second neural network have to be carried out using dynamic backpropagation. However, dynamic gradient methods are computationally intensive, so a static method is used to generate approximate gradients. The resulting tracking performance is shown in figure 14.10 where the reference trajectory was chosen to be  $y^*(k+1) = \sin(2\pi k/10) + \sin(2\pi k/20)$ .

When comparing the results of the last two chapters one realizes, that most arguments from linear theory have nonlinear counterparts. Although the results are valid only in a certain neighborhood of the equilibrium state, they provide the justification for transferring linear methods to certain nonlinear problems. In fact, the problem of rejecting disturbances which are the output of an unforced system was solved for both linear and nonlinear systems by using the same concept of extending the order of the controller.

While dealing with stochastic disturbances, it was seen that the above method results in good performance provided that the disturbance is highly correlated with its past values. In addition, it was shown that the estimation error could be used to cancel out the effect of past values of the noise. This resulted in a minimum variance control law. It is only natural to assume, that a similar method exists in the nonlinear domain, if the latter is restricted to a neighborhood of an equilibrium state. Further work is needed to investigate whether the variance of the control error can be minimized even as the plant is nonlinear

$$y(k) = f[y(k-1), \dots, y(k-n_A), u(k-1), \dots, u(k-n_B), w(k), \dots, w(k-n_C)] \quad (14.66)$$

and  $\{w(k)\}$  is a white noise sequence.

### 14.3 Time-Varying Disturbances

The third class of disturbances considered in this chapter is due to large and sudden changes in the parameters of the system. In fact, one of the primary reasons for considering adaptive control in practical applications is to compensate for large variations of the plant parameters. Despite this fact, a coherent theory of adaptive control exists only when the unknown system is time-invariant. The accepted philosophy is that if an adaptive system is fast and accurate when the plant parameters are constant but unknown, it would also prove satisfactory when the parameters vary with time, provided the latter occurred on a relatively slower time-scale.

The analytical difficulties encountered while dealing with unknown, time-varying systems are due to the following factors. If  $\underline{\theta} : \mathbb{N}_0 \rightarrow \mathbb{R}^{2n}$  is the (time-dependent) control parameter vector which is used to compensate for variations in plant parameters, it must first be shown that a vector function  $\underline{\theta}^* : \mathbb{N}_0 \rightarrow \mathbb{R}^{2n}$  exists such that the control error is zero when  $\underline{\theta}(k) \equiv \underline{\theta}^*(k)$ . In practice, this is not easy to accomplish since the analysis of the resulting system involves time-varying operators. Even when  $\underline{\theta}^*(k)$  has been shown to exist, the derivation of a stable adaptive law is difficult since the error equations involved are nonlinear and non-autonomous. Using the parameter error  $\tilde{\theta}(k) = [\hat{\theta}(k) - \underline{\theta}^*(k)]$  one may attempt to apply a standard adaptive algorithm to adjust  $\hat{\theta}(k)$ . Due to the time-variation of  $\underline{\theta}^*$ , an additional term  $\Delta\underline{\theta}^*(k) = [\underline{\theta}^*(k) - \underline{\theta}^*(k-1)]$  appears on the right hand side of the update equation which makes the system non-autonomous. Moreover, convergence has to be proven (in a pointwise fashion) in function space.

Results have been obtained in the continuous-time case under the assumption that the unknown plant parameter vector is the output of an asymptotically stable linear time-invariant system with constant input. In this case, stability was established without modifications of the adaptive law (see Narendra, 1989 [158]). Simulation studies have been carried out for the case where the dynamics of the plant parameters is governed by a linear second-order system. When the frequency of the parameter variation is low, a standard adaptive law was seen to result in a small output error. For general time-variations, a robust adaptive control approach has been proposed by (Tsakalis, 1987 [226]). Discrete-time results have been obtained by (Kreisselmeier, 1982 [129]) and (Narendra, 1987 [163]). In the latter case, a bound on the parameter  $\underline{\theta}^*(k)$  and its variation  $\Delta\underline{\theta}^*(k)$  was established for which the standard adaptive law ensures stability. For an overview, see (Ioannou, 1996 [109]).

All the above results are based on the assumption that the parameter variation is slow in some sense. If this is not satisfied, but the parameter perturbations are small in magnitude, robust control is generally preferred. However, as systems become more complex, situations where the parameters vary both rapidly and by large amounts arise with increasing frequency. Typical examples are mechanical processes with large variations in load, actuator failures or transition control tasks in chemical systems. It was to cope with such situations that a methodology based multiple models has been introduced by Narendra and Balakrishnan in 1992 [159]. The general methodology as well as the principal results are summarized below (see also Narendra et al., 2003 [162]).

### 14.3.1 Multi-Model Adaptive Control

From the very beginning, the interest of adaptive control theorists was centred around adaptation in changing environments. However, due to mathematical tractability, they confined their attention to time-invariant systems with unknown parameters. The control algorithms that have proven stability properties are characterized by an adaptive law which is given in the form of a differential equation. Once the evolution of the parameters towards their true values (or –more generally– towards a solution manifold) is governed by a differential equation, the long-term behavior of the adaptive system can be studied in terms of a dynamical system and becomes amenable to powerful tools from the qualitative theory of nonlinear differential equations (e.g. convergence of the parameters is studied in terms of the stability of an associated dynamical system). It is clear, though, that this may not be the fastest way to determine the parameters since adaptation thus defined is inherently incremental. Note that when the unknown parameters appear nonlinearly, it may even be impossible to find a differential equation that describes the evolution of the parameter estimates towards their true values. In these cases, the parameters are determined by optimizing an error function, e.g.  $\hat{\theta} = \operatorname{argmin}\{\varepsilon(\hat{\theta})^2\}$ , and the resulting trajectory in parameter space

is generally a non-smooth curve. It is in the same spirit that one may wish to allow for switching while adapting the parameters of an unknown system.

The advantage of switching would be to provide fast response to time-variations occurring in the system. Even, if no such variations take place, the tracking error is quite often unacceptably large and oscillatory during the initial phase of parameter convergence, when there are large errors in the estimates. In this section we present a methodology, which –in addition to providing improved response in the case of time-invariant parameters– opens up new ways of controlling systems with large and sudden variations in the parameters.

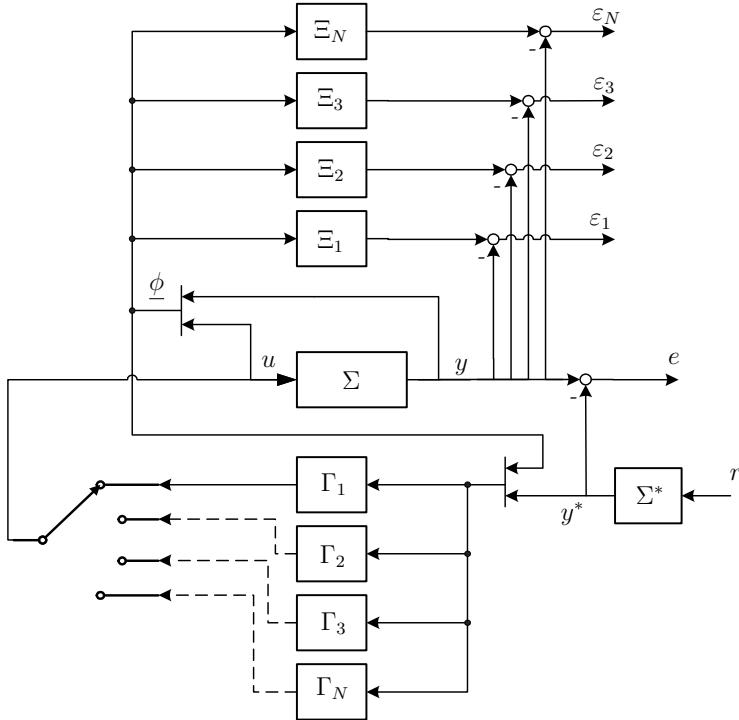
Since the method was originally proposed in 1992, many other approaches have been reported in the adaptive control literature. However, only a few have been demonstrated to be stable. Since it is well known that efficient design methods for different classes of control systems can be developed only when their stability properties are well understood, we focus our attention on stability issues in multi-model based adaptive control. In particular, we will be interested in the assumptions that have to be made to assure stability.

#### 14.3.1.1 General Methodology

In this section the motivation of the approach as well as its basic concepts are presented. The basic idea for using multiple models comes from biology: every biological system is faced with a multiplicity of choices at any instant of time. As the environment changes, it demonstrates an ability to rapidly modify its strategy such as to maintain optimal performance. Such an ability involves recognizing the specific situation that has arisen and taking an appropriate control action which is selected from a set of available strategies. In order to acquire a repertoire of strategies the biological system learns from past experience. Hence, it is the ability of the system to learn and store information, and combine it with adaptation that is responsible for its performing satisfactorily in rapidly varying situations. It is this feature of biological system behavior that underlies the idea of multi-model adaptive control.

A mathematical description that captures the essential aspects of a physical system is generally referred to as a *model* of that system. It is clear that the kind of models generated for a given system may vary significantly according to what the designer *regards to be* its essential characteristics. Since it is reasonable to assume that diverse models may be appropriate for describing different environments, the use of multiple models arises naturally. The basic architecture for adaptive control based on multiple models (MMST) may be described as follows:

The structure of the control system is shown in figure 14.11. The plant  $\Sigma$  to be controlled has an input  $u$  and output  $y$ . A reference model with piecewise continuous input  $r$  provides the desired output  $y^*$ , and the objective is to cause the control error  $e = y - y^*$  to tend to zero, (or lie within specified bounds) for large values of time.  $N$  identification models  $\Xi_1, \Xi_2, \dots, \Xi_N$  are used in parallel to estimate the parameters of the plant each of which generates an estimated output  $\hat{y}_i$ ,  $i = 1, 2, \dots, N$ . The estimation error of the  $j$ th model  $\Xi_j$  is defined as



**Abb. 14.11:** Multi-Model Adaptive Control (MMST)

$\varepsilon_j = y - \hat{y}_j$ . Corresponding to each model  $\Xi_j$  there exists a controller  $\Gamma_j$  such that  $\Xi_j$  together with  $\Gamma_j$  in the feedback-path behaves like the reference model  $\Sigma^*$ . At every instant, based on a switching criterion, one of the model/controller pairs, e.g.  $[\Xi_j, \Gamma_j]$ , is chosen. Consequently, the output  $u_j$  of the  $j$ th controller is used to control the plant. Given prior information about the plant (e.g. that the latter is linear, nonlinear, stochastic, slowly time-varying etc.), the design problem is to choose the models  $\Xi_j$  and the controllers  $\Gamma_j$  together with the rules for switching to obtain best performance, and demonstrate that the overall system is stable.

For mathematical convenience, as well as for a precise definition of the control problem, it is assumed that the plant and all the identification models (unless otherwise stated) can be parameterized in the same fashion. If the unknown plant parameter is a vector  $\underline{\theta}_0$  and the estimates of  $\underline{\theta}_0$  given by the models are  $\hat{\underline{\theta}}_i$ , we assume that  $\underline{\theta}_0$  and  $\hat{\underline{\theta}}_i (i = 1, 2, \dots, N) \in S \subset \mathbb{R}^p$  where  $S$  is a compact set. For ease of exposition, we shall refer to  $\underline{\theta}_0$  as the plant and  $\hat{\underline{\theta}}_i$  as a model. The general problem of MMST can also be considered as one of choosing  $\hat{\underline{\theta}}_i (i = 1, 2, \dots, N)$  so that for any  $\underline{\theta}_0 \in S$  the control objectives can be achieved and the overall

system is stable. In the paragraphs that follow, further details concerning the models, the switching criterion, and the manner in which the control input is to be computed are discussed.

#### 14.3.1.2 Models

As stated in the introduction, a large variety of models can be chosen including continuous-time or discrete-time, linear and nonlinear models. If the plant is subjected to disturbances, a model of extended order may be called for. Note that if models with feedback are used in the estimation procedure, there is the possibility that the models become unstable for some values of their parameters. In such cases, even when the controlled plant is stable and has a bounded output, the outputs of the models may grow in an unbounded fashion. To avoid this, the models are chosen to be of the *series-parallel type* [158], i.e. the regression vector contains measured input–output values of the plant and is the same for all the models.

Models with constant parameters  $\underline{\theta}_i$  are referred to as fixed models while those which are continuously updated based on input–output data are referred to as adaptive models. Fixed models require very little computational overhead and are mainly used to provide better initial conditions for parameter estimation. Using computer simulations, the use of  $N - 2$  fixed models and two adaptive models was found to be a reasonable compromise between computational complexity and performance in many adaptive problems [160]. If a fixed model  $\hat{\underline{\theta}}_i$  is selected (according to a performance criterion) at any instant  $k_0$ , an adaptive model with parameter vector  $\hat{\underline{\theta}}_a$  is initiated using  $\hat{\underline{\theta}}_i$  as an initial condition. If at a later time  $k_1$  a different fixed model  $\hat{\underline{\theta}}_{i+1}$  is chosen, the adaptive model  $\hat{\underline{\theta}}_a$  is discarded and a new one is initiated at  $\hat{\underline{\theta}}_{i+1}$ . Details regarding this procedure can be found in (Narendra and Balakrishnan 1997, [161]). If fixed models are to play an important role in the adaptive process their parameters must lie close to those of the plant, even as the latter varies with time. Since the plant parameter vector  $\underline{\theta}_0$  is unknown and can vary arbitrarily, this is possible only if a very large number of fixed models is used within the compact set  $S$ . Alternatively, the location of the models is determined based on past experience assuming that the latter is available.

#### 14.3.1.3 Switching and Tuning

Tuning is the process of incrementally adjusting the parameters of the control or estimation model. This is the method used in classical adaptive control where the evolution of the parameters is described by a differential equation. When multiple models are used, parameters can change discontinuously. To motivate the need for switching in an adaptive control system, we consider the case where it is known a priori that the plant can assume only one of two values (i.e. the plant parameter vector satisfies:  $\underline{\theta}_0 \in \{\underline{\theta}_1, \underline{\theta}_2\}$ ). Corresponding to each of the above, it is also known that controllers  $\Gamma_1$  and  $\Gamma_2$  exist such that  $\underline{\theta}_i$  together with  $\Gamma_i$  in the feedback path matches a stable reference model. It is further assumed that the

plant  $\underline{\theta}_i$  together with controller  $\Gamma_j (j \neq i)$  results in instability. If the plant were to switch rapidly between  $\theta_1$  and  $\theta_2$  the adaptive method used must be able to detect the change in the plant and switch to the appropriate controller to avoid instability. Since only the inputs and outputs of the plant are assumed to be known, detection of the change in the plant has to be concluded from the output estimation errors  $\varepsilon_i$ . Switching is desirable to react to rapid changes in the plant characteristics and avoid instability. Since the number of models is finite, while the number of values that the plant can assume within  $S$  is infinite, tuning is necessary if the control error is to tend to zero asymptotically. The essence of MMST is to combine switching and tuning efficiently such as to improve the performance of the closed-loop system while keeping all signals bounded. The control input  $u$  resulting from this procedure is in general piecewise-continuous.

A crucial role in the design of MMST is played by the switching criterion which determines when to switch from one model to another and which of the models should be the new one. A number of different performance indices can be defined based on the identification error  $\varepsilon(k)$  to determine which of the models best fits the plant at any instant. These may assume the following forms:

$$\begin{aligned}
 \text{(i)} \quad J_i(k) &= \varepsilon_i^2(k) \\
 \text{(ii)} \quad J_i(k) &= \sum_{\nu=0}^k \varepsilon_i^2(\nu) \\
 \text{(iii)} \quad J_i(k) &= \alpha \varepsilon_i^2(k) + \beta \sum_{\nu=0}^k \varepsilon_i^2(\nu) \\
 \text{(iv)} \quad J_i(t) &= \alpha \varepsilon_i^2(k) + \beta \sum_{\nu=0}^k \rho^{k-\nu} \varepsilon_i^2(\nu)
 \end{aligned} \tag{14.67}$$

The criterion (i) represents instantaneous values of the square of the output error, (ii) the integral of the former, (iii) a linear combination of (ii) and (iii), and (iv) a modification of (iii) which includes a forgetting factor determined by the parameter  $\rho \in [0; 1]$ . Based on one of the criteria (i)–(iv) the model/controller pair  $[\Xi_j, \Gamma_j]$  is used at every instant of time using  $J_j(k) = \min_i J_i(k)$ . Criterion (i) invariably results in rapid switching between controllers while relatively slow switching is achieved using (ii). As a consequence, the control errors are larger in the latter case. For satisfactory operation in rapidly time-varying environments, criterion (iv) is generally preferred and the parameter  $\rho$  is selected to achieve a compromise between speed and performance.

#### 14.3.1.4 Control

The control input to the system is computed based on the parameters of the model that performs best according to one of the above criteria. The approach

of using the parameters of an estimation model to determine the control input is referred to as an indirect one, in contrast to the direct approach where the controller parameters are identified directly based on the control error  $e$ . From the above discussion it is clear that the indirect approach is inherent to the MMST methodology. While a multiplicity of models can be used concurrently to estimate the plant, only one controller can be used at any instant to determine the control input.

Notice that the overall performance of the system will be judged on the basis of the *control error*  $e$ , whereas the choice of the controller at any instant is based on a performance index that depends upon the *estimation error*  $\varepsilon$ . As seen in chapter 13, equation (13.69), the control error deviates from the identification error by an amount depending upon the time-variation of the parameter. If a fixed controller is used, the two are equivalent. However, in general, the controller has to be adaptive in order for  $e$  to vanish. But this means that the control based on the model with the smallest estimation error need not result in the smallest control error. This is the principal difficulty encountered when attempting to derive a quantitative measure of the performance improvement obtained through MMST.

#### 14.3.1.5 Benefits

While being a natural extension of conventional adaptive control to time-varying situations, the MMST methodology offers a number of intuitively appealing and practically relevant advantages:

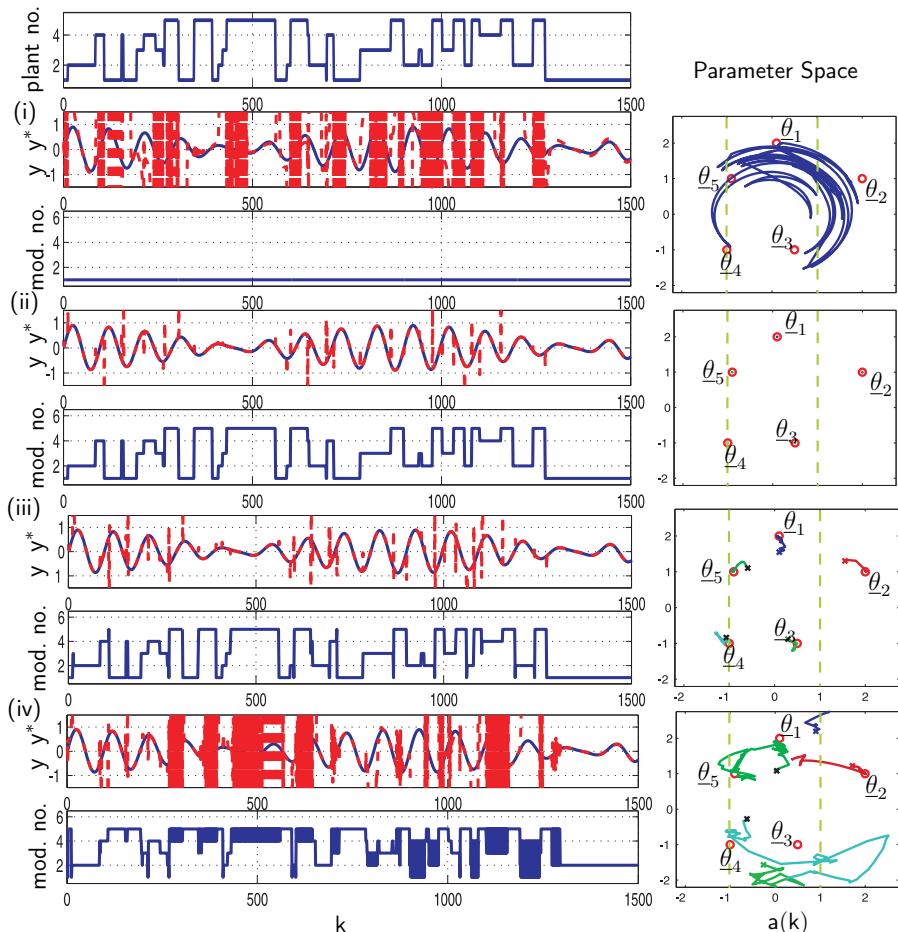
- Its principal use is to detect changes in the environment (e.g. due to time-varying disturbances) and initiate appropriate control action.
- When the information needed to design an adaptive controller (e.g. its relative degree, order of the plant) is missing, multiple models can be used to obtain it.
- An application is to combine the advantages of different models. One of the models, designed analytically, may assure stability, while another, designed heuristically, may have better performance. A proper combination of the two results in a stable system with improved performance.
- By placing fixed models in the neighborhood of the points in parameter space which the plant is likely to assume, the time for the parameters to converge can be reduced substantially and, hence, performance improved. Recall that, in the absence of a persistently exciting input, the models do not have to be close to the plant, since the identification error is small anywhere near the hypersurface defined in equation (13.29). But if the models are close, the error will be small for any input.

**Example:**

Given a discrete-time plant described by

$$y(k+1) = a(k)y(k) + b(k)u(k) \quad (14.68)$$

where the plant parameter vector  $\underline{\theta}(k) = [a(k) \ b(k)]$  is piecewise constant and switches between the elements of a finite set  $S = \{\underline{\theta}_1, \underline{\theta}_2, \underline{\theta}_3, \underline{\theta}_4, \underline{\theta}_5\}$  of unknown vectors. As indicated in figure 14.12,  $\underline{\theta}_2$  corresponds to an unstable plant, while  $\underline{\theta}_4$  is stable with eigenvalue  $z = -1$ , and all other parameter vectors are strictly stable.



**Abb. 14.12:** Multi-Model Adaptive Control (MMST)

The objective is to control the plant such that the output tracks the desired output  $y^*(k+1) = \sin(2\pi k/100) + \sin(2\pi k/90)$  with small errors even as the plant parameters change discontinuously from one element in  $S$  to another at random instants of time. The switching is assumed to be governed by an ergodic Markov chain, where  $p = 0.9$  is the probability of remaining at a given parameter  $\underline{\theta}_i$ ,  $i = 1 \dots 5$  and  $q = 0.025$  is the probability of switching to one of the other parameters.

The first row of the figure displays the nature of the time-variation of the plant. Part (i) shows the response of the system using classical adaptive control with a single model. The performance is clearly unacceptable and the parameters do not converge anywhere. In part (ii), the location of the plant parameters is assumed to be known so that five fixed models having parameters identical to the plant can be used. At any instant, the current plant parameter is detected and the system is controlled using the parameters associated with the best model. In order to ensure a fast response, the instantaneous switching criterion is used to decide which of the models performs best. It is seen that an error occurs at the instants following a switching of the plant parameters, which persists over an interval of length  $\delta$ , where  $\delta$  is the relative degree of the system. To avoid the error, the “new” control input  $u$  would have to be computed  $\delta$  instants before a switching occurs. It is clear that this is impossible since the switching sequence is unknown. Hence, the error is inherent. In part (iii), the fixed models are replaced by adaptive ones which are initialized within a small neighborhood  $N = \{\hat{\theta} \mid \|\hat{\theta} - \underline{\theta}_i\| < \varepsilon, i = 1 \dots 5\}$  of the plant parameters, i.e.  $\varepsilon = 0.5$ . If the identification error of a given model  $\hat{\theta}_i$ ,  $i = 1 \dots 5$ , is small, this model is updated while all other models retain their current position in parameter space. The combined performance criterion (iv) in equation (14.67) with  $\alpha = \beta = 1/2$  and  $\rho = 0.3$  was found to result in acceptable performance. In addition, observe that the parameters approach the true parameters of the system. Finally, in part (iv) of the figure the experiment is repeated, but in this case, the models are initialized in a larger neighborhood,  $\varepsilon = 1$ . The performance is seen to be similar to case (i). The example demonstrates, that the performance of the MMST approach in a time-varying environment critically depends on the location of the models and, hence, the prior information about the plant.

### 14.3.2 Proof of Stability

Stability proofs have been derived for linear deterministic and stochastic systems as well as certain classes of nonlinear systems [162]. In all these cases, the unknown parameters are assumed to be constant. The principal stability question encountered in MMST can be qualitatively described by considering two models  $\Xi_1$  and  $\Xi_2$  and two controllers  $\Gamma_1$  and  $\Gamma_2$ . The estimated output given by the two models is  $\hat{y}_1$  and  $\hat{y}_2$  respectively and the corresponding estimation errors are  $\varepsilon_1$  and  $\varepsilon_2$ . We assume that the plant  $\Sigma$  is stable with controller  $\Gamma_1$  in the feedback path and unstable with  $\Gamma_2$ . If, based on the switching criterion,  $\Xi_2$  is chosen at

a specific instant, the controller  $\Gamma_2$  would be used in the feedback path resulting in an unstable closed loop. The question that has to be addressed is how the switching criterion should be chosen such as to result in the controller switching to  $\Gamma_1$  in a finite time. In what follows, we present the proof of linear deterministic adaptive control using multiple models which builds upon the arguments presented in chapter 13 in the single model case. Three different configurations of models are considered:

- (i)  $N$  adaptive models
- (ii) One fixed model and one adaptive model
- (iii)  $N - 2$  models and 2 adaptive models

The plant is described by the deterministic equation

$$y(k + \delta) = \sum_{i=0}^{n-1} a_i y(k - i) + \sum_{i=0}^{n-1} b_i u(k - i) \quad (14.69)$$

where the **constant** parameters  $a_i$  and  $b_i$  are unknown. The plant is assumed to be minimum-phase and its relative degree  $\delta$  (delay) and order is known. The objective of the control is to determine a bounded input  $u(k)$  such that the output of the plant  $y(k + \delta)$  asymptotically tracks a given bounded reference output  $y^*(k + \delta)$ . It is assumed that  $y^*(k + \delta)$  is known at time  $k$  (or alternately if  $y^*(k + \delta) = r(k)$ ,  $y^*(k + \delta)$  is specified). The reference model in this case is a pure delay of  $\delta$  units, i.e. it has the transfer function  $z^{-\delta}$ .

#### 14.3.2.1 Case (i): All adaptive models

From our discussion in chapter 13 it is clear that a single model is sufficient to control an unknown plant in a stable fashion. The objective of this paragraph is to show that this is also the case when multiple models are used. More importantly, the closed-loop system is stable even as the switching between the models (and, hence, the corresponding controllers) is carried out in a random fashion.

If  $N$  models are used to estimate the plant parameters, and at time  $k$  the  $i$ th model  $\hat{\theta}_i$  is chosen, the control input  $u(k)$  is computed from the equation

$$y^*(k + \delta) = \underline{\phi}(k)^T \hat{\theta}_i(k) \quad (14.70)$$

The control error at time  $k$  is given by

$$e(k) = \varepsilon_i(k) + \underline{\phi}(k - \delta)^T [\hat{\theta}_i(k - 1) - \hat{\theta}_i(k - \delta)] \quad (14.71)$$

where  $\varepsilon_i(k)$  is the identification error of the  $i$ th model. If the model at the next instant is chosen randomly as  $\hat{\theta}_j$ , the control error  $e$  satisfies an equation similar to (14.71) with  $i$  replaced by  $j$ . This process can be repeated at every instant and a model chosen randomly from  $\hat{\theta}_i$ ,  $i = 1 \dots N$ . The important fact to note

is that the regression vector is the same for all of them. The parameters of all the models are estimated using the same algorithm, only the initial conditions are different. From the properties of the estimation algorithm we know that  $\|\hat{\theta}_i(k-1) - \hat{\theta}_i(k-\delta)\| \rightarrow 0$  for all  $i$ . Hence both terms on the right hand side of equation (14.71), when normalized, i.e.  $\frac{\varepsilon_i(k)}{[1 + \underline{\phi}(k-\delta)^T \underline{\phi}(k-\delta)]^{1/2}}$  and  $\frac{\underline{\phi}(k-\delta)^T [\hat{\theta}_i(k-1) - \hat{\theta}_i(k-\delta)]}{[1 + \underline{\phi}(k-\delta)^T \underline{\phi}(k-\delta)]^{1/2}}$  tend to zero. Hence, using the same arguments as in the single model case, it follows that  $e(k)$  grows at a slower rate than  $\|\underline{\phi}(k)\|$ ,  $\|\underline{\phi}(k)\|$  is bounded, and  $\lim_{k \rightarrow \infty} e(k) = 0$ .

In the proof, any one of the adaptive controllers is chosen randomly to control the system. This implies that with multiple models, the question of stability can be decoupled from that of performance and the switching procedure can be based entirely on the latter. Notice, that the proof can be directly extended to the case where adaptive models are either introduced or removed, provided that at least one model (referred to as a free-running model) is not disturbed during the adaptive process.

#### 14.3.2.2 Case (ii): One adaptive model and one fixed model

If the plant is time-invariant, the benefit of having multiple models is that if the latter are initialized at different locations in parameter space, one model  $\hat{\theta}_i$ ,  $i = 1 \dots N$ , may be close to the plant and will result in a smaller error  $\varepsilon_i$ . However, if no adaptive model is close to the plant, there may be no improvement in performance. It is clear that keeping all models adaptive requires considerable computational overhead and is not efficient since a single adaptive model has already been found to be adequate for stability. The idea is to replace  $N - 1$  adaptive models by fixed ones. These can be thought of as providing convenient initial conditions for adaptation. In this section we consider the case of  $N = 2$  models, one adaptive and one fixed.

The adaptive and the fixed model result in error equations  $\varepsilon(k) = \tilde{\theta}(k)^T \underline{\phi}(k)$  and  $\varepsilon_f(k) = \tilde{\theta}_f^T \underline{\phi}(k)$  respectively where, in general, only the error of the adaptive model  $\varepsilon(k)$  tends to zero. If  $|e_f(0)| < |e(0)|$  and switching criterion (iii) in equation (14.67) is used, the system will initially start with the fixed model, and the control corresponding to it will be applied. However, since  $J_i(k)$  is bounded while  $J_f(k)$  (of the fixed model) grows monotonically with  $k$ , the system will switch to the adaptive model in a finite time. In the non-generic case that the fixed model is identical to the plant,  $\varepsilon_f(k)$  is zero and no switching occurs. The above arguments are still valid if an arbitrary number  $N$  of fixed models are used. Notice that  $\varepsilon_f(k)$  can be zero on a subsequence  $\{k_t\}$ ,  $t = 1 \dots \infty$  because the regression vector  $\underline{\phi}(k_t)$  is orthogonal to  $\tilde{\theta}_f$  at those instants of time. If criterion (i) is used, switching will not stop in finite time since  $\varepsilon(k_t) > \varepsilon_f(k_t)$ . This emphasizes the importance of using an integral criterion in a time-invariant situation.

### 14.3.2.3 Case (iii): (N-2) fixed models and 2 adaptive models

If the fixed model in the previous paragraph is closer to the plant than the adaptive model (as determined by the performance criterion  $J$ ), the system switches to it and switches back to the adaptive model once  $\varepsilon(k)$  has become small. However, faster tracking can be obtained if a new adaptive model is initialized at the same value as the fixed model. In view of the above discussion, the fixed models have to be located based on the past performance of the plant. One adaptive model is free-running and is included to assure stability. A second model is initiated whenever a fixed model is found to be superior. A large number of fixed models may be needed to obtain one model with distinguished performance at which the adaptation can be initiated. The reinitialization of the second adaptive model is central for improving the performance of the system. If at any instant  $k_0$ ,  $J_f(k_0)$  is found to be a minimum, the adaptive model is initialized using the parameter vector  $\underline{\theta}_f$  and the performance index  $J_f(k_0)$ . The adaptive process is continued. If, at a later instant  $k_1$ , a different fixed model (with parameter  $\underline{\theta}_g \neq \underline{\theta}_f$ ) is superior, the adaptive model is discarded and a new one is initialized at  $\underline{\theta}_g$  and  $J_g(k_1)$ .

The introduction of the additional (re-initialized) model does not adversely affect the stability of the overall system. Let the system switch between fixed and adaptive models at every instant of time. From the arguments provided in section 14.3.2.2, this can only last for a finite interval of time, since the performance index of the free-running adaptive model will eventually become smaller than the indices of all the fixed models. It is seen that the existence of the free-running model is the key argument for convergence. If the plant is rapidly time-varying, the error of a single adaptive model cannot tend to zero and the arguments are substantially more difficult. In this case, all models can be made adaptive and the switching criterion modified by using a finite window (i.e.  $\sum_{k-T}^k \varepsilon^2(\nu)$ ) or a forgetting factor  $0 < \rho \leq 1$ . In the ideal case,  $J_i(k) \rightarrow 0$  for all adaptive models, while  $J_f(k) > 0$  for the fixed models. The fundamental assumption made in this context is that at least one adaptive model is close to every parameter that the plant is likely to assume. This has already been observed in cases (iii) and (iv) of example displayed in figure 14.12.

## 14.4 Mathematical background

In the following chapter we collect some mathematical tools used throughout the book. The material may be seen as a self-contained *user's guide* to the theory of ordinary differential equations and Lyapunov stability. For a more detailed exploration of nonlinear systems theory, the reader is referred to [122].

### 14.4.1 Nonlinear Differential Equations

We begin by reviewing some basic facts about ordinary differential equations (ODEs) of the form

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, \dots, x_n, t) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n, t).\end{aligned}\tag{14.72}$$

Defining  $\underline{x} = [x_1, \dots, x_n]^T$  we obtain the simpler form with a vector-valued function  $\underline{f}$

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t), \quad \underline{x} \in \mathbb{R}^n.\tag{14.73}$$

It may happen that  $\underline{f}$  does not depend explicitly on  $t$ , i.e.

$$\dot{\underline{x}} = \underline{f}(\underline{x}), \quad \underline{x} \in \mathbb{R}^n\tag{14.74}$$

in which case the system is said to be *autonomous*. Except for some special cases, it is impossible to determine a closed form solution of a nonlinear ODE. We will try to determine bounds on the solution of a given equation without actually solving it. It is then possible to determine, at each instant of time, a region in  $\mathbb{R}^n$  in which the solution must lie. In this section we develop the key mathematical tools for carrying out this program. We begin with a quick review of some basic concepts of analysis.

### 14.4.2 Concepts from Analysis

#### Definition

A linear vector space over the field  $\mathbb{K}$  (either  $\mathbb{R}$  or  $\mathbb{C}$ ) is a set  $V$  of elements called vectors, together with two operations

- addition       $+ : V \times V \rightarrow V$
- multiplication     $\cdot : \mathbb{K} \times V \rightarrow V$

such that for any  $\underline{x}, \underline{y}, \underline{z} \in V$  and  $\alpha, \beta \in \mathbb{K}$

- $\underline{x} + \underline{y} = \underline{y} + \underline{x}$ ,  $(\underline{x} + \underline{y}) + \underline{z} = \underline{x} + (\underline{y} + \underline{z})$
- there is a zero vector  $\underline{0} \in V$  s.t.  $\underline{x} + \underline{0} = \underline{x} \quad \forall \underline{x} \in V$
- there is an inverse vector  $-\underline{x} \in V$  s.t.  $\underline{x} + (-\underline{x}) = \underline{0} \quad \forall \underline{x} \in V$
- $(\alpha\beta)\underline{x} = \alpha(\beta\underline{x})$ ,  $\alpha(\underline{x} + \underline{y}) = \alpha\underline{x} + \alpha\underline{y}$  and  $(\alpha + \beta)\underline{x} = \alpha\underline{x} + \beta\underline{x}$
- $1 \cdot \underline{x} = \underline{x}$ ,  $0 \cdot \underline{x} = \underline{0}$

**Note:** The definition implies that whether a linear vector space is real or complex is determined not by its elements (vectors  $\underline{x}$ ) but by the associated set of scalars ( $\alpha \in \mathbb{R}$ ) or ( $\alpha \in \mathbb{C}$ ).

**Example:**

The set  $\mathbb{R}^n$  of all ordered  $n$ -tupels of real numbers becomes a *real* linear vector space if for  $\underline{x}, \underline{y} \in \mathbb{R}^n$  and  $\alpha \in \mathbb{R}$  addition and scalar multiplication are defined as:

$$\begin{aligned}\underline{x} + \underline{y} &= \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix} \\ \alpha \underline{x} &= \begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_n \end{bmatrix}\end{aligned}$$

**Note:** Similarly, the set of all ordered  $n$ -tupels of complex numbers,  $\mathbb{C}^n$ , becomes a **real** linear vector space if addition and scalar multiplication is defined as above (component-wise) and  $\alpha \in \mathbb{R}$ .

**Example:**

The set  $F[a, b]$  of all real-valued functions defined over an interval  $[a, b]$  in  $\mathbb{R}$  becomes a real linear vector space if addition and scalar multiplication are defined as follows:

For  $x(\cdot), y(\cdot) \in F[a, b]$  and  $\alpha \in \mathbb{R}$

$$\begin{aligned}(x + y)(t) &= x(t) + y(t) \quad \forall t \in [a, b] \\ (\alpha x)(t) &= \alpha x(t) \quad \forall t \in [a, b]\end{aligned}$$

→ “pointwise” addition and scalar multiplication. Notice that  $F[a, b]$  has dimension  $n$  provided that there are at most  $n$  linearly independent vectors in  $F[a, b]$ . Any set of  $n$  such vectors is called a basis that spans  $F[a, b]$  (i.e. any other vector in  $F[a, b]$  can be written as a linear combination of these). For example, let  $F[-1, 1]$  be the vector space of  $n$ -th order polynomials on  $[-1, 1]$  with real coefficients. Then, the monomials  $\{1, t^2, \dots, t^n\}$  span  $F[-1, 1]$ . If one does not restrict the degree of the polynomials, the dimension of  $F'[-1, 1]$  (the space of all polynomials on  $[-1, 1]$ ) is infinite.

**Definition 14.4.1** A subset  $M \subset V$  is called a subspace of  $V$  if

$$1. \quad \underline{x}, \underline{y} \in M \quad \Rightarrow \quad \underline{x} + \underline{y} \in M \quad (14.75)$$

$$2. \quad \underline{x} \in M, \alpha \in \mathbb{K} \quad \Rightarrow \quad \alpha \underline{x} \in M \quad (14.76)$$

**Example:**

The set  $M$  of all vectors  $\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$ ,  $x_1, x_2 \in \mathbb{R}$  is a subspace of  $\mathbb{R}^3$ .

**Example:**

The set  $M$  of all vectors  $\begin{bmatrix} x_1 \\ x_2 \\ 2 \end{bmatrix}$ ,  $x_1, x_2 \in \mathbb{R}$  is *not* a subspace of  $\mathbb{R}^3$  because this set is not closed under addition, i.e.

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \in M : \quad \begin{bmatrix} 1+1 \\ 1+0 \\ 2+2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} \notin M$$

So far, we do not have any notion of distance in a linear vector space. Hence we cannot discuss basic concepts like convergence or continuity. The device which makes this possible is the concept of a *norm*.

**Definition 14.4.2** Let  $X$  be any vector space over  $\mathbb{K}$ . A non-negative function  $\|\cdot\| : X \rightarrow \mathbb{R}$  is a *norm on  $X$*  if the following axioms hold

$$1. \quad \|\underline{x}\| \geq 0 \quad \forall \underline{x} \in X, \quad \|\underline{x}\| = 0 \Leftrightarrow \underline{x} = 0 \quad (14.77)$$

$$2. \quad \|\alpha \underline{x}\| = |\alpha| \|\underline{x}\| \quad \forall \underline{x} \in X, \quad \forall \alpha \in \mathbb{K} \quad (14.78)$$

$$3. \quad \|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\| \quad \forall \underline{x}, \underline{y} \in X \quad (\text{triangle inequality}) \quad (14.79)$$

A vector space  $X$  equipped with a norm is called a normed vector space. For a given  $X$  there are many ways to define a norm. If  $X = \mathbb{R}^n$  a popular class of norms is of the form

$$\|\underline{x}\|_p = \left[ \sum_{i=1}^n |x_i|^p \right]^{\frac{1}{p}} \quad (14.80)$$

where  $p$  is a positive integer.  $\|\cdot\|_p$  is called the *p-norm on  $\mathbb{R}^n$* .

**Example:**

Let  $p = 1$ . Then

$$\|\underline{x}\|_1 = \sum_{i=1}^n |x_i| \quad (14.81)$$

$\|\cdot\|$  clearly satisfies (14.78) and (14.79). We know, by the triangle inequality for real numbers, that  $|x_i + y_i| \leq |x_i| + |y_i| \forall i$ . Therefore, for any  $\underline{x}, \underline{y} \in \mathbb{R}^n$

$$\begin{aligned} \|\underline{x} + \underline{y}\|_1 &= \sum_{i=1}^n |x_i + y_i| \leq \sum_{i=1}^n (|x_i| + |y_i|) \\ &= \sum_{i=1}^n |x_i| + \sum_{i=1}^n |y_i| = \|\underline{x}\|_1 + \|\underline{y}\|_1. \end{aligned}$$

For  $p = 2$  we obtain

$$\|\underline{x}\|_2 = \left[ \sum_{i=1}^n |x_i|^2 \right]^{\frac{1}{2}}, \quad (14.82)$$

i.e. the Euclidean norm on  $\mathbb{R}^n$ .

As  $p \rightarrow \infty$ ,  $\|\cdot\|$  approaches

$$\|\underline{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|. \quad (14.83)$$

A function  $\underline{f} : S \rightarrow \mathbb{R}^n$  is *bounded* with respect to the  $p$ -norm  $\|\cdot\|$  if there is a finite number  $c$  such that  $\|\underline{f}(\underline{x})\|_p < c$  for all  $\underline{x} \in S$ . The smallest  $c$  satisfying the inequality is called the supremum of  $\underline{f}$  over  $S$  (with respect to  $\|\cdot\|$ ) denoted by

$$\sup_S \|\underline{f}\|_p. \quad (\text{If } \underline{f} \text{ is not bounded, } \sup_S \|\underline{f}\|_p = \infty.)$$

The  $p$ -norm of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  can be defined as

$$\|\mathbf{A}\|_p = \sup_{\underline{y} \in \mathbb{R}^m} \frac{\|\mathbf{A}\underline{y}\|_p}{\|\underline{y}\|_p} \quad (m > 1). \quad (14.84)$$

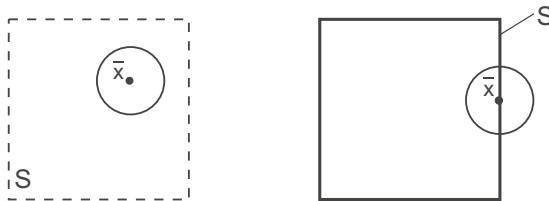
$\|\mathbf{A}\|_p$  can be thought of as the “gain” of the linear mapping  $\underline{f}(\underline{y}) = \mathbf{A}\underline{y}$ ,  $\underline{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ .

It can be shown that  $\|\mathbf{A}\|_2$  is the largest “singular value” of  $\mathbf{A}$  (i.e. eigenvalue of the symmetric matrix  $\mathbf{A}^T \mathbf{A}$ ). The  $p$ -norm defined in (3.13) has the useful property that for all  $p > 1$ , including  $p = \infty$ :

$$\|\mathbf{AB}\|_p \leq \|\mathbf{A}\|_p \|\mathbf{B}\|_p \quad (14.85)$$

**Definition 14.4.3** Let  $X$  be any vector space and let  $\|\cdot\|$  be a norm defined on  $X$ . A *neighborhood* or *open ball* of radius  $r$  about a vector  $\bar{x} \in X$  is defined as

$$B_r = \{\underline{x} \mid \|\underline{x} - \bar{x}\| < r\} \quad (14.86)$$



**Abb. 14.13:** Example for open and closed set

An *open set*  $S \subset X$  is simply any subset of  $X$  whose elements all have neighborhoods which lie completely inside of  $S$ .

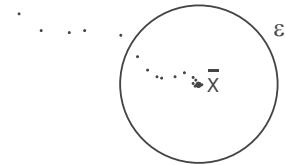
The interior of a rectangle is an open set in  $\mathbb{R}^2$ . The rectangle *and* its boundary is not open (because it is impossible to construct an open neighborhood of any point on the boundary which completely lies within  $S$ ).  $S \subset X$  is closed if its complement in  $X$  is open.

Let  $X$  be a normed linear space and  $\{\underline{x}_k\}$  be a sequence of points in  $X$ . Then  $\{\underline{x}_k\}$  is said to converge to  $\bar{x} \in X$  if for any  $\varepsilon > 0$  there exists a number  $N$  such that

$$\|\underline{x}_k - \bar{x}\| < \varepsilon \quad \forall k \leq N. \quad (14.87)$$

If  $B_\varepsilon = \{\underline{x} \in X \mid \|\underline{x} - \bar{x}\| < \varepsilon\}$  is a neighborhood of  $\bar{x}$  then  $\{\underline{x}_k\}$  converges to  $\bar{x} \in X$  if  $B_\varepsilon$  contains all but a finite number of elements of  $\{\underline{x}_k\}$ . We write

$$\lim_{k \rightarrow \infty} \underline{x}_k = \bar{x}. \quad (14.88)$$

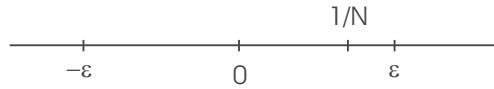


**Abb. 14.14:** Convergence of  $\{\underline{x}_k\}$

**Example:**  
 $x_k = 1/k$  then  $\lim_{k \rightarrow \infty} x_k = 0$ .  
*Proof* Let  $\varepsilon > 0$  Find  $N > 0$  such that  $1/N < \varepsilon$ :  $N > 1/\varepsilon$ . We have

$$0 < x_k = \frac{1}{k} \leq \frac{1}{N} < \varepsilon \quad \forall k \geq N$$

$\Rightarrow x_k$  converges to 0.

**Abb. 14.15:** Proof

Using the concept of convergence we obtain a different characterization of closedness: A set  $S$  is *closed* if and only if every convergent sequence  $\{\underline{x}_k\}$  with elements in  $S$  converges to a point in  $S$ .

A set  $S$  is *bounded* if there is an  $r > 0$  such that

$$\|\underline{x}\| \leq r \quad \forall \underline{x} \in S. \quad (14.89)$$

A set  $S$  is *compact* if it is closed and bounded. A point  $\underline{p}$  is a *boundary point* of a set  $S$  if every neighborhood of  $\underline{p}$  contains at least one point of  $S$  and one point not in  $S$ . A closed set contains all its boundary points, an open set contains none of its boundary points.

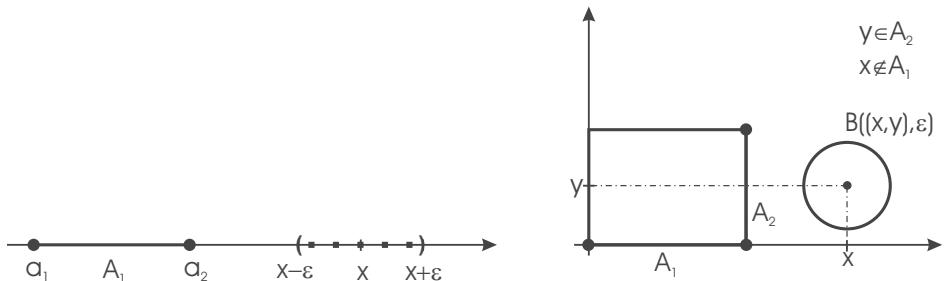
**Example:**

The set  $\mathbb{R}$  of all real numbers is open: If  $x$  is any real number, then the open ball  $(x - \varepsilon, x + \varepsilon) \subset \mathbb{R}$ . Since  $\mathbb{R}$  is open, its complement  $\emptyset$  is closed (and empty).

**Example:**

Let  $A_1 = [a_1, b_1] \subset \mathbb{R}$  and  $A_2 = [a_2, b_2] \subset \mathbb{R}$  be closed. Then  $A_1 \times A_2 \subset \mathbb{R}^2$  is also a closed set.

*Proof* Let  $(x, y) \notin A_1 \times A_2$ . This means that  $x \notin A_1$  or  $y \notin A_2$ . Assume  $x \notin A_1$ . Since  $A_1$  is closed,  $\mathbb{R} \setminus A_1$  is open, i.e. there is an  $\varepsilon > 0$  s. t.  $B(x, \varepsilon) \subset \mathbb{R} \setminus A_1$ .

**Abb. 14.16:** Proof

But this implies that

$$B((x, y), \varepsilon) \subset \mathbb{R}^2 \setminus (A_1 \times A_2) \quad \forall (x, y) \in \mathbb{R}^2 \setminus (A_1 \times A_2).$$

The complement of  $A_1 \times A_2$  is open.  $\Rightarrow$   $A_1 \times A_2$  is closed.

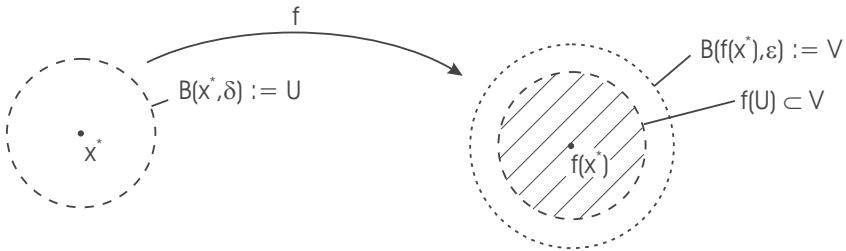
**Note:** Any cuboid  $Q := \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i \text{ for } i = 1 \dots n\}$ ,  $a_i, b_i \in \mathbb{R}$ ,  $a_i \leq b_i$  defines a closed set in  $\mathbb{R}^n$ .

Another important concept is *continuity*. A function mapping a set  $S_1$  into a set  $S_2$  is denoted by  $\underline{f} : S_1 \rightarrow S_2$ .

**Definition 14.4.4** Let  $S_1$  and  $S_2$  be normed vector spaces. A function  $\underline{f} : S_1 \rightarrow S_2$  is continuous at a point  $\underline{x}^* \in S_1$  if for any given  $\varepsilon > 0$  there is a  $\delta > 0$  such that

$$\|\underline{f}(\underline{x}) - \underline{f}(\underline{x}^*)\| < \varepsilon \quad \forall \underline{x} \in S_1 \text{ satisfying } \|\underline{x} - \underline{x}^*\| < \delta. \quad (14.90)$$

The function  $\underline{f}$  is continuous on  $S$  if it is continuous at every point of  $S$ .



**Abb. 14.17: Definition of continuity**

According to Fig. (14.17) we say: “For any neighborhood  $V$  of  $\underline{f}(\underline{x}^*)$  there is a neighborhood  $U$  of  $\underline{x}^*$  such that  $\underline{f}(U) \subset V$ .”

The following two statements are equivalent:

1.  $\underline{f}$  is continuous at  $\underline{x}^*$ .
2. For any sequence  $\{\underline{x}_k\}$  with

$$\lim_{k \rightarrow \infty} \underline{x}_k = \underline{x}^* : \quad \lim_{k \rightarrow \infty} \underline{f}(\underline{x}_k) = \underline{f}(\underline{x}^*) \quad (14.91)$$

#### Example:

$f(x) = c$  ( $c \in \mathbb{R}, x \in \mathbb{R}$ ) is continuous in  $x_0 \in \mathbb{R}$ .

For any given  $\varepsilon > 0$ ,  $\delta > 0$  is arbitrarily large, i.e. for all  $x$  satisfying  $|x - x_0| < \delta$

$$|f(x) - f(x_0)| = |c - c| = 0 < \varepsilon$$

#### Example:

$f(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$  is not continuous at  $x_0 = 0$ .

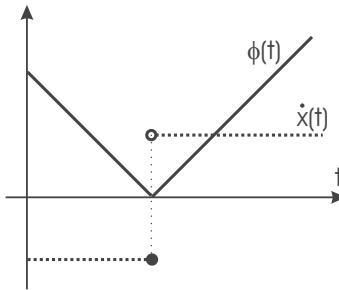


Abb. 14.18: Step function

Given  $\varepsilon > 0$  (e.g.  $\varepsilon = \frac{1}{2}$ ) we cannot find a  $\delta > 0$  such that

$$\|f(x) - f(0)\| < \frac{1}{2} \quad \text{for all } \|x\| < \delta.$$

(Any neighborhood of 0 contains points which are not mapped into a neighborhood of  $f(0) = 1$ .)

#### 14.4.3 Existence and uniqueness

Let us return to our study of nonlinear differential equations of the form (14.73). For a long time, mathematicians were not concerned with the fundamental problem of the very existence of a solution to (14.73). Let us first clarify what it means for the system of differential equations to have a “solution”.

**Definition 14.4.5** By a solution to the system of differential equation (14.73) on an interval  $I \subset [0, \infty)$  of positive, finite length is meant a continuous function  $\underline{\zeta} : I \rightarrow \mathbb{R}^n$  which satisfies  $\dot{\underline{\zeta}} = \underline{f}(\underline{\zeta}, t)$ , wherever  $\underline{\zeta}$  exists.

##### Example:

$\zeta(t) = |t - 2|$ ,  $t \geq 0$  is a solution to  $\dot{x} = \begin{cases} -1 & \text{if } t \leq 2 \\ +1 & \text{if } t > 2 \end{cases}$ .

The initial value problem (IVP) can be stated as follows:

Given an initial time  $t_0 \in [0, \infty)$  and an initial state vector  $\underline{x}_0 \in \mathbb{R}^n$  find an interval  $I$  which contains  $t_0$  and a solution  $\underline{\zeta}$  to (14.73) on  $I$  which passes through  $\underline{x}_0$  at  $t = t_0$ .

The IVP turns out to always have a solution  $\underline{\zeta}$  on some interval  $I$  if  $f$  is a continuous function of  $\underline{x}$  and a piecewise-continuous function of  $t$ . It turns out though, that *continuity in  $\underline{x}$  is not a strong enough condition to ensure uniqueness of solutions*.

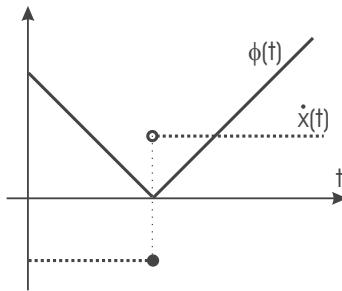


Abb. 14.19: Example 14.4.3

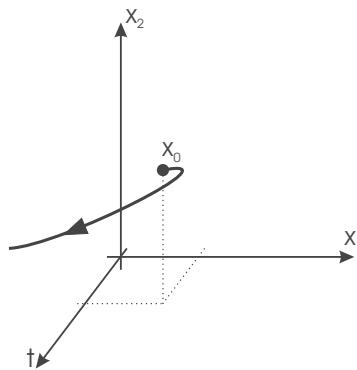


Abb. 14.20: Solution of an IVP

**Example:**  
The IVP

$$\dot{x} = \sqrt[3]{x}, \quad x(0) = 0$$

demonstrates non-uniqueness. Both

$$\zeta(t) = \left(\frac{2}{3}t\right)^{\frac{3}{2}} \quad \text{and} \quad \zeta(t) = 0$$

satisfy the differential equation and the initial condition  $\zeta(0) = 0$ .

A slightly stronger smoothness condition has to be imposed on  $f$  in order to obtain uniqueness.

**Definition 14.4.6** Let  $\underline{f}$  have the following properties:

- For each closed, bounded subset  $S \subset \mathbb{R}^n$  and each bounded interval  $I \subset [0; \infty)$  there exists a constant  $L$  such that

$$\|\underline{f}(\underline{x}, t) - \underline{f}(\underline{y}, t)\| \leq L\|\underline{x} - \underline{y}\| := \varepsilon \quad (14.92)$$

for all  $\underline{x}, \underline{y} \in S$  and all  $t \in I$ .

- $\underline{f}(\underline{x}, \cdot) : t \rightarrow \underline{f}(\underline{x}, t)$  has at most a finite number of discontinuities and at each such point  $\underline{f}(\underline{x}, \cdot)$  has unique limits when approached from above and from below.

$\Rightarrow$  Then  $\underline{f}$  is said to be *locally Lipschitz continuous on  $\mathbb{R}^n$*  (and piecewise continuous on  $[0; \infty]$ ).  $L$  is called a Lipschitz constant. If  $S \equiv \mathbb{R}^n$  then  $\underline{f}$  is *globally Lipschitz continuous on  $\mathbb{R}^n$* .

**Example:**

$f(x) = x^2$  is locally Lipschitz on  $\mathbb{R}$  because

$$\|x^2 - y^2\| = \|(x+y)(x-y)\| \leq \|x+y\| \|x-y\| \leq \underbrace{\max_{x,y \in S} \|x+y\|}_{L < \infty} \cdot \|x-y\|$$

If  $S \equiv \mathbb{R}$ , then  $L$  is infinite  $\Rightarrow f(x)$  is only locally Lipschitz.

**Example:**

$f(\underline{x}) = \mathbf{A}\underline{x} + \underline{b}$  (linear) is globally Lipschitz on  $\mathbb{R}^n$  because

$$\|\mathbf{A}\underline{x} + \underline{b} - \mathbf{A}\underline{y} - \underline{b}\| = \|\mathbf{A}(\underline{x} - \underline{y})\| \leq \|\mathbf{A}\| \|\underline{x} - \underline{y}\| \quad \forall \underline{x}, \underline{y} \in \mathbb{R}^n$$

It can be shown that *any continuously differentiable function* is locally Lipschitz (e.g.  $f(x) = x^2$ ) and that any locally Lipschitz function is continuous.

**Theorem 14.4.1** Suppose that  $\underline{f}(\underline{x}, t)$  is locally Lipschitz in  $\underline{x}$  and piecewise continuous in  $t$ .

For each  $\underline{x}_0 \in \mathbb{R}^n$  and each  $t_0 \in [0, \infty)$  there exists a largest interval  $I^* \subset [0, \infty)$  and *exactly one* continuous solution  $\underline{\zeta}$  to

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t)$$

on  $I^*$  passing through  $\underline{x}_0$  at  $t = t_0$ .

The possibility that  $I^*$  might have to be something less than the whole real interval  $[0, \infty)$  is illustrated by the following example:

**Example:**

$$\dot{x} = x^2, \quad x(t_0) = x_0 \quad (14.93)$$

$$\theta(t) = \begin{cases} \frac{x_0}{1-x_0(t-t_0)} & \text{if } t \neq \bar{t} = t_0 + \frac{1}{x_0} \\ 0 & \text{if } t = \bar{t} \end{cases}$$

$\theta$  satisfies the initial condition and also the differential equation except at the time  $t = \bar{t}$ . Since  $\theta$  is not continuous at  $t = \bar{t}$  it cannot be a solution of the IVP. On the other hand, its restriction to  $[0, \bar{t}]$ :  $\zeta(t) = \theta(t)$ ,  $t \in [0, \bar{t}]$  must be a solution.

$I^* = [0, \bar{t}]$  in the above example is the largest possible interval on which a solution to the problem exists. The corresponding  $\underline{\zeta}^*$  is called the maximal solution and  $I^*$  is its maximal interval of existence.

It can be shown that  $I^* = [0, \infty)$  either if

- $\underline{f}$  is *globally Lipschitz continuous* in  $\underline{x}$  or
- $\underline{f}$  is *locally Lipschitz* and the maximal solution  $\underline{\zeta}^*(t)$  is *bounded* i.e. every solution of  $\dot{x} = \underline{f}(\underline{x}, t)$ ,  $\underline{x}(t_0) = \underline{x}_0$  lies entirely in some compact subset  $W$  of  $\mathbb{R}^n$ .

**Example:**

$$\dot{x} = -x^3$$

$$\begin{cases} \dot{x} < 0 & \text{if } x > 0 \\ \dot{x} > 0 & \text{if } x < 0 \end{cases}$$

Hence, starting from any initial condition  $x(0) = a$ , the solution cannot leave the compact set  $\{x \in \mathbb{R} \mid |x| \leq |a|\}$   $\Rightarrow$  (Without calculating the solution) we conclude that the equation has a unique solution for all  $t \geq 0$ .

**Example:**

$$\dot{x} = x^2, \quad x(0) = x_0$$

As  $t \rightarrow \infty$ ,  $x(t)$  leaves any compact set.

For the solution of (14.73) to be of any practical interest, it must depend continuously on the initial condition  $\underline{x}_0$  and the right-hand side  $\underline{f}(\underline{x}, t)$ . In fact it can be shown that if  $\underline{f} = \underline{f}(t, \underline{x}, \underline{\eta})$  where  $\underline{\eta} \in \mathbb{R}^p$  (a vector describing e.g. the influence of  $p$  different parasitic effects) and  $\underline{f}$  depends continuously on  $\underline{\eta}$ , the solution of

$$\dot{\underline{x}} = \underline{f}(t, \underline{x}, \underline{\eta}_0), \quad \underline{x}(t_0, \underline{\eta}_0) = \underline{x}_0 \quad (14.94)$$

for all  $t \in I$  is denoted by  $\underline{\zeta}(t, \underline{\eta}_0)$  and satisfies:

**Theorem 14.4.2** Given  $\varepsilon > 0$  there is  $\delta > 0$  such that if

$$\|\underline{\xi}_0 - \underline{x}_0\| < \delta \quad \text{and} \quad \|\underline{\eta} - \underline{\eta}_0\| < \delta$$

then there is a unique solution  $\bar{\underline{\zeta}}(t, \underline{\eta})$  through  $\underline{\xi}_0$  which satisfies

$$\|\bar{\underline{\zeta}}(t, \underline{\eta}) - \underline{\zeta}(t, \underline{\eta}_0)\| < \varepsilon \quad \forall t \in I \quad (14.95)$$

i.e. the solution  $\underline{\zeta}$  is continuous in  $\underline{x}_0$  and  $\underline{\eta}$ .

#### 14.4.4 Lyapunov's direct method

The idea behind Lyapunov's direct method is as follows: If the energy contained in a physical system is continuously dissipated, then the system must eventually settle down to an equilibrium point. The approach was pictured by Lagrange (1788: Mécanique Celeste, Dunod, Paris) who showed that an equilibrium of a mechanical system is stable if it corresponds to a local minimum of the potential energy function. Lyapunov (1892: Problème General de la Stabilité du Mouvement, Princeton University Press) derived from this idea a general theory applicable to *any* differential equation.

**Example:**

Investigate the stability of the origin of

$$\begin{aligned} \dot{x} &= -y - x^3 \\ \dot{y} &= x - y^3 \end{aligned} \quad (14.96)$$

The phase portrait displays a stable focus. We use an energyconcept to arrive at the same conclusion. The (energy) function

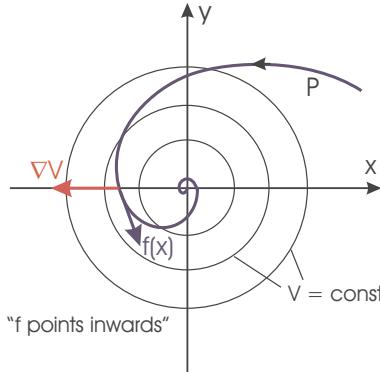
$$V = x^2 + y^2 = c, \quad c > 0$$

defines a family of concentric circles in the phase plane.

Stability implies that if we choose any point on the circle, the phase path  $p$  through that point is directed towards the interior.

$$\begin{aligned} \left( \frac{dV}{dt} \right) |_p &= \left( \frac{\partial V}{\partial x} \dot{x} + \frac{\partial V}{\partial y} \dot{y} \right) |_p \\ &= 2x[-y - x^3] + 2y[x - y^3] + 2y[x - y^3] \\ &= -2[x^4 + y^4] < 0 \end{aligned}$$

This is equivalent to the fact that the scalar function  $V$  (energy) decreases along the solutions of (14.96).



**Abb. 14.21:** Phase path  $P$  with concentric circles  $V = \text{const.}$

**Definition 14.4.7** Let

$$\dot{\underline{x}} = \underline{f}(\underline{x}), \quad \underline{x}(0) = \underline{x}_0 \quad (14.97)$$

$\underline{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a continuously differentiable function. Then the *orbital derivative* of  $V$  is defined by

$$\frac{d}{dt}V(\underline{x}) = \nabla V^T \underline{f}(\underline{x}) \quad (14.98)$$

(derivative of  $V$  along solutions of the dynamic system)

**Note:** (14.98) results simply by applying the chain rule to  $V(\underline{x})$  which depends implicitly on  $t$ :  $V(\underline{x}(t))$ .

Let  $\mathcal{I}$  be a compact invariant set. For simplicity we think of  $\mathcal{I}$  as the origin  $\mathcal{I} = \{\underline{0}\}$ . Hence the origin is an equilibrium point of (14.97).

**Definition 14.4.8** Let  $D \subset \mathbb{R}^n$  be an open neighborhood of  $\mathcal{I}$ . A *Lyapunov function* is a  $C^1$ -function  $V : D \rightarrow \mathbb{R}$  with the following properties

1.

$$V(\underline{x}) \geq 0 \quad \forall \underline{x} \in D \quad \text{and} \quad V(\underline{x}) = 0 \Leftrightarrow \underline{x} \in \mathcal{I} \quad (14.99)$$

(I.e., since  $\mathcal{I} = \{\underline{0}\}$ ,  $V$  is locally positive definite.)

2.

$$\dot{V}(\underline{x}) \leq 0 \quad \forall \underline{x} \in D \quad (14.100)$$

The basic stability theorem of Lyapunov's direct method is as follows:

**Theorem 14.4.3** If there exists a Lyapunov function  $V$  defined on a neighborhood of  $\mathcal{I}$ , then  $\mathcal{I}$  is stable.

Note that the theorem provides only a sufficient condition for stability.

**Example:**

[Pendulum]

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin x_1 \\ V &= \underbrace{(1 - \cos x_1)}_{\text{potential energy}} + \underbrace{\frac{1}{2}x_2^2}_{\text{kinetic energy}}\end{aligned}$$

$V$  is  $C^1$  and positive definite  $\rightarrow$  it is a candidate.

$$\dot{V} = \sin x_1 \dot{x}_1 + x_2 \dot{x}_2 = \sin x_1 x_2 - \sin x_1 x_2 = 0$$

$\Rightarrow$  Equilibrium 0 is stable.

**Example:**

[Pendulum with friction]

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin x_1 - x_2\end{aligned}$$

Using the same Lyapunov-function as in the case without friction, i.e.

$$V = (1 - \cos x_1) + \frac{1}{2}x_2^2$$

yields <sup>3)</sup>

$$\dot{V} = -x_2^2 \leq 0$$

$\Rightarrow$  origin is stable

Actually, we know (from the phase portrait) that the origin is (uniformly) asymptotically stable. Our choice of  $V$  fails to show this fact.

However,  $\dot{V}$  is negative everywhere except on the line  $x_2 = 0$ . For  $\dot{V} = 0$  to hold, the trajectories of the system must be confined to the  $x_2 = 0$  line. It is clear that this is only possible at the origin. Therefore  $V$  must decrease along the solutions and tend to zero, which is consistent with the fact that in the presence of friction, the energy cannot remain constant while the system is in motion.

The following theorem formalizes our observations:

---

<sup>3)</sup>  $\dot{V}$  is negative semidefinite because  $\dot{V} = 0$  for all  $\{\underline{x} \mid x_2 = 0\}$  irrespective of the value of  $x_1$ .

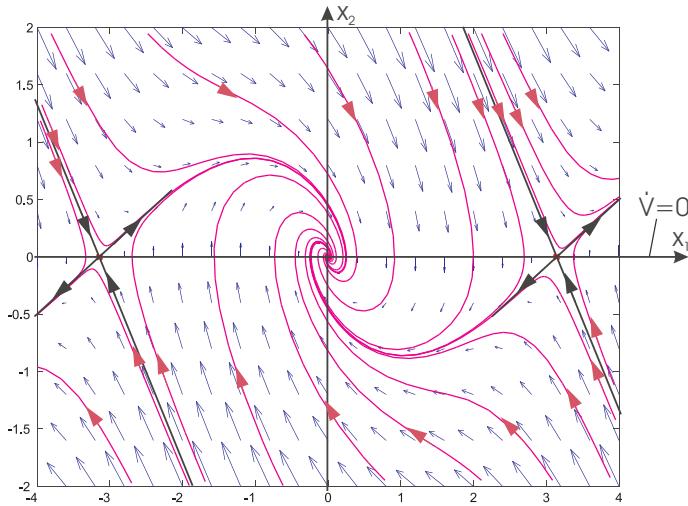


Abb. 14.22: Phase portrait of pendulum with friction

**Theorem 14.4.4 [La Salle 1960]** Let  $\Omega \subset D$  be a compact, positive invariant set. Suppose there is a  $C^1$ -function<sup>4)</sup>  $V : D \rightarrow \mathbb{R}$  such that  $\dot{V} \leq 0$  in  $\Omega$ . Let  $E = \{\underline{x} \in \Omega \mid \dot{V}(\underline{x}) = 0\}$  and let  $M$  be the largest invariant set in  $E$ . Then every solution starting in  $\Omega$  approaches  $M$  as  $t \rightarrow \infty$ .

The theorem is known as *La Salle's Invariance Principle*. Unlike Lyapunov's theorem, it does not require the function  $V(\underline{x})$  to be positive definite. In the analysis of the pendulum, we have used a special version of the Invariance Principle, where  $V(\underline{x}) > 0$  and  $M = \{0\}$ .

**Corollary 14.4.1** Let  $\underline{x} = 0$  be an equilibrium of (14.97) and  $V : D \rightarrow \mathbb{R}$  be a Lyapunov function. Define  $S = \{\underline{x} \in D \mid \dot{V} = 0\}$  and suppose that no solution can stay in  $S$  other than the trivial solution  $\zeta_t(\underline{x}) \equiv 0$ . In other words, no positive orbit  $\gamma^+(\underline{x})$  is contained in the set  $S' = \{\underline{x} \in D \setminus \{0\} \mid \dot{V}\}$ . Then,  $\{0\}$  is asymptotically stable.

An elegant way of proving asymptotical stability is to find a *strict* Lyapunov function.

**Definition 14.4.9** Let  $V$  be a Lyapunov function and, in addition,  $\dot{V} = 0 \Leftrightarrow \underline{x} \in \mathcal{I}$ .

Then,  $V$  is called a *strict* Lyapunov function.

**Theorem 14.4.5** Let  $\mathcal{I} = \{0\}$  (for simplicity), i.e. the origin is an equilibrium point of (14.97). Let  $V : D \rightarrow \mathbb{R}$  defined on a neighborhood of 0 be a strict Lyapunov function. Then the origin is asymptotically stable.

The proof is a direct consequence of the above corollary.

By definition  $\dot{V}(\underline{x}) = 0 \Leftrightarrow \underline{x} = 0$ , so  $S' = \{\underline{x} \in D \setminus \{0\} \mid \dot{V} = 0\} = \emptyset$ .

**Example:**

Let us look for a strict Lyapunov function for the pendulum with friction.

$$V = (1 - \cos x_1) + \underbrace{\frac{1}{2} \underline{x}^T \mathbf{P} \underline{x}}_{\text{instead of } \frac{1}{2} x_2^2}$$

$\mathbf{P} > 0$ : positive definite, symmetric matrix

$$\begin{bmatrix} p_1 & p_3 \\ p_3 & p_2 \end{bmatrix} > 0 \Leftrightarrow \begin{array}{l} p_1 > 0 \\ p_1 p_2 - p_3^2 > 0 \end{array}$$

$$\dot{V} = (1 - p_2) x_2 \sin x_1 - p_3 x_1 \sin x_1 + (p_1 - p_3) x_1 x_2 + (p_3 - p_2) x_2^2$$

Cancel the two summands with indefinite sign by taking  $p_2 = 1$ ,  $p_1 = P_3$ . Now, from  $\mathbf{P} \stackrel{!}{>} 0$ ,  $0 < p_3 < 1$ , e.g.  $p_3 = \frac{1}{2}$

$$\dot{V}(\underline{x}) = -\frac{1}{2} \left( \underbrace{x_1 \sin x_1}_{>0, -\pi < x_1 < \pi} + x_2^2 \right)$$

Taking  $D = \{\underline{x} \in \mathbb{R}^2 \mid |x_1| < \pi\}$  we see that  $V(\underline{x}) > 0$  and  $\dot{V} < 0 \quad \forall \underline{x} \in D \Rightarrow \{0\}$  is asymptotically stable.

When the origin is asymptotically stable we are often interested in determining, how far from the origin the trajectory can be and still converge to the origin as  $t \rightarrow \infty$ . This gives rise to the following definition.

**Definition 14.4.10** The set  $A(0) = \{\underline{x} \in X \mid \text{dist}(\zeta_t(\underline{x}), 0) \rightarrow 0\}$  is called *basin (region) of attraction* of 0.

Finding the exact region of attraction  $A$  may be difficult. However, Lyapunov's method can be used to find  $A$  or an estimate of it. As a first step we make the following conjecture:

Let  $V : D \rightarrow \mathbb{R}$  be a strict Lyapunov function. Then  $D \subset A$ , i.e.  $D$  is an estimate of the region of attraction.

In the above example,  $V$  is a strict Lyapunov function on  $D = \{\underline{x} \in \mathbb{R}^2 \mid |x_1| < \pi\}$  but in the phase portrait we observe trajectories starting in  $D$  which

do not remain in  $D$ . Once a trajectory leaves  $D$  there is no guarantee that  $\dot{V}$  is negative. The problem does not arise if  $A$  is estimated by a compact, positive invariant  $\Omega \subset D$ .

From La Salle's principle we know that  $\Omega \subset A$  (since every solution starting in  $\Omega$  approaches the largest invariant set  $M$  as  $t \rightarrow \infty$ ). The simplest such estimate is the set

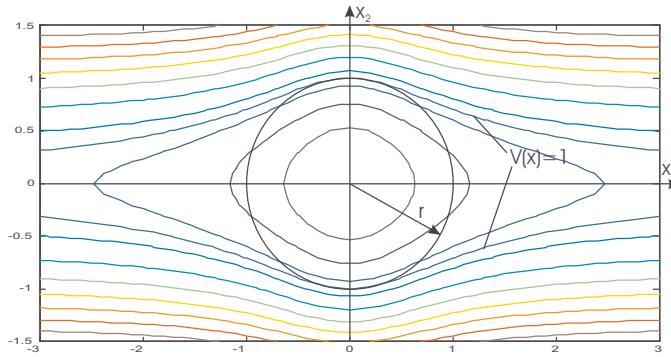
$$\Omega_\gamma = \{\underline{x} \in \mathbb{R}^n \mid V(\underline{x}) \leq \gamma\} \quad (14.101)$$

provided that it is bounded and contained in  $D$ . From the proof of Lyapunov's direct method we know that  $\Omega_\gamma$  is contained in the interior of a ball  $B_r$  if  $\gamma$  satisfies

$$\gamma < \mu = \min_{\|\underline{x}\|=r} V(\underline{x})$$

### Example:

$$V = \frac{x_1^2}{1+x_1^2} + x_2^2$$



**Abb. 14.23:** Contour lines of  $V(\underline{x}) = \text{const.}$

$$\lim_{r \rightarrow \infty} \min_{\|\underline{x}\|=r} \left( \frac{x_1^2}{1+x_1^2} + x_2^2 \right) = \lim_{x_1 \rightarrow \infty} \frac{x_1^2}{1+x_1^2} = 1$$

$\Omega_\gamma$  is bounded only for  $\gamma < 1$ .

Suppose  $\gamma > 1$ :

Then the radius of a ball in which  $\Omega_\gamma$  is contained is infinite, since the level curve  $V(\underline{x}) = \gamma$  is not closed.

A condition that ensures that  $\Omega_\gamma$  is bounded for all values of  $\gamma > 0$  is that for all  $\underline{x} \in \mathbb{R}^n$

$$V(\underline{x}) \rightarrow \infty \quad \text{as} \quad \|\underline{x}\| \rightarrow \infty \quad (14.102)$$

Such a  $V$  is called **radially unbounded**.

(14.102) implies that for any  $\gamma > 0$  there is a  $r > 0$  such that  $V(\underline{x}) > \gamma$  whenever  $\|\underline{x}\| > r$ . This means that any level curve  $V(\underline{x}) = \gamma$  is closed and there is a ball  $B_r$  such that  $\Omega_\gamma \subset B_r$ . Hence,  $\Omega_\gamma$  is bounded (and closed by definition). Further,  $\zeta_t(\Omega_\gamma) \subset \Omega_\gamma$ , since  $V(\underline{x})$  is decreasing along the solution of (14.97), i.e.  $\Omega_\gamma$  is an estimate of the region of attraction  $A$  of the origin. Since any  $\underline{x} \in \mathbb{R}^n$  can be included in  $\Omega_\gamma$  (by choosing  $\gamma$  large enough), we see that  $A = \mathbb{R}^n$ .

We conclude:

**Theorem 14.4.6** Let  $\underline{x} = 0$  be an equilibrium of (14.97) and  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  be a radially unbounded strict Lyapunov function. Then  $\underline{x} = 0$  is *globally* asymptotically stable.

If  $V$  is not radially unbounded, the estimate  $\Omega_\gamma$  is usually quite conservative. From La Salle's principle we know that any compact set  $\Omega \subset D$  is contained in  $A$ , provided that we can show that  $\Omega$  is positive invariant.

#### 14.4.5 LTI Systems and Lyapunov Stability

The stability of LTI systems can be determined using Lyapunov's direct method, without explicit knowledge of the solutions, and is stated in the following theorem.

**Theorem 14.4.7** The equilibrium state  $\underline{x} = 0$  of the linear time-invariant system

$$\dot{\underline{x}} = \mathbf{A}\underline{x} \quad (14.103)$$

is asymptotically stable if, and only if, given any symmetric positive definite matrix  $\mathbf{Q}$ , there exists a symmetric positive definite matrix  $\mathbf{P}$ , which is the unique solution of the set of  $n(n+1)/2$  linear equations

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}. \quad (14.104)$$

Therefore,  $V(\underline{x}) = \underline{x}^T \mathbf{P} \underline{x}$  is a Lyapunov function for Eq. (14.103).

*Proof* Sufficiency follows directly by choosing a matrix  $\mathbf{P} = \mathbf{P}^T > 0$  satisfying Eq. (14.104). If  $V(\underline{x}) = \underline{x}^T \mathbf{P} \underline{x} > 0$ , the time derivative of  $V$  along the solutions of Eq. (14.103) is given by

$$\dot{V}(\underline{x}) = \underline{x}^T [\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A}] \underline{x} = -\underline{x}^T \mathbf{Q} \underline{x} < 0.$$

Hence,  $V(\underline{x})$  is a Lyapunov function so that the asymptotic stability of  $\underline{x} = 0$  of Eq. (14.103) follows.

To prove necessity, assume that  $\underline{x} = 0$  in Eq. (14.103) is asymptotically stable. Then a matrix  $\mathbf{P}$  defined as

$$\mathbf{P} \triangleq \int_0^\infty \exp(\mathbf{A}^T t) \mathbf{Q} \exp(\mathbf{A} t) dt \quad (14.105)$$

exists, is symmetric and positive definite. To show that  $\mathbf{P}$  as defined by Eq. (14.105) is the unique solution of Eq. (14.104), let  $\bar{\mathbf{P}}$  be any other solution of Eq. (14.104). Then

$$\begin{aligned} \mathbf{P} &= - \int_0^\infty \exp(\mathbf{A}^T t) (\mathbf{A}^T \bar{\mathbf{P}} + \bar{\mathbf{P}} \mathbf{A}) \exp(\mathbf{A} t) dt \\ &= - \int_0^\infty \frac{d}{dt} [\exp(\mathbf{A}^T t) \bar{\mathbf{P}} \exp(\mathbf{A} t)] dt \text{ since } \exp(\mathbf{A} t) \text{ commutes with } \mathbf{A} \ \forall t \\ &= - \exp(\mathbf{A}^T t) \bar{\mathbf{P}} \exp(\mathbf{A} t) |_0^\infty = \bar{\mathbf{P}} \quad \text{since } \underline{x} = 0 \text{ in Eq. (14.103)} \end{aligned}$$

Eq. (14.104) is referred to as the *Lyapunov equation*.

Quite often, only the output of a system (rather than its state) is available for measurement; the output vector  $\underline{y} \in \mathbb{R}^m$  is defined by the following mapping

$$\underline{y} = \mathbf{C}\underline{x} \quad \mathbf{C} \in \mathbb{R}^{m \times n}, \quad \underline{x} \in \mathbb{R}^n \quad (14.106)$$

A key concept is *observability* of the pair  $(\mathbf{C}, \mathbf{A})$  which means that by observing the output we may uniquely identify the evolution of the state. It can be shown that a necessary and sufficient condition for  $(\mathbf{C}, \mathbf{A})$  to be observable is

$$\text{rank} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} = n.$$

**Corollary 14.4.2** If  $\mathbf{C} \in \mathbb{R}^{m \times n}$  and  $(\mathbf{C}, \mathbf{A})$  is observable, the origin of Eq. (14.103) is asymptotically stable if, and only if, there exists a symmetric positive definite matrix  $\mathbf{P}$  which is the unique solution of the equation

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{C}^T \mathbf{C}$$

Let  $V = \underline{x}^T \mathbf{P} \underline{x}$  where  $\mathbf{P}$  is a symmetric positive definite matrix such that  $\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{C}^T \mathbf{C}$ .  $\dot{V}$  can be evaluated along the solutions of eq. (14.103) as  $\dot{V} = -\|\mathbf{C}\underline{x}\|^2 \leq 0$ . Uniform asymptotic stability follows from La Salle's principle as shown below. The set  $E$  is defined by

$$E \triangleq \{\underline{x} \mid \mathbf{C}\underline{x} = 0\}$$

On an invariant set in  $E$ ,  $\dot{V}(\underline{x}(t)) \equiv 0$  i.e.  $\underline{y}(t) \equiv 0$  for all  $t \geq 0$ . This implies,

$$\underline{y}(t) = \dot{\underline{y}}(t) = \ddot{\underline{y}}(t) = \dots = \frac{d^i \underline{y}(t)}{dt^i} = 0 \quad (14.107)$$

for some  $i > 0$  (which is determined as follows). First, notice that  $\underline{y} = \mathbf{C}\underline{x}$ ,  $\dot{\underline{y}} = \mathbf{CA}\underline{x}, \dots, d^i\underline{y}/dt^i = \mathbf{CA}^i\underline{x}$ . From the Cayley–Hamilton Theorem<sup>5)</sup>,

$$\mathbf{A}^n + a_1\mathbf{A}^{n-1} + \cdots + a_{n-1}\mathbf{A} + a_n\mathbf{I} = 0 \quad (14.108)$$

where  $a_1, \dots, a_n$  are the coefficients of the characteristic polynomial  $a(\lambda) = \det(\lambda I - \mathbf{A})$ . It follows that

$$\mathbf{CA}^n\underline{x} = -a_1\underbrace{\mathbf{CA}^{n-1}\underline{x}}_{\frac{d^{n-1}\underline{y}}{dt^{n-1}}} - \cdots - a_{n-1}\underbrace{\mathbf{CA}\underline{x}}_{\dot{\underline{y}}} - a_n\underbrace{\mathbf{Cx}}_{\underline{y}} \quad (14.109)$$

This means that  $d^n\underline{y}/dt^n = \mathbf{CA}^n\underline{x}$  is a linear combination of derivatives of  $\underline{y}$  up to order  $n-1$ . Hence,  $i = n-1$  and  $\underline{y}(t) \equiv 0$  whenever

$$\begin{bmatrix} \underline{y}(t) \\ \dot{\underline{y}}(t) \\ \vdots \\ \frac{d^{n-1}\underline{y}(t)}{dt^n} \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \underline{x}(t) \equiv 0 \quad (14.110)$$

Since  $(\mathbf{C}, \mathbf{A})$  is observable, the last expression is zero if and only if  $\underline{x} = 0$ . Hence the largest invariant set contained in  $E$  is the origin.

Necessity follows directly by substituting  $\mathbf{Q} = \mathbf{C}^T\mathbf{C}$  in Eq. (14.105).

The Lyapunov equation can be used to test whether or not a matrix  $\mathbf{A}$  is Hurwitz, i.e.  $\text{Re}\lambda_i < 0$  for all eigenvalues of  $\mathbf{A}$ , as an alternative to calculating the eigenvalues of  $\mathbf{A}$ .

#### 14.4.6 Barbalat's Lemma

If  $f : \mathbb{R} \rightarrow \mathbb{R}$  is uniformly continuous for  $t \geq 0$  and if the limit of the integral

$$\lim_{t \rightarrow \infty} \int_0^t f(\tau) d\tau$$

exists and is finite then

$$\lim_{t \rightarrow \infty} f(t) = 0.$$

The key point here is that continuity is uniform, i.e. the size of the  $\delta$ -neighborhood (around some point  $t_1 > 0$ ) for which  $|f(t) - f(t_1)| < \varepsilon$  for all  $t - \delta < t_1 < t + \delta$  does **not** depend on  $t_1$ .

---

<sup>5)</sup> The Cayley–Hamilton Theorem states that any square matrix satisfies its own characteristic equation.

**Example 14.4.1**

$$\int_0^t f(\tau) d\tau = 1 - e^{-t} \cos(t) \rightarrow 1 \quad \text{as } t \rightarrow \infty \quad (14.111)$$

$$f(t) = e^{-t}[\cos(t) + \sin(t)] \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad (14.112)$$

**Example 14.4.2**

$$\int_0^t f(\tau) d\tau = e^{-t} \cos(e^t) - \cos(1) \rightarrow -\cos(1) \quad \text{as } t \rightarrow \infty \quad (14.113)$$

$$f(t) = e^{-t} \cos(e^t) - \sin(e^t) \quad (14.114)$$

Notice that  $\delta \rightarrow 0$  as  $t_1 \rightarrow \infty$ , i.e.  $f(t)$  is not uniformly continuous.

*Proof.* [122] Let  $\lim_{t \rightarrow \infty} f(t) \neq 0$ . Then there exists a positive constant  $\varepsilon$  such that for every  $T > 0$ , we can find  $T_1 \geq T$  with  $|f(T_1)| \geq \varepsilon$ . Since  $f(t)$  is uniformly continuous, there is a positive constant  $\delta$  such that  $|f(t + \tau) - f(t)| < \varepsilon/2$  for all  $t \geq 0$  and all  $0 \leq \tau < \delta$ . Hence,

$$\begin{aligned} |f(t)| &= |f(t) - f(T_1) + f(T_1)| \\ &\geq |f(T_1)| - |f(t) - f(T_1)| \\ &> \varepsilon - \frac{1}{2}\varepsilon = \frac{1}{2}\varepsilon, \quad \forall t \in [T_1, T_1 + \delta] \end{aligned}$$

Therefore,

$$\left| \int_{T_1}^{T_1 + \delta} f(\tau) d\tau \right| = \int_{T_1}^{T_1 + \delta} |f(\tau)| d\tau > \frac{1}{2}\varepsilon\delta$$

where the equality holds since  $f(t)$  retains the same sign for  $T_1 \leq t \leq T_1 + \delta$ . Thus,  $\int_0^t f(\tau) d\tau$  cannot converge to a finite limit as  $t \rightarrow \infty$ . Hence the above assumption, that  $\lim_{t \rightarrow \infty} f(t) \neq 0$  was incorrect.

**Digression: Input-Output Stability**

It is sometimes useful to regard the system as a mapping (or operator) between signal spaces. For example, given an LTI-system

$$\begin{aligned} \dot{\underline{x}} &= \mathbf{A}\underline{x} + \underline{b}u & \underline{x}(0) = 0 \\ \underline{y} &= \underline{c}^T \underline{x} + \underline{d}u \end{aligned} \quad (14.115)$$

where  $\mathbf{A}$  is asymptotically stable, one can define the linear (convolution) operator

$$(Hu)(t) = y(t) = \int_0^t \underline{c}^T e^{\mathbf{A}(t-\tau)} \underline{b} u(\tau) d\tau \quad (14.116)$$

If a system is described by an operator  $H$  that maps an input space  $\mathcal{U}$  into an output space  $\mathcal{Y}$ , the concept of stability is based on the properties of  $\mathcal{U}$  and  $\mathcal{Y}$ :

If a property  $\mathcal{L}$  of the input is invariant under the transformation  $H$ , the system is said to be  $\mathcal{L}$ -stable.

Some care must be taken when defining the signal spaces  $\mathcal{U}$  and  $\mathcal{Y}$ . On one hand, these spaces must contain all technically relevant functions of time (e.g. the set  $C[0; \infty)$  of all continuous functions on  $[0; \infty)$ ). On the other hand, the space must be complete, in the sense that every Cauchy sequence converges to a point in that space.

For example, the set of continuous real-valued functions is *not* complete since we all know a Cauchy sequence that may converge to a *discontinuous* function  $\hat{f}(t)$  (e.g. a square wave): the sequence  $\{s_N\}_{N \geq 0}$  of partial sums of the Fourier series where  $s_N$  is given by  $s_N(t) = \sum_{n=-N}^N c_n e^{int}$ . Notice that this sequence converges to  $\hat{f}(t)$  in a mean square sense, i.e.

$$\lim_{N \rightarrow \infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} [\hat{f}(t) - s_N(t)]^2 dt = 0$$

The question is how to complete the set  $C[0; \infty)$  such that the resulting space is complete even in the mean square norm. It turns out that the set of all measurable<sup>6)</sup> functions  $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}$  which satisfy  $\int_0^\infty |f(\tau)|^p d\tau < \infty$  has all the required properties.

**Definition 14.4.11** For any fixed  $p \in [1, \infty)$ ,  $f : \mathbb{R}^+ \rightarrow \mathbb{R}$  is said to belong to the space  $\mathcal{L}^p$  iff  $f$  is locally integrable and

$$\|f\|_p \triangleq \left( \int_0^\infty |f(t)|^p dt \right)^{\frac{1}{p}} < \infty.$$

When  $p = \infty$ ,  $f \in \mathcal{L}^\infty$  iff

$$\|f\|_\infty \triangleq \sup_{t \geq 0} |f(t)| < \infty.$$

Using the definition, the system represented by the operator  $H$  is said to be  $\mathcal{L}^p$ -stable if  $u \in \mathcal{L}^p$  is mapped into  $y \in \mathcal{L}^p$ . When  $p = \infty$ ,  $\mathcal{L}^p$ -stability is also referred to as bounded-input bounded-output (BIBO) stability.

**Example 14.4.3** If  $A$  is an asymptotically stable matrix, it can be shown that the operator  $H$ , defined in (14.116), is BIBO stable.

Using the notation introduced in the above definition we can state the following useful version of Barbalat's lemma.

*Corollary.* If  $g \in \mathcal{L}^2 \cap \mathcal{L}^\infty$  (i.e.  $g$  is square integrable and uniformly bounded), and  $\dot{g}$  is bounded, then  $\lim_{t \rightarrow \infty} g(t) = 0$ .

---

<sup>6)</sup> A function  $f(t)$  is measurable iff  $f(t)$  is the limit of a sequence of staircase functions at all  $t$  except a set of measure zero

*Proof.* Choose  $f(t) = g^2(t)$ . Then  $f(t)$  satisfies the conditions of Barbalat's Lemma.

(Here, we use the fact that if  $\dot{g}$  is bounded, i.e.  $|\dot{g}| \leq \bar{g}$  then  $g(t)$  is uniformly continuous. *Proof:* For all  $t, t_1$  there is a  $t_2$  between  $t$  and  $t_1$  such that  $g(t) - g(t_1) = \dot{g}(t_2)(t - t_1)$ . Hence  $|g(t) - g(t_1)| < \varepsilon$  for all  $|t - t_1| < \varepsilon/\bar{g} := \delta$ . Hence  $\delta$  is independent of  $t_1$ .  $\triangleleft$ )

# 15 Lernende Automaten

Paul Kotyczka, Matthias Feiler

## 15.1 Einleitung

Das vorige Kapitel beschäftigte sich mit der Unterdrückung von Störungen, die Ausgang eines exogenen Zustandsraummodells sind. Dabei wurde sowohl der Fall autonomer Störmodelle betrachtet, als auch die Situation, in der das Exosystem mit weißem Rauschen am Eingang beaufschlagt wird. Mit verschiedenen, auch adaptiven Verfahren ist sehr gute Störunterdrückung zu erreichen, wobei bei stochastischen Störungen die Dynamik des Exosystems eine entscheidende Rolle für den Erfolg der Maßnahme spielt (vgl. etwa Abb. 14.5). Zur Anpassung an zeitvariante Störungen bietet sich die adaptive Variante der Regelung mit multiplen Modellen (Multiple Model Switching and Tuning) an.

Das vorliegende Kapitel beschäftigt sich mit *Automaten*, die als selbstständige und lernende Entscheidungsmechanismen in *stochastischen Umgebungen* operieren. Gerade in Regelsystemen können sie in Ergänzung bestehender Architekturen eingesetzt werden, um die Regelgüte unter dem Einfluss bestimmter stochastischer Störungen zu erhöhen. Eine typische Aufgabe für Automaten ist das Erkennen von Regelmäßigkeiten zeitlicher oder räumlicher Muster bei der Zeitreihenanalyse oder Bildverarbeitung.

Die Mustererkennung lässt sich als Problem der theoretischen Informatik formulieren. Eine *formelle Sprache* ist durch eine Menge von *Worten* definiert, die aus *Symbolfolgen* eines endlichen *Alphabets* gebildet werden. Nach der Klassifikation von Chomsky [215] heißt eine Sprache regulär, wenn sie durch einen endlichen Automaten darstellbar ist. Die Sprache besteht dabei aus der Menge vom Automaten akzeptierter Worte. Worte werden dadurch erkannt (akzeptiert), dass der Automat beim „Lesen“ des Wortes eine Folge von Zuständen (und damit eine Symbolfolge) durchläuft. Die Transitionen von einem Zustand  $\psi[n]$  nach  $\psi[n + 1]$  hängen von  $\psi[n]$  und dem nächsten Buchstaben des Wortes ab. Liegt der letzte Zustand  $\psi[N]$  (bei einer Wortlänge  $N$ ) in einer Menge vordefinierter Endzustände, so wird das Wort akzeptiert und zählt zur vom Automaten dargestellten Sprache. Im sogenannten Wortproblem der theoretischen Informatik muss entschieden werden, ob ein gegebenes Wort zu einer gegebenen regulären Sprache gehört. Ein erkanntes Wort entspricht damit einem bestimmten Muster in einer Symbolfolge.

Das in diesem Beitrag vorgestellte Anwendungsbeispiel für (lernende) Automaten in Regelsystemen ist eine Erweiterung des Ansatzes der multiplen Modelle (vgl. Kap. 14.3) um einen Prognosemechanismus für durch eine *Markov-Kette* beschriebene, sprunghafte Parameteränderungen. Für den unbekannten Parametervektor  $\underline{\theta}(k)$  existieren  $N$  über den Wertebereich von  $\underline{\theta}(k)$  verteilte Modelle  $\{\hat{\underline{\theta}}_1, \dots, \hat{\underline{\theta}}_N\}$ , die durch Elemente der Menge  $\Omega = \{1, \dots, N\}$  indiziert werden. Zu jedem diskreten Zeitpunkt  $k$  wird die Abstandsmetrik (vgl. Gl. (14.67), Kap. 14.3), die über dem Prädiktionsfehler  $\varepsilon_i = y - \hat{y}_i$  definiert ist, ausgewertet und jenes Modell  $\hat{\underline{\theta}}_i$  zur Regelung genutzt, welches dem gemessenen Ausgang am nächsten kommt. Durch die inhärente Verzögerung bei der Identifikation des gültigen Modells entsteht ein Regelfehler, der nur durch richtige und rechtzeitige Parameterprognose eliminiert werden kann. Aufgabe der Automaten ist in diesem Fall, dominante Sequenzen im Verlauf der identifizierten Modelle  $\underline{\theta}_i$  zu erkennen und darauf basierend Prognosen für den nächsten gültigen Parameter zu liefern. Im Kontext der formalen Sprachen bedeutet das, in der mit Symbolen aus  $\Omega$  indizierten Parameterfolge Worte zu erkennen und eine Grammatik abzuleiten, die auf zukünftige Sequenzen angewendet wird.

Der Beitrag ist wie folgt gegliedert: In Abschnitt 2 werden für die Formulierung der lernenden Automaten wichtige Tatsachen aus der Wahrscheinlichkeitsrechnung zusammengefasst und eine kurze Einführung in die Theorie der Markov-Ketten gegeben. Abschnitt 3 behandelt vor allem stochastische Automaten mit veränderlicher Struktur, sowie einige grundlegende lineare Lerngesetze, mit denen sich ihr Verhalten anpassen lässt. Ferner wird ein typischer deterministischer Automat vorgestellt, der trotz seiner einfachen Struktur ebenfalls *asymptotisch optimales* Verhalten aufweist. Das Hauptaugenmerk bei der Analyse der verschiedenen Automaten liegt auf der Konvergenz ihrer Handlungen gegen die jeweils optimale. Im vierten Abschnitt werden die Automaten schließlich zur Parameterprognose bei der Regelung mit multiplen Modellen eingesetzt.

## 15.2 Mathematische Grundlagen

Um das Verhalten von lernenden Automaten unter verschiedenen Lerngesetzen beschreiben zu können, sind Grundlagen aus der Wahrscheinlichkeitsrechnung und insbesondere der Theorie der Markov-Ketten notwendig. In diesem Abschnitt sind einige Definitionen und Sätze zusammengefasst, die dem Buch *Markov Chains – Theory and Applications* [110] entnommen sind, das auch die entsprechenden Beweise enthält.

### 15.2.1 Stochastische Prozesse

Ein *stochastischer Prozess* ist eine Folge von Zufallsexperimenten, deren Ergebnisse durch Zufallsvariablen in Form reeller Zahlen dargestellt werden. Das Ergebnis eines Experiments wird als *Ereignis*  $\omega$  bezeichnet und ist Element des

*Ereignisraumes*  $\Omega$ . Jedem Ereignis ist die Wahrscheinlichkeit  $\Pr(\omega)$  zugeordnet, wobei die Summe aller Wahrscheinlichkeiten auf dem Ereignisraum 1 ist. Die *Zufallsvariable*  $X(\omega)$ , eine Funktion  $\Omega \rightarrow S \subset \mathbb{R}$ , ordnet jedem Ereignis eine reelle Zahl zu, abstrahiert also das Ergebnis des Zufallsexperiments. Wir beschränken uns auf eine *diskrete Zufallsvariable*  $X : \Omega \rightarrow S \subset \mathbb{N}$ , deren Wertemenge die natürlichen Zahlen sind. Die Wertemenge der Zufallsvariablen wird als *Zustandsraum* des stochastischen Prozesses bezeichnet. Ist dieser abzählbar oder endlich, dann spricht man von einer stochastischen *Kette*. Ein *zeitdiskreter* stochastischer Prozess wird durch die Folge von Zufallsvariablen  $X[1], X[2], X[3], \dots$ , die über dem Ereignisraum  $\Omega$  definiert sind, notiert. Der Index bezeichnet dabei die zeitliche Abfolge der Ereignisse.

**Beispiel 15.2.1** Ein geläufiges Beispiel für eine endliche stochastische Kette ist Würfeln. Das Zufallsexperiment ist der Wurf, die möglichen Ereignisse  $\omega \in \Omega$  sind die geworfenen Augenzahlen. Die Zufallsvariable  $X = X(\omega)$  ordnet jedem Ereignis die entsprechende Zahl zwischen 1 und 6 zu. Der Zustandsraum der Kette ist also  $S = \{1, \dots, 6\}$ . Jedem Ereignis ist (bei einem fairen Wurf) die gleiche Wahrscheinlichkeit  $\Pr(\text{"gewürfelte Zahl ist } i) = 1/6$  mit  $i \in S$  zugeordnet. Anders gesagt ist die Wahrscheinlichkeit, dass die Zufallsvariable  $X(\omega)$  einen Wert  $i \in S$  annimmt, gegeben durch:  $\Pr(X(\omega) = i) = 1/6$ . Eine mögliche Folge der Zufallsvariablen bei vier aufeinander folgenden Würfen ist durch  $\{X[1], X[2], X[3], X[4]\} = \{2, 3, 6, 1\}$  gegeben.

Kenngröße einer stochastischen Kette mit Zustandsraum  $S \subset \mathbb{N}$  ist die *Verteilung* der Wahrscheinlichkeiten der Zufallsvariable. Sie lässt sich etwa durch die *Verteilungsfunktion* von  $X$  beschreiben

$$F(x) = \Pr(X \leq x) = \sum_{y \leq x} \Pr(X = y) \quad (15.1)$$

oder die *Wahrscheinlichkeits-* oder Dichtefunktion der Verteilung.

$$f(x) = \Pr(X = x) \quad (15.2)$$

**Beispiel 15.2.2** Wird zwei Mal gewürfelt und die Anzahl der Augen addiert, dann ist die Wertemenge der Zufallsvariable  $S = \{2, 3, \dots, 12\}$ . Es ergeben sich die in Abb. 15.1 gezeigte Verteilungsdichte- und Verteilungsfunktion.

### 15.2.2 Markov-Ketten

Die im Kontext der lernenden Automaten bedeutendste Unterklasse stochastischer Prozesse sind die *Markov-Prozesse*. Besitzen sie einen endlichen Zustandsraum, spricht man von *Markov-Ketten*. Die Zustandstransitionen eines deterministischen Automaten in einer stochastischen Umgebung werden etwa durch eine solche Markov-Kette beschrieben.

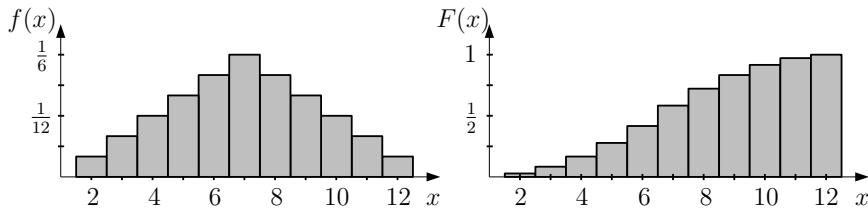


Abb. 15.1: Verteilungsdichte- und Verteilungsfunktion beim zweimaligen Würfeln

**Definition 15.2.1** Ein Zufallsprozess mit den folgenden Eigenschaften wird Markov-Kette genannt:

- Der Zufallsprozess ist zeitdiskret, wird also durch die Folge der Zufallsvariablen  $\{X[n]\}$ ,  $n = 1, 2, \dots$  notiert.
- Der Zustandsraum des Zufallsprozesses sind die natürlichen Zahlen oder eine Untergruppe:  $S \subseteq \mathbb{N}$ .
- Der Zufallsprozess erfüllt die Markov-Eigenschaft

$$\begin{aligned} \Pr(X[n] = i[n] \mid X[n-1] = i[n-1], \dots, X[0] = i[0]) &= \\ &= \Pr(X[n] = i[n] \mid X[n-1] = i[n-1]) \end{aligned} \quad (15.3)$$

wobei  $i[n]$  den Wert der Zufallsvariablen  $X[n]$  darstellt.

Die Wahrscheinlichkeit der Zustandstransition  $X[n-1] \rightarrow X[n]$  einer Markov-Kette wird also ausschließlich durch den unmittelbaren Ausgangszustand  $X[n-1]$ , jedoch nicht durch weiter zurück liegende Werte der Zufallsvariablen, also die „Vorgeschichte“ des stochastischen Prozesses bestimmt. Sind die Übergangswahrscheinlichkeiten über der Zeit konstant, gilt also für beliebige Zeitpunkte  $n$  und ganze Zahlen<sup>1)</sup>  $k$  sowie Zustände  $i, j \in S$

$$\Pr(X[n] = j \mid X[n-1] = i) = \Pr(X[n+k] = j \mid X[n+k-1] = i) \quad (15.4)$$

dann spricht man von einer *stationären*, andernfalls von einer *instationären* Markov-Kette.

**Grundbegriffe** Die Übergangswahrscheinlichkeiten der Markov-Kette werden in der sog. *Transitionsmatrix*  $\mathbf{P}$  geschrieben. Diese ist eine stochastische Matrix, d. h. alle Einträge sind nichtnegativ und die Summe der Elemente einer Zeile ist jeweils 1. Die  $i$ -te Zeile enthält die Übergangswahrscheinlichkeiten ausgehend vom Zustand  $i$ . Für eine Markov-Kette mit  $N$  Zuständen ist also

<sup>1)</sup> so dass  $n+k-1 \geq 0$

$$\mathbf{P} = \begin{bmatrix} p_{11} & \dots & p_{1N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \dots & p_{NN} \end{bmatrix} \quad \text{mit} \quad p_{ij} = \Pr(X[n] = j \mid X[n-1] = i) \quad (15.5)$$

Mit gegebenem Zustandsraum und bekannter Transitionsmatrix lassen sich nun die Wahrscheinlichkeiten angeben, mit denen sich die Markov-Kette zum Zeitpunkt  $n$  im Zustand  $i$  befindet:

$$a_i[n] = \Pr(X[n] = i), \quad i \in S \quad (15.6)$$

Im Vektor

$$\underline{a}[n] = [a_1[n] \ a_2[n] \ \dots \ a_N[n]]^T \quad \sum_{i=1}^N a_i[n] = 1 \quad (15.7)$$

sind damit alle Wahrscheinlichkeiten zusammengefasst, mit denen sich die Markov-Kette zur Zeit  $n$  in einem bestimmten Zustand befindet. In Anlehnung an die Verteilungsdichtefunktion einer Zufallsvariable wird dieser Vektor als *Verteilung der Markov-Kette* zur Zeit  $n$  bezeichnet. Der *Startvektor*  $\underline{a}[0]$  enthält die Wahrscheinlichkeiten für die möglichen Anfangszustände der Markov-Kette. Die Wahrscheinlichkeit, dass sich die Markov-Kette bei gegebener Verteilung  $\underline{a}[n-1]$  zur Zeit  $n$  im Zustand  $j$  befindet, lässt sich durch

$$\begin{aligned} a_j[n] &= \Pr(X[n-1] = 1) \cdot \Pr(X[n] = j \mid X[n-1] = 1) + \dots + \\ &\quad + \Pr(X[n-1] = N) \cdot \Pr(X[n] = j \mid X[n-1] = N) = \\ &= \sum_{i=1}^N a_i[n-1] \cdot p_{ij} \end{aligned} \quad (15.8)$$

berechnen. Der ganze Vektor der Verteilung zur Zeit  $n$  lautet entsprechend

$$\underline{a}[n] = \mathbf{P}^T \underline{a}[n-1] \quad (15.9)$$

Wiederholtes Anwenden dieser Vorschrift ergibt die Verteilung der Markov-Kette bei gegebenem Startvektor  $\underline{a}[0]$ :

$$\underline{a}[n] = (\mathbf{P}^T)^n \underline{a}[0] = (\mathbf{P}^n)^T \underline{a}[0] \quad (15.10)$$

Die Elemente dieser Mehrschritt-Transitionsmatrix  $\mathbf{P}^n$  sind die Wahrscheinlichkeiten

$$p_{ij}^{(n)} = \Pr(X[k] = j \mid X[k-n] = i) \quad (15.11)$$

mit denen ein Zustand  $i$  nach  $n$  Transitionen in den Zustand  $j$  übergeht. Für die Wahrscheinlichkeit, mit der sich die Kette zur Zeit  $n$  im Zustand  $j$  befindet, gilt also

$$a_j[n] = \sum_{i=1}^n a_i[0] p_{ij}^{(n)} \quad (15.12)$$

Beim Rechnen mit diesen Mehrschritt-Transitionswahrscheinlichkeiten ist die sog. Chapman-Kolmogorov-Identität hilfreich. Es gilt für alle natürlichen Zahlen  $l$  und  $m$ :

$$p_{ij}^{(l+m)} = \sum_{k=1}^N p_{ik}^{(l)} p_{kj}^{(m)} \quad (15.13)$$

Die eine Markov-Kette beschreibende Transitionsmatrix lässt sich in Form eines gerichteten Graphen oder Digraphen darstellen (vgl. Abb. 15.2). Ein Zustand der Markov-Kette wird als Kreis dargestellt. Die Kreise untereinander sind durch mit den Transitionswahrscheinlichkeiten gewichtete Pfeile (Kanten) verbunden. Aus der graphischen Darstellung lassen sich viele der nachfolgend vorgestellten Eigenschaften sofort ablesen.

*Eigenschaften* Eine wesentliche Frage ist, ob die Verteilung der Kette im Grenzwert, also  $\lim_{n \rightarrow \infty} \underline{a}[n]$  unabhängig vom Startvektor  $\underline{a}[0]$  ist. Betrachtet man Gl. (15.12), so erkennt man, dass dies der Fall ist, wenn für alle  $j$ , unabhängig von  $i$ , der Grenzwert

$$\pi_j := \lim_{n \rightarrow \infty} p_{ij}^{(n)} \quad \text{mit} \quad \sum_{j=1}^n \pi_j = 1 \quad (15.14)$$

existiert.

**Definition 15.2.2** Gilt für eine Markov-Kette Gleichung (15.14), dann ist sie ergodisch.

$\pi_j$  ist der vom Anfangszustand  $i$  unabhängige Grenzwert der  $n$ -Schritt-Transitionswahrscheinlichkeit zum Zustand  $j$  der ergodischen Markov-Kette. Damit ist  $\pi_j$  gleichzeitig die Wahrscheinlichkeit, dass der Zustand  $j$  für  $n \rightarrow \infty$  eingenommen wird. Aus Gl. (15.12) wird deutlich, dass

$$\lim_{n \rightarrow \infty} a_j[n] = \sum_{i=1}^N (a_i[0] \pi_j) = \pi_j \quad (15.15)$$

Im Vektor  $\underline{\pi}$  zusammengefasst stellen diese Grenzwerte die sog. *stationäre Verteilung* der Markov-Kette dar, die sich aus dem durch die Forderungen

$$\underline{\pi} = \mathbf{P}^T \underline{\pi}, \quad \sum_{i=1}^N \pi_i = 1 \quad (15.16)$$

resultierenden Gleichungssystem berechnet. Wie später gezeigt wird, ist Ergodizität eine wichtige Eigenschaft von Lerngesetzen stochastischer Automaten. Die Ergodizität einer Markov-Kette lässt sich anhand der Eigenschaften *Irreduzibilität*, *positive Rekurrenz* und *Aperiodizität* untersuchen, die im Folgenden kurz eingeführt werden.

**Definition 15.2.3** Eine Markov-Kette ist reduzibel, falls der Zustandsraum  $S$  eine abgeschlossene Untermenge echt enthält. Die Kette ist irreduzibel, falls keine nicht-leere abgeschlossene Menge außer der Zustandsraum  $S$  selbst existiert.

Abgeschlossen bedeutet also, dass für alle Zustände einer Untermenge  $C \subset S$  des Zustandsraumes die Transitionswahrscheinlichkeit aus dieser Untermenge heraus Null ist. Formell also  $p_{ik} = 0$  für alle  $i \in C$  und alle  $k \notin C$ . Ist die abgeschlossene Untermenge ein einziger Zustand, so wird dieser als *absorbierender Zustand* bezeichnet. Die obige Definition der Reduzibilität gilt auch für Untermengen des Zustandsraums.

Zwei Zustände  $i$  und  $j$  sind *kommunizierende Zustände*, wenn die Wahrscheinlichkeit, dass es *irgendwann* zu einer Zustandstransition zwischen ihnen kommt, größer Null ist. Es müssen also Zahlen  $n \geq 0$  und  $m \geq 0$  existieren, so dass  $p_{ij}^{(n)} > 0$  und  $p_{ji}^{(m)} > 0$  gilt. Kommunizieren alle Zustände einer Markov-Kette (oder einer Untermenge  $C \subset S$ ), dann genau ist die Kette (oder die Untermenge  $C$ ) irreduzibel.

**Definition 15.2.4** Der Zustand  $j$  einer Markov-Kette hat die Periode  $d$ , falls  $p_{jj}^{(n)} = 0$  außer für  $n = md$  mit  $m \in \mathbb{N}$ . Die Periodendauer  $d$  ist die größte ganze Zahl, für die die genannte Bedingung gilt.

Die Periode  $d$  lässt sich ermitteln als der größte gemeinsame Teiler aller Zeitpunkte  $n$ , für die  $p_{jj}^{(n)} > 0$  gilt. Für einen 2-periodischen Zustand ist also die Wahrscheinlichkeit einer Rückkehr zum selben Zustand nur größer Null zu den Zeitpunkten 2, 4, 6, .... Ist  $d = 1$ , so ist die Periode 1 und damit der Zustand *aperiodisch*.

Eine weitere wichtige Kenngröße für Zustände einer Markov-Kette ist die Wahrscheinlichkeit, mit der ein Zustand  $j$ , ausgehend vom Zustand  $i$  zum ersten Mal nach  $n$  Schritten erreicht wird. Diese Wahrscheinlichkeit wird mit  $f_{ij}^{(n)}$  notiert:

$$f_{ij}^{(n)} = \Pr(X[k+n] = j \mid X[k+n-1] \neq j, \dots, X[k+1] \neq j, X[k] = i) \quad (15.17)$$

Für  $i = j$  bezeichnet  $f_{ii}^{(n)}$  die Wahrscheinlichkeit der ersten Rückkehr zum Zustand  $i$  zur Zeit  $n$ . Per Definition gilt  $f_{ij}^{(0)} = f_{ii}^{(0)} = 0$ . Bildet man den Grenzwert

$$f_{ij}^* = \sum_{n=1}^{\infty} f_{ij}^{(n)} \quad (15.18)$$

so erhält man die Wahrscheinlichkeit, dass ausgehend von  $i$  der Zustand  $j$  überhaupt erreicht wird.  $f_{ii}^*$  ist die Wahrscheinlichkeit der Rückkehr zu einem Zustand  $i$ . Die Auswertung dieser Wahrscheinlichkeit führt zur Definition der *Rekurrenz*:

**Definition 15.2.5** Ein Zustand  $i$  ist rekurrent, falls  $f_{ii}^* = 1$ , falls also die Markov-Kette mit Wahrscheinlichkeit 1 den Ausgangszustand  $i$  wieder erreichen wird. Ist  $f_{ii}^* < 1$ , so ist der Zustand transient.

Der Erwartungswert  $\mu_i$  des Zeitpunkts der ersten Rückkehr zu einem rekurrenten Zustand  $i$  berechnet sich aus

$$\mu_i = \sum_{n=1}^{\infty} n f_{ii}^{(n)} \quad (15.19)$$

Ist diese *Rekurrenzzeit* für einen rekurrenten Zustand endlich, so wird dieser als *positiv rekurrent* bezeichnet, ist sie unendlich, dann ist der Zustand *nullrekurrent*.

Die bisher definierten Eigenschaften von Zuständen lassen sich durch folgenden Satz auf Untermengen des Zustandsraums ausweiten:

**Satz 15.2.1** *Kommunizieren zwei Zustände  $i$  und  $j$ , sind sie vom gleichen Typ.*

Aus der Eigenschaft dass eine Markov-Kette (oder eine Untermenge des Zustandsraums) irreduzibel ist, wenn alle Zustände paarweise kommunizieren, lässt sich folgern, dass alle Zustände, die zu einer irreduziblen Untermenge des Zustandsraums  $S$  gehören, vom gleichen Typ sind, also etwa positiv rekurrent sind und die gleiche Periode haben.

Die hier zusammengefassten Eigenschaften lassen sich zur Überprüfung der Ergodizität einer endlichen<sup>2)</sup> Markov-Kette heranziehen:

**Satz 15.2.2** *Eine endliche Markov-Kette ist genau dann ergodisch, wenn genau eine irreduzible Untermenge positiv rekurrenter Zustände existiert und all diese Zustände aperiodisch sind.*

Die Herleitung und der Beweis sind ausführlich in [110] dargestellt. Da sich schließlich noch zeigen lässt, dass alle Zustände einer endlichen irreduziblen Markov-Kette positiv rekurrent sind, reicht zum Nachweis der Ergodizität aus, zu überprüfen ob die Markov-Kette irreduzibel und aperiodisch ist. Im Folgenden soll die Plausibilität dieser Aussage anhand zweier Gegenbeispiele gezeigt werden.

Existiert mehr als eine irreduzible Untermenge in  $S$ , so wie in Abb. 15.2 (b) der absorbierende Zustand 1 und die Untermenge aus den Zuständen 3 und 4, dann ist die Kette auf keinen Fall ergodisch, da  $\lim_{n \rightarrow \infty} a_j[n]$  vom Startvektor  $a[0]$  abhängt. Im Beispiel heißt das, je höher die Anfangswahrscheinlichkeit für Zustand 1 ist, umso höher ist auch die Wahrscheinlichkeit, dass die Markov-Kette für  $n \rightarrow \infty$  diesen Zustand annimmt. Das gleiche gilt für die Zustände der zweiten irreduziblen Unermenge.

Die in (c) dargestellte Markov-Kette ist periodisch mit der Periode  $d = 3$ . Für jeden Zustand  $j$  gilt  $p_{jj}^{(n)} = 0$  für alle  $n$  außer  $n = 3m$ ,  $m \in \mathbb{N}$ . Damit existiert aber auch der Grenzwert  $\pi_j = \lim_{n \rightarrow \infty} p_{jj}^{(n)}$  nicht, da  $p_{jj}^{(n)}$  immer wieder Null wird. Die Markov-Kette ist daher ebenfalls nicht ergodisch.

---

<sup>2)</sup> Der Fall eines abzählbar unendlichen Zustandsraums wird hier nicht betrachtet.

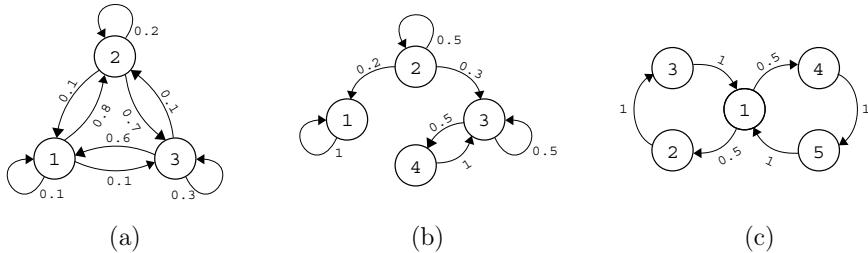


Abb. 15.2: Digraphen ergodischer (a) und nicht-ergodischer (b,c) Markov-Ketten

Die Markov-Kette zum Digraphen (a) hingegen ist ergodisch. Der Zustandsraum selbst ist die irreduzible Untermenge rekurrenter und aperiodischer Zustände. Alle Zustände kommunizieren und keine der Transitionswahrscheinlichkeiten ist Null. Aus der Transitionsmatrix und Gl. (15.16) folgt die stationäre Verteilung der Markov-Kette:

$$P = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.1 & 0.2 & 0.7 \\ 0.6 & 0.1 & 0.3 \end{bmatrix} \Rightarrow \pi = \begin{bmatrix} 49/170 \\ 57/170 \\ 64/170 \end{bmatrix} \quad (15.20)$$

Weiterhin kann gezeigt werden, dass die erwarteten Rekurrenzzeiten  $\mu_j$  einer ergodischen Markov-Kette sich aus den die reziproken Wahrscheinlichkeiten der stationären Verteilung  $\pi$  berechnen lassen:

$$\mu_j = \frac{1}{\pi_j}, \quad j \in S \quad (15.21)$$

So liegen im Beispiel die Rekurrenzzeiten bei  $[\mu_1 \ \mu_2 \ \mu_3] = [3.47 \ 2.98 \ 2.66]$ .

### 15.2.3 Konvergenzbegriffe

Speziell für die im folgenden Abschnitt vorgestellten Lerngesetze für stochastische Automaten sind Konvergenzbegriffe einzuführen (siehe z. B. [164]), mit deren Hilfe der Verlauf einer Zufallsgröße  $\{X_n\}$  für  $n \rightarrow \infty$  beschrieben werden kann. Der Übersicht halber schreiben wir hier  $X_n := X[n]$  für das  $n$ -te Glied einer Folge.

Die geläufigste Definition beschreibt die Konvergenz von Zahlenfolgen:

**Definition 15.2.6** Eine Folge reeller Zahlen  $\{x_n\}$  konvergiert gegen den Grenzwert  $x$ , falls es für jedes  $\varepsilon > 0$  eine Zahl  $N = N(\varepsilon)$  gibt, so dass  $|x_n - x| < \varepsilon$  für alle  $n \geq N$ . Man schreibt

$$\lim_{n \rightarrow \infty} x_n = x \quad (15.22)$$

Eine Zahlenfolge  $\{x_n\}$  ist also konvergent, wenn sie einen Grenzwert  $x$  besitzt. Die Konvergenz stochastische Prozesse, die durch eine Folge von Ereignissen  $\{\omega_n\}$  beschrieben werden, bzw. durch die Folge der Zufallsvariablen  $\{X_n\}$ , welche ja Abbildungen der Ereignisse auf die reellen Zahlen sind, lässt sich besser durch die sog. *fast sichere Konvergenz* oder *Konvergenz mit Wahrscheinlichkeit 1* beschreiben.

**Definition 15.2.7** Eine Folge von Zufallsvariablen  $\{X_n\}$  konvergiert mit Wahrscheinlichkeit 1 gegen eine Zufallsvariable  $X$ , falls für jedes  $\delta > 0$  und  $\varepsilon > 0$  eine Zahl  $N = N(\delta, \varepsilon)$  existiert, so dass

$$\Pr(|X_n - X| < \varepsilon) > 1 - \delta \quad (15.23)$$

für alle  $n \geq N$ . Da  $\delta$  und  $\varepsilon$  beliebig klein sein dürfen, muss alternativ gelten:

$$\Pr\left(\lim_{n \rightarrow \infty} X_n = X\right) = 1 \quad (15.24)$$

Für einen in dieser Weise konvergenten stochastischen Prozess strebt also die Wahrscheinlichkeit, dass das Zufallsereignis  $\omega_n$  auf die Zufallsvariable  $X_n = X$  abgebildet wird für  $n \rightarrow \infty$  gegen 1. Die Handlungswahrscheinlichkeiten des sog. *Linear Reward-Inaction*-Lerngesetzes konvergieren etwa mit Wahrscheinlichkeit 1.

Andere (ergodische) Lerngesetze, wie die *Linear Reward-Penalty*-Regel konvergieren in schwächerer Weise. Statt der Konvergenz der Folge von Zufallsvariablen  $\{X_n\}$  gegen eine Zufallsvariable  $X$  wird die Konvergenz der Verteilungsfunktionen  $\{F_n(x)\}$  der Zufallsvariable gegen eine Verteilungsfunktion  $\{F(x)\}$  betrachtet.

**Definition 15.2.8** Die Folge von Zufallsvariablen  $\{X_n\}$  konvergiert in Verteilung gegen eine Zufallsvariable  $X$ , falls die Verteilungsfunktionen  $F_n(x)$  der Zufallsgrößen  $X_n$  gegen eine Verteilungsfunktion  $F(x)$  einer Zufallsgröße  $X$  konvergieren, falls also

$$\lim_{n \rightarrow \infty} F_n(x) = F(x) \quad (15.25)$$

für alle  $x$ , an denen  $F(x)$  stetig ist.

Es existieren noch einen Reihe weiterer Konvergenzdefinitionen für stochastische Prozesse, die aber nicht für die in diesem Beitrag betrachteten Lernalgorithmen von Bedeutung sind.

### 15.3 Automaten

Automaten werden eingeführt als mathematische Konstrukte, die basierend auf einfachen Regeln, imstande sind, in einer unbekannten Umgebung *autonom Entscheidungen zu treffen* und dadurch mit der Umgebung zu interagieren. Ihre

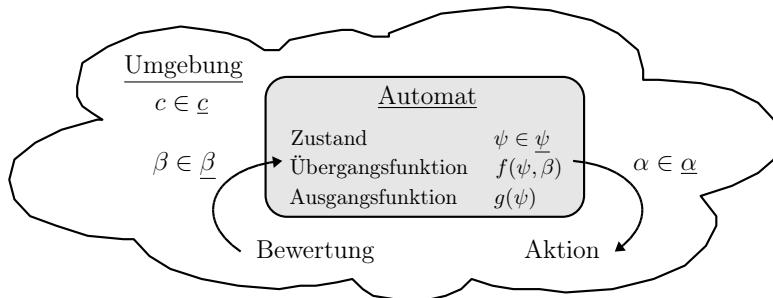


Abb. 15.3: Automat und Umgebung

*Handlungsgrundlage* ist durch ihre Struktur und ggf. die Anpassung der Verhaltensweise durch *Lernalgorithmen* bestimmt. Eine typische Aufgabe, die sich von Automaten bewältigen lässt, ist etwa die Identifikation und Prognose dominanter Sequenzen aus einer durch eine Markov-Kette modellierten Folge zeitveränderlicher Parameter. Durch den Einsatz von Automaten wird auf „heuristisches“ Vorgehen<sup>3)</sup> verzichtet.

Dieser Abschnitt enthält einige grundlegenden Definitionen und Tatsachen aus der Automatentheorie, sowie einen kurzen Überblick über typische Strukturen von Automaten sowie ihrer Lerngesetze. Dabei stützt sich die Darstellung auf das sehr anschauliche und weiterführende Buch [164], das auch die hier nicht aufgeführten Beweise enthält.

### 15.3.1 Automat und Umgebung

Der Automat wirkt durch sein Handeln auf seine *Umgebung*, d. h. er wählt jeweils eine der ihm zur Verfügung stehenden *Aktionen* entsprechend seiner Handlungsgrundlage. Für jede Aktion erhält er eine Rückmeldung der Umgebung. Wir betrachten *stochastische Umgebungen*, die jede Aktion des Automaten mit einer gewissen Wahrscheinlichkeit *bestrafen* oder *belohnen*. Die Rückmeldung der Umgebung induziert entweder eine Zustandstransition des Automaten oder veranlasst die Adaption seiner Entscheidungsgrundlage. Wie der so angestoßene Lernprozess verläuft, hängt entscheidend von der Wahl des *Lerngesetzes* ab. Im Folgenden werden die Elemente der Interaktion des Automaten mit seiner Umgebung vorgestellt:

*Eingang* Das Bewertungssignal der Umgebung dient als Eingang des Automaten. Die Bewertung  $\beta$  nimmt Werte aus einer Menge  $\underline{\beta}$  an. Im einfachsten und meist praktikablen Fall ist  $\beta$  binär, d. h.

<sup>3)</sup> Ein solches Vorgehen wäre z. B. das Abzählen der Häufigkeit verschiedener Parametertransitions über einen Zeithorizont und die anschließende Auswertung der beobachteten Daten.

$$\underline{\beta} = \{0, 1\} \quad (15.26)$$

Mit  $\beta = 0$  wird die vorangegangene Aktion des Automaten belohnt, mit  $\beta = 1$  bestraft.

*Ausgang* Der Automat wirkt durch das Ausführen einer Aktion  $\alpha$  aus einer Menge von  $M$  ihm zur Verfügung stehenden Aktionen

$$\underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_M\} \quad (15.27)$$

auf die Umgebung. Durch die Auswahl der Struktur und den Entwurf des Lerngesetzes für den Automaten soll ein möglichst geringer Erwartungswert der Bestrafung der gewählten Aktionen erreicht werden.

*Zustand* Jeder Automat wird durch eine Menge von  $N$  möglichen Zuständen

$$\underline{\psi} = \{\psi_1, \psi_2, \dots, \psi_N\} \quad (15.28)$$

beschrieben. Stellt der Ausgang  $\alpha$  das äußerlich sichtbare Verhalten des Automaten dar, so ist die Menge der Zustände  $\underline{\psi}$  mit ihren gegenseitigen Verknüpfungen, zusammen mit der Übergangsfunktion  $f : \underline{\psi} \times \underline{\beta} \rightarrow \underline{\psi}$  und der Ausgangsfunktion  $g : \underline{\psi} \rightarrow \underline{\alpha}$  die innere Entscheidungsgrundlage des Automaten. Durch

$$\psi[n+1] = f(\psi[n], \beta[n]) \quad (15.29)$$

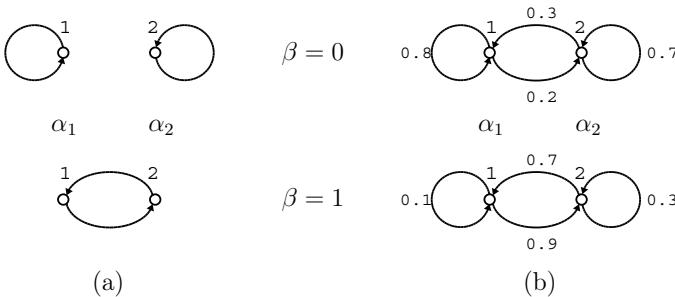
werden Zustandstransitionen auf Grund der Bewertung der vergangenen Aktion induziert, mit  $\alpha[n] = g(\psi[n])$  findet eine Abbildung des Zustandsraums auf den Ausgang statt. Häufig ist die Ausgangsfunktion  $g(\cdot)$  die Identität, wie etwa bei den später vorgestellten stochastischen Automaten veränderlicher Struktur. Zu beachten ist, dass sowohl  $f$  als auch  $g$  deterministische oder stochastische Funktionen sein dürfen.

*Transitionsmatrizen* Die Übergangsfunktion  $f$  lässt sich durch von der Bewertung abhängige Transitionsmatrizen  $\mathbf{F}^\beta$ ,  $\beta \in \underline{\beta}$  darstellen. Sie enthalten die Wahrscheinlichkeiten, mit denen der Automat bei gegebener Bewertung  $\beta$  vom Zustand  $i$  (Zeilen) in den Zustand  $j$  (Spalten) übergeht. Für einen Automat mit zwei Zuständen und binärer Bewertung stellen die Matrizen

$$\mathbf{F}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{F}^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (15.30)$$

eine deterministische Übergangsfunktion dar, während die Matrizen

$$\mathbf{F}^0 = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}, \quad \mathbf{F}^1 = \begin{bmatrix} 0.1 & 0.9 \\ 0.7 & 0.3 \end{bmatrix} \quad (15.31)$$



**Abb. 15.4:** Digraphen des deterministischen (a) und stochastischen (b)  $L_{2,2}$ -Automaten

eine stochastische Übergangsfunktion beschreiben. Entsprechend dieser Unterscheidung werden Automaten in deterministische und stochastische klassifiziert. In Abb. 15.4 sind die Transitionsdiagramme (Digraphen) der beiden Automaten<sup>4)</sup> dargestellt.

*Umgebung* Die Umgebung wirkt durch ihr Bewertungssignal  $\beta$  auf den Automaten zurück. Ist  $\beta$  binär, so ist durch

$$c_i = \Pr(\beta[n] = 1 \mid \alpha[n] = \alpha_i), \quad i = 1, \dots, M \quad (15.32)$$

die Bestrafungswahrscheinlichkeit einer Aktion  $\alpha_i$  gegeben. Die Umgebung wird durch die Menge der Bestrafungswahrscheinlichkeiten

$$\underline{c} = \{c_1, c_2, \dots, c_M\} \quad (15.33)$$

beschrieben. Alternativ lassen sich durch

$$d_i = \Pr(\beta[n] = 0 \mid \alpha[n] = \alpha_i) = 1 - c_i \quad (15.34)$$

die einzelnen Belohnungswahrscheinlichkeiten angeben. Bleiben die so definierten Wahrscheinlichkeiten über die Zeit konstant, so spricht man von einer *stationären* Umgebung. Im Folgenden wird nur dieser Fall betrachtet.

Sind die Bestrafungswahrscheinlichkeiten der Umgebung bekannt, so lassen sich die (bedingten) Transitionsmatrizen  $\mathbf{F}^0$  und  $\mathbf{F}^1$  zu *absoluten* Transitionsmatrizen kombinieren, die die Wahrscheinlichkeiten angeben, dass in der vorliegenden Umgebung *überhaupt* Transitionen zwischen Zuständen des Automaten erfolgen. Die Elemente der stochastischen Matrix

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{f}_{11} & \dots & \tilde{f}_{1N} \\ \vdots & \ddots & \vdots \\ \tilde{f}_{N1} & \dots & \tilde{f}_{NN} \end{bmatrix} \quad (15.35)$$

<sup>4)</sup> Es handelt sich um sog.  $L_{2,2}$ -Automaten, da jedem der zwei Zustände genau eine Aktion zugeordnet ist

berechnen sich aus

$$\tilde{f}_{ij} = f_{ij}^0 \cdot \Pr(\beta[n] = 0 | \psi[n] = \psi_i) + f_{ij}^1 \cdot \Pr(\beta[n] = 1 | \psi[n] = \psi_i) \quad (15.36)$$

Im Fall der identischen Abbildung zwischen Zuständen und Aktionen gilt  $\psi_i = \alpha_i$  und man erhält

$$\tilde{f}_{ij} = f_{ij}^0 d_i + f_{ij}^1 c_i \quad (15.37)$$

Für den Automaten mit den Transitionsmatrizen (15.31) in einer durch  $c_1 = 0.3$  und  $c_2 = 0.6$  beschriebenen Umgebung ergibt sich

$$\tilde{\mathbf{F}} = \begin{bmatrix} 0.59 & 0.41 \\ 0.54 & 0.46 \end{bmatrix} \quad (15.38)$$

*Aktionswahrscheinlichkeiten* Statt durch mit Wahrscheinlichkeiten behaftete Zustandsübergänge und die Abbildung der Zustände auf Aktionen lassen sich stochastische Automaten direkt mit Hilfe von *Aktionswahrscheinlichkeiten* beschreiben. Durch

$$p_i[n] = \Pr(\alpha[n] = \alpha_i), \quad i = 1, \dots, N \quad (15.39)$$

(wir nehmen wieder die Identität als Ausgangsfunktion, es gilt also  $M = N$ ) ist die Wahrscheinlichkeit gegeben, dass der Automat zum Zeitpunkt  $n$  die Aktion  $\alpha_i$  wählt. Die Entwicklung des Vektors der Aktionswahrscheinlichkeiten

$$\underline{p}[n] = \begin{bmatrix} p_1[n] \\ \dots \\ p_N[n] \end{bmatrix} \quad (15.40)$$

hängt entscheidend vom *Lerngesetz* ab, mit dem die Aktionswahrscheinlichkeiten angepasst werden. Wie später deutlich wird, ist der Verlauf von  $\underline{p}[n]$  unter einem gegebenen Lernalgorithmus in einer stationären Umgebung ein Markov-Prozess, da die Wahrscheinlichkeit, dass die vektorwertige Zufallsgröße  $\underline{p}[n]$  einen Wert  $\underline{p}$  annimmt, nur von  $\underline{p}[n-1]$  und den Bestrafungswahrscheinlichkeiten der Umgebung abhängt:

$$\Pr(\underline{p}[n] = \underline{p}) = \Pr(\underline{p}[n] = \underline{p} | \underline{p}[n-1]) \quad (15.41)$$

*Arten von Automaten* Automaten werden in *deterministische* und *stochastische* eingeteilt, je nach Art der Übergangs- und Ausgangsfunktion. Daneben wird zwischen Automaten *fester Struktur* und Automaten *veränderlicher Struktur* unterschieden. Letztere werden auch als *lernende Automaten* bezeichnet, da ihre Entscheidungsgrundlage durch entsprechende Lernalgorithmen angepasst wird.

### 15.3.2 Nützlichkeit und Optimalität

Ein Gütekriterium zur Beurteilung der Leistung eines Automaten in einer stationären Umgebung ist der Erwartungswert der Bestrafung  $\beta[n]$  für einen gegebenen Wahrscheinlichkeitsvektor  $\underline{p}[n]$ . Diese *durchschnittliche Bestrafung* wird mit  $M[n]$  abgekürzt und berechnet sich aus

$$M[n] := E(\beta[n] \mid p[n]) = \Pr(\beta[n] = 1 \mid p[n]) = \sum_{i=1}^N c_i p_i[n] \quad (15.42)$$

als Summe der Produkte aus Aktions- und Bestrafungswahrscheinlichkeit. Als Referenz für alle anderen Automaten wird die *rein zufällige* Wahl der Aktionen mit gleicher Wahrscheinlichkeit

$$p_i[n] = \frac{1}{N}, \quad i = 1, \dots, N, \quad \forall n \quad (15.43)$$

betrachtet. Solch ein reiner Zufallsautomat erfährt die durchschnittliche Bestrafung

$$M_0 := \sum_{i=1}^N c_i p_i[n] = \frac{1}{N} \sum_{i=1}^N c_i \quad (15.44)$$

Jeder für eine bestimmte Aufgabe entworfene Automat muss sich (wenigstens nach einer anfänglichen Lernphase) besser verhalten, also eine geringe durchschnittliche Bestrafung erzielen.

**Definition 15.3.1** Ein Automat ist nützlich, wenn  $\lim_{n \rightarrow \infty} E(M[n]) < M_0$ .

Es sei angemerkt, dass der Erwartungswert der *durchschnittlichen Bestrafung*  $M[n]$  gleich dem Erwartungswert der Bestrafung  $\beta[n]$  ist. Bildet man nämlich zu beiden Seiten des ersten Gleichheitszeichens in Gl. (15.42) den Erwartungswert, so erhält man

$$E(M[n]) = E(\beta[n]) \quad (15.45)$$

Nützlichkeit eines Automaten ist zwar eine Verbesserung gegenüber dem reinen Zufall, erstrebenswert ist jedoch, dass die erwartete Bestrafung minimiert wird. Das passiert genau dann, wenn der Automat stets die Aktion  $\alpha_l$  wählt, die mit der geringsten Wahrscheinlichkeit  $c_l$  bestraft wird:

**Definition 15.3.2** Strebt die erwartete durchschnittliche Bestrafung  $E(M[n])$  gegen die geringste Bestrafungswahrscheinlichkeit der Umgebung, gilt also

$$\lim_{n \rightarrow \infty} E(M[n]) = c_l \quad \text{mit} \quad c_l = \min\{c_1, \dots, c_N\} \quad (15.46)$$

so nennt man den Automaten optimal.

Ein optimaler Automat wählt also (nach abgeschlossenem Lernvorgang) stets die Aktion, die die geringste Bestrafung erwarten lässt. Betrachten wir einen optimalen Automaten, der durch einen Vektor von Aktionswahrscheinlichkeiten beschrieben wird, dann muss dieser gegen den Einheitsvektor der am schwächsten bestraften Aktion konvergieren:

$$\lim_{n \rightarrow \infty} \underline{p}[n] = \underline{e}_l \quad (15.47)$$

Optimalität des Automaten ist in der Regel nicht zu erreichen. Bei Anwendung des  $L_{R-I}$ -Algorithmus etwa konvergiert der Vektor der Aktionswahrscheinlichkeiten gegen einen Einheitsvektor. Ob die dann mit Wahrscheinlichkeit 1 gewählte Aktion tatsächlich die am geringsten bestraft ist, hängt entscheidend vom Startvektor  $\underline{p}[0]$  ab. Oft muss man sich mit *suboptimalem* oder  $\varepsilon$ -optimalem Verhalten zufrieden geben.

**Definition 15.3.3** *Kommt die erwartete durchschnittliche Bestrafung  $E(M[n])$  der geringsten Bestrafungswahrscheinlichkeit beliebig nahe, gilt also für beliebige  $\varepsilon > 0$*

$$\lim_{n \rightarrow \infty} E(M[n]) < c_l + \varepsilon \quad (15.48)$$

dann ist der Automat  $\varepsilon$ -optimal.

Durch geeignete Wahl der Entwurfsparameter (Gedächtnistiefe beim Tsetlin-Automat, Lernschrittweite des  $L_{R-I}$ -Algorithmus) lässt sich  $\varepsilon$  jedoch beliebig klein machen, so dass optimales Verhalten beliebig angenähert werden kann.

Eine Eigenschaft, die besonders den Erfolg des Lernfortschritts illustriert, ist *absolute Nützlichkeit*. Sie fordert schrittweise Verbesserung des Verhaltens des Automaten:

**Definition 15.3.4** *Gilt für die durchschnittliche Bestrafung  $M[n]$  eines Lernautomaten*

$$E(M[n+1] | p[n]) < M[n] \quad (15.49)$$

bzw. (*Erwartungswert auf beiden Seiten*)

$$E(M[n+1]) < E(M[n]) \quad (15.50)$$

so nennt man den Automaten absolut nützlich.

Absolute Nützlichkeit verzichtet auf den Vergleich mit dem reinen Zufallsautomaten. Nimmt man jedoch an, dass durch die Wahl des Startvektors  $\underline{p}[0]$  für irgendein  $n$ , z. B.  $n = 0$  Nützlichkeit gegeben ist, also  $E(M[n]) < M_0$ , dann ist absolute Nützlichkeit eine deutlich stärkere Eigenschaft, da  $E(M[n])$  streng monoton abnimmt. Es kann gezeigt werden, dass in einer stationären Umgebung absolute Nützlichkeit tatsächlich asymptotische Optimalität impliziert.

### 15.3.3 Stochastische Automaten veränderlicher Struktur

Die erste betrachtete Klasse von Automaten sind *stochastische Automaten veränderlicher Struktur*, die in einer stationären Umgebung arbeiten. Jede Aktion  $\alpha_i$  wird mit der Wahrscheinlichkeit  $p_i[n]$  ausgeführt und gemäß den Bestrafungswahrscheinlichkeiten  $\{c_1, \dots, c_N\}$  der Umgebung durch  $\beta[n] \in \{0, 1\}$  bewertet. Auf Grund dieser Bewertung findet eine Anpassung des Vektors der Aktionswahrscheinlichkeiten

$$\underline{p}[n] = \begin{bmatrix} p_1[n] \\ \vdots \\ p_N[n] \end{bmatrix} \quad (15.51)$$

nach einem vorher definierten Lerngesetz statt. Das *allgemeine Lerngesetz*

$$\alpha[n] = \alpha_i \Rightarrow \begin{cases} \beta[n] = 0 : & \begin{cases} p_j[n+1] = p_j[n] - g_j(\underline{p}[n]), & j \neq i \\ p_i[n+1] = p_i[n] + \sum_{j \neq i} g_j(\underline{p}[n]) \end{cases} \\ \beta[n] = 1 : & \begin{cases} p_j[n+1] = p_j[n] + h_j(\underline{p}[n]), & j \neq i \\ p_i[n+1] = p_i[n] - \sum_{j \neq i} h_j(\underline{p}[n]) \end{cases} \end{cases} \quad (15.52)$$

wertet die Wahrscheinlichkeiten aller Aktionen  $\alpha_j \neq \alpha_i$  um die Korrekturterme  $g_j$  ab bzw. um  $h_j$  auf, je nachdem ob die gerade durchgeführte Aktion belohnt oder bestraft wird. Deren Wahrscheinlichkeit erhöht bzw. verringert sich um die Summe der Korrekturterme, so dass die Summe aller Wahrscheinlichkeiten 1 bleibt. Damit alle Wahrscheinlichkeiten im offenen Intervall  $(0, 1)$  bleiben, müssen die Korrekturterme die Ungleichungen

$$0 < g_j(\underline{p}[n]) < p_j[n] \quad \text{und} \quad 0 < \sum_{j \neq i} (p_j[n] + h_j(\underline{p}[n])) < 1 \quad (15.53)$$

erfüllen.

Beim *allgemeinen linearen Lerngesetz* erfolgt die Anpassung von  $\underline{p}[n]$  über die Parameter  $a$  und  $b$  zur Belohnung und Bestrafung. Um das  $a$ -fache werden im Falle der Belohnung von  $\alpha[n] = \alpha_i$  alle übrigen Aktionswahrscheinlichkeiten reduziert. Analog wird  $p_i[n]$  um das  $b$ -fache herabgesetzt, wenn die Aktion  $\alpha_i$  bestraft wird. Aus der Forderung  $\sum_{i=1}^N p_i = 1$  folgen die Adoptionsvorschriften

$$\alpha[n] = \alpha_i \Rightarrow \begin{cases} \beta[n] = 0 : & \begin{cases} p_j[n+1] = p_j[n] - ap_j[n], & j \neq i \\ p_i[n+1] = p_i[n] + a(1 - p_i[n]) \end{cases} \\ \beta[n] = 1 : & \begin{cases} p_j[n+1] = p_j[n] + b(\frac{1}{N-1} - p_j[n]), & j \neq i \\ p_i[n+1] = p_i[n] - bp_i[n] \end{cases} \end{cases} \quad (15.54)$$

Für das lineare Lerngesetz lauten also die Funktionen

$$g_j(\underline{p}[n]) = ap_j[n] \quad \text{und} \quad h_j(\underline{p}[n]) = b(\frac{1}{N-1} - p_j[n]) \quad (15.55)$$

In Vektorschreibweise lauten die Aktualisierungsvorschriften:

$$\alpha[n] = \alpha_i \Rightarrow \begin{cases} \beta[n] = 0 : & \underline{p}[n+1] = (1-a)\underline{p}[n] + ae_i \\ \beta[n] = 1 : & \underline{p}[n+1] = (1-b)\underline{p}[n] + b\frac{1}{N-1}\bar{e}_i \end{cases} \quad (15.56)$$

$e_i$  ist der  $i$ -te  $N$ -dimensionale Einheitsvektor. Der Vektor, dessen Elemente gegenüber  $e_i$  logisch invertiert sind (also 1 und 0 vertauscht), wird mit  $\bar{e}_i$  bezeichnet.

Ziel aller Lerngesetze ist die Verbesserung des Automaten hin zu nützlichem, absolut nützlichem oder gar optimalem Verhalten. Der folgende Satz [164] liefert hinreichende und notwendige Bedingungen für die *absolute Nützlichkeit* und damit die *asymptotische Optimalität* eines stochastischen Automaten unter einem bestimmten Lerngesetz.

**Satz 15.3.1** Ein stochastischer Automat veränderlicher Struktur unter dem allgemeinen Lerngesetz (15.52) ist genau dann absolut nützlich, wenn für alle  $i$  und  $j$  folgende Symmetriebedingungen gelten:

$$\frac{g_i(p)}{p_i} = \frac{g_j(p)}{p_j} \quad \text{und} \quad \frac{h_i(p)}{p_i} = \frac{h_j(p)}{p_j} \quad (15.57)$$

Bezogen auf lineare Lerngesetze wird deutlich, dass absolute Nützlichkeit nur durch Festlegung von  $b \equiv 0$  erreicht werden kann, wie etwa beim *Linear Reward-Inaction*-Algorithmus. Die Konvergenz unter diesem Lerngesetz hängt jedoch vom Startvektor der Aktionswahrscheinlichkeiten  $\underline{p}[0]$  ab. Im Gegensatz dazu ist der zunächst vorgestellte *Linear Reward-Penalty*-Algorithmus, der zwar nicht absolut nützlich ist, *ergodisch*. Sein Konvergenzverhalten ist also von  $\underline{p}[0]$  unabhängig.

### 15.3.3.1 Ein ergodisches Lerngesetz: *Linear Reward-Penalty*

Für das sog. *Linear Reward-Penalty*- oder kurz  $L_{R-P}$ -Lerngesetz setzt man gleiche Faktoren für Belohnung und Bestrafung an:  $a = b > 0$ . Bei einer positiv bewerteten Aktion  $\alpha_i$  werden die Wahrscheinlichkeiten aller anderen Aktionen im selben Maße erniedrigt, wie  $p_i$  bei negativ bewerteter Aktion gesenkt wird. Umgekehrt entspricht das Anheben von  $p_i$  bei Belohnung genau der Erhöhung der Summe  $\sum_{j \neq i} p_j$  bei Bestrafung der Aktion  $\alpha_i$ .

Die beiden Gleichungen von (15.56) lassen sich durch Einfügen der Terme  $\beta[n]$  und  $(1 - \beta[n])$  zu

$$\underline{p}[n+1] = \underline{p}[n] + a \cdot (\underline{e}_i - \underline{p}[n]) \cdot (1 - \beta[n]) + a \cdot \left( \frac{1}{N-1} \bar{\underline{e}}_i - \underline{p}[n] \right) \cdot \beta[n] \quad (15.58)$$

zusammenfassen.  $\beta[n]$  ist eine Zufallsgröße, deren zeitlicher Verlauf durch die Aktions- und Bestrafungswahrscheinlichkeiten bestimmt wird. Der Wahrscheinlichkeitsvektor  $\underline{p}[n+1]$  hängt nur vom Vorgänger  $\underline{p}[n]$  ab. Damit ist die Folge  $\{\underline{p}[n]\}$ ,  $n = 0, 1, 2, \dots$  ein zeitdiskreter, wertkontinuierlicher Markov-Prozess. Dass dieser Prozess ergodisch ist und keinen absorbierenden Zustand besitzt (also kein Vektor  $\underline{p}^* = \lim_{n \rightarrow \infty} \underline{p}[n]$  existiert), wird im Folgenden skizziert.

*Existenz eines absorbierenden Zustands* Lässt sich der Markov-Prozess (15.58) auf einen absorbierenden Zustand  $\underline{p}^*$  reduzieren, dann folgt aus dem Zustand  $\underline{p}[n] = \underline{p}^*$  mit Wahrscheinlichkeit 1 der Zustand  $\underline{p}[n+1] = \underline{p}^*$ . In diesem Fall muss die Gleichung

$$\underline{p}^* = \underline{p}^* + a \cdot (\underline{e}_i - \underline{p}^*) \cdot (1 - \beta[n]) + a \cdot \left( \frac{1}{N-1} \bar{\underline{e}}_i - \underline{p}^* \right) \cdot \beta[n] \quad (15.59)$$

sowohl für  $\beta[n] = 0$  als auch für  $\beta[n] = 1$  gelten, da alle Aktionen  $\alpha_i$  mit von 0 und 1 verschiedener Wahrscheinlichkeit bestraft werden. Da sich die Lösungen für beide Fälle

$$\underline{p}^* = \underline{e}_i \quad \text{und} \quad \underline{p}^* = \frac{1}{N-1} \bar{\underline{e}}_i \quad (15.60)$$

widersprechen, existiert kein absorbierender Zustand  $\underline{p}^*$  des durch (15.58) beschriebenen Markov-Prozesses.

*Ergodizität* Existiert für die Zufallsgröße  $\underline{p}[n]$  eine vom Anfangswert unabhängige (vektorwertige) Verteilungsfunktion  $\underline{F}^*(\underline{p}) = \lim_{n \rightarrow \infty} \underline{F}(\underline{p}[n])$  mit dem Erwartungswert  $\underline{E}^* := \lim_{n \rightarrow \infty} \underline{E}(\underline{p}[n])$ , dann ist der Markov-Prozess ergodisch. In 15.2.2 wurden als Kriterien für die Ergodizität einer endlichen Markov-Kette Irreduzibilität und Aperiodizität genannt. Da der Zustandsraum des durch (15.58) beschriebenen Markov-Prozesses jedoch kontinuierlich ist, muss die Ergodizität auf anderem Wege gezeigt werden, nämlich über die Kontraktionsabbildung des Abstands zweier Anfangszustände  $\underline{p}[0]$  und  $\underline{q}[0]$ . Berechnet man nach Gleichung (15.58) deren Folgezustände  $\underline{p}[1]$  und  $\underline{q}[1]$ , dann gilt sowohl für  $\beta[0] = 0$  als auch  $\beta[0] = 1$

$$\underline{p}[1] - \underline{q}[1] = (1-a)(\underline{p}[0] - \underline{q}[0]) \quad (15.61)$$

Geht man zur Abstandsnorm der beiden Vektoren über und betrachtet den Prozess  $n$  Schritte in der Zukunft, so ergibt sich

$$\|\underline{p}[n] - \underline{q}[n]\| = (1-a)^n \|\underline{p}[0] - \underline{q}[0]\| \quad (15.62)$$

Wegen  $0 < a < 1$  nimmt der Betrag mit jedem Schritt ab und man erkennt, dass  $\underline{p}[n] \rightarrow \underline{q}[n]$  für  $n \rightarrow \infty$ , und zwar unabhängig von den Anfangswerten  $\underline{p}[0]$  und  $\underline{q}[0]$ .

*Erwartungswert der Aktionswahrscheinlichkeiten* Der Vektor der Aktionswahrscheinlichkeiten  $\underline{p}[n]$  (als vektorwertige Zufallsvariable) konvergiert wegen des fehlenden absorbierenden Zustandes nicht mit Wahrscheinlichkeit 1 sondern *in Verteilung*. Zwar ist dies eine schwache Konvergenzeigenschaft, doch existieren wegen der Ergodizität eine vom Anfangswert *unabhängige* Verteilungsfunktion  $\underline{F}^*(\underline{p})$  und der zugehörige Erwartungswert  $\underline{E}^*$  für  $n \rightarrow \infty$ . Dieser lässt sich aus der stationären Lösung eines Systems von Differenzengleichungen für die Erwartungswerte der einzelnen Aktionswahrscheinlichkeiten berechnen.

Der bedingte Erwartungswert  $\underline{E}(p_i[n+1] | p_i[n])$  berechnet sich gemäß dem allgemeinen linearen Lerngesetz (15.54) mit der abkürzenden Schreibweise  $\Pr(\alpha_i) := \Pr(\alpha[n] = \alpha_i)$  und  $\Pr(0 | \alpha_i) := \Pr(\beta[n] = 0 | \alpha[n] = \alpha_i)$  als

$$\begin{aligned}
E(p_i[n+1] | p_i[n]) = & \quad p_i[n] + \\
& + \Pr(\alpha_i) \cdot [\Pr(0 | \alpha_i) \cdot (a - ap_i[n]) + \Pr(1 | \alpha_i) \cdot (-ap_i[n])] + \\
& + \sum_{j \neq i} \Pr(\alpha_j) [\Pr(0 | \alpha_j) \cdot (-ap_i[n]) + \Pr(1 | \alpha_j) \cdot (\frac{a}{N-1} - ap_i[n])] \\
\end{aligned} \tag{15.63}$$

Setzt man mit  $c_i = \Pr(1 | \alpha_i)$  die Bestrafungswahrscheinlichkeiten der Umgebung ein und formt das Ergebnis um, so erhält man

$$E(p_i[n+1] | p_i[n]) = p_i[n] \cdot (1 - ac_i) + \frac{a}{N-1} \sum_{j \neq i} c_j p_j[n] \tag{15.64}$$

Beidseitiges Bilden der Erwartungswerte und Umstellen ergibt den Ausdruck

$$\frac{E(p_i[n+1]) - E(p_i[n])}{a} = -c_i E(p_i[n]) + \sum_{j \neq i} \frac{c_j}{N-1} E(p_j[n]) \tag{15.65}$$

für alle  $i$ . Die Forderungen  $\lim_{n \rightarrow \infty} (E(p_i[n+1]) - E(p_i[n])) = 0$  für eine statio-näre Verteilung der im Vektor  $\underline{p}[n]$  zusammengefassten Zufallsgrößen liefern das lineare Gleichungssystem

$$\underbrace{\begin{bmatrix} -c_1 & \frac{c_2}{N-1} & \dots & \frac{c_N}{N-1} \\ \frac{c_1}{N-1} & -c_2 & \dots & \frac{c_N}{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{c_1}{N-1} & \frac{c_2}{N-1} & \dots & -c_N \end{bmatrix}}_{\mathbf{A}} \cdot \begin{bmatrix} \lim_{n \rightarrow \infty} E(p_1[n]) \\ \lim_{n \rightarrow \infty} E(p_2[n]) \\ \vdots \\ \lim_{n \rightarrow \infty} E(p_N[n]) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{15.66}$$

Da die Matrix  $\mathbf{A}$  Rang  $N-1$  hat, wird das Gleichungssystem erst durch Einfügen der Nebenbedingung

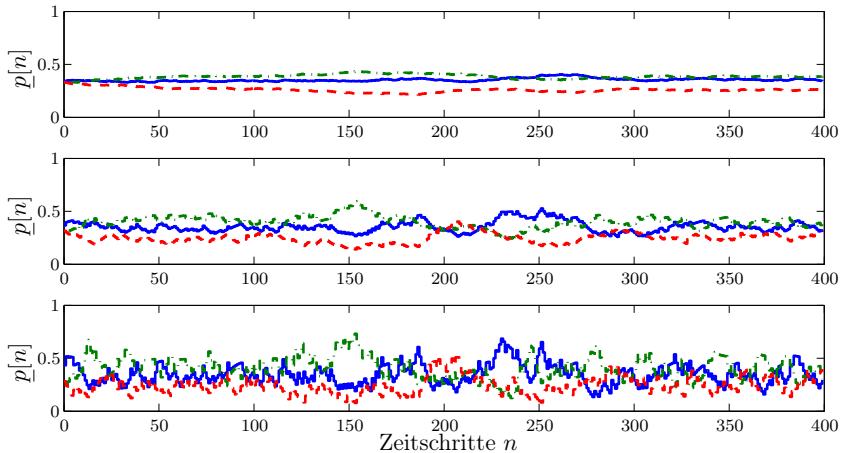
$$\sum_{i=1}^N p_i[n] = 1 \quad \text{für alle } n \tag{15.67}$$

eindeutig bestimmt. Die Lösung berechnet sich schließlich aus

$$\begin{bmatrix} -c_1 & \dots & \frac{c_{N-1}}{N-1} & \frac{c_N}{N-1} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{c_1}{N-1} & \dots & -c_{N-1} & \frac{c_N}{N-1} \\ 1 & \dots & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} E_1^* \\ \vdots \\ E_{N-1}^* \\ E_N^* \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{15.68}$$

und man erhält den Grenzwert der Erwartungswerte der Aktionswahrscheinlichkeiten des Automaten unter dem  $L_{R-P}$ -Lerngesetz:

$$\underline{E}^* = \lim_{n \rightarrow \infty} E(p[n]) = \begin{bmatrix} 1/c_1 \\ \sum_{j=1}^N 1/c_j \\ 1/c_2 \\ \sum_{j=1}^N 1/c_j \\ \vdots \\ 1/c_N \\ \sum_{j=1}^N 1/c_j \end{bmatrix}^T \tag{15.69}$$



**Abb. 15.5:** Lernprozess unter dem  $L_{R-P}$ -Algorithmus mit Lernschrittweiten  $a = b = 0.01/0.05/0.15$  (von oben nach unten).  $p_1[n]$  (-),  $p_2[n]$  (-·),  $p_3[n]$  (---).

Es sei nochmals angemerkt, dass es sich um Konvergenz der vektorwertigen Zufallsgröße  $\underline{p}[n]$  in Verteilung handelt.  $\underline{p}[n]$  strebt also nicht selbst gegen einen Grenzwert, vielmehr konvergiert die Verteilung der Zufallsgröße und somit ihr Erwartungswert  $E(\underline{p}[n])$ . Weiterhin kann man zeigen, dass die Varianz der Zufallsgröße mit der Lernschrittweite  $a$  zunimmt.

**Beispiel 15.3.1** Abbildung 15.5 zeigt den Verlauf der Aktionswahrscheinlichkeiten  $\underline{p}[n]$  eines stochastischen Automaten mit drei Aktionen unter dem  $L_{R-P}$ -Lerngesetz in einer durch  $\underline{c} = \{0.6, 0.5, 0.9\}$  beschriebenen Umgebung. Deutlich wird die schwache Konvergenz in Verteilung des ergodischen Algorithmus. Damit ist auch nur eine grobe Abschätzung der Bestrafungswahrscheinlichkeiten durch Auflösen von Gl. (15.69) möglich. Mit einer Erhöhung der Lernschrittweite  $a$  vergrößert sich zudem die Varianz der einzelnen Aktionswahrscheinlichkeiten. Da kein absorzierender Zustand von  $\underline{p}[n]$  in Form eines Einheitsvektors erreicht wird, ist auch klar, dass der  $L_{R-P}$ -Algorithmus nicht optimal arbeitet.

**Nützlichkeit** In einer stationären Umgebung mit konstanten  $c_i > 0$  ergibt sich als der Erwartungswert der durchschnittlichen Bestrafung

$$\lim_{n \rightarrow \infty} E(M[n]) = \lim_{n \rightarrow \infty} E\left(\sum_{i=1}^N c_i p_i[n]\right) = \sum_{i=1}^N c_i E_i^* = \frac{N}{\sum_{j=1}^N \frac{1}{c_j}} \quad (15.70)$$

Mit Hilfe der Ungleichung  $\sum_{j=1}^N c_j \cdot \sum_{j=1}^N \frac{1}{c_j} \geq N^2$ , die echt erfüllt ist, falls nicht alle  $c_j$  identisch sind, lässt sich die erwartete Bestrafung durch

$$\lim_{n \rightarrow \infty} E(M[n]) = \frac{N}{\sum_{j=1}^N \frac{1}{c_j}} < \frac{\sum_{j=1}^N c_j}{N} = M_0 \quad (15.71)$$

abschätzen. Das  $L_{R-P}$ -Lerngesetz liefert damit einen Automaten, der besser als der reine Zufall funktioniert, er verhält sich somit nützlich. Der Vektor der Aktionswahrscheinlichkeiten konvergiert jedoch nicht gegen einen festen Wert, und somit auch nicht gegen den Einheitsvektor der am seltensten bestraften Aktion. Dies kann unter gewissen Voraussetzungen und mit ausreichend hoher Wahrscheinlichkeit der im Folgenden vorgestellte  $L_{R-I}$ -Algorithmus leisten.

### 15.3.3.2 Ein absolut nützliches Lerngesetz: *Linear Reward-Inaction*

Der *Linear Reward-Inaction*- oder  $L_{R-I}$ -Algorithmus verzichtet im Gegensatz zum  $L_{R-P}$ -Lerngesetz auf die Bestrafung (*penalty*) im Fall einer als falsch bewerteten Aktion. Dieser durch  $b \equiv 0$  im allgemeinen linearen Lerngesetz (15.54) formulierte Umstand führt auf grundlegend verschiedene Konvergenzeigenschaften. Unter dem  $L_{R-I}$ -Lerngesetz konvergiert der Vektor der Aktionswahrscheinlichkeiten  $\underline{p}[n]$  mit der Wahrscheinlichkeit 1 gegen einen Einheitsvektor. Dieser ist jedoch nicht zwingend der Einheitsvektor  $\underline{e}_l$  zur am schwächsten bestraften Aktion  $\alpha_l$ , d. h. das Lerngesetz ist nicht optimal. Da die Konvergenzwahrscheinlichkeit gegen einen beliebigen Einheitsvektor zudem vom Anfangsvektor  $\underline{p}[0]$  abhängt, ist der Algorithmus auch nicht ergodisch. Es handelt sich vielmehr um ein absolut nützliches (und somit auch  $\varepsilon$ -optimales) Lerngesetz, mit dem durch ausreichend kleine Lernschrittweite  $a$  beliebig hohe Konvergenzwahrscheinlichkeit gegen die beste Aktion realisiert werden kann. In Vektorschreibweise lautet der Algorithmus

$$\underline{p}[n+1] = \underline{p}[n] + a \cdot (1 - \beta[n]) \cdot (\underline{e}_i - \underline{p}[n]) \quad (15.72)$$

Konvergenz Besitzt der Markov-Prozess (15.72) einen absorbierenden Zustand, so muss dort  $\underline{p}^* \equiv \underline{p}[n] = \underline{p}[n+1]$  gelten. Gesucht ist also ein Vektor  $\underline{p}^*$ , der für beliebige Bestrafung  $\beta[n]$  die Gleichung

$$\underline{p}^* = \underline{p}^* + a \cdot (1 - \beta[n]) \cdot (\underline{e}_i - \underline{p}^*) \quad (15.73)$$

erfüllt. Offensichtlich ist die Gleichung für  $\beta[n] = 1$  immer wahr, schließlich findet keine Bestrafung falscher Aktionen statt. Für  $\beta[n] = 0$  gilt die Gleichung für  $\underline{p}^* = \underline{e}_i$ , also den Einheitsvektor zur zuletzt belohnten Aktion  $\alpha[n] = \alpha_i$ . Dabei muss  $\underline{e}_i$  nicht zwingend der Einheitsvektor  $e_l$  der optimalen Aktion  $\alpha_l$  sein. Dies ist zwar meist der Fall, es ist jedoch nicht garantiert und hängt entscheidend vom Anfangsvektor  $\underline{p}[0]$  ab. Die Wahrscheinlichkeit dafür wird durch die vom Startvektor  $\underline{p} = \underline{p}[0]$  abhängige Funktion  $\Gamma_l(\underline{p})$  angegeben. Die Funktion lässt sich

nicht direkt berechnen, sondern nur durch sog. super- und subreguläre Funktionen abschätzen. Für die Herleitung sei auf [164] verwiesen. Im Folgenden wird die Berechnungsvorschrift für die Schranken bei Verwendung des  $L_{R-I}$ -Lerngesetzes zusammengefasst.

*Schranken für die Konvergenzwahrscheinlichkeit* Die Wahrscheinlichkeit  $\Gamma_i(\underline{p})$ , mit der der  $L_{R-I}$ -Algorithmus, vom Startvektor  $\underline{p} = \underline{p}[0]$  ausgehend, für  $n \rightarrow \infty$  gegen den Einheitsvektor  $e_i$  konvergiert, lässt sich durch die Ungleichung

$$\Gamma_i^-(\underline{p}) \leq \Gamma_i(\underline{p}) \leq \Gamma_i^+(\underline{p}) \quad (15.74)$$

mit der unteren und oberen Schranke

$$\Gamma_i^-(\underline{p}) = \frac{1 - e^{-a_i p_i}}{1 - e^{-a_i}} \quad \text{und} \quad \Gamma_i^+(\underline{p}) = \frac{1 - e^{-b_i p_i}}{1 - e^{-b_i}} \quad (15.75)$$

abschätzen.  $p_i$  ist dabei die anfängliche Aktionswahrscheinlichkeit  $p_i = p_i[0]$ . Die Werte  $a_i$  und  $b_i$  lassen sich für das lineare  $L_{R-I}$ -Lerngesetz mit gegebener Schrittweite  $a$  durch folgenden Algorithmus bestimmen:

- Die Bestrafungswahrscheinlichkeiten  $c_i, i = 1, \dots, N$  der Umgebung werden in Belohnungswahrscheinlichkeiten  $d_i = 1 - c_i$  umgerechnet. Es werden die Werte

$$\max_{j \neq i} \frac{d_j}{d_i} \quad \text{und} \quad \min_{j \neq i} \frac{d_j}{d_i} \quad (15.76)$$

bestimmt.

- Die Funktion  $V(x)$  ist definiert durch

$$V(x) = \begin{cases} \frac{e^x - 1}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (15.77)$$

- Die Werte  $a_i$  und  $b_i$  ergeben sich aus der Lösung der Gleichungen

$$\frac{1}{\max_{j \neq i} \frac{d_j}{d_i}} = V(ax)|_{x=a_i} \quad \text{und} \quad \min_{j \neq i} \frac{d_j}{d_i} = V(-ax)|_{x=b_i} \quad (15.78)$$

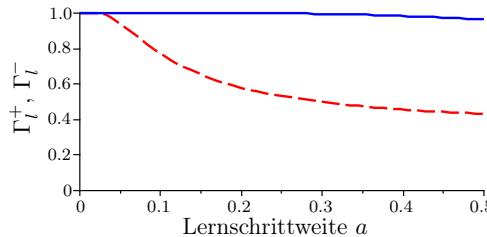
und werden in die Abschätzung (15.75) eingesetzt.

Der vorliegende Algorithmus kann für die Abschätzung aller Wahrscheinlichkeiten  $\Gamma_i(\underline{p})$ , genutzt werden, in der Regel ist jedoch die Konvergenz gegen die optimale Aktion, also  $\Gamma_l$  von Interesse. Die allgemeine Fassung der Bestimmungsvorschrift für die Schranken im Fall nichtlinearer Lerngesetze ist in [164] beschrieben.

Die Wahrscheinlichkeiten  $\Gamma_i(\underline{p})$  hängen zum einen von der Initialisierung  $\underline{p}[0]$  ab. In der Regel (wenn kein Vorwissen über die Umgebung vorliegt) werden jedoch alle Anfangswahrscheinlichkeiten gleich gesetzt. Interessanter ist die Abhängigkeit, insbesondere von  $\Gamma_l(\underline{p})$  von der Lernschrittweite  $a$ . Je kleiner diese ist, umso näher rückt  $\Gamma_l(\underline{p})$  gegen 1.

$a$	0.05	0.1	0.2	0.3	0.4	0.5
$\Gamma_l^+(\underline{p})$	1	1	0.9997	0.9960	0.9840	0.9635
$\Gamma_l^-(\underline{p})$	0.9436	0.7725	0.5795	0.4990	0.4574	0.4322

**Tabelle 15.1:** Schranken für  $\Gamma_l(\underline{p})$  in Abhängigkeit der Lernschrittweite  $a$



**Abb. 15.6:** Schranken der Konvergenzwahrscheinlichkeit gegen die optimale Aktion für gleich initialisierte Startwahrscheinlichkeiten  $p_i[0] = 1/N$

**Beispiel 15.3.2** Für das Beispiel eines Automaten mit 3 Aktionen und Bestrafungswahrscheinlichkeiten der Umgebung  $\{0.6, 0.5, 0.9\}$  ergeben sich die in Tabelle 15.1 angegebenen Schranken für  $\Gamma_l(\underline{p}) = \Gamma_2(\underline{p})$  bei mit  $p_i[0] = \frac{1}{3}$  gleich initialisierten Aktionswahrscheinlichkeiten, abhängig von der Lernschrittweite  $a$ . Abbildung 15.6 zeigt die Abhängigkeit der Schranken von  $a$  graphisch.

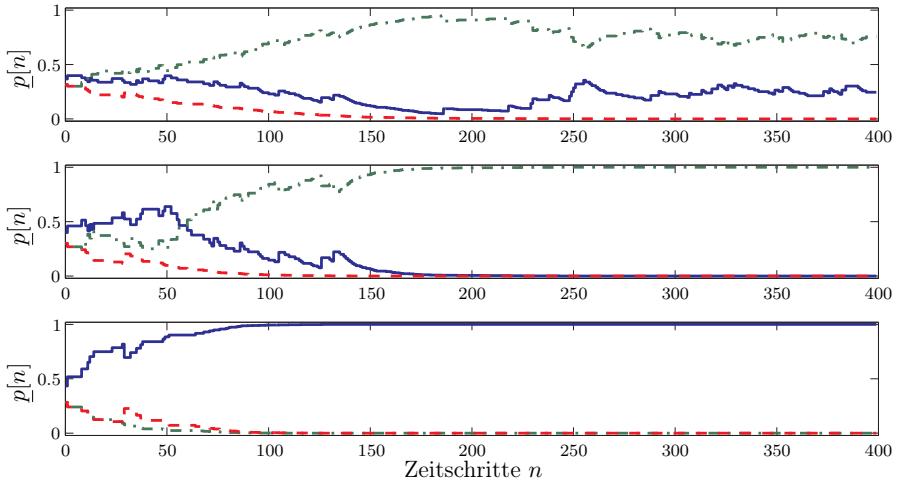
**Beispiel 15.3.3** Im Gegensatz zum  $L_{R-P}$ -Algorithmus konvergiert der Vektor der Aktionswahrscheinlichkeiten unter dem  $L_{R-I}$ -Lerngesetz gegen einen Einheitsvektor und hat damit einen absorbierenden Zustand. Dass es sich dabei nicht um den Vektor handeln muss, der der optimalen Aktion entspricht, zeigt Abb. 15.7. In den beiden oberen Zeitverläufen strebt die Wahrscheinlichkeit  $p_2[n]$  für die am seltensten bestrafte Aktion gegen 1, und zwar umso schneller, je größer  $a$  ist. Im unten dargestellten Fall  $a = 0.15$  ist die Lernschrittweite so groß gewählt, dass der Automat zügig gegen die zweitbeste und somit nicht optimale Aktion  $\alpha_1$  konvergiert. Ist  $p_1[n]$  bei 1, wird nie mehr eine andere Aktion gewählt. Da  $p_1[n]$  auch nicht abgewertet wird, wenn  $\alpha_1$  als falsch bewertet wird, bleibt der Wahrscheinlichkeitsvektor in diesem falschen absorbierenden Zustand gefangen.

Nützlichkeit Betrachtet man das  $L_{R-I}$ -Lerngesetz, so ist mit Satz 15.3.1 absolute Nützlichkeit des Lerngesetzes offensichtlich, da

$$\frac{g_i(\underline{p})}{p_i} = \lambda(\underline{p}) = a, \quad \frac{h_i(\underline{p})}{p_i} = \mu(\underline{p}) = 0 \quad \forall i = 1, \dots, N \quad (15.79)$$

Nachprüfen lässt sich diese Tatsache (für den allgemeinen Fall eines absolut nützlichen Algorithmus), wenn man das Inkrement der erwarteten Bestrafung<sup>5)</sup>

<sup>5)</sup> Erklärung, dass der bedingte Erwartungswert auftaucht



**Abb. 15.7:** Lernprozess unter dem  $L_{R-I}$ -Algorithmus mit Lernschrittweiten  $a = 0.05/0.1/0.15$  (von oben nach unten).  $p_1[n]$  (-),  $p_2[n]$  (-·),  $p_3[n]$  (-·)

$$\Delta M[n] = E(M[n+1] - M[n] \mid p[n]) = \sum_{i=1}^N c_i \Delta p_i[n] \quad (15.80)$$

berechnet. Zunächst ist analog zu (15.63), das erwartete Inkrement der einzelnen Aktionswahrscheinlichkeiten unter dem allgemeinen Lerngesetz (15.52)

$$\Delta p_i[n] = E(p_i[n+1] - p_i[n] \mid p[n]) \quad (15.81)$$

zu ermitteln. Mit  $\lambda(\underline{p}) = \frac{g(p_i)}{p_i}$  und  $\mu(\underline{p}) = \frac{h(p_i)}{p_i}$  ergibt sich nach einigen Umformungen

$$\Delta p_i[n] = -(\lambda(\underline{p}) + \mu(\underline{p})) \cdot p_i[n] \cdot \left( \sum_{j=1}^N (c_j - c_i) p_j[n] \right) \quad (15.82)$$

Einsetzen in (15.80) liefert schließlich

$$\Delta M[n] = (\lambda(\underline{p}) + \mu(\underline{p})) \cdot \sum_{i=1}^N \sum_{j=1}^N (c_i(c_j - c_i) p_i[n] p_j[n]) \quad (15.83)$$

was sich als quadratische Form  $\Delta M[n] = \frac{1}{2} \underline{p}^T[n] \mathbf{C} \underline{p}[n]$  mit der symmetrischen Matrix  $\mathbf{C}$  schreiben lässt. Die Elemente  $c_{ij} = -(c_i - c_j)^2$  sind bis auf die Diagonalelemente  $c_{ii} = 0$  negativ, so dass sicher  $\Delta M[n] < 0$  gilt, solange  $\underline{p}$  kein Einheitsvektor ist.

### 15.3.3.3 Ein Kompromiss: Der $L_{R-\varepsilon P}$ -Algorithmus

Der Nachteil des  $L_{R-I}$ -Lerngesetzes, dass für  $n \rightarrow \infty$  die Konvergenz gegen die optimale Aktion nicht *sicher* ist, lässt sich durch das Einführen einer vergleichsweise kleinen Bestrafung mildern. Durch Ansetzen von  $b = \varepsilon a$  entsteht das  $L_{R-\varepsilon P}$  Lerngesetz.  $\varepsilon$  stellt also die Relation zwischen Bestrafung und Belohnung im allgemeinen linearen Lerngesetz (15.54) dar.

*Konvergenz* Zur Beurteilung des Konvergenzverhaltens wird wiederum der Erwartungswert des Inkrements der Wahrscheinlichkeiten  $p_i[n+1] - p_i[n]$  bzw. des gesamten Wahrscheinlichkeitsvektors  $\underline{p}[n+1] - \underline{p}[n]$  bei gegebenem  $\underline{p}[n]$  entsprechend (15.81) betrachtet. Unter dem allgemeinen linearen Lerngesetz in der vektoriellen Form (15.56) erhält man

$$\Delta \underline{p}[n] = \sum_{j=1}^N p_j[n] \left[ a(1 - c_j)(\underline{e}_j - \underline{p}[n]) + bc_j \left( \frac{1}{N-1} \bar{\underline{e}}_j - \underline{p}[n] \right) \right] =: aw(\underline{p}) \quad (15.84)$$

Für die  $\varepsilon$ -Bestrafung  $b = \varepsilon a$  ergibt sich schließlich

$$\Delta \underline{p}[n] = \sum_{j=1}^N ap_j[n] \left[ (1 - c_j)(\underline{e}_j - \underline{p}[n]) + \varepsilon c_j \left( \frac{1}{N-1} \bar{\underline{e}}_j - \underline{p}[n] \right) \right] =: aw(\underline{p}) \quad (15.85)$$

Für  $\varepsilon = 0$ , was dem  $L_{R-I}$ -Algorithmus entspricht, verschwindet  $\Delta \underline{p}[n]$ , wenn  $\underline{p}[n]$  ein beliebiger Einheitsvektor ist<sup>6)</sup>. Für genügend kleine Werte von  $\varepsilon$  lässt sich zeigen, dass genau eine Lösung  $\underline{p}^*$  der Gleichung  $\Delta \underline{p}[n] = 0$  in der Nähe des Einheitsvektors  $\underline{e}_l$  liegt, der zur geringsten Bestrafungswahrscheinlichkeit  $c_l$  gehört und dabei die Eigenschaften eines Wahrscheinlichkeitsvektors hat. Alle übrigen Lösungen verletzen diese. Unter Vernachlässigung von Termen zweiter Ordnung lassen sich die Elemente von  $\underline{p}^*$  durch

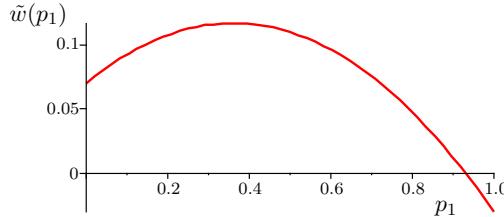
$$\begin{aligned} p_i^* &= \varepsilon \frac{c_l}{(N-1)(c_i - c_l)}, & i \neq l \\ p_l^* &= 1 - \sum_{i \neq l} p_i^* \end{aligned} \quad (15.86)$$

berechnen [164]. Durch ausreichend kleines  $\varepsilon$  kann  $\underline{p}^*$  also dem optimalen Einheitsvektor  $\underline{e}_l$  beliebig angenähert werden. Der  $L_{R-\varepsilon P}$ -Algorithmus ist somit  $\varepsilon$ -optimal, und dies trotz seiner Ergodizität – es existiert ja kein absorbierender Zustand und  $\underline{p}[n]$  konvergiert unabhängig von  $\underline{p}[0]$  in Verteilung (Nachweis über Kontraktionsabbildung).

Ob  $\underline{p}^*$  tatsächlich der Grenzwert von  $E(\underline{p}[n])$  für  $n \rightarrow \infty$  ist, muss durch Stabilitätsanalyse der Ruhelage  $\underline{p}^*$  des Systems von Differenzengleichungen (15.85)

---

<sup>6)</sup> vgl. die Diskussion von Gleichung (15.73)



**Abb. 15.8:**  $\tilde{w}(p_1)$  für den Automaten mit 2 Aktionen

gezeigt werden. Durch die Bedingung  $\sum_{i=1}^N p_i[n] = 1$  lässt sich eine Wahrscheinlichkeit (z. B.  $p_N$ ) eliminieren und man betrachtet das Ersatzsystem

$$\frac{\Delta \tilde{p}}{a} = \underline{\tilde{w}}(\tilde{p}) \quad (15.87)$$

mit dem verkürzten Wahrscheinlichkeitsvektor  $\tilde{p} = [p_1 \dots p_{N-1}]^T$ . Im Grenzübergang für  $a \rightarrow 0$  ergibt sich ein Satz von  $N - 1$  Differentialgleichungen erster Ordnung

$$\lim_{a \rightarrow 0} \left( \frac{\Delta \tilde{p}}{a} \right) = \frac{d \tilde{p}}{da} = \underline{\tilde{w}}(\tilde{p}) \quad \hat{=} \quad \dot{x} = \underline{f}(x) \quad (15.88)$$

Für den  $L_{R-\varepsilon P}$ -Algorithmus ist dadurch ein nichtlineares dynamisches System definiert. Das folgende Beispiel zeigt, wie sich für den Fall  $N = 2$  Stabilität der Ruhelage  $\underline{p}^*$  und damit  $\varepsilon$ -Optimalität des  $L_{R-\varepsilon P}$ -Lerngesetzes nachweisen lässt.

**Beispiel 15.3.4** Für einen Automaten mit 2 Aktionen erhält man aus (15.85) und mit  $p_2 = 1 - p_1$

$$\Delta p_1 = a(c_2 - c_1)p_1(1 - p_1) + \varepsilon(c_2(1 - p_1)^2 - c_1p_1^2) \quad (15.89)$$

Die zugehörige Differentialgleichung ist

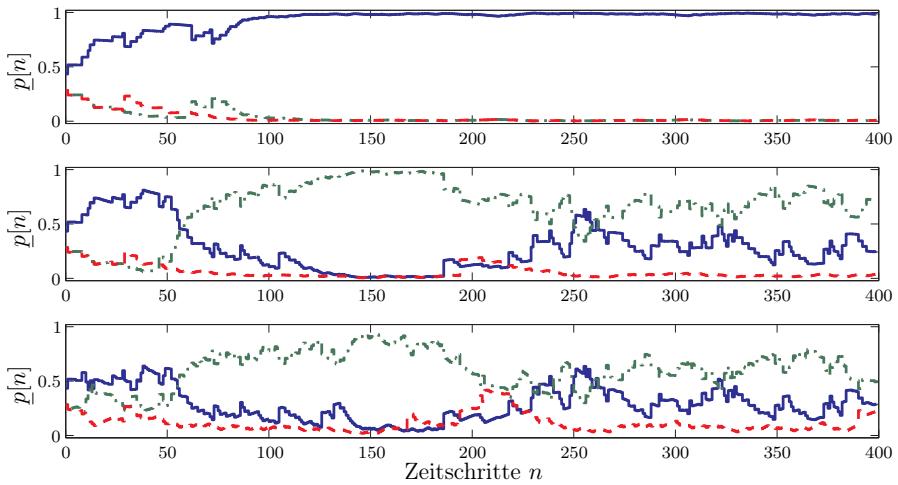
$$\dot{x} = f(x) = -(1 - \varepsilon)(c_2 - c_1)x^2 + ((c_2 - c_1) - 2\varepsilon c_2)x + \varepsilon c_2 \quad (15.90)$$

Für  $c_1 = 0.3$ ,  $c_2 = 0.7$  und  $\varepsilon = 0.1$  ist

$$f(x) = -0.36x^2 + 0.26x + 0.07 \quad (15.91)$$

so dass sich die Ruhelage  $p_1^* = x^* = -0.931$  ergibt. In Abb. 15.8 ist die Funktion  $f(x) = \tilde{w}(p_1)$  dargestellt. Im Intervall  $(0, 1)$  ist  $p_1^*$  offensichtlich stabile Ruhelage, da  $\tilde{w}(p_1) > 0$  für  $p_1 \in (0, p_1^*)$  und  $\tilde{w}(p_1) < 0$  für  $p_1 \in (p_1^*, 1)$  gilt.

Der analytische Stabilitätsnachweis – insbesondere das Finden einer Lyapunov-Funktion, mit der Stabilität für beliebige Anfangswerte gezeigt werden kann – gestaltet sich für eine beliebige Anzahl von Aktionen  $N$  erheblich schwieriger.



**Abb. 15.9:** Lernprozess unter dem  $L_{R-\varepsilon P}$ -Algorithmus mit  $a = 0.15$  und  $\varepsilon = 0.01/0.05/0.15$  (von oben nach unten).  $p_1[n]$  (–),  $p_2[n]$  (–·),  $p_3[n]$  (–·–)

**Beispiel 15.3.5** Die Leistungsfähigkeit des  $L_{R-\varepsilon P}$ -Algorithmus als Kompromiss zwischen den beiden zuerst vorgestellten Lerngesetzen zeigt Abb. 15.9. Während im obersten Plot  $\varepsilon$  so schwach ausfällt, dass der Algorithmus trotzdem gegen die falsche Aktion  $\alpha_1$  konvergiert, ermöglicht in den beiden unteren Verläufen sichtbar die  $\varepsilon$ -Bestrafung bei etwa  $n = 50$  eine Umkehr im Lernvorgang und verhindert das Festhalten des Automaten an der nicht optimalen Aktion. Offensichtlich ist auch, dass der  $L_{R-\varepsilon P}$ -Algorithmus als ergodisches Lerngesetz in Verteilung konvergiert und die Varianz der Wahrscheinlichkeiten  $p_i[n]$  mit wachsendem  $\varepsilon$  zunimmt.

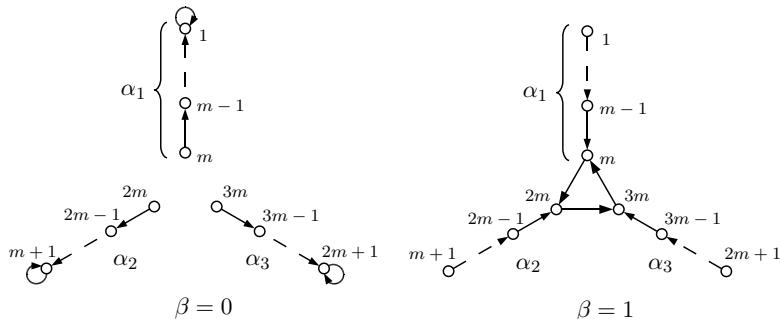
Abschließend sei angemerkt, dass die mit  $\varepsilon$  stark anwachsende Streuung der Wahrscheinlichkeiten in Abb. 15.9 auch auf die eng beieinander liegenden Bestrafungswahrscheinlichkeiten der Umgebung zurückzuführen ist. Je dichter die niedrigsten Bestrafungswahrscheinlichkeiten beieinander liegen, umso schwerer fällt es dem Algorithmus, sich für die günstigste Aktion zu entscheiden.

### 15.3.4 Ein deterministischer Automat mit fester Struktur

Als Alternative zu lernenden Automaten, wie den bisher betrachteten stochastischen Automaten mit veränderlicher Struktur, bieten sich Automaten mit festem Verhaltensmuster an. Ein Beispiel hierfür ist der in Abb. 15.4 dargestellte  $L_{2,2}$ -Automat, der auf Michail L. Tsetlin (1924–1966), einen der Pioniere der Automatentheorie in der ehemaligen Sowjetunion, zurückgeht. Dieser *Tsetlin-Automat*

besitzt 2 Zustände (erster Index) und 2 Aktionen (zweiter Index). Die Transitionen zwischen den Zuständen erfolgt, abhängig von der Bewertung  $\beta$ , nach einem festen Schema. Beim deterministischen  $L_{2,2}$ -Automaten bleibt im Fall  $\beta = 0$  der Automat im selben Zustand, für  $\beta = 1$  wechselt der Zustand. Ist der Automat stochastisch, so ist jede Transition mit einer Wahrscheinlichkeit belegt.

Die einfache Struktur des deterministischen  $L_{2,2}$ -Automaten hat in einer stochastischen Umgebung den Nachteil, dass jede als negativ bewertete Aktion einen sofortigen Wechsel des Zustands und damit gleichzeitig der Aktion zur Folge hat. Dadurch fehlt dem Automaten Robustheit gegenüber der stochastischen Natur seiner Umgebung. Er wird nicht in der Lage sein, wie von einem (zumindest annähernd) optimalen Automaten verlangt, im Grenzfall die am seltensten bestrafte Aktion zu wählen. Abhilfe schafft die Erweiterung des Automaten um ein *Gedächtnis*. Im Folgenden wird beispielhaft ein  $L_{3m,3}$ -Automat mit  $N = 3$  Aktionen und der Gedächtnistiefe  $m$  betrachtet.



**Abb. 15.10:** Digraphen des  $L_{3m,3}$ -Automaten

*Funktionsweise* Die  $3m$  Zustände des Automaten sind, wie in Abb. 15.10 zu erkennen, den 3 Aktionen in Form von Ästen zugeordnet. Für  $\beta = 0$  wandert der Zustand nach außen. Von der Astwurzel ist durch  $m - 1$  aufeinander folgende positive Bewertungen der äußerste Zustand zu erreichen. Der Automat braucht nun mindestens  $m$  negative Bewertungen seiner Aktion in Folge, um die Handlung ändern zu können. Dieses Gedächtnis von  $m$  Zuständen versetzt den Automaten in die Lage, annähernd optimal zu arbeiten.

*Nützlichkeit und Optimalität* Betrachtet man den  $L_{2,2}$ -Automaten aus Abb. 15.4, der in einer Umgebung mit Bestrafungswahrscheinlichkeiten  $\{c_1, c_2\}$  operiert, so berechnet sich die erwartete Bestrafung wie folgt. Mit den bedingten Transitionsmatrizen

$$\mathbf{F}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{F}^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (15.92)$$

lässt sich die Matrix  $\tilde{\mathbf{F}}$  der absoluten Transitionswahrscheinlichkeiten gemäß (15.37) als

$$\tilde{\mathbf{F}} = \begin{bmatrix} d_1 & c_1 \\ c_2 & d_2 \end{bmatrix} \quad (15.93)$$

mit  $d_i = 1 - c_i$  angeben. Die Wahrscheinlichkeiten der Zustandstransitionen  $\psi[n] \rightarrow \psi[n+1]$  hängen neben  $c_1$  und  $c_2$  nur vom gerade aktuellen Zustand  $\psi[n]$  ab. Damit ist die Folge der Zustände (und gleichzeitig Aktionen) eine Markov-Kette mit der Transitionsmatrix  $\tilde{\mathbf{F}}$ . Deren stationäre Verteilung, die den Grenzwert der Zustands- bzw. Aktionswahrscheinlichkeiten angibt, berechnet sich aus

$$\underline{\pi} = \tilde{\mathbf{F}}^T \underline{\pi} \quad (15.94)$$

Für den  $L_{2,2}$ -Automaten ergibt die Lösung von Gl. (15.94)

$$\begin{aligned} \pi_1 &= \lim_{n \rightarrow \infty} \Pr(\alpha[n] = \alpha_1) = \lim_{n \rightarrow \infty} \Pr(\psi[n] = 1) = \frac{c_2}{c_1 + c_2} \\ \pi_2 &= \lim_{n \rightarrow \infty} \Pr(\alpha[n] = \alpha_2) = \lim_{n \rightarrow \infty} \Pr(\psi[n] = 2) = \frac{c_1}{c_1 + c_2} \end{aligned} \quad (15.95)$$

Daraus folgt die im Grenzwert erwartete Bestrafung

$$M(L_{2,2}) = \lim_{n \rightarrow \infty} M[n] = c_1\pi_1 + c_2\pi_2 = \frac{2c_1c_2}{c_1 + c_2} \quad (15.96)$$

Für  $c_1 \neq c_2$  liegt  $M(L_{2,2})$  unter dem Vergleichswert  $M_0$  des reinen Zufallsautomaten. Der  $L_{2,2}$ -Automat ist somit nützlich. Die Leistung des Automaten lässt sich durch Hinzufügen des Gedächtnisses steigern. Der daraus resultierende  $L_{Nm,N}$ -Automat mit  $N$  Aktionen und  $N \cdot m$  Zuständen verhält sich sogar  $\varepsilon$ -optimal.

**Definition 15.3.5** Ein deterministischer Automat ist  $\varepsilon$ -optimal, wenn für jedes  $\varepsilon > 0$  eine Zahl  $m_1$  existiert, so dass für  $m > m_1$

$$M = \lim_{n \rightarrow \infty} M[n] \leq \min\{c_1, \dots, c_N\} + \varepsilon \quad (15.97)$$

gilt, wobei  $m$  die Gedächtnistiefe ist und  $c_1, \dots, c_N$  im abgeschlossenen Intervall  $[0, 1]$  liegen.

Für einen beliebigen  $L_{Nm,N}$ -Automaten lässt sich, werden die Zustände analog zu Abb. 15.10 angeordnet, die absolute Zustandstransitionsmatrix

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{\mathbf{F}}_{11} & \dots & \tilde{\mathbf{F}}_{1N} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{F}}_{N1} & \dots & \tilde{\mathbf{F}}_{NN} \end{bmatrix} \quad (15.98)$$

aufstellen, mit den  $m \times m$ -Untermatrizen  $\tilde{\mathbf{F}}_{ij}$ , die die Wahrscheinlichkeiten enthalten, dass der Zustand vom  $i$ -ten in den  $j$ -ten Zweig übergeht. Für den  $L_{2m,2}$ -Automaten etwa sind

$$\tilde{\mathbf{F}}_{ii} = \begin{bmatrix} d_i & c_i & 0 & \dots & 0 \\ d_i & 0 & c_i & \ddots & \vdots \\ 0 & d_i & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_i \\ 0 & \dots & 0 & d_i & 0 \end{bmatrix} \quad \text{und} \quad \tilde{\mathbf{F}}_{ij} = \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & c_i \end{bmatrix} \quad (15.99)$$

für  $i, j = 1, 2$  und  $i \neq j$ . Mit Gleichung (15.94) lässt sich wiederum die stationäre Verteilung  $\underline{\pi}$  der Zustandswahrscheinlichkeiten berechnen, aus der durch

$$\lim_{n \rightarrow \infty} p_i[n] = \sum_{j=(i-1) \cdot m + 1}^{i \cdot m} \pi_j, \quad i = 1, \dots, N \quad (15.100)$$

die Verteilung der Aktionswahrscheinlichkeiten im Grenzwert folgt. Für die erwartete Bestrafung ergibt sich schließlich, vgl. [227]

$$M(L_{Nm,N}) = \lim_{n \rightarrow \infty} \sum_{i=1}^N c_i p_i[n] = \frac{\sum_{i=1}^N \frac{c_i^m - d_i^m}{c_i^{m-1}(c_i - d_i)}}{\sum_{i=1}^N \frac{c_i^m - d_i^m}{c_i^m(c_i - d_i)}} \quad (15.101)$$

Man zeigen, dass unter der Voraussetzung

$$\min\{c_1, \dots, c_N\} = c_l \leq \frac{1}{2} \quad (15.102)$$

dass also mindestens eine Aktion häufiger belohnt als bestraft wird, sich die erwartete Bestrafung mit der Größe des Gedächtnisses  $m$  der geringsten Bestrafungswahrscheinlichkeit annähert, also

$$\lim_{m \rightarrow \infty} M(L_{Nm,N}) = \min\{c_1, \dots, c_N\} \quad (15.103)$$

Damit ist der  $L_{Nm,N}$ -Automat  $\varepsilon$ -optimal. In der Tat reichen bereits niedrige Gedächtnistiefe  $m$ , um die erwartete Bestrafung des Automaten deutlich zu verringern, wie das folgende Beispiel zeigt.

**Beispiel 15.3.6** Ein  $L_{3m,3}$ -Automaten operiert in einer durch  $\underline{c} = \{0.8, 0.3, 0.9\}$  beschriebenen Umgebung. Für verschiedene Werte von  $m$  ergeben sich die in Tabelle 15.2 dargestellten Erwartungswerte der Bestrafung. Schon für  $m = 3$  liegt die erwartete Bestrafung bei etwa 0.3 und somit nahe  $c_2 = \min\{0.8, 0.3, 0.9\}$ .

**Beispiel 15.3.7** Das Verhalten eines  $L_{3m,3}$ -Automaten in einer durch  $\underline{c} = \{0.8, 0.3, 0.9\}$  beschriebenen stochastischen Umgebung zeigt Abb. 15.11. Mit der Gedächtnistiefe  $m = 2$  ausgestattet (oben) verlässt der Zustand  $\psi[n]$  des Automaten noch recht häufig den zur optimalen Aktion  $a_2$  gehörenden Zweig. Immerhin

$m$	1	2	3	3	5	10
$M(L_{3m,3})$	0.5268	0.4094	0.3488	0.3212	0.3091	0.3001

Tabelle 15.2: Erwartete Bestrafung für den  $L_{3m,3}$ -Automaten

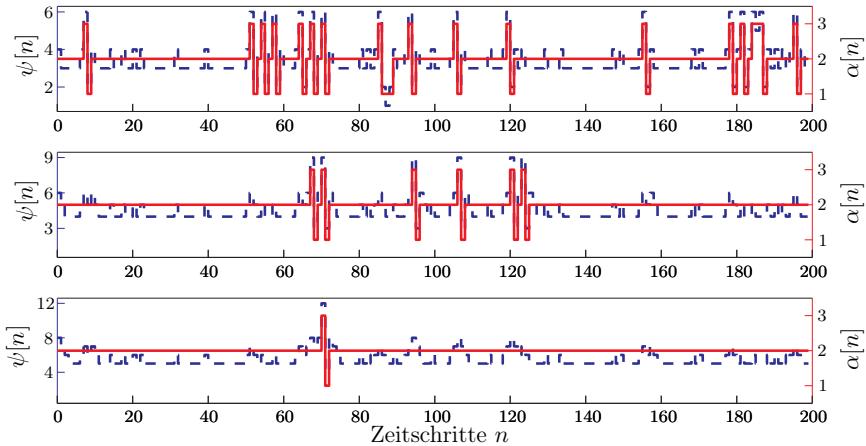
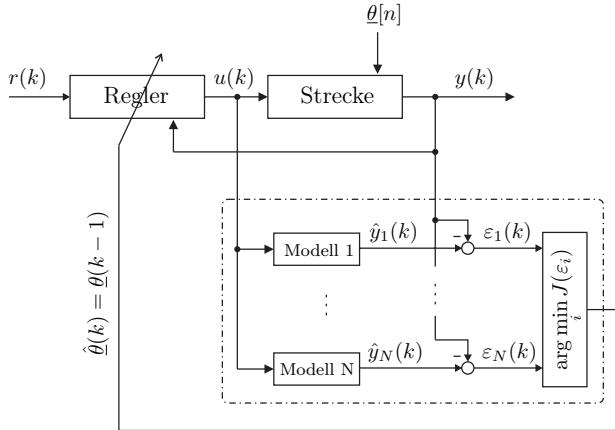


Abb. 15.11: Lernprozess dreier  $L_{3m,3}$ -Automaten mit Gedächtnistiefe  $m = 2/3/4$ . Zustand  $\psi[n]$  (- -) und Aktion  $\alpha[n]$  (—)

wird in 30% der Fälle die Aktion  $\alpha_2$  bestraft. Da jedoch die beiden anderen Aktionen wesentlich öfter sanktioniert werden, kehrt der Zustand bald wieder in den richtigen Zweig zurück. Eine Vergrößerung der Gedächtnistiefe mildert die Auswirkung dieser Zustandstransitionen. Schon für  $m = 3$  (Mitte) und erst recht für  $m = 4$  behält der Automat, trotz gelegentlicher Bestrafung, die optimale Aktion  $\alpha_2$  bei.

## 15.4 Prognose stochastischer Parameterwechsel

Die im vorigen Abschnitt vorgestellten Automaten und Lerngesetze zum Anpassen ihrer Handlungsgrundlage lassen sich vorteilhaft zur Milderung der Auswirkung stochastischer Störungen ausnutzen, die durch eine Markov-Kette beschrieben werden. Als Anwendungsbeispiel dient die Regelung mit multiplen Modellen, wie sie auch in Kap. 14.3 beschrieben ist. Die Störgröße wird als zeitveränderlicher Parameter  $\theta(k)$  verstanden, dessen Wert sich gemäß der Transitionsmatrix einer unbekannten Markov-Kette zu bekannten Zeitpunkten verändert. In den folgenden Abschnitten wird das Problem des inhärenten Regelfehlers bei sprunghafter Parameteränderung dargelegt, eine Erweiterung der Regelung mit multiplen Modellen um einen Prognosemechanismus vorgestellt, sowie die erreichbare



**Abb. 15.12:** Prinzip der Regelung mit multiplen Modellen

Verbesserung der Regelgüte quantifiziert und an einem Simulationsbeispiel illustriert.

#### 15.4.1 Regelung mit multiplen Modellen

Betrachtet wird die zeitdiskrete Ausgangsregelung mit multiplen Modellen für ein Eingrößensystem mit veränderlichem Parameter(vektor)  $\underline{\theta}(k)$ . Dazu wird im Digitalrechner eine Anzahl  $N$  von Streckenmodellen mit den festen Parametern  $\{\hat{\underline{\theta}}_1, \dots, \hat{\underline{\theta}}_N\}$  realisiert. Die jeweils aktuelle Identifikation des Modells  $\hat{\underline{\theta}}$  beruht auf Messungen des Ein- und Ausgangs der Strecke  $u(k)$  und  $y(k)$ . Mit diesen Messwerten zu diskreten Zeitpunkten  $k$  erzeugt jedes Modell eine *Prädiktion*  $\hat{y}_i(k)$  des Ausgangs der Strecke. Die Prädiktionen werden mit der Messung verglichen und das Modell, welches gemäß einem zuvor festgelegten Gütemaß das Systemverhalten am besten repräsentiert, wird zur Regelung der Strecke verwendet. Diesem Modell entsprechend werden die Parameter der zeitdiskreten Regelung eingestellt.

Ein Eingrößensystem  $n$ -ter Ordnung lässt sich in zeitdiskreter Form durch das ARX-Modell

$$y(k) = \sum_{\nu=0}^{n-1} a_\nu(k-d)y(k-\delta-\nu) + \sum_{\nu=0}^{n-1} b_\nu(k-d)u(k-\delta-\nu) \quad (15.104)$$

beschreiben.  $\delta$  ist der (diskrete) relative Grad des Ausgangs und  $d$  die Verzögerung in Zeitschritten, mit der Änderungen der Parameter  $a_\nu$  und  $b_\nu$  auf den Ausgang wirken. Eine alternative Darstellung ist

$$y(k) = \underline{\phi}^T(k-\delta)\underline{\theta}(k-d) \quad (15.105)$$

mit dem Regressionsvektor der  $n$  letzten Messungen des Ein- und Ausgangs

$$\underline{\phi}(k) = [y(k), \dots, y(k-n+1), u(k), \dots, u(k-n+1)]^T \quad (15.106)$$

und dem Parametervektor

$$\underline{\theta}(k) = [a_0(k), \dots, a_{n-1}(k), b_0(k), \dots, b_{n-1}(k)]^T \quad (15.107)$$

Ohne Beschränkung der Allgemeinheit betrachten wird den Fall  $d = \delta = 1$ . Somit lautet Gleichung (15.105)

$$y(k) = \underline{\phi}^T(k-1) \underline{\theta}(k-1) \quad (15.108)$$

Die Prädiktionen der  $N$  parallelen Streckenmodelle werden durch

$$\hat{y}_i(k) = \underline{\phi}^T(k-1) \hat{\underline{\theta}}_i, \quad i = 1, \dots, N \quad (15.109)$$

dargestellt. Unter der Annahme, dass jeder mögliche Wert von  $\underline{\theta}(k)$  durch genau ein identisches Modell  $\hat{\underline{\theta}}(k)$  repräsentiert wird, liefert die Wahl des besten Modells den wahren Wert des Parameters im letzten Zeitschritt:

$$\hat{\underline{\theta}}(k) = \underline{\theta}(k-1) \quad (15.110)$$

Eingesetzt in die um einen Zeitschritt verschobene Gleichung (15.108) ergibt sich die Forderung für den zeitdiskreten Regler mit Einstellzeit  $\delta = 1$  (*Deadbeat-Regler*<sup>7)</sup>)

$$y(k+1) = \underline{\phi}^T(k) \underline{\theta}(k-1) \stackrel{!}{=} r(k+1) \quad (15.111)$$

Dabei steht der Regelung wegen  $d = 1$  das um einen Zeitschritt verzögerte Modell  $\underline{\theta}(k-1)$  zur Verfügung. Wird  $u(k)$  aus dem Regressionsvektor isoliert, so ergibt sich das Regelgesetz

$$u(k) = \frac{1}{b_0(k-1)} \left( r(k+1) - \sum_{\nu=0}^{n-1} a(k-1)y(k-\nu) - \sum_{\nu=1}^{n-1} b_\nu(k-1)u(k-\nu) \right) \quad (15.112)$$

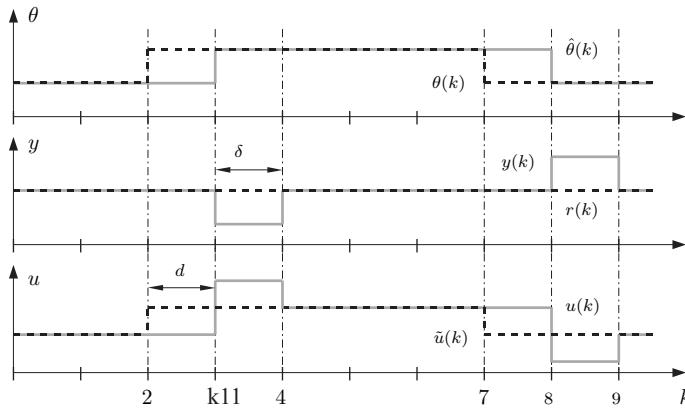
Die verspätete Identifikation von  $\theta$  führt zur Ausbildung eines inhärenten Regelfehlers, wie im Folgenden beschrieben. Den Zeitverläufen in Abbildung 15.13 liegt eine abgetastete Strecke mit  $d = \delta = 1$  und einem veränderlichen Parameter  $\theta(k)$  zu Grunde. Zum Zeitpunkt  $k = 2$  ändert sich  $\theta$ , was einen Zeitschritt später auf die Regelgröße wirkt. Zu diesem Zeitpunkt  $k = 3$  wird aus dem Ein-Ausgangs-Verhalten der Parameter  $\theta(2)$  identifiziert und in der Stellgröße  $u(3)$  berücksichtigt (durchgezogene Linie). Dies führt zum Verschwinden des Regelfehlers nach einem weiteren Schritt ( $k = 4$ ). Durch die verspätete Identifikation des Streckenmodells zum Zeitpunkt  $k = 2$  mit

$$\hat{\theta}(2) = \theta(1) \neq \theta(2) \quad (15.113)$$

entsteht also ein Regelfehler der Dauer  $d = 1$ . Gelingt es, bereits zum Zeitpunkt  $k = 2$  eine richtige Prognose  $\tilde{\theta}(2) = \theta(2)$  zu erhalten, kann der Regelfehler durch das richtige Regelgesetz  $\tilde{u}(k)$  (strichlierte Linie) verhindert werden. Die Überlegungen lassen sich entsprechend auf den Fall  $d \neq \delta \neq 1$  verallgemeinern.

---

<sup>7)</sup> Wegen der Inversion der Systemdynamik muss die Strecke für die Anwendung dieses Regelgesetzes minimalphasig sein.



**Abb. 15.13:** Parameter, Regelgröße und Stellgröße bei Regelung mit multiplen Modellen

### 15.4.2 Stochastische Parameterwechsel

Wie bereits gezeigt, lässt sich der inhärente Fehler dann vermeiden, wenn rechtzeitig der richtige Folgeparameter prognostiziert wird.

Finden die Parameterwechsel rein zufällig statt, resultieren sie etwa aus einem stochastischen (Rausch-)Prozess, dann kann keine Prognose gelingen, da der Prozess keine zeitliche Regularität aufweist. Für eine erfolgreiche Prognose sind somit eine gewisse Regelmäßigkeit der (stochastischen) Parameterwechsel und die Kenntnis der Zeitpunkte der Parameterwechsel Voraussetzung. Es werden also folgende Annahmen getroffen:

- Die Dynamik der Parameterwechsel gehorcht einer Markov-Kette mit unbekannter Transitionsmatrix

$$\mathbf{P}^\theta = \begin{bmatrix} p_{11}^\theta & \dots & p_{1N}^\theta \\ \vdots & \ddots & \vdots \\ p_{N1}^\theta & \dots & p_{NN}^\theta \end{bmatrix} \quad (15.114)$$

Die Transitionswahrscheinlichkeiten sind

$$p_{ij}^\theta = \Pr(\underline{\theta}[n+1] = \underline{\theta}_j \mid \underline{\theta}[n] = \underline{\theta}_i) \quad (15.115)$$

Mit  $[n]$  wird die übergeordnete Dynamik der Parameterwechsel dargestellt, gegenüber  $(k)$  für die (abgetastete) Dynamik der Strecke.

- Die Zeitpunkte  $k_n$ ,  $n = 0, 1, 2, \dots$  der Parameterwechsel sind bekannt. Es gilt für alle  $n$ :

$$\underline{\theta}(k) = \underline{\theta}[n], \quad k \in [k_n, k_{n+1}) \quad (15.116)$$

d. h. zwischen den bekannten Zeitpunkten  $k_n$  ändert sich der Parameter  $\underline{\theta}$  nicht.

Damit lässt sich durch richtige Vorhersage von  $\underline{\theta}[n]$  zum Zeitpunkt  $k_n$  der inhärente Regelfehler zwischen  $k_n + 1$  und  $k_n + 2$  eliminieren. Selbst eine falsche Vorhersage zum Zeitpunkt  $k_n$  verlängert nicht die Dauer des Regelfehlers, da bei  $k_n + 1$  der aktuelle Parametervektor identifiziert und das Regelgesetz richtig angepasst wird, was sich zum Zeitpunkt  $k_n + 2$  auswirkt. Ein auf Automaten basierender Prognosemechanismus für die Parameterwechsel wird im folgenden vorgestellt.

### 15.4.3 Erweiterte Regelungsstruktur

Abbildung 15.14 zeigt im Blockschaltbild die um eine Parameterprognose erweiterte Regelung mit multiplen Modellen. Mit Hilfe eines Automaten soll zu den Zeitpunkten  $k_n$  aus den durch die Parallelmodelle identifizierten Parametern eine Vorhersage des zur Zeit  $k_n$  gültigen Parametervektors erzeugt werden, mit der statt des identifizierten Parametervektors der Regler parametriert wird. Im folgenden Zeitschritt wird die Richtigkeit der Prognose durch Vergleich mit dem identifizierten Modell bewertet und daraufhin die Handlungsgrundlage des Automaten angepasst. Der Prognosemechanismus besteht aus folgenden Bausteinen:

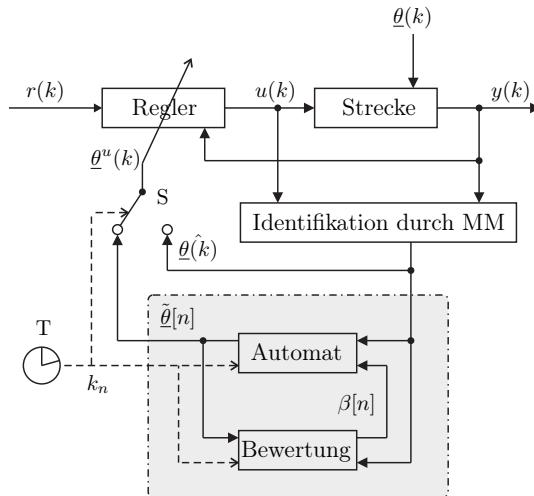
- Die Uhr  $T$  liefert die Zeitpunkte  $k_n$  der Wechsel des Parametervektors  $\underline{\theta}(k)$ .
- Durch den Schalter  $S$  wird der für die Regelung verwendete Parametervektor  $\underline{\theta}^u(k)$  gemäß der Vorschrift

$$\underline{\theta}^u(k) = \begin{cases} \hat{\underline{\theta}}(k), & k \neq k_n \\ \tilde{\underline{\theta}}(k_n), & k = k_n \end{cases} \quad (15.117)$$

umgeschaltet. Zur Zeit des Parameterwechsels erhält die Regelung die Prognose  $\tilde{\underline{\theta}}(k_n) =: \tilde{\underline{\theta}}[n]$  des Automaten, ansonsten mit  $\hat{\underline{\theta}}(k) = \underline{\theta}(k - 1)$  den zuletzt identifizierten Parametervektor. Dabei wird wie in 15.4.1 unterstellt, dass jedes Modell  $\hat{\underline{\theta}}_i$ ,  $i = 1, \dots, N$  genau einen möglichen Wert des Parametervektors  $\underline{\theta}_i$ ,  $i = 1, \dots, N$  repräsentiert.

- Der Automat erzeugt seine Prognose  $\tilde{\underline{\theta}}(k_n)$  auf Grund des zuletzt identifizierten Modells  $\hat{\underline{\theta}}(k_n) = \underline{\theta}(k_n - 1)$ . Dabei besteht der Automat selbst aus  $N$  Teilautomaten  $A_i$ ,  $i = 1, \dots, N$ , wobei jeder für die Vorhersage, ausgehend vom ihm zugeordneten identifizierten Modell  $\hat{\underline{\theta}}_i$ , zuständig ist. Jeder Teilautomat  $A_i$  prognostiziert also den Parameterwechsel

$$\underline{\theta}(k_n - 1) = \underline{\theta}_i \quad \rightarrow \quad \underline{\theta}(k_n) \quad (15.118)$$



**Abb. 15.14:** Um Prognosemechanismus erweiterte Regelung mit multiplen Modellen

- Die Bewertung überprüft die Richtigkeit der Prognose  $\hat{\theta}(k_n)$  zu den Zeitpunkten  $k_n + 1$  und erzeugt das Bestrafungssignal  $\beta(k_n + 1) =: \beta[n]$ . Dieses führt zur Anpassung des Vektors  $p_i[n + 1]$  der Handlungswahrscheinlichkeiten des stochastischen Automaten  $A_i$  bzw. zur Zustandstransition  $\psi_i[n] \rightarrow \psi_i[n + 1]$  im Fall eines deterministischen Automaten.

Da die Transitionen des Parametervektors  $\theta$  durch eine Markov-Kette beschrieben werden, erzielt der Prognosemechanismus das *optimale* Ergebnis, wenn jeder Teilautomat *immer die häufigste Transition* vom ihm zugeordneten Ausgangszustand vorhersagt. Durch die Zeilen der Transitionsmatrix  $\mathbf{P}^\theta$  ist für jeden Teilautomaten  $A_i$  eine stochastische Umgebung mit den Bestrafungswahrscheinlichkeiten

$$c_{ij} = 1 - p_{ij}^\theta, \quad j = 1, \dots, N \quad (15.119)$$

gegeben. Jeder optimale Teilautomat wird im Grenzwert die häufigste Transition mit der entsprechend niedrigsten Bestrafungswahrscheinlichkeit

$$c_{il} = \min\{c_{i1}, \dots, c_{iN}\} \quad (15.120)$$

vorhersagen. Praktisch reicht, wie im vorigen Abschnitt beschrieben, asymptotische oder  $\varepsilon$ -Optimalität bei der Auslegung der Teilautomaten aus, um der optimalen Prognose beliebig nahe zu kommen. Im folgenden Abschnitt wird die Regelgüte bei Anwendung des automatenbasierten Prognosemechanismus – gemessen an der Häufigkeit des auftretenden Regelfehlers – quantifiziert.

### 15.4.4 Quantifizierung der erreichten Regelgüte

Für die Markov-Kette der Modelltransitionen lässt sich aus

$$\underline{\pi}^\theta = (\mathbf{P}^\theta)^T \underline{\pi}^0 \quad (15.121)$$

die stationäre Verteilung  $\underline{\pi}^\theta$  angeben. Mit den Wahrscheinlichkeiten  $\pi_i^\theta$ ,  $i = 1, \dots, N$  befindet sich die Markov-Kette im Zustand  $i$  und der aktuelle Parametervektor ist  $\underline{\theta} = \underline{\theta}_i$ . Mit der Wahrscheinlichkeit  $\pi_i^\theta$  wird damit auch der  $i$ -te Teilautomat  $A_i$  des Prognosemechanismus aktiv, der in der durch  $\underline{c}_i = \{c_{i1}, \dots, c_{iN}\}$  gemäß Gl. (15.119) beschriebenen stochastischen Umgebung arbeitet.

Handelt es sich um  $\varepsilon$ -optimale Teilautomaten, dann ergibt sich als Grenzwert der zu erwartenden Bestrafung von  $A_i$

$$M_i = \lim_{n \rightarrow \infty} \Pr(\beta_i[n] = 1) = c_{il} + \varepsilon_i \quad (15.122)$$

$\varepsilon_i > 0$  lässt sich für jeden Automaten durch Wahl seiner Parameter (Lernschrittweite, Gedächtnistiefe) beliebig klein machen. Dann führt jeder Teilautomat mit einer Wahrscheinlichkeit beliebig nahe 1 die günstigste Aktion in Form der Prognose der jeweils häufigsten Transition von  $\underline{\theta}$  aus. Die Wahrscheinlichkeit einer falschen Vorhersage für den gesamten Prognosemechanismus und damit die Häufigkeit des inhärenten Fehlers ist dann

$$M = \lim_{n \rightarrow \infty} \Pr(\beta[n] = 1) = \sum_{i=1}^N \pi_i^\theta (c_{il} + \varepsilon_i) = \underbrace{\sum_{i=1}^N \pi_i^\theta c_{il}}_{=: M^{opt}} + \underbrace{\sum_{i=1}^N \pi_i^\theta \varepsilon_i}_{=: \varepsilon} \quad (15.123)$$

Bei  $M^{opt}$  handelt es sich um die untere Schranke der erreichbaren erwarteten Fehlerhäufigkeit. Durch geeignete Auslegung der Teilautomaten wird auch  $\varepsilon$  beliebig klein, so dass mit einer Wahrscheinlichkeit beliebig nahe 1 die jeweils häufigsten Modelltransitionen vorhergesagt werden. Mit  $\sum_{i=1}^N \pi_i^\theta = 1$  und  $c_{il} = 1 - \max_j p_{ij}^\theta$  kann man also formulieren:

**Satz 15.4.1** Bei der um einen Prognosemechanismus erweiterten Regelung mit multiplen Modellen lässt sich unter den getroffenen Annahmen und beim Einsatz  $\varepsilon$ -optimaler Teilautomaten die Häufigkeit des inhärenten Regelfehlers auf

$$M = 1 - \sum_{i=1}^N \pi_i^\theta \max_j p_{ij}^\theta + \varepsilon = M^{opt} + \varepsilon \quad (15.124)$$

reduzieren. Dabei sind  $\max_j p_{ij}^\theta$  die größten Wahrscheinlichkeiten für eine Modelltransition ausgehend von  $\underline{\theta}_i$  und  $\varepsilon > 0$  beliebig klein.

Im Vergleich zu  $M^{opt}$  lassen sich noch die erwartete Fehlerhäufigkeit bei rein zufälliger Prognose

$$M^{rand} = \sum_{i=1}^N \pi_i^\theta \sum_{j=1}^N \frac{1}{N} c_{ij} = \frac{1}{N} \sum_{i=1}^N \pi_i^\theta \sum_{j=1}^N (1 - p_{ij}^\theta) = 1 - \frac{1}{N} \quad (15.125)$$

und die erwartete Fehlerhäufigkeit ohne Prognose (es wird jeweils der aktuelle Parametervektor „vorhergesagt“)

$$M^{ohne} = \sum_{i=1}^N \pi_i^\theta c_{ii} = \sum_{i=1}^N \pi_i^\theta (1 - p_{ii}^\theta) = 1 - \sum_{i=1}^N \pi_i^\theta p_{ii}^\theta \quad (15.126)$$

berechnen.

**Beispiel 15.4.1** Für Parameterwechsel, die durch die Markov-Kette mit der Transitionsmatrix

$$\mathbf{P}^\theta = \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.1 & 0.2 & 0.7 \\ 0.9 & 0.05 & 0.05 \end{bmatrix} \quad (15.127)$$

beschrieben sind, ergeben sich folgende erwartete Fehlerhäufigkeiten:

$$M^{ohne} = 0.8915, \quad M^{rand} = \frac{2}{3}, \quad M^{opt} = 0.3058 \quad (15.128)$$

Die erwartete Häufigkeit des Regelfehlers lässt sich also durch Prognose der häufigsten Transition auf weniger als ein Drittel reduzieren.

### 15.4.5 Simulationsbeispiel

Als Beispielsystem wird die zeitdiskrete Strecke erster Ordnung

$$y(k+1) = 0.8y(k) + 0.2u(k) + \theta(k) \quad (15.129)$$

mit einem als Störung begriffenen zeitveränderlichen Parameter  $\theta(k) \in \{-1, 0, 1\}$  betrachtet. Die Parameterwechsel finden zu den Zeitpunkten  $k_n = 2n, n = 1, 2, \dots$  gemäß der Markov-Kette mit der Transitionsmatrix (15.127) statt. Es gilt:

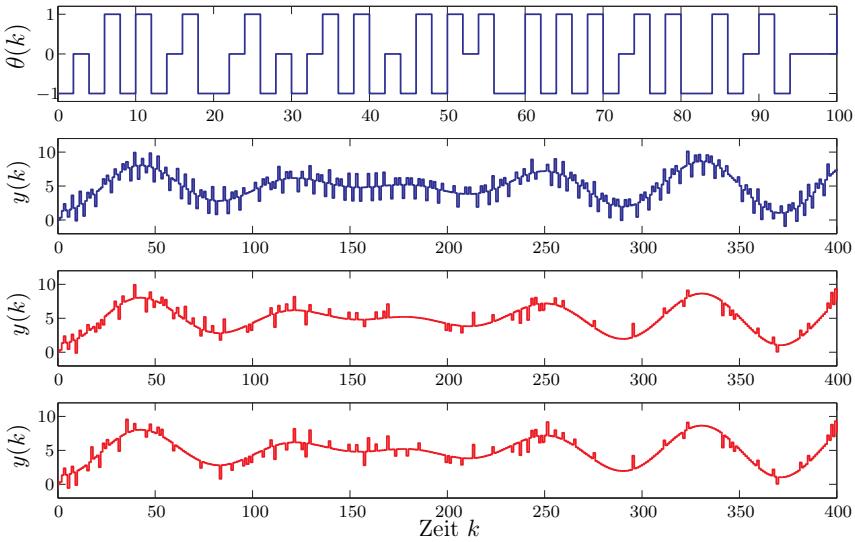
$$p_{ij}^\theta = \Pr(\theta[n] = \theta(k_n) = \theta_j \mid \theta[n-1] = \theta_i) \quad (15.130)$$

Jeder Wert des Parameters ist durch ein identisches Modell  $\hat{\theta}_i$  repräsentiert, welches die Prädiktion  $\hat{y}_i(k)$  erzeugt. Durch Vergleich mit  $y(k)$  erhält man den verzögert identifizierten Parameter  $\hat{\theta}(k) = \theta(k-1)$ . Das Regelgesetz für exaktes Folgeverhalten bei der um den Prognosemechanismus erweiterten Regelung lautet

$$u(k) = 5(r(k+1) - 0.8y(k) - \theta^u(k)), \quad (15.131)$$

mit

$$\theta^u(k) = \begin{cases} \hat{\theta}(k), & k = 2n - 1 \\ \tilde{\theta}[n] & k = 2n \end{cases} \quad n = 1, 2, \dots \quad (15.132)$$



**Abb. 15.15:** Zeitveränderlicher Parameter (Ausschnitt), Regelgröße ohne Prognose, sowie mit Prognose durch  $L_{R-I}$ -Algorithmus ( $a = 0.3$ ) und  $L_{3m,3}$ -Automaten ( $m = 3$ )

Abbildung 15.15 zeigt neben einem Ausschnitt des Störsignals (oben) die Verläufe der Regelgröße für die aus sin-Funktionen unterschiedlicher Frequenz zusammengesetzte Führungsgröße. Dabei steht die Regelung ohne Parameterprognose (2. Plot) der erweiterten Regelung mit multiplen Modellen gegenüber. Sowohl die Prognose durch stochastische Automaten (3. Plot,  $L_{R-I}$ -Algorithmus,  $a = 0.3$ ,  $p_i[0] = 1/3$ ), als auch durch deterministische Automaten (4. Plot,  $L_{3m,3}$ ,  $m = 3$ ), deren Startzustände in den Astwurzeln liegen, liefern schnell eine Verringerung der Fehlerhäufigkeit. Letztlich konvergieren die Automaten des Prognosemechanismus gegen die Vorhersage der häufigsten Transition.

Der vorgestellte Prognosemechanismus ist ein einfaches Beispiel für die Leistungsfähigkeit von Automaten. Auf Grundlage einfacher Adoptionsregeln entscheiden sie sich nach einem Lernprozess für die jeweils günstigste Aktion in einer unbekannten Umgebung, hier die bevorzugte Parametertransition. Die Einfachheit der Regeln, nach denen Automaten arbeiten, bewährt sich gerade dort, wo eine vollständige Modellierung des technischen (oder auch sozioökonomischen) Systems unmöglich ist und eine hohe Anzahl von unbekannten Einflussgrößen wirkt. Als Beispiele seien der Betrieb von Telekommunikationsnetzen, die Bildverarbeitung oder die Mustererkennung in Zeitreihen ökonomischer Kenngrößen genannt.

## 16 Hochverstärkungsbasierte Regelung

Dieses Kapitel behandelt die Zustandsregelung von nichtlinearen Strecken, deren Parameter unbekannt sind. Am Beispiel eines reibungsbehafteten — und damit nichtlinearen — Zweimassensystems wird ein zeitvarianter Zustandsregler entworfen, der gänzlich ohne Identifikation der Streckenparameter auskommt. Weder die linearen Parameter (Massenträgheitsmomente, Federhärte, Dämpfung) noch die nichtlinearen Parameter (Reibkennlinie) werden in irgendeiner Weise identifiziert oder gelernt. Selbes gilt für unbekannte Störgrößen wie Lastmomente, Messrauschen oder unmodellierte Verfälschung der Stellgröße durch den Aktor. Es kommt ein hochverstärkungsbasierter Regler zum Einsatz, der den Aufwand für die Implementierung aufwändiger und komplexer Lerngesetze vermeidet und dadurch eine sehr einfache Struktur erhalten kann. Neben dem Entwurf und der theoretischen Analyse eines solchen Reglers wird gezeigt, dass eine vorgegebene Fehlertoleranzgrenze eingehalten werden kann und eine aktive Bedämpfung von Oszillationen in der Antriebswelle erfolgt.

Die Ausführungen zur hochverstärkungsbasierten Regelung stützen sich auf die Dissertation [208] sowie auf die Veröffentlichung [108].

In der klassischen Regelungstheorie wurden bislang die Regelstrecken stets als hinreichend genau bekannt vorausgesetzt. Dies bedeutet, dass zum einen die Struktur der Strecke (also auch Ordnung und Relativgrad) und zum anderen die Werte der Streckenparameter (z.B. Verstärkungsfaktoren) exakt bekannt sind. Mit diesem detaillierten Wissen können Aussagen bezüglich des dynamischen Verhaltens der Regelstrecke getroffen werden, was eine geeignete Reglerauslegung vereinfacht. Allerdings ist die Voraussetzung von exakt bekannten Regelstrecken in der Praxis nur schwer zu erfüllen. Eine überwiegende Mehrheit von industriellen Prozessen kann mit erträglichem Aufwand nur näherungsweise modelliert werden. Dennoch ist auch in solchen Fällen eine präzise Regelung erforderlich. Aus dieser Notwendigkeit heraus sind verschiedene adaptive Regelungskonzepte entwickelt worden, u.a. auch die hochverstärkungsbasierte Regelung (engl. high-gain-feedback ).

Wegen ihrer bestechenden Einfachheit ist die hochverstärkungsbasierte Regelung eine elegante Methode, um eine Stabilisierung von unbekannten Regelstrecken zu erreichen. Hierfür wird im einfachsten Fall die Verstärkung eines Proportionalreglers so lange erhöht, wie eine Regeldifferenz besteht. Dazu muss keine Identifikation der Regelstrecke stattfinden, weswegen auch kein Prozessmodell

erforderlich ist. Deshalb lässt sich mit diesem Regelungskonzept ein schnelleres Einschwingen erzielen als bei herkömmlichen Optimierungsverfahren. Des Weiteren muss keine störende Anregung vorhanden sein, um die PE-Bedingung (persistence of excitation) zu erfüllen, damit Parameterkonvergenz auf die korrekten Werte garantiert werden kann. Allerdings ist der Anwendungsbereich von hochverstärkungsbasierten Reglern durch harte Anforderungen an die Regelstrecke sehr eingeschränkt. Global stabiles Regelkreisverhalten kann nur für einen „high-gain-fähigen“ Prozess erreicht werden, welcher

1. einen Relativgrad  $\delta$  von Eins besitzt,
2. minimalphasig ist und
3. dessen Vorzeichen der instantanen Verstärkung bekannt ist.

Voraussetzung (3) schränkt die regelbare Systemklasse nur geringfügig ein und kann meist als erfüllt betrachtet werden. Nachdem nur das Vorzeichen entscheidend ist, wird nur Kenntnis über die Wirkungsrichtung der Stellgröße gefordert. Für eine Gleichstrommaschine bedeutet dies zum Beispiel, dass die Drehrichtung der Antriebswelle bekannt ist, wenn ein positiver Strom in den Ankerkreis des Motors eingeprägt wird. Das Vorzeichen der instantanen Verstärkung lässt sich in physikalischen Systemen oftmals angeben, ohne auf die exakten Parameter der Strecke zurückgreifen zu müssen.

Die vorausgesetzte Minimalphasigkeit stellt zwar eine gewisse Einschränkung dar, liegt jedoch bei den meisten mechatronischen Regelstrecken vor.

Das hauptsächliche Hindernis für die Anwendung der hochverstärkungsbasierten Regelung ist die Annahme, dass der Relativgrad Eins ist. Bereits durch die Berücksichtigung einer elastischen Welle bei einem Zweimassensystem wird Forderung (1) nicht mehr erfüllt. Verschärft wird dieses Problem, wenn zusätzlich ein Umrichter als Stellglied in das Prozessmodell integriert wird. Aus diesem Grund ist das Konzept der hochverstärkungsbasierten Regelung im Umfeld elektrischer Antriebe in der dargestellten Form nicht einsetzbar. Um dieses attraktive Verfahren dennoch für die Antriebstechnik nutzbar zu machen, wurde die Arbeit [208] durchgeführt, in der mittels Zustandsrückkopplung eine realisierbare Reduktion des Relativgrades unter Einhaltung der Minimalphasigkeit erreicht wird.

## 16.1 Grundidee der hochverstärkungsbasierten Regelung

Die Grundidee des hochverstärkungsbasierten Konzeptes soll anhand eines linearen Systems erster Ordnung erklärt werden. Diese Regelstrecke hat PT<sub>1</sub>-Verhalten und wird durch folgende Differentialgleichung beschrieben:

$$\begin{aligned}\dot{x}(t) &= a x(t) + u(t), & x(0) \in \mathbb{R} \\ y(t) &= x(t)\end{aligned}\tag{16.1}$$

Das Signal  $u(t)$  ist der Steuereingang,  $x(t)$  ist der Zustand des Systems und  $a \in \mathbb{R}$  der Eigenwert der Differentialgleichung,  $y(t)$  bezeichnet den Ausgang des Systems. Für den Reglerentwurf ist der Eigenwert  $a$  gänzlich unbekannt, also auch dessen Vorzeichen. Damit ist auch unklar, ob die Strecke stabil ist. Mit Hilfe eines einfachen P-Reglers

$$u(t) = -k y(t) \quad (16.2)$$

soll die Differentialgleichung stabilisiert werden. Daraus resultiert die Beschreibung

$$\dot{x}(t) = (a - k) \cdot x(t) \quad (16.3)$$

für den geschlossenen Kreis. Wenn eine konstante Reglerverstärkung  $k \in \mathbb{R}$  so gewählt wird, dass  $a - k < 0$  gilt, dann ist der Regelkreis stabil und dessen Eigenwert liegt in der linken Halbebene der Laplace-Ebene. Da der Eigenwert  $a$  der Regelstrecke aber nicht bekannt ist, kann kein geeigneter Wert für die Reglerverstärkung  $k$  explizit angegeben werden. Allerdings verschiebt eine Erhöhung von  $k$  den Eigenwert des Regelkreises  $a - k$  in Richtung der linken Halbebene, d.h. in Richtung des stabilen Gebietes. Hier greift das Prinzip der hochverstärkungsbasierten Regelung an: es wird ein Adoptionsgesetz gewählt, das die Reglerverstärkung  $k$  solange erhöht, also das geregelte System immer „stabil“ macht, bis der Ausgang der Regelstrecke seine Gleichgewichtslage  $y = 0$  erreicht. Ein solches Adoptionsgesetz lautet:

$$\dot{k}(t) = y(t)^2, \quad k(0) = k_0 \in \mathbb{R} \quad (16.4)$$

Als Folge ergibt sich ein adaptiver Stabilisator für den Prozess (16.1). Das Adoptionsgesetz (16.4) verschiebt den Eigenwert des geregelten Systems entlang der reellen Achse in die stabile linke Halbebene. Der Beweis, dass die Regelstrecke hierdurch stabilisiert werden kann, und dass die monoton wachsende Verstärkung  $k(t)$  auf einen endlichen Wert konvergiert, wird hier nicht wiedergegeben und ist in [208, S. 65ff] und in [99] zu finden. Für Regelstrecken höherer Ordnung ist die selbe Vorgehensweise möglich, so dass sich allgemein der Stabilisator (16.2), (16.4) auf lineare Strecken beliebiger Ordnung verallgemeinern lässt [208, S. 71ff], [101, S. 23ff]. Nachdem keine Identifikation stattfindet, ist eine hohe Robustheit gegenüber diversen Störeinflüssen erzielbar.

Die beschriebene Methode wurde ursprünglich entwickelt zur Stabilisation unbekannter Prozesse. Für die Regelung auf beliebige Sollwerte ist eine Modifikation nötig. Anstelle der Ausgangsgröße  $y(t)$  ist sowohl im Regelgesetz (16.2), als auch im Adoptionsgesetz (16.4) der Regelfehler  $e = y^* - y = x_d$  einzusetzen, mit dem Sollwert  $y^*$  und der Regeldifferenz  $x_d$ . Sofern die Strecke global integrales Verhalten aufweist, entstehen durch eine solche Abänderung keine Schwierigkeiten. Falls jedoch global proportionales Verhalten vorliegt, führt die Modifikation zum unbeschränkten Anwachsen der Verstärkung. Eine mögliche Abhilfe für dieses Problem ist der Einsatz eines  $\lambda$ -Trackers, wie beispielsweise in [26] oder [101] beschrieben.

## 16.2 Auswirkung großer Verstärkungen im Regelkreis

Der Regler

$$u(t) = -k(t)y(t), \quad \dot{k}(t) = y(t)^2 \quad (16.5)$$

gründet sich auf einem adaptiven Regelungskonzept, das wegen  $\dot{k}(t) = y(t)^2 \geq 0$  eine monoton anwachsende Verstärkungsfunktion generiert. Angewendet auf Regelstrecken mit Relativgrad  $\delta = 1$  wird die steigende Verstärkung den Regelkreis immer weiter stabilisieren, bis ein weiteres Ansteigen nicht mehr notwendig ist. In diesem Abschnitt soll nun gezeigt werden, dass für hinreichend große Verstärkung alle Regelkreise sich dem Verhalten eines inversen Reglers annähern. Dazu wird eine Strecke höherer Ordnung betrachtet, welche jedoch einen relativen Grad von eins besitzt. Die Übertragungsfunktion der zu untersuchenden Strecke lautet:

$$G_s(s) = \frac{B(s)}{A(s)} \quad (16.6)$$

Die beiden Polynome  $A(s)$  und  $B(s)$  sind vom Grad  $n$  bzw.  $m$  und unterscheiden sich somit in ihrer Ordnung, wobei  $A(s)$  eine Ordnung höher ist als  $B(s)$ , d.h.

$$n = \deg(A(s)) = \deg(B(s)) + 1 \quad (16.7)$$

Der Regelkreis wird durch einen Proportionalregler im Vorwärtzweig mit der konstanten Verstärkung  $k$  geschlossen. Dadurch ergibt sich für den Regelkreis eine Übertragungsfunktion von

$$G_{RK}(s) = \frac{kB(s)}{A(s) + kB(s)} = \frac{1}{\frac{1}{k} \frac{A(s)}{B(s)} + 1} \quad (16.8)$$

Im Zusammenhang mit konstanten, hohen Verstärkungen ist der Grenzwert dieser Übertragungsfunktion für  $k \rightarrow \infty$  interessant. Für dessen Berechnung eignet sich eine Systemdarstellung gemäß Kap. 12.3 (Byrnes-Isidori Form, BINF), die sich aus der Übertragungsfunktion (16.6) durch eine Polynomdivision ergibt. Um die Herleitung allgemeingültig zu halten, seien hier die beiden Polynome

$$A(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0 \quad (16.9)$$

und

$$B(s) = s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_1s + b_0 \quad (16.10)$$

angesetzt. Weil der Relativgrad per Voraussetzung gleich eins ist, gilt  $m = n - 1$ , d.h. der Grad des Zählerpolynoms ist um eins kleiner, als die Systemordnung. Im hier betrachteten Fall ergibt daher die Division ein Polynom  $\gamma(s)$  vom Grad  $\delta = 1$  und ein Restpolynom  $\rho(s)$  vom Grad  $m - 1 = n - 2$ , das durch  $B(s)$  zu teilen ist:

$$\frac{A(s)}{B(s)} = \frac{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0}{s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_1s + b_0} = s + (a_{n-1} - b_{n-2}) + \frac{\rho(s)}{B(s)} \quad (16.11)$$

Mit der Gewichtung  $1/k$  lautet das Ergebnis der Polynomdivision:

$$\frac{1}{k} \frac{A(s)}{B(s)} = \frac{1}{k} s + \frac{1}{k} (a_{n-1} - b_{n-2}) + \frac{1}{k} \frac{\rho(s)}{B(s)} \quad (16.12)$$

Trotz der Multiplikation mit dem Faktor  $1/k$ , darf nicht allein von einem unendlich großen  $k$  auf das Verschwinden des gesamten Quotienten geschlossen werden. Der Grund hierfür liegt in der Tatsache, dass für sehr große Frequenzen ( $s \rightarrow \infty$ ) der erste Summand sowohl einen unendlich großen Zähler, als auch einen unendlich großen Nenner erhält. Der dritte Summand weist im Nenner das unbestimmte Produkt aus Null und Unendlich auf, wenn  $s$  eine Wurzel von  $B(s)$  ist. Daher ist eine tiefergehende Analyse erforderlich:

- Der Bruch  $\rho(s)/B(s)$  ist der Quotient aus einem Polynom der Ordnung  $m-1$  und einem Polynom der Ordnung  $m$ . Folglich strebt dieser für wachsendes  $s \rightarrow \infty$  gegen Null. Der Bruch bleibt beschränkt, solange  $s$  nicht Wurzel des Nennerpolynomes ist, d.h. solange für  $s$  nicht die Nullstellen der Regelstrecke eingesetzt werden. Durch die Annahme einer minimalphasigen Regelstrecke ist garantiert, dass die Übertragungsnullstellen einen negativen Realteil besitzen und somit nicht auf der imaginären Achse liegen können. Nachdem der Frequenzgang entlang der imaginären Achse bestimmt wird und nur technische Frequenzen ( $s = j\omega$ ) eingesetzt werden, kann demzufolge das Polynom  $B(s)$  nicht verschwinden, so dass der betreffende Bruch beschränkt bleibt<sup>1)</sup>. Die ungewollte Konstellation, nämlich das Produkt von Null und Unendlich, tritt nicht auf, ein unbestimmter Grenzwert wird dadurch ausgeschlossen. Wegen der Gewichtung mit  $\lim_{k \rightarrow \infty} 1/k = 0$  verschwindet der Bruch.
- Der zweite Summand  $(a_{n-1} - b_{n-2})/k$  hängt nicht von  $s$  ab und wird für steigendes  $k$  vernachlässigbar.
- Über den Term  $s/k$  kann zunächst keine Aussage getroffen werden, dieser stellt den (unbestimmten) Grenzwert dar.

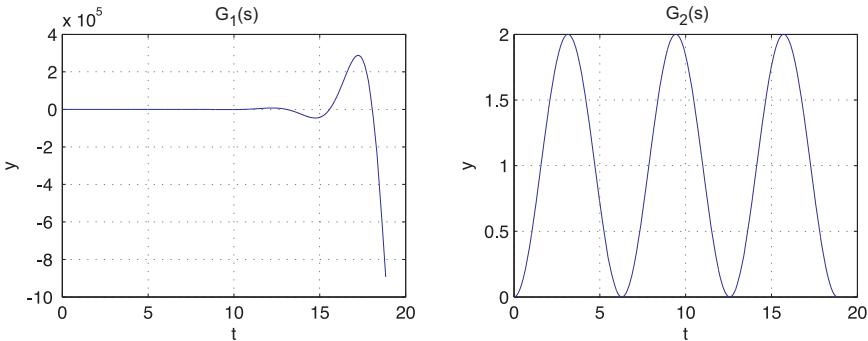
Aus dieser Untersuchung folgt, dass damit einzig noch der erste Summand verbleibt. Somit nähert sich der zu untersuchende Term in der Übertragungsfunktion des geschlossenen Regelkreises an

$$\lim_{k \rightarrow \infty} \frac{1}{k} \frac{A(s)}{B(s)} = \lim_{k \rightarrow \infty} \frac{1}{k} s \quad (16.13)$$

an. Mit der Einführung einer Zeitkonstanten  $T := 1/k$  ergibt sich daher für den geschlossenen Regelkreis das Verhalten eines PT<sub>1</sub>-Gliedes

---

<sup>1)</sup> Der Bruch bleibt auch dann beschränkt, wenn  $B(s)$  Wurzeln in der rechten Halbebene besitzt, also wenn die Bedingung der Minimalphasigkeit verletzt ist. Dann allerdings tritt im Regelkreis eine instabile Pol-Nullstellen-Kompensation ein, wie im folgenden Abschnitt dargestellt wird.



**Abb. 16.1:** Sinusantwort der ungeregelten Strecken (16.17) und (16.18).

$$G_{RK}(s) = \frac{1}{\frac{1}{k}s + 1} = \frac{1}{Ts + 1} \quad (16.14)$$

wobei dessen Zeitkonstante  $T$  mit wachsendem  $k$  kleiner wird. Daher wird die Dynamik dieses Systems durch eine steigende Verstärkung beschleunigt, im Grenzfall  $k \rightarrow \infty$  verschwindet die Zeitkonstante  $T$

$$\lim_{k \rightarrow \infty} T = \lim_{k \rightarrow \infty} \frac{1}{k} = 0 \quad (16.15)$$

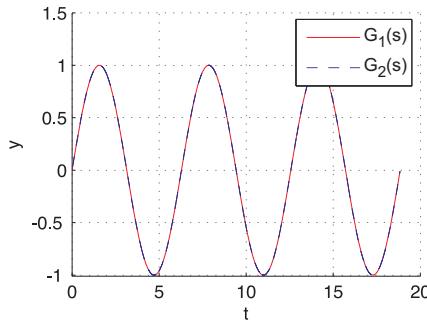
vollständig. Daher resultiert die Übertragungsfunktion im Grenzfall  $k \rightarrow \infty$  in einem unendlich schnellen PT<sub>1</sub>-Glied, was einer konstanten Verstärkung

$$G_{RK}(s)|_{k \rightarrow \infty} = \frac{1}{0s + 1} = 1 \quad (16.16)$$

gleicht. Weil eine konstante Übertragungsfunktion mit dem Faktor Eins exakt derjenigen des inversen Reglers entspricht, sind folglich Sollwert und Istwert identisch.

Dieses Ergebnis überrascht nicht, wenn die zugehörigen Wurzelortskurven betrachtet werden: da für anwachsende Kreisverstärkung jede Nullstelle gemäß ihrer Vielfachheit entsprechend viele Pole anzieht, entstehen für hinreichend hohe Verstärkungen Pol-Nullstellen-Kompensationen. Wegen dem Relativgrad  $\delta$  von Eins, bleibt genau eine Polstelle übrig, welche nicht in eine Nullstelle, sondern gegen  $-\infty$  strebt. Daran ist der Grenzfall des unendlich schnellen PT<sub>1</sub>-Gliedes ersichtlich, in dem alle hochverstärkungsbasierten Regelkreise münden.

Mittels Simulation soll bestätigt werden, dass alle high-gain-fähigen Regelstrecken für hohe Kreisverstärkungen einem PT<sub>1</sub>-Glied ähneln und damit stets



**Abb. 16.2:** Sinusantwort der geregelten Strecken (16.17) und (16.18) für  $k = 10^4$ .

die gleiche Dynamik erhalten. Beispielhaft sei dies für die beiden linearen Systeme

$$G_1(s) = \frac{s^4 + 10s^3 + 35s^2 + 50s + 34}{s^5 + 1s^4 + 20s^3 + 2s^2 + 4s + 60} \quad (16.17)$$

und

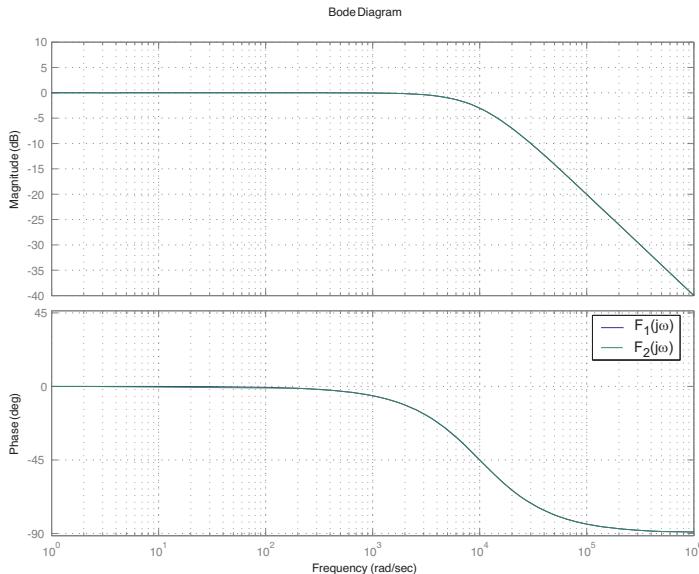
$$G_2(s) = \frac{1}{s} \quad (16.18)$$

betrachtet. Die Antwort auf eine sinoidale Anregung beider Strecken ist in Abbildung 16.1 gezeigt. Deutlich zu erkennen ist der hauptsächliche Unterschied, nämlich die Tatsache, dass die Übertragungsfunktion (16.17) eine instabile Strecke beschreibt.

Die beiden Strecken (16.17) und (16.18) werden nun durch einen einfachen P-Regler mit der Verstärkung  $k = 10000$  geregelt. Wie Abbildung 16.2 zeigt, sind bei identischer Anregung die Ausgänge der beiden Systeme fast gleich. Diese Eigenschaft kann auch anhand des Bodediagrammes eingesehen werden. In Abbildung 16.3 ist kaum ein Unterschied zwischen den beiden geregelten Systemen erkennbar, obwohl die Strecken selbst durchaus unterschiedliche Eigenschaften aufweisen, wie aus den Abbildungen 16.1 und 16.4 hervorgeht.

Eine weitere Erhöhung der Verstärkung bewirkt eine Verschiebung der Eckfrequenz zu höheren Frequenzen und eine noch stärkere Annäherung der beiden Regelkreise für niedrige Frequenzen. Dies ist aus Abbildung 16.5 ersichtlich. Diese Ergebnisse legen den Schluss nahe, dass durch Erhöhung der Reglerverstärkung  $k$  der Regelkreis einem PT<sub>1</sub>-System immer ähnlicher wird, wobei die Zeitkonstante des PT<sub>1</sub>-Systems mit steigender Verstärkung abnimmt. Damit wird die Regelstrecke durch große Verstärkungen dynamisch immer schneller.

Rechnerisch lässt sich diese Aussage sehr einfach für die Regelstrecke (16.18) zeigen. Wird deren Ausgang durch einen P-Regler mit der Übertragungsfunktion  $G_R(s) = k$  zurückgeführt, ergibt sich für den Regelkreis die Übertragungsfunktion:

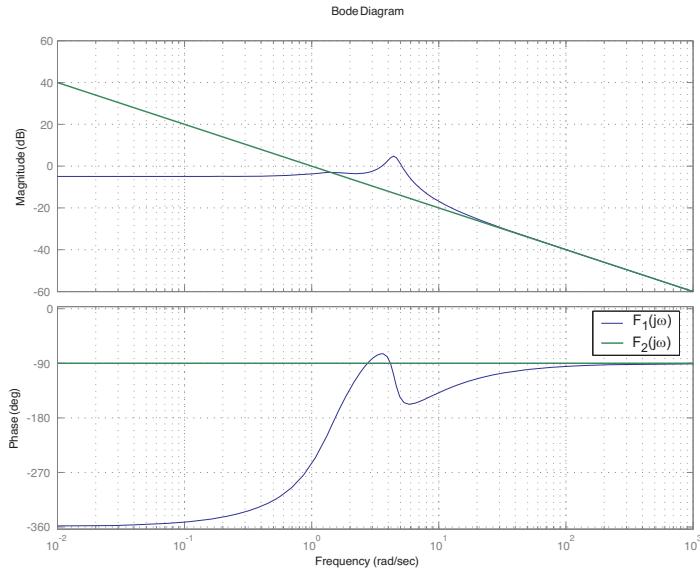


**Abb. 16.3:** Bodediagramm für die geregelten Strecken (16.17) und (16.18) für  $k = 10^4$ .

$$G(s) = \frac{k \frac{1}{s}}{1 + k \frac{1}{s}} = \frac{1}{\frac{1}{k}s + 1} \quad (16.19)$$

Dies ist die Beschreibung eines PT<sub>1</sub>-Systems mit der stationären Verstärkung  $K = 1$  und der Zeitkonstanten  $T = 1/k$ . Die Eckfrequenz  $\omega_E$  ist umgekehrt proportional zur Zeitkonstanten und ergibt sich daher zu  $\omega_E = 1/T = k$ . Damit erhöht sich die Bandbreite des Regelkreises mit steigender Verstärkung  $k$ .

Damit ist sowohl simulativ als auch rechnerisch gezeigt, was auch durch eine Argumentation mit der Wurzelortskurve erklärt werden kann: Jede der  $m = n - \delta$  Nullstellen zieht für  $k \rightarrow \infty$  einen Pol an, dies ruft eine  $m$ -fache Pol-Nullstellen-Kürzung in der Übertragungsfunktion des geschlossenen Kreises hervor. Damit sind die zugehörigen Zustände nicht mehr steuerbar bzw. beobachtbar und treten im Ein-Ausgangs-Verhalten nicht mehr in Erscheinung. Der übrige Pol wird nicht gekürzt und wandert gegen Unendlich. Wegen der Symmetrieeigenschaften der WOK muss dieser Pol auf der reellen Achse liegen, weswegen sich das geregelte System einem PT<sub>1</sub>- Übertragungsverhalten annähert. Für die Strecke  $G_1(s)$  in Gleichung (16.17) wird dieser Sachverhalt graphisch durch die WOK in Abbildung 16.6 dargestellt.



**Abb. 16.4:** Bodediagramm für die ungeregelten Strecken (16.17) und (16.18).

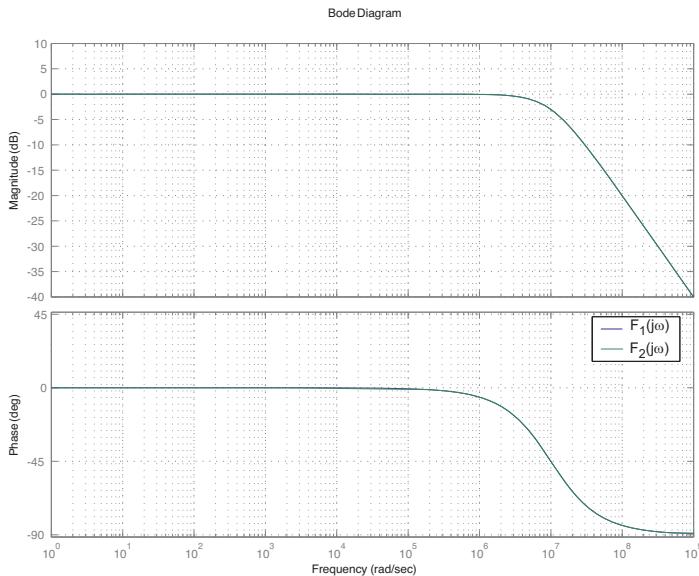
### 16.3 Empfindlichkeit gegenüber hohen Relativgraden

Die vorangegangenen Ausführungen beziehen sich auf Strecken mit Relativgrad eins. Im Falle einer linearen Strecke darf die Anzahl der Pole damit nur um eins größer sein als die Anzahl der Nullstellen. Diese Voraussetzung stellt eine harte Anforderung an die Regelstrecke dar und engt die regelbare Systemklasse beträchtlich ein. Beispielsweise im Hinblick auf die Regelung von Zweimassensystemen ist der Relativgrad hinderlich. Für eine zugängliche Darstellung des Relativgradproblems beschränkt sich die Argumentation hier auf lineare Regelstrecken. Im Falle nichtlinearer Strecken versagt die verwendete Argumentationsweise, die sich auf Übertragungsfunktionen und Wurzelortskurven stützt. Dennoch ist auch bei nichtlinearen Strecken das Relativgradproblem in analoger Weise zu finden.

Als Beispiel sei hier eine Strecke mit der Zustandsbeschreibung

$$\begin{aligned}\dot{\underline{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \cdot \underline{x}(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot u(t), \quad \underline{x}(0) \in \mathbb{R}^3 \\ y(t) &= (3 \ 1 \ 0) \cdot \underline{x}(t)\end{aligned}\quad (16.20)$$

zu Grunde gelegt, die das charakteristische Verhalten des linearen Zweimassen- system aus Kapitel 2 beschreibt. Dabei ist zu beachten, dass in Kapitel 2 die

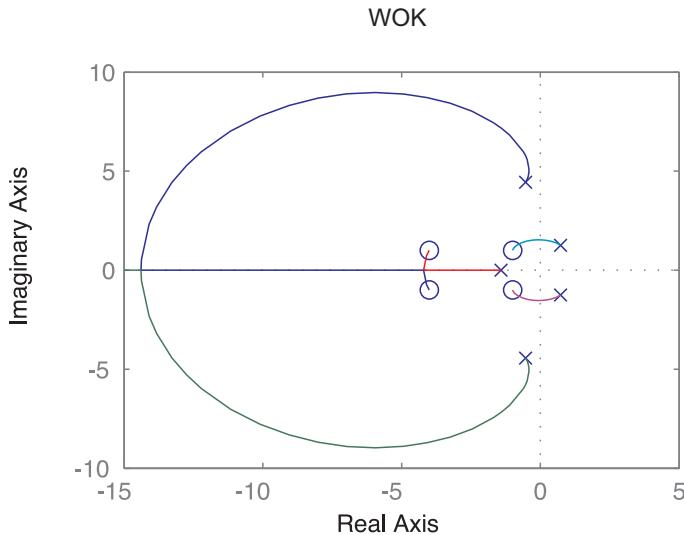


**Abb. 16.5:** Bodediagramm für die geregelten Strecken (16.17) und (16.18) für  $k = 10^7$ .

physikalische Zustandsrepräsentation angegeben ist, hier jedoch die Darstellung in Regelungsnormalform gewählt wurde. Beide Beschreibungen können durch eine Ähnlichkeitstransformation ineinander überführt werden.

Das lineare Differentialgleichungssystem (16.20) hat drei Zustände ( $n = 3$ ) und ist so parametriert, dass die drei Eigenwerte/Pole bei  $s = 0$  und bei  $s = -1 \pm j$  liegen. Zusätzlich ist eine ( $m = 1$ ) Nullstelle bei  $s = -3$  vorhanden. Daraus resultiert ein Relativgrad ( $\delta = n - m = 3 - 1 = 2$ ) von zwei, weswegen die Strecke nicht high-gain-fähig ist. Die Wurzelortskurve WOK dieser Strecke ist in Abbildung 16.7 aufgetragen. Es ist sofort ersichtlich, dass ab der Verstärkung  $k > 4$  das konjugiert komplexe Polpaar in die instabile rechte Halbebene eintritt. Erhöht man die Verstärkung der Rückführung, entsteht dadurch ein instabiler Regelkreis. Damit kann in diesem Fall die Verstärkung nicht beliebig erhöht werden, und der Grundgedanke, dass eine Vergrößerung der Rückführverstärkung den Regelkreis stabiler macht, trifft nicht zu. Der einfache Grund hierfür ist der zu hohe Relativgrad. Wie von der Theorie der Wurzelortskurven her bekannt ist, zieht jede Nullstelle für steigende Kreisverstärkung entsprechend ihrer Vielfachheit Pole an. Daraus entstehen zwei Konsequenzen:

1. Es darf höchstens *ein* Pol existieren, der nicht von einer Nullstelle angezogen wird, da dieser entlang der reellen Achse in die stabile Halbebene hineinläuft. Im Beispiel (16.20) ist das wegen einem Relativgrad  $\delta = 2$



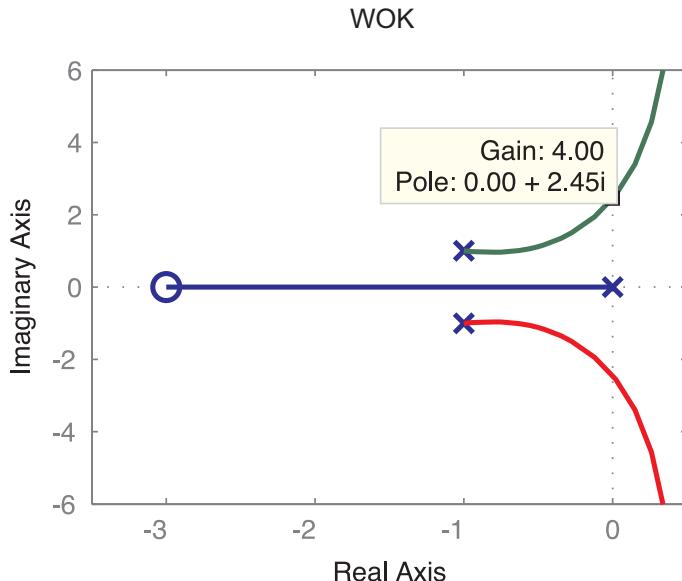
**Abb. 16.6:** WOK für die Strecke (16.17).

nicht erfüllt.

2. Weiterhin müssen notwendigerweise alle Nullstellen in der stabilen Halbebene liegen, da ansonsten Pole in die instabile Halbebene gelangen könnten. Dies ist die Erklärung für die geforderte Minimalphasigkeit.

Zu beachten ist bei dieser Argumentation jedoch, dass es sich wegen der Adaption um ein *zeitvariantes* Gesamtsystem handelt. Solange eine Adaption der Verstärkung stattfindet, ist die Position der Pole völlig belanglos, da die Pole ihre Bedeutung hinsichtlich einer Stabilitätsbeurteilung verlieren. Zeitvariante Systeme können durchaus instabil sein, obwohl ihre Pole stets in der linken Halbebene verbleiben. Allerdings muss aus praktischen Gesichtspunkten die Rückführverstärkung konvergieren. Die Wahl des Adaptionsgesetzes (16.4) verursacht eine monoton wachsende (bzw. nicht fallende) Funktion  $k(t)$ . Weil die Verstärkung nicht abnehmen kann, muss Konvergenz gewährleistet sein, um ein unbeschränktes Anwachsen zu verhindern. Die konvergierende Verstärkung führt schließlich auf ein zeitinvariantes System, dessen Pole zur Beurteilung der Stabilität wieder herangezogen werden können. Aus diesem Grund muss die Wurzelortskurve die Eigenschaft aufweisen, dass sich ab einer (unbekannten) Mindestverstärkung alle Pole in der linken Halbebene befinden. Dies ist genau dann gegeben, wenn die Voraussetzungen

1. Relativgrad  $\delta$  Eins,



**Abb. 16.7:** Wurzelortskurve für die Regelstrecke (16.20)

2. Minimalphasigkeit und
3. bekanntes Vorzeichen der instantanen Verstärkung

erfüllt sind. Der Beweis, dass der Adoptionsvorgang selbst stabil ist, kann beispielsweise mit der direkten Methode nach Lyapunov geführt werden.

## 16.4 Funnel-Control

Das Adoptionsgesetz in den Gleichungen (16.4) bzw. (16.5) weist den gravierenden Nachteil einer monotonen, nicht abnehmenden Verstärkung auf, so dass dieses Konzept in der Praxis wegen vorhandenem Messrauschen und Störungen nicht anwendbar ist. Weil jeder beliebig kleine Regelfehler die Verstärkung erhöht, ist eine Konvergenz der Verstärkung praktisch nicht möglich und  $k(t)$  driftet gegen Unendlich. Als Abhilfe kann hier Funnel-Control (engl. Schornstein-Regler) eingesetzt werden, das als Variante des beschriebenen hochverstärkungsbasierten Ansatzes auf der gleichen Grundidee basiert, jedoch eine Reduktion der Verstärkung zulässt. Daher ist Funnel-Control für den Einsatz in der Praxis besser geeignet. Darüberhinaus ist vielfach eine Verringerung der Verstärkung sinnvoll bzw. notwendig, wenn sich beispielsweise Parameter in der Regelstrecke ändern oder unbekannte, zeitvariante Störgrößen auftreten. Auch hier ist

Funnel-Control überlegen, da im Gegensatz zum etablierten High-Gain-Konzept und zum aufwändigen Identifikationsansatz nicht nur nichtlineare, sondern auch gestörte Strecken mit zeitvarianten Parametern zulässig sind. Ein signifikanter Vorteil von Funnel-Control besteht jedoch in der Eigenschaft, dass die Verstärkung niedrige Werte aufweisen kann (sofern dies die aktuelle Situation erlaubt) und ausschließlich dann hohe Werte erreicht, wenn sich der Regelfehler einer vorgegebenen Grenze nähert.

Zusätzlich besitzt dieses Konzept den bemerkenswerten Vorteil, dass nicht nur asymptotische (also für  $t \rightarrow \infty$ ) Aussagen über den Regelfehler möglich sind, sondern darüberhinaus eine Schranke für den Regelfehler festgelegt werden kann und zu jedem Zeitpunkt eingehalten wird. Erstmals vorgestellt wurde Funnel-Control in [107] von Ilchmann, Ryan und Sangwin.

Das Charakteristikum von Funnel-Control ist, dass neben dem Sollwertverlauf  $y^*(t)$  gleichzeitig ein Grenzwert für den maximal erlaubten Regelfehler  $e(t) = y^*(t) - y(t) = x_d$  festgelegt wird. Um den Fehlerverlauf zu beschränken, wird eine strikt positive (d.h. von Null wegbeschränkte Funktion), stetige und beschränkte Funktion  $\partial\mathcal{F}(t)$  gewählt, die der Vorgabe

$$\forall t \geq 0 : 0 < \mu \leq \partial\mathcal{F}(t) \leq \Gamma < \infty \quad (16.21)$$

genügt. Üblicherweise ist in realen, regelungstechnischen Aufgabenstellungen zunächst ein gewisser Regelfehler zulässig, der im weiteren Verlauf durch geeignete Aktionen des Reglers zu verringern ist. Diese Tatsache schlägt sich direkt in der Spezifikation für den Regelkreis nieder, die damit bereits eine Schranke für den Regelfehler festlegt. Gewöhnlich besitzt der maximal zulässige Regelfehler als Funktion über der Zeitachse eine trichterförmige Gestalt, die für die Namensgebung des Regelkonzeptes verantwortlich ist. Im Folgenden sei vorausgesetzt, dass eine Spezifikation des Regelkreises vorliegt, aus der sich eine geeignete Trichterrandfunktion  $\partial\mathcal{F}(t)$  direkt ableiten lässt. Exemplarisch ist eine mögliche Trichterrandfunktion in Abbildung 16.8 gezeigt.

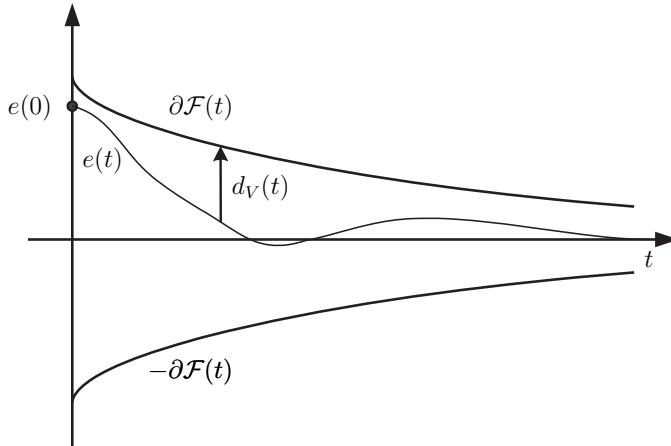
Die Anforderungen gängiger Anwendungsfälle werden erfüllt, wenn als Trichterrand die Funktion

$$\partial\mathcal{F}(t) = a \cdot e^{-\lambda t} + \mu, \quad a, \mu > 0, \quad \lambda \geq 0 \quad (16.22)$$

gewählt wird. Dabei stellen die Parameter  $\mu$  und  $\lambda$  den maximalen Fehler für  $t \rightarrow \infty$  bzw. die Abklingrate des Fehlers ein. Durch  $a$  und  $\lambda$  kann daher das transiente Verhalten vorgeschrieben werden.

Sinn und Zweck der Regelstrategie ist, dem Fehlerverlauf durch eine vorab festzulegende Funktion eine Schranke vorzugeben. Essentiell hierfür ist, dass zum Startzeitpunkt  $t = 0$  der Fehler  $e(0)$  bereits innerhalb des Trichters liegt.

Mit Bezug auf den in Gleichung (16.22) definierten Trichterrand muss die Konstante  $a$  in solcher Weise eingestellt werden, dass  $|e(0)| < \partial\mathcal{F}(0) = a + \mu$



**Abb. 16.8:** Exemplarischer Trichterrand und Fehlersignal.

erfüllt ist. Falls durch diese zusätzliche Forderung Schwierigkeiten aufgeworfen werden, kann auch eine Trichterrandfunktion der Art

$$\partial\mathcal{F}(t) = \frac{((1-\epsilon)t + \epsilon\tau)\lambda}{t} \quad (16.23)$$

mit:  $\lambda > 0, \tau > 0, \epsilon \in (0, 1)$

eingesetzt werden [107]. Für die Parameterwerte  $\epsilon = 0.2, \tau = 1$  und  $\lambda = 1$  ist die Funktion aus Gleichung (16.23) in Abbildung 16.9 dargestellt. Diese beginnt für  $t = 0$  im Unendlichen und umfasst somit jeden endlichen Wert des Fehlers zum Zeitpunkt  $t = 0$ . Damit muss bei der Parametrierung der Trichterrandfunktion nicht auf den anfänglichen Regelfehler geachtet werden.

Mathematisch formuliert drückt sich ein Verlauf des Regelfehlers innerhalb des Trichters durch die Ungleichung

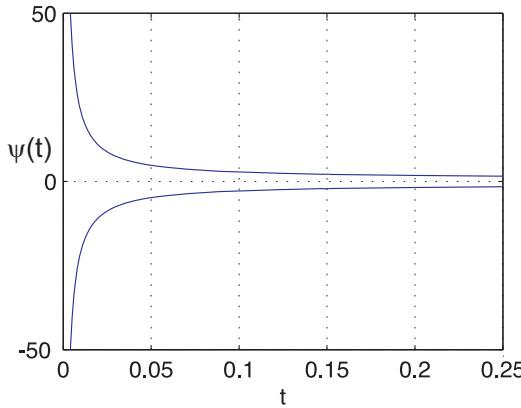
$$|e(t)| < \partial\mathcal{F}(t) \Leftrightarrow \frac{|e(t)|}{\partial\mathcal{F}(t)} < 1 \quad \forall t \geq 0 \quad (16.24)$$

aus. Graphisch dargestellt ist diese Konstellation in Abbildung 16.8.

Zwischen den beiden Funktionen, die den Regelfehler und Trichterrand beschreiben, wird zu jedem Zeitpunkt die Differenz bestimmt. Jene entspricht dem vertikalen Abstand

$$d_V(t) := \partial\mathcal{F}(t) - |e(t)| \quad (16.25)$$

der beiden Graphen. Dieser Parameter stellt das essentielle Kriterium für die Berechnung der Reglerverstärkung dar. Das Ziel, den Fehler im Trichter zu halten, wird gefährdet, wenn die vertikale Distanz kleine Werte erreicht. In diesem Fall muss die Verstärkung ausreichend ansteigen, um den Fehler zu verringern. Dieser



**Abb. 16.9:** Graph der Trichterrandfunktion (16.23) für die Parameterwerte  $\epsilon = 0.2$ ,  $\tau = 1$  und  $\lambda = 1$ .

Mechanismus greift, da eine high-gain-fähige Strecke zugrundegelegt ist. Sobald eine große Distanz zwischen Fehler und Trichterrand besteht, ist der aktuelle Betriebsbereich unkritisch, und es kann eine geringe Verstärkung (und dadurch nur geringes Eingreifen des Reglers) akzeptiert werden. Insofern ist mit  $d_V(t)$  eine geeignete Größe gegeben, um daraus die benötigte Reglerverstärkung zu ermitteln. Es bietet sich an, den Wert der Verstärkung durch die Inverse des vertikalen Abstandes festzulegen

$$k(t) := \frac{1}{d_V(t)} = \frac{1}{\partial \mathcal{F}(t) - |e(t)|} = \frac{\frac{1}{\partial \mathcal{F}(t)}}{1 - \frac{|e(t)|}{\partial \mathcal{F}(t)}} \quad (16.26)$$

und ein proportionales, zeitvariantes Regelgesetz

$$u(t) = k(t) e(t) \quad (16.27)$$

anzusetzen. Für den mathematischen Beweis der oben beschriebenen Strategie sei dem interessierten Leser die Veröffentlichung [107] zum Studium empfohlen.

An dieser Stelle sei lediglich der Grundgedanke skizziert: durch Gleichung (16.26) ist gewährleistet, dass die Reglerverstärkung eine geeignete Vergrößerung erfährt, wenn der Regelfehler nahe am Trichterrand verläuft. In der Folge steigt die Stellgröße an und hält den Fehler im Trichter. Eine wesentliche Aufgabe in der Beweisführung ist, zu zeigen, dass der Fehler vom Trichterrand wegbeschränkt bleibt, und somit ein Mindestabstand besteht. Dadurch wird eine Berührung des Trichterrandes ausgeschlossen, mit der Folge, dass der Fehler den Rand des Trichters nicht überqueren kann und daher die Verstärkung beschränkt bleibt. Sobald ein großer Abstand zwischen Rand und Fehler entsteht, ergibt sich wieder

ein geringer Wert für die Verstärkung  $k(t)$ . Durch diesen Automatismus wird der bisherige Schwachpunkt der hochverstärkungsbasierten Regelung beseitigt, nämlich das monotone Wachstum der Verstärkungsfunktion.

Nachdem für den Trichterrand eine von Null wegbeschränkte Funktion gefordert ist ( $\partial\mathcal{F}(t) \geq \lambda > 0$ ), wird der Trichter stets eine Mindestbreite beibehalten. Der Hintergrund dieser Forderung ist sofort verständlich, wenn die Berechnung der Verstärkung betrachtet wird. Wegen  $k(t) = 1/d_V(t)$  muss eine Distanz  $d_V(t) > 0$  verbleiben, um eine Verstärkung mit beschränkten Werten zu erhalten. Folglich ist nicht gesichert, dass der Regelfehler gegen Null konvergiert. Unter Umständen muss ein stationärer Fehler hingenommen werden, der häufig in proportionalen Regelkreisen auftritt. Insofern ist die remanente Regeldifferenz nicht untrennbar mit Funnel-Control verknüpft, sondern ist vielmehr kennzeichnend für das proportionale Regelgesetz. Dem von Francis und Wonham vorgestellten Prinzip des internen Modells [59], [244], [245] folgend, ist ein integraler Anteil in den Regelkreis einzubringen, um den stationären Fehler auszulöschen. Die zentrale Frage, wie ein integraler Anteil in den Regelkreis einzufügen ist, ohne dessen Stabilitätseigenschaften zu zerstören, wird im Abschnitt 16.8 eingehend diskutiert.

Aus einer theoretischen Sichtweise heraus ist ein Integrator im Regelkreis nicht zwingend notwendig. Weil der Trichter eine beliebig schmale Breite besitzen darf, kann der bleibende Regelfehler beliebig weit minimiert werden. Es ist jedoch zu beachten, dass Messrauschen einen umso größeren Einfluss erhält, je enger der Trichter wird. Daher wird in der Praxis ein positiver Effekt erzielt, wenn ein breiterer Trichter zum Einsatz kommt, und die stationäre Regeldifferenz durch einen integralen Anteil beseitigt wird.

Obwohl sich im Gegensatz zum Adoptionsgesetz (16.4) die Einstellung der Verstärkung grundsätzlich ändert, besteht die Idee der hochverstärkungsbasierten Regelung prinzipiell fort. Deshalb werden auch im Falle von Funnel-Control high-gain-fähige Strecken vorausgesetzt, die

1. einen Relativgrad  $\delta$  von Eins besitzen,
2. minimalphasig sind und
3. deren Vorzeichen der instantanen Verstärkung bekannt sind.

Ein nicht zu unterschätzender Vorteil von Funnel-Control besteht darin, dass auch nichtlineare Strecken geregelt werden können. Im Falle nichtlinearer Prozesse ist der Begriff „minimalphasig“ nicht auf natürliche Weise bestimmt und bedarf daher einer detaillierten Definition. Hierfür eignet sich die Darstellung eines dynamischen Systems als (nicht-)linearer Operator.

### **Definition 16.1** (Operatorklasse $\mathcal{N}$ )

Ein Operator  $N$  (gleichbedeutend Funktional), der eine stetige, reelle Funktion in den Raum der lokal essentiell beschränkten Funktionen abbildet, gehört zur Operatorklasse  $\mathcal{N}$ , wenn folgende Forderungen zutreffen:

1. Zu jeder vorgegebenen Grenze  $\delta > 0$  kann eine Zahl  $\Delta > 0$  angegeben werden, so dass alle beschränkten Eingangssignale  $\zeta(t) < \infty$  durch  $N$  gemäß

$$\sup_{t \geq 0} |\zeta(t)| \leq \delta \Rightarrow |(N\zeta)(\tau)| \leq \Delta \quad \text{für fast alle } \tau \geq 0 \quad (16.28)$$

in ein beschränktes Ausgangssignal abgebildet werden.

2. Für alle stetigen Zeitfunktionen  $\zeta(t), \xi(t)$  gilt zu jedem Zeitpunkt  $T \geq 0$ :

$$\zeta(t) = \xi(t) \forall t \in [0, T] \Rightarrow (N\zeta)(\tau) = (N\xi)(\tau) \quad \text{für fast alle } \tau \in [0, T] \quad (16.29)$$

3. Zu jedem gegebenen Zeitpunkt  $T \geq 0$  können für jede stetige Funktion  $\beta(t)$  Konstante  $\tau > 0$ ,  $\delta > 0$  und  $c > 0$  angegeben werden, die für alle stetigen Signale  $\zeta(t)$  und  $\xi(t)$  mit den Eigenschaften

$$\zeta(t) = \beta(t) = \xi(t) \quad \forall t \in [0, T]$$

und

$$\zeta(s), \xi(s) \in [\beta(T) - \delta, \beta(T) + \delta] \quad \forall s \in [T, T + \tau]$$

der Ungleichung

$$\text{ess-sup}_{s \in [T, T + \tau]} |(Z\zeta)(s) - (Z\xi)(s)| \leq c \cdot \sup_{s \in [T, T + \tau]} |\zeta(s) - \xi(s)| \quad (16.30)$$

genügen.

Die folgenden Anmerkungen führen die Eigenschaften der Operatorklasse in Textform aus:

### Anmerkung:

1. Jeder Operator aus  $\mathcal{N}$  bildet eine stetige Funktion in eine stückweise stetige und lokal essentiell beschränkte Funktion ab.
2. Nachdem eine Funktion mit beschränkter Norm in eine Funktion mit wiederum beschränkter Norm abgebildet wird, lässt sich jeder Operator aus  $\mathcal{N}$  als E/A-stabiles System interpretieren.
3. Es liegt Kausalität vor, weil  $N$  eine Ausgangsfunktion erzeugt, die nur vom Verlauf der Eingangsfunktion in der Vergangenheit abhängt. Die zukünftige Entwicklung des Einganges beeinflusst das momentane Verhalten des Ausgangs nicht. Jedes reale System ist kausal (= realisierbar), so dass durch diese technische Voraussetzung keine weitere Restriktion hinzugefügt wird.
4. Es liegt in gewissem Sinne Stetigkeit vor. Naturgemäß erzeugen zwei identische Eingangssignale auch zwei identische Ausgangssignale. Wenn sich die Eingänge nur „wenig“ voneinander unterscheiden, so trifft dies auch für die

Ausgänge zu. Damit ist der Unterschied, der zwischen zwei verschiedenen Ausgängen auftreten kann, durch den Unterschied zwischen zwei gegebenen Eingängen begrenzt. Hier liegt ein ähnliches Prinzip zu Grunde wie bei linearen Systemen, die bekanntlich eine stetige Abhängigkeit der Lösung von den Anfangswerten aufweisen. Zwei Eingangsfunktionen, die fast gleich sind, führen zu zwei Ausgangsfunktionen, die ebenfalls fast gleich sind.

Mit einem solchen Operator kann das Verhalten eines dynamischen Systems nachgebildet werden. Im Hinblick auf hochverstärkungsbasierte Regelung wird der Operator  $N$  genutzt, um die Nulldynamik der Regelstrecke zu beschreiben. Aus der E/A-Stabilität des Operators folgt direkt die Stabilität der Nulldynamik. Lineare Systeme, deren Nulldynamik durch  $N$  erfasst werden kann, sind demzufolge minimalphasig.

Mit Hilfe eines solchen Operators bzw. mit Hilfe der Operatorklasse  $\mathcal{N}$  lässt sich schließlich die zulässige Systemklasse definieren.

**Definition 16.2** (Systemklasse  $\mathcal{S}$ )

Ein dynamisches System (physikalischer Prozess) wird der Systemklasse  $\mathcal{S}$  zugerechnet, wenn es einer Differentialgleichung der Form

$$\dot{y}(t) = f(p(t), (Ny)(t), u(t)) \quad (16.31)$$

genügt und letztere zusätzlich folgende Eigenschaften besitzt:

1.  $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  ist stetig.
2.  $N$  stammt aus  $\mathcal{N}$ .
3.  $p(t)$  ist eine essentiell beschränkte Funktion.
4. Der High-Frequency-Gain des Systems besitzt ein konstantes, bekanntes Vorzeichen. Vereinfachend sei im Folgenden ein positives Vorzeichen vorausgesetzt. Weil ein System mit negativem Vorzeichen durch Vorzeichenumkehr in der Stellgröße ein positives Vorzeichen enthält, stellt die Festlegung auf ein positives Vorzeichen keinerlei Einschränkung der Allgemeinheit dar. Formulieren lässt sich diese Forderung dadurch, indem für jede beliebige nicht-leere, kompakte Menge  $C \subset \mathbb{R} \times \mathbb{R}$  und für jede Folge  $(u_n) \subset \mathbb{R} \setminus \{0\}$ ,  $n \in \mathbb{N}$

$$|u_n| \rightarrow \infty \text{ wenn } n \rightarrow \infty \Rightarrow \min_{(v,y^*) \in C} (\text{sign}(u_n) \cdot f(v, y^*, u_n)) \rightarrow \infty$$

wenn  $n \rightarrow \infty$  gefordert wird.

Bedingung 4 stellt sicher, dass die zeitliche Änderung des Ausgangs mit einer geeigneten Stellgröße stets auf jeden beliebigen Wert geführt werden kann. Dadurch kann  $y(t)$  beliebig schnell verändert werden. Diese Eigenschaft der Systemklasse ist ein wesentlicher Kernpunkt. Es ist erforderlich, dass das Fehlerignal (das sich aus der Differenz von Sollwert und Ausgang errechnet) jedem

Abfallen des Trichterrandes folgen kann. Hierzu ist offensichtlich nötig, dass der Ausgang mit jeder beliebigen Rate verstellt werden kann.

In Gleichung (16.31) ist die Einwirkung eines Signales  $p(t)$  betrachtet, um auch externe Störeinflüsse zu berücksichtigen. Auch an dieser Stelle zeigt sich das Potential, das dieses Verfahren beinhaltet. Für praxisnahe Anwendungen muss in jedem Falle eine ausreichende Robustheit gegenüber Störeinflüssen gewährleistet sein. Im Gegensatz zu vielen Identifikationsalgorithmen, die unter Störeinwirkungen gewöhnlicherweise zu instabilen Regelkreisen führen, besteht bei Funnel-Control diese Schwierigkeit nicht. Bezogen auf Antriebe ist festzustellen, dass unbekannte Lastmomente und Reibkennlinien problemlos in das Konzept integriert werden können und keine negativen Auswirkungen auf die Stabilität des Gesamtsystems besitzen.

Die Hauptaufgabe beim Entwurf einer solchen Regelung ist, die high-gain-Fähigkeit der Strecke zu prüfen und gegebenenfalls durch weitere Maßnahmen zu erreichen. Hierbei stehen strukturelle Eigenschaften im Mittelpunkt, die vielfach ohne genaue Kenntnis der Streckenparameterwerte untersucht werden können. Meist hängt die Wirkungsrichtung der Stellgröße (v.a. in Antriebssystemen) nicht von unbekannten Parameterwerten ab, sondern lässt sich by inspection ablesen. Ähnliches trifft für den Relativgrad zu. Für dessen mathematische Definition und Berechnung sei auf Kapitel 12 verwiesen. Oftmals ist die Anwendung des dort vorgestellten Formalismus nicht notwendig, weil sich der Relativgrad anhand der Anzahl der Integratoren auf dem kürzesten Pfad zwischen Eingang und Ausgang aus dem Signalflussplan unmittelbar ergibt. Dabei ist jedoch zu beachten, dass diese Aussage nur zutrifft, sofern keine parallelen Zweige im Signalflussplan enthalten sind. Als Gegenbeispiele dienen die Jordanform und die Modalform, deren kürzeste Pfade nicht zwangsläufig den Relativgrad angeben.

Etwas mehr Aufmerksamkeit erfordert die Stabilität der Nulldynamik. Für diese Analyse kann die Kenntnis einiger Parameterwerte hilfreich oder sogar unumgänglich sein. Für eine große Klasse von Systemen ist die Kenntnis der Parameter nicht erforderlich. Ebenso sind die Beträge dieser Größen völlig unerheblich, weshalb deren Identifikation unnötig ist.

Wird ein gegebenes Streckenmodell hinsichtlich der high-gain-Fähigkeit untersucht, zeigt das Ergebnis meist eine Überschreitung des Relativgrades. Bedingt durch Trägheiten im System, erhält ein brauchbares Modell im Allgemeinen einen Relativgrad, der den zulässigen Wert Eins überschreitet. Auch bei elektrischen Antriebssystemen ist dieses Problem in weiter Verbreitung zu finden. Eine praktikauglicher Lösungsansatz hierfür wird in Abschnitt 16.9.3 diskutiert.

## 16.5 Hochverstärkungsbasierte Regelung mit zeitvarianter, nicht-monotoner Verstärkung

Für Funnel-Control wird die theoretische Basis durch Theorem 7 in [107] gelegt, das wegen seiner Bedeutung hier wiedergegeben wird:

### Theorem 16.1

Auf ein beliebiges System aus der Klasse  $\mathcal{S}$  lässt sich der hochverstärkungsbasierte Regler

$$u(t) = k(t)e(t) \quad \text{mit} \quad k(t) = \frac{1}{\partial\mathcal{F}(t) - |e(t)|} \quad \text{und} \quad e(t) = y^*(t) - y(t) = x_d(t) \quad (16.32)$$

anwenden, so dass ein Regelkreis resultiert, dessen Verhalten durch die Differentialgleichung

$$\dot{y}(t) = f(p(t), (Ny)(t), k(t) \cdot e(t)) \quad (16.33)$$

wiedergegeben wird. Diese besitzt eine Lösung auf dem Intervall  $[0, \infty[$ , wobei unter den Voraussetzungen

1. die beschränkte, stetige und positive Funktion  $\partial\mathcal{F}(t)$  besitzt die Eigenschaften  $\partial\mathcal{F}(t) > 0$  für alle  $t > 0$  und  $\liminf_{t \rightarrow \infty} \partial\mathcal{F}(t) > 0$
2. der Sollwert  $y^*(t)$  ist ebenfalls eine beschränkte, stetige und positive Funktion
3. der anfängliche Fehler  $e(0)$  befindet sich im Trichter

folgende Eigenschaften vorliegen:

1. Es existiert ein  $\varepsilon \in ]0, 1[$ , das  $|e(t)| \leq (1 - \varepsilon)\partial\mathcal{F}(t)$  für alle  $t \geq 0$  erfüllt. Der Fehler bleibt also innerhalb des Trichters und ist von dessen Rand wegbeschränkt.
2. Weder Stellgröße, noch Verstärkung gehen gegen Unendlich. Beide Funktionen sind stetig.

Der Beweis dieses Theorems ist geprägt von mathematischen Kunstgriffen und zum Abdruck in einem ingenieurnahen Lehrbuch ungeeignet. Dem Interessierten sei die Arbeit in [107] nahegelegt.

Obiges Theorem besagt, dass der Regelfehler durch einen weitgehend frei wählbaren Trichter eingegrenzt werden kann. Dadurch kann einer Regelstrecke nicht nur ein beliebiges asymptotisches Verhalten, sondern als Besonderheit auch ein beliebiges Verhalten in der transienten Phase aufgezwungen werden.

## 16.6 Anwendung am Beispiel Einmassensystem

Sofern im Antriebssystem eine starre Welle vorliegt, eignet sich ein Einmassensystem zur Modellierung der Anlage. Durch das Symbol  $J$  werde das Massenträgheitsmoment beschrieben,  $\omega(t)$  kennzeichnet die Drehgeschwindigkeit und stimmt mit der Zustandsgröße und dem Ausgangssignal  $y(t)$  überein. Als Eingang wirkt ein Drehmoment  $u(t)$ . Eine mögliche Störung (beispielsweise in Form eines Lastmomentes) wird durch die Größe  $z(t)$  bezeichnet. Als Zustandsraummodell eines solchen Prozesses lässt sich die Gleichung

$$\dot{\omega}(t) = \frac{1}{J} \cdot u(t) - \frac{1}{J} \cdot z(t), \quad \omega(0) = \omega_0 \in \mathbb{R}, \quad y(t) = \omega(t) \quad (16.34)$$

angeben. Deren High-Frequency-Gain ist durch  $1/J$  gegeben und physikalisch bedingt auf positive Werte beschränkt. Aus dem selben Grund lässt sich sofort der Relativgrad zu  $\delta = 1$  ermitteln. Weil der Relativgrad mit der Ordnung des Gesamtsystems übereinstimmt, ist keine Nulldynamik vorhanden. Trivialerweise ist damit auch kein instabiles Subsystem im Rückführweg als Nulldynamik enthalten. An dieser Stelle zeigt sich, dass die Kenntnis der Parameterwerte nicht notwendig ist, um die erforderlichen Eigenschaften der Strecke abzuprüfen. Einzig physikalische Naturgesetze (Massen haben positive Werte) und die Struktur der Differentialgleichung wurden herangezogen.

## 16.7 Internes Modell für die Realisierung einer stationär genauen Regelung

Neben der Forderung nach Stabilität, d.h. nach beschränkten Verläufen der Eingangs-, Zustands- und Ausgangsgrößen des Regelkreises, ist normalerweise auch die asymptotische Konvergenz des Ausgangssignales gegen den Sollwertverlauf gewünscht. Für Einsatzzwecke in der industriellen Praxis ist daneben auf Robustheit zu achten, in dem Sinne, dass die geforderten Eigenschaften selbst unter Störgrößeneinfluss und bei Unterschieden zwischen nominellen und realen Parameterwerten erhalten bleiben.

Wird das Konzept Funnel-Control an obigen Vorgaben gemessen, zeichnet sich folgendes Bild ab: Weil jeder Prozess aus der Klasse  $\mathcal{P}$  beherrscht werden kann, insofern, dass der Regelfehler durch die Trichterrandfunktion begrenzt ist, wird die Forderung nach robuster Stabilität erfüllt. Robust deshalb, weil die exakte Parametrierung keine Rolle spielt und damit Abweichungen zwischen nominellen und realen Parameterwerten keinen destabilisierenden Effekt bewirken. Auch Störungen werden von diesem Konzept toleriert und sind somit unkritisch.

Allerdings ist nicht gesichert, dass der Regelfehler tatsächlich asymptotisch verschwindet, lediglich ein Verlauf im Trichter ist garantiert. Die gewünschte asymptotische Konvergenz zwischen Ausgangssignal und Sollwertverlauf wird

demzufolge nicht erreicht. An dieser Stelle werden Erweiterungen notwendig, die hier beschrieben werden sollen.

Die grundlegende Theorie, wann ein Regelkreis die Eigenschaft der asymptotischen Konvergenz auf den Sollwertverlauf besitzt, ist durch die Arbeiten von B. A. Francis und W. M. Wonham [59], [244], [245] gegeben. Darin wird unter der Bezeichnung „Internal Model Principle“ beschrieben, wann der Ausgang einer linearen Strecke einem gegebenen Sollwert asymptotisch folgt und der Regelfehler damit asymptotisch verschwindet. Es wird gezeigt, dass im Regelkreis ein internes Modell vorliegen muss, das in der Lage ist, den Sollwert zu generieren.

Um das Prinzip des internen Modells konkret darstellen zu können, wird im betrachteten Regelkreis Stabilität vorausgesetzt, die eine notwendige Bedingung für das Erreichen des gewünschten Folge- und Störverhaltens darstellt. Zugrunde liegt hier die Tatsache, dass ausschließlich im stabilen Regelkreis transiente Vorgänge abklingen, so dass asymptotisch das gewünschten Verhalten eintreten kann. Um im Folgenden eine Unterscheidung zwischen Folgeregelung und Störgrößenunterdrückung<sup>2)</sup> des in Abbildung 16.10(a) gezeichneten Regelkreises zu vermeiden, wird als Ausgangssignal anstelle von  $y(\cdot)$  der Regelfehler  $e(\cdot)$  gewählt, der in beiden Aufgabenstellungen gleichermaßen verschwinden soll. Das Eingangssignal ist abhängig von der Problemstellung entweder die Soll- oder die Störgröße und wird einheitlich als Anregung  $w$  bezeichnet. Damit ergeben sich die Verhältnisse, die in den Abbildungen 16.10(b) bzw. 16.10(c) dargestellt sind. Aus dieser Sichtweise heraus sind beide Aufgaben identisch und bedeuten die asymptotische Eliminierung des Anregungssignales  $w(\cdot)$ . Dies gestattet die vorliegende Aufgabenstellung wie in Abbildung 16.11 kompakt zu beschreiben, wobei im System RK mit der Übertragungsfunktion

$$S(s) := \frac{e(s)}{w(s)} \quad (16.35)$$

der gesamte Regelkreis zusammengefasst ist. Die zu unterdrückende Anregung  $w(\cdot)$  sei beschränkt auf Signale, die der folgenden Klasse angehören.

### Definition 16.3 (Funktionsraum $\mathcal{W}$ )

Die Gesamtheit aller möglichen, maximalen Lösungen  $w : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ , die der homogenen Differentialgleichung

$$w^{(l)}(t) + \gamma_1 w^{(l-1)}(t) + \cdots + \gamma_{l-1} \dot{w}(t) + \gamma_l w(t) = 0, \quad \gamma_1, \dots, \gamma_l \in \mathbb{R}, \quad l \in \mathbb{N} \quad (16.36)$$

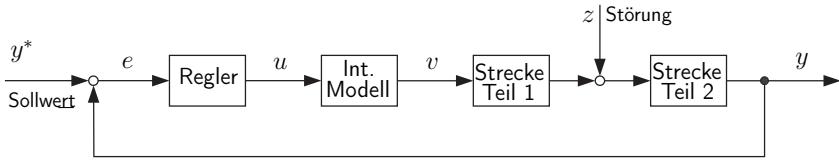
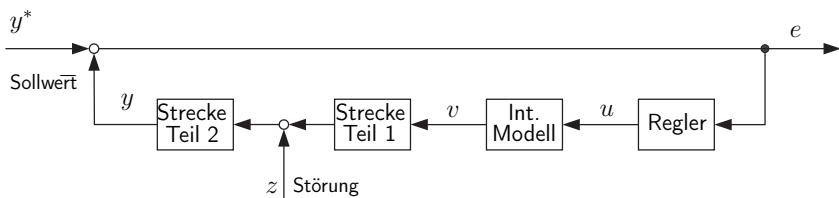
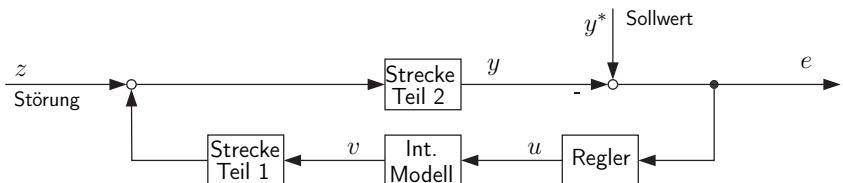
mit dem Anfangswert

$$w_o := (w(0) \quad \dot{w}(0) \quad \dots \quad w^{(l-1)}(0)) \in \mathbb{R}^l \quad (16.37)$$

genügen, spannt die Funktionenklasse  $\mathcal{W}$  auf.

---

<sup>2)</sup> Um einen Eingriff der Störung  $z$  an beliebiger Stelle in der Regelstrecke zuzulassen, wird die Strecke in zwei Teilsysteme aufgespalten. Falls die Störung nicht *in* der Strecke, sondern *vor* oder *nach* der Strecke einwirkt, ist das Teilsystem 1 bzw. 2 die Einheitsverstärkung und kann daher entfallen.

(a) Regelkreis mit Führungsgröße  $y^*$ , Störgröße  $z$  und Ausgang  $y$ .(b) Regelkreis für Folgeregelung mit Sollwert als Führungsgröße und Ausgang  $e$ .(c) Regelkreis für Störgrößenunterdrückung mit Störung als Führungsgröße und Ausgang  $e$ .

**Abb. 16.10:** Betrachteter Regelkreis mit Führungs- und Störgröße in aufgabenspezifischer Darstellung.

Im stabilen Regelkreis RK erregt jedes asymptotisch verschwindende Eingangssignal eine Antwort, die ebenfalls asymptotisch abklingt. Es folgt aus  $\lim_{t \rightarrow \infty} w(t) = 0$  unmittelbar  $\lim_{t \rightarrow \infty} e(t) = 0$ , der Effekt der Anregung wird für  $t \rightarrow \infty$  allein durch die vorausgesetzte Stabilität bereits vollständig unterdrückt. Der positive Einfluss eines internen Modells tritt daher nur im Zusammenhang mit dauerhaft bestehenden Anregungen zu Tage. Aus diesem Grund sei die Differentialgleichung (16.36) mit den Konstanten  $\gamma_i$  derart parametrisiert, dass keine asymptotisch abklingenden Anteile in deren Lösung enthalten sind. Durch diese Festsetzung konzentriert sich die Aufgabenstellung auf die asymptotische Auslösung beständig wirkender Anregungen.

Das anzustrebende Verhalten (Folgeverhalten/Störgrößenunterdrückung) liegt vor, wenn ein beliebiges Anregungssignal (Soll- bzw. Störsignal)  $w(\cdot) \in \mathcal{W}$  zu einem asymptotisch verschwindenden Regelfehler führt. Den folgenden Ausführungen werde ein Regler zu Grunde gelegt, der dieses Regelziel erreicht.



**Abb. 16.11:** Verallgemeinerte Darstellung für das Problem der Folgeregelung und Störgrößenunterdrückung.

Zusammen mit den beiden Polynomen  $Z$  und  $N$  lässt sich die Übertragungsfunktion

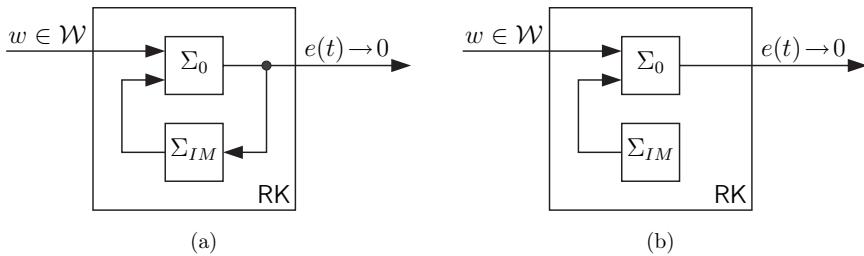
$$S(s) = \frac{Z(s)}{N(s)} = \frac{e(s)}{w(s)} \quad (16.38)$$

angeben, die das Ein-/Ausgangsverhalten des Regelkreises beschreibt. Unmittelbar kann aus der vorausgesetzten Stabilität des Regelkreises die Aussage

$$N(s) \neq 0 \quad \text{für alle } s \text{ mit } \operatorname{Re}\{s\} \geq 0 \quad (16.39)$$

gefollgert werden.

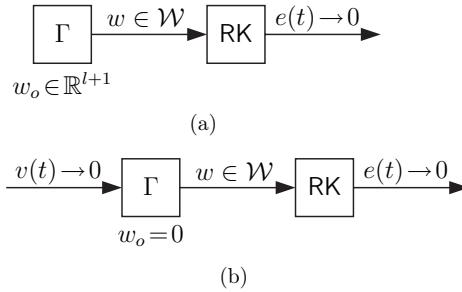
Durch eine Transformation in BINF ist eine Aufteilung des Regelkreises in zwei Subsysteme gemäß Abbildung 16.12(a) möglich, wobei das interne Modell  $\Sigma_{IM}$  ausschließlich vom Fehlersignal  $e$  beeinflusst wird und unter der Einwirkung von  $e$  eine Art Schätzung der Anregung erzeugt. Jene ist zur Kompensation von  $w$  geeignet, wenn der Regler die gestellte Aufgabe erfüllt und  $e(t) \rightarrow 0$  gilt. Dadurch verliert das interne Modell asymptotisch sein Eingangssignal und muss in der Konsequenz die Fähigkeit besitzen, als homogenes System alle Signale aus der Klasse  $\mathcal{W}$  generieren zu können. Der Fall  $t \rightarrow \infty$  ist in Abbildung 16.12(b) dargestellt. Eine „Anregungsphase“ bringt das interne Modell zunächst in einen



**Abb. 16.12:** (a) Regelkreis in BINF. (b) Regelkreis in BINF im Falle  $e = 0$ .

Zustand, so dass von diesem ausgehend, ohne weitere Beeinflussung ein kompensierendes Signal erzeugt wird.

Es kann allgemein gezeigt werden [217], dass die Nulldynamik als homogenes Modell mit entsprechenden Anfangswerten genau die Klasse von Signalen generiert, denen der Regelkreis RK asymptotisch folgen kann, bzw. die asymptotisch unterdrückt werden können.



**Abb. 16.13:** (a) Nachbildung der Anregung  $w(\cdot)$  durch einen homogenen Generator  $\Gamma$  mit Anfangswert  $w_o$ . (b) Nachbildung der Anregung  $w(\cdot)$  durch ein Übertragungssystem  $\Gamma$  mit Eingangssignal  $v(\cdot)$ .

Äquivalent zum homogenen Generator mit dem Anfangswert  $w_o$  in Abbildung 16.13(a) kann jedoch auch ein Übertragungssystem mit entsprechendem Eingangssignal  $v(\cdot)$  die Anregung  $w(\cdot)$  des Regelkreises erzeugen, wie in Abbildung 16.13(b) dargestellt ist. Dazu muss das homogene Modell um eine Einkopplung erweitert werden. Unter der Annahme, dass ein Einkoppelvektor festgelegt wird, mit dem der Generator  $\Gamma$  vollständig zustandssteuerbar wird, folgt daraus unmittelbar, dass ein Signalverlauf  $v : [0, 1] \rightarrow \mathbb{R}$  gefunden werden kann, der ausgehend vom Anfangszustand  $w_o = 0$  den Zustandsvektor des Generators in den Punkt

$$w_1 := \begin{pmatrix} w(1) & \dot{w}(1) & \dots & {}^{(l-1)}w(1) & {}^l w(1) \end{pmatrix}^T \in \mathbb{R}^{l+1} \quad (16.40)$$

steuert. Wird das Eingangssignal durch  $v(t) = 0$  für alle  $t \in ]1, \infty]$  fortgesetzt, erzeugt das Übertragungssystem für alle  $t > 1$  ein zum homogenen Generator identisches Ausgangssignal. Gleichzeitig gilt die für spätere Überlegungen wichtige Bedingung  $\lim_{t \rightarrow \infty} v(t) = 0$ . Damit ist das externe Übertragungssystem

$$G(s) = \frac{w(s)}{v(s)} = \frac{1}{\pi(s)} = \frac{1}{s^l + b_1 s^{l-1} + \dots + b_{l-1} s + b_l} \quad (16.41)$$

mit Anfangswert  $w_o = 0$  und Eingang  $v(\cdot)$  nun prinzipiell in der Lage, die zu unterdrückende Anregung  $w(\cdot)$  zu generieren.

Bedingt durch die Parametrierung des Signalgenerators (die keine abklingenden Anteile in der Lösung von Gleichung (16.36) bzw. im Ausgangssignal  $w(\cdot)$  von Übertragungssystem (16.41) zulässt), besitzt das Polynom  $\pi(s)$  keine Wurzel mit negativem Realteil, die Übertragungsfunktion  $G(s)$  ist nicht asymptotisch stabil. Durch diese Festsetzung konzentriert sich die Aufgabenstellung auf die asymptotische Auslöschung beständig wirkender Anregungen. Besitzt der Regler die Fähigkeit,  $w(\cdot)$  erfolgreich zu unterdrücken, gilt  $e(t) \rightarrow 0$ .

Die Serienschaltung aus Signalgenerator für die Anregung und Regelkreis wird durch das Produkt  $G(s) \cdot S(s) = Z(s)/(N(s)\pi(s))$  beschrieben. Nachdem voraussetzungsgemäß der verwendete Regler die Anregung  $w(\cdot)$  erfolgreich unterdrückt, gilt  $e(t) \rightarrow 0$ . Zusammen mit der Tatsache, dass das Eingangssignal  $v(\cdot)$  der Kombination aus Regelkreis und Generator asymptotisch gegen Null strebt (es gilt sogar  $v(t) = 0$  für alle  $t > 1$ ), folgt hieraus unmittelbar die Stabilität des Gesamtsystems  $G(s) \cdot S(s)$ . Wegen der Parametrierung der Differentialgleichung (16.36) besitzt das Polynom  $\pi(s)$  keine Wurzel mit negativem Realteil, so dass die Übertragungsfunktion  $G(s)$  und damit das Teilsystem  $\Gamma$  nicht asymptotisch stabil sind.

Aus diesem Grund muss im Produkt  $G(s) \cdot S(s) = Z(s)/(N(s)\pi(s))$  eine Nullstellen-Kürzung auftreten, die das nicht stabile Polynom  $\pi(s)$  im Nenner von  $G$  eliminiert.

Die Übertragungsfunktion  $S(s)$  des Regelkreises RK muss demzufolge in der Menge ihrer Nullstellen sämtliche Pole des Anregungsgenerators (die Wurzeln von  $\pi$ ) enthalten. Es ist sofort ersichtlich, dass der Schluss

$$\pi(s) = 0 \quad \Rightarrow \quad Z(s) = 0 \quad (16.42)$$

zutreffen muss, wenn die Anregung unterdrückt wird. Bei Betrachtung des Regelkreises in BINF stimmen die Polstellen der Nulldynamik mit den Nullstellen von  $S(s)$  und daher auch mit den Polen des Generators  $\Gamma$  überein. Dadurch zeigt sich, dass das interne Modell auch im Vorwärtszweig realisiert sein kann, weil dessen Pole bei einer Transformation in BINF stets in den Rückwärtszweig verschoben sind. Dadurch ist die Übereinstimmung mit der Aussage des Internen-Modell-Prinzips gegeben.

Das Subsystem im Rückführzweig umfasst daher die Dynamik des Generators und kann folglich bei geeigneter Vorgabe eines Anfangswertes das zu unterdrückende Anregungssignal nachbilden. Damit ist in der Nulldynamik des Regelkreises ein Generator für die Anregung bzw. ein Modell der Anregung enthalten, das sog. interne Modell. Zusammenfassend ist anzumerken, dass ein Regelkreis genau diejenigen Anregungen unterdrückt, die dessen Nulldynamik zu erzeugen im Stande ist.

### Anmerkungen:

- Das interne Modell kann als dynamisches Teilsystem im Regelkreis die Anregung nachbilden und lässt sich daher als Vorsteuerung auffassen. Solange ein Regelfehler besteht, generiert der Regler eine zweckmäßige Anregung für das interne Modell. Unter deren Einfluss wird asymptotisch eine ideale Stellgröße erzeugt, die den Regelfehler gegen Null konvergiert lässt. Für  $e = 0$  ist das interne Modell keiner Erregung unterworfen (unter der Annahme eines proportionalen Reglers) und muss, um diesen Zustand beizubehalten, die Strecke in geeigneter Weise steuern können. Wenn beispielsweise durch eine Störung ein Regelfehler verursacht wird, greift der Regler erneut ein und beeinflusst das interne Modell derart, dass die ideale Stellgröße ausgegeben wird. Werden das interne Modell und die Regelstrecke als Einheit

aufgefasst, erhält der erweiterte Prozess eine Eigendynamik, die unter der Voraussetzung geeignet gewählter Anfangswerte für  $u(\cdot) = 0$  auf  $e(t) \rightarrow 0$  führt.

- Das Zählerpolynom im internen Modell wirkt als Filter und verfälscht während des transienten Vorganges die vom Regler erzeugte Stellgröße. Asymptotisch ist der Zähler dagegen ohne Einfluss, da für  $t \rightarrow \infty$  der Ausgang des Reglers gegen Null konvergiert.
- Hinsichtlich eines Einsatzes im Funnel-Regelkreis ist die Verwendung eines internen Modells vorwiegend dann sinnvoll, wenn (z.B. bedingt durch Messrauschen) ein breiter Trichter gewählt werden muss. Im Falle eines sehr engen Trichters würde nämlich ein kleiner Regelfehler resultieren, weshalb mit einem internen Modell naturgemäß keine wesentliche Steigerung der Performance mehr möglich ist.

Damit ist die Frage beantwortet, wann eine Regelung stationär genaues Verhalten aufweist: es lässt sich festhalten, dass ein Subsystem im Regelkreis vorhanden sein muss (das sogenannte Interne Modell), das als homogenes System das Referenzsignal generieren kann. Dabei ist es aus Sicht des Führungsverhaltens unerheblich, ob das Interne Modell im Regler, oder in der Regelstrecke enthalten ist. Unter dem Einfluss von Störungen besteht dagegen ein Unterschied, d.h. es ist relevant, ob das Interne Modell im Regler oder in der Strecke liegt.

Vor diesem Hintergrund zeigt sich, dass der in Abschnitt 16.4 beschriebene Funnel-Regler (der in die Klasse der Proportionalregler einzuordnen ist), vor allem bei Störeinwirkung keine stationäre Genauigkeit erreichen kann. Dieser Nachteil wird nicht durch die spezielle Art hervorgerufen, wie die zeitvariante Reglerverstärkung bei Funnel-Control berechnet wird, sondern hängt mit dem proportionalen Verhalten (das gleichbedeutend ist mit dem Fehlen eines Internen Modells) des Reglers zusammen.

Aus theoretischer Sicht ist das fehlende Interne Modell kein relevanter Nachteil, weil durch eine entsprechende Verengung des Trichters der Betrag des Regelfehlers auf beliebig kleine Werte gezwungen werden kann. Bedingt durch Messunsicherheiten und Auflösungsgrenzen von Sensoren und Messaufnehmern kann ein reales Regelsystem niemals exakt arbeiten. Insofern bilden die physikalisch bedingten Grenzen für die erreichbare Genauigkeit die hauptsächliche Einschränkung, ein Regelfehler im Bereich der Auflösungsgrenze ist daher kein schwerwiegender Nachteil.

Dennoch ist ein Internes Modell bei praktischen Einsätzen der Regelung unabdingbar, was im Messrauschen begründet liegt. Weil jeder Sensor ein Minimum an Rauschen besitzt, ist das gemessene Fehlersignal ein „Band“, in dem der Regelfehler liegt. Würde nun ein Trichter angesetzt werden, der schmäler als das Band des Messrauschen ist, so würde der Regler den Rauschanteil als Sollwert miss verstehen und durch eine unrealistisch hohe Stellgröße dem Rauschen zu folgen versuchen. Um diesen unerwünschten Fall auszuschließen, muss der

Trichter breiter als das Rauschen bleiben. Aus diesem Grund ist eine beliebig große Reduktion des Regelfehlers in der Praxis ausgeschlossen, stationäre Genauigkeit (auch im Rahmen der physikalisch erreichbaren Genauigkeit) ist somit im Allgemeinen nicht erreichbar.

## 16.8 Allgemeines zur Regelung von Zweimassensystemen mit Funnel-Control

Praktische Problemstellungen der Regelungstechnik verlangen häufig die Auslegung des Reglers für eine Strecke mit nicht bekannten Parametern. In diesem Kontext ist Funnel-Control ein attraktives und hilfreiches Konzept, da für die Reglersynthese keine Streckenparameter benötigt werden.

In diesem Abschnitt wird die Regelung eines Zweimassensystems mit Funnel-Control betrachtet. Dabei wird zunächst das lineare Modell (2.15) zu Grunde gelegt. Auf den gewonnenen Erkenntnissen aufbauend wird im Anschluss die Regelung des nichtlinearen Zweimassensystems diskutiert.

An dieser Stelle soll herausgehoben werden, dass die Werte der linearen Parameter (Massenträgheitsmoment des Antriebes  $J_1$ , Massenträgheitsmoment der Last  $J_2$ , Federsteifigkeit der Kopplungswelle  $c$  und Dämpfungskoeffizient  $d$  der Kopplungswelle) nicht verwendet werden und daher gänzlich unbekannt sein dürfen. Gleiches gilt für die Reibcharakteristik. Ausschließlich die Struktur der Strecke ist Einschränkungen unterworfen, die sich durch

- bekanntes und konstantes Vorzeichen der instantanen Verstärkung,
- stabile Nulldynamik und
- Relativgrad eins

formulieren lassen. Dabei ist festzustellen, dass diese Eigenschaften nicht nur von der Eigendynamik eines Systems abhängen, sondern ebenfalls von der Wahl des Ausgangssignals beeinflusst werden.

Obige Eigenschaften liegen vor, wenn eine geeignete Linearkombination der Zustände oder die Motordrehzahl als Ausgang festgesetzt werden. Deshalb ist Funnel-Control für Zweimassensysteme zur Realisierung einer Zustandsregelung und zur Regelung der Motordrehzahl geeignet.

Für den Betrieb von Antriebssystemen, die sich durch Zweimassensysteme modellieren lassen, sind neben allgemeinen Anforderungen für Funnel-Regelkreise auch spezielle Vorgaben zu erfüllen:

- Das Fehlersignal  $e(t) = y^*(t) - y(t) = x_d(t)$  bleibt im vorgegebenen Trichter.
- Sämtliche Signalverläufe im Regelkreis sind beschränkt (stabiles Verhalten).
- Asymptotisch, d.h. für „große“  $t$  ist der Regelfehler „klein“.

- Schwingungen der flexiblen Verbindungswelle werden durch die Regelung gedämpft.
- Eine Störgröße  $z$ , die additiv zur Stellgröße wirkt, wird in dem Sinne unterdrückt, dass obige Forderungen dennoch erfüllt werden.

Mit Hilfe des zeitvarianten Proportionalreglers (Abb. 16.14)

$$\begin{aligned} v(t) &= k(t) e(t) \\ k(t) &= \frac{1}{\partial \mathcal{F}(t) - |e(t)|} \end{aligned} \tag{16.43}$$

sollen obige Ziele erreicht werden.

Unter Ausnutzung von Theorem 16.1 ergibt sich unmittelbar die Tatsache, dass der Fehler im Trichter verläuft, sowie die Stabilität des Regelkreises. Ein „kleiner“ Regelfehler ist dagegen nur dann garantiert, wenn ein entsprechend enger Trichter eingesetzt werden darf. Falls dies – wegen starkem Messrauschen – nicht zulässig ist, muss der Regelfehler unter Zuhilfenahme eines Internen Modells „klein“ gemacht werden.

Die aktive Bedämpfung von Torsionsschwingungen in der Transmissionswelle lässt sich durch eine Zustandsregelung erreichen, wobei der Verdrehwinkel ausreichend hoch zu gewichten ist [201].

Wie Theorem 16.1 zeigt, ist der Funnel-Regler grundsätzlich in der Lage, additive Eingangsstörungen zu tolerieren. Es bleibt hier zu beweisen, dass die Erweiterung um ein Internes Modell diese Eigenschaft nicht beeinträchtigt.

## 16.9 Funnel-Regelung für das lineare Zweimassensystem

Eine ausführliche Darstellung des Zweimassensystems findet sich in Kapitel 2, daher wird hier auf die Herleitung der Zustandsdifferentialgleichung verzichtet und direkt das Zustandsraummodell

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A} \underline{x}(t) + \underline{b} u(t) + \underline{b}_N M_W(t), & \underline{x}(0) = \underline{x}_o \in \mathbb{R}^3 \\ y(t) &= \underline{c}^T \underline{x}(t) \end{aligned} \tag{16.44}$$

angesetzt. Dabei bezeichnen die Eingangssignale  $u(t)$  das Antriebsmoment und  $M_W(t)$  das Widerstandsmoment. Die Zustandsmatrix  $\underline{A} \in \mathbb{R}^{3 \times 3}$  und die Eingangsvektoren  $\underline{b}$  bzw.  $\underline{b}_N$  sind durch

$$\underline{A} = \begin{bmatrix} -d/J_2 & c/J_2 & d/J_2 \\ -1 & 0 & 1 \\ d/J_1 & -c/J_1 & -d/J_1 \end{bmatrix}, \quad \underline{b} = \begin{pmatrix} 0 \\ 0 \\ 1/J_1 \end{pmatrix} \quad \text{und} \quad \underline{b}_N = \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \end{pmatrix} \tag{16.45}$$

gegeben, der Zustandsvektor ist mit

$$\underline{x} = \begin{pmatrix} \omega_2 \\ \Delta\varphi \\ \omega_1 \end{pmatrix} \quad \begin{array}{l} \text{Arbeitsmaschinendrehzahl} \\ \text{Wellenverdrehwinkel} \\ \text{Antriebsdrehzahl} \end{array} \quad (16.46)$$

gegeben. Für spätere Zwecke sei hier angemerkt, dass aus physikalischen Gründen für die Parameter

$$J_1 > 0, \quad J_2 > 0, \quad c > 0, \quad d > 0 \quad (16.47)$$

gilt.

Als Regelgröße (Ausgangssignal der Differentialgleichung) kommen die drei Signale Motordrehzahl, Lastdrehzahl und Linearkombination aller drei Zustände in Betracht. Daher wird der Auskoppelvektor  $c^T \in \mathbb{R}^3$  von der gewählten Regelungsart bestimmt.

### 16.9.1 Antriebsdrehzahl als Regelgröße

Die Regelung der Motordrehzahl erfordert als Ausgang den Zustand  $\omega_1(t)$ , weswegen für den Auskoppelvektor

$$\underline{c}^T = (0 \ 0 \ 1) \quad (16.48)$$

zu wählen ist.

Im Folgenden sind die strukturellen Eigenschaften der Strecke zu ermitteln, die sich aus den Gleichungen (16.44), (16.45), (16.48) zusammensetzt und deren high-gain-Fähigkeit zu überprüfen.

Das Ein-Ausgangs-Verhalten des Zweimassensystems mit Ausgang  $\omega_1(t)$  wird durch die Übertragungsfunktion

$$G(s) = \frac{\omega_1(s)}{u(s)} = \frac{y(s)}{u(s)} = \underline{c}^T (sI - \underline{\mathbf{A}})^{-1} \underline{b} = \frac{\frac{1}{J_1} s^2 + \frac{d}{J_1 J_2} s + \frac{c}{J_1 J_2}}{s^3 + d \frac{J_1 + J_2}{J_1 J_2} s^2 + c \frac{J_1 + J_2}{J_1 J_2} s} \quad (16.49)$$

wiedergegeben. Anhand dieser Übertragungsfunktion können die erforderlichen Merkmale der Strecke leicht ermittelt werden.

- (i) Instantane Verstärkung:

Die Instantane Verstärkung ist der Quotient aus dem höchsten Koeffizienten des Zählerpolynomes und dem höchsten Koeffizienten des Nennerpolynomes. Damit lautet die instantane Verstärkung:

$$V_0 = \frac{\frac{1}{J_1}}{1} = \frac{1}{J_1} \quad (16.50)$$

Aus physikalischen Gründen gilt  $J_1 > 0$  (es gibt keine negativen Massenträgheitsmomente in der Natur), daher ist die instantane Verstärkung positiv, unabhängig von der exakten Parametrierung der Strecke.

## (ii) Relativgrad:

Wie im Kapitel 12 eingehend beschrieben ist, bildet sich der Relativgrad aus der Differenz von Nennerordnung und Zählerordnung:  $\delta = 3 - 2 = 1$ .

## (iii) Minimalphasigkeit:

Da mit wachsendem  $k$  die Pole des Nennerpolynoms zu den Nullstellen des Zählerpolynoms streben, ist die Minimalphasigkeit des Nennerpolynoms bei hoher Verstärkung  $k$  nicht mehr relevant. Wesentlich ist aber daher die Minimalphasigkeit der Nullstellen des Zählerpolynoms der Übertragungsfunktion. Aufgrund  $\delta = 1$  strebt zusätzlich ein Pol gegen reell negativ unendlich. Das Zählerpolynom ist zweiter Ordnung und daher ein Hurwitzpolynom, wenn sämtliche Polynomkoeffizienten positive Vorzeichen besitzen. Es ist leicht einzusehen, dass diese Forderung wegen (16.47) zutrifft. Folglich liegt ein minimalphasiges System vor.

Zusammenfassend gilt, dass die Motordrehzahl von Zweimassensystemen mit Funnel-Control regelbar ist, weil solche Zweimassensysteme high-gain-fähig sind.

Prinzipbedingt enthalten proportionale Regler kein Internes Modell und sind daher nicht für die Beseitigung stationärer Regelfehler ausgelegt. Mit dem Ziel, die vorteilhaften Eigenschaften von Funnel-Control um stationäre Genauigkeit zu ergänzen, wird ein Integrator in den Regler eingefügt. Dabei ist sicherzustellen, dass durch die zusätzliche Dynamik des Integrators die Stabilität des Regelkreises nicht gefährdet wird.

Als Internes Modell eignet sich in diesem Zusammenhang ein PI-Baustein mit der Übertragungsfunktion:

$$G_{PI}(s) = \frac{u(s)}{v(s)} = \frac{k_p \cdot s + k_i}{s} \quad (16.51)$$

Im Gegensatz zu einem reinen Integrator enthält dieser den Durchgriff  $k_p$  und besitzt im Zähler sowie im Nenner die Ordnung eins; daher ist der Relativgrad null. Unter den Voraussetzungen

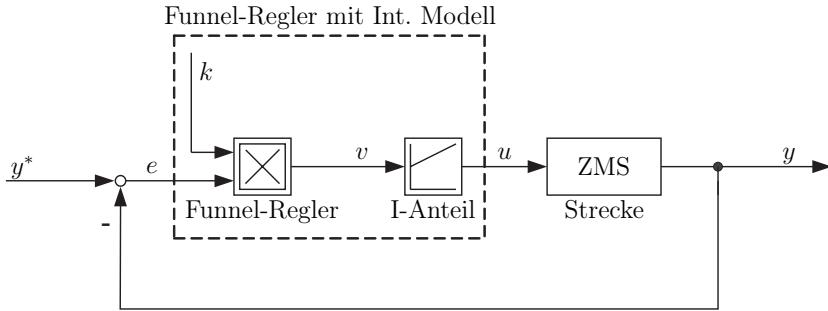
$$k_i > 0 \quad \text{und} \quad k_p > 0 \quad (16.52)$$

ist der Zähler ein Hurwitzpolynom, weshalb der PI-Baustein minimalphasiges Verhalten hat. Der daraus resultierende Regelkreis ist in Abbildung 16.14 dargestellt. Zielführend ist die beschriebene Erweiterung, weil im Gleichgewichtspunkt Integratoreingänge den Wert Null aufweisen müssen. Mit Blick auf Abbildung 16.14 bedeutet dies  $v \equiv 0$  als Voraussetzung für das Beibehalten einer Gleichgewichtslage. Wird darüberhinaus das funnel-typische Regelgesetz

$$v(t) = k(t) \cdot e(t) \quad (16.53)$$

mit

$$0 < k(t) < \infty \quad (16.54)$$



**Abb. 16.14:** Erweiterung des Regelkreises um einen PI-Block.

eingesetzt, ergibt sich aus  $v \equiv 0$  unmittelbar  $e \equiv 0$ . Daran zeigt sich direkt, dass durch den zusätzlich hinzugefügten Integrator (bzw. PI-Baustein) stationäre Genauigkeit hervorgerufen wird.

Allerdings ist nun zu überprüfen, ob durch das Hinzufügen des PI-Bausteins die Stabilität des Regelkreises verloren geht. Es kann nicht grundsätzlich davon ausgegangen werden, dass ein solches Vorgehen ohne Folgen bleibt. Für die erforderliche Stabilitätsuntersuchung wird das Interne Modell – das gemäß Abbildung 16.14 aus Sicht der Realisierung zum Regler gehört – der Strecke hinzugerechnet. Dadurch ändern sich die Gegebenheiten im Regelkreis nicht. Weil aber das Regelgesetz formal unverändert bleibt, kann weiterhin mit Theorem 16.1 argumentiert werden, was die Vorgehensweise erheblich erleichtert.

Deshalb sind im Folgenden erneut die strukturellen Eigenschaften der Strecke zu ermitteln, wobei unter Strecke nun die Serienschaltung aus PI-Block und Zweimassensystem zu verstehen ist. Das Ein-Ausgangs-Verhalten dieses Systems kann beschrieben werden durch die Übertragungsfunktion:

$$\begin{aligned}
 G(s) = \frac{\omega_1(s)}{v(s)} &= \frac{k_p \cdot s + k_i}{s} \cdot \frac{\frac{1}{J_1}s^2 + \frac{d}{J_1 J_2}s + \frac{c}{J_1 J_2}}{s^3 + d\frac{J_1+J_2}{J_1 J_2}s^2 + c\frac{J_1+J_2}{J_1 J_2}s} = \\
 &= \frac{\frac{k_p}{J_1}s^3 + \frac{k_p d + k_i J_2}{J_1 J_2}s^2 + \frac{k_p c + k_i d}{J_1 J_2}s + \frac{k_i c}{J_1 J_2}}{s^4 + d\frac{J_1+J_2}{J_1 J_2}s^3 + c\frac{J_1+J_2}{J_1 J_2}s^2} \quad (16.55)
 \end{aligned}$$

Jene besitzt die Eigenschaften:

- (i) Instantane Verstärkung:

Die Instantane Verstärkung ist der Quotient aus dem höchsten Koeffizienten des Zählerpolynomes und dem höchsten Koeffizienten des Nennerpolynomes. Damit lautet die instantane Verstärkung:

$$V_0 = \frac{k_p}{J_1} \quad (16.56)$$

Wegen  $J_1 > 0$  und mit  $k_p > 0$  ist die instantane Verstärkung wiederum positiv.

(ii) Relativgrad:

Die Differenz von Nennerordnung und Zählerordnung lautet  $\delta = 4 - 3 = 1$ , daher bleibt der Relativgrad bei eins. Weil das interne Modell über einen Durchgriff verfügt, ändert sich der ursprüngliche Relativgrad des Zweimassensystems nicht.

(iii) Minimalphasigkeit:

Die Prüfung der Minimalphasigkeit erfordert hier keinen Mehraufwand, obwohl ein Polynom dritter Ordnung zu prüfen ist; laut Hurwitzbedingungen müssen in diesem Falle nicht nur die einzelnen Koeffizienten positiv sein, sondern zusätzlich muss auch

$$\frac{k_p d + k_i J_2}{J_1 J_2} \cdot \frac{k_p c + k_i d}{J_1 J_2} - \frac{k_p}{J_1} \cdot \frac{k_i c}{J_1 J_2} > 0 \quad (16.57)$$

gelten. Zwar ist deren Gültigkeit nach Ausmultiplizieren direkt offensichtlich. Der Aufwand kann allerdings erspart bleiben, weil in Gleichung (16.55) zwei Polynome im Zähler multipliziert werden. Das Produkt hat trivialerweise genau dann seine Nullstellen in der linken Halbebene, wenn dies auch auf beide Faktoren zutrifft, d.h. wenn zwei Hurwitzpolynome miteinander multipliziert werden. Von dieser Tatsache haben wir uns aber bereits oben überzeugt. Folglich liegt ein minimalphasiges Gesamtsystem vor.

Aus diesem Grund darf der PI-Baustein als Internes Modell in den Regelkreis eingebbracht werden, ohne dass dadurch Stabilitätsprobleme verursacht würden.

### 16.9.2 Arbeitsmaschinendrehzahl als Regelgröße

Die Regelung der Lastdrehzahl erfordert als Ausgang den Zustand  $\omega_2(t)$ , weswegen für den Auskoppelvektor

$$\underline{c}^T = (1 \ 0 \ 0) \quad (16.58)$$

zu wählen ist.

Im Folgenden wird gezeigt, dass die erforderlichen strukturellen Eigenschaften nicht gegeben sind, so dass die Strecke (16.44), (16.45), (16.58) mit Funnel-Control nicht regelbar ist.

Aus dem Modell (16.44), (16.45), (16.58) ergibt sich die Übertragungsfunktion:

$$G(s) = \frac{\omega_2(s)}{u(s)} = \frac{y(s)}{u(s)} = \underline{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \underline{b} = \frac{\frac{d}{J_1 J_2} s + \frac{c}{J_1 J_2}}{s^3 + d \frac{J_1 + J_2}{J_1 J_2} s^2 + c \frac{J_1 + J_2}{J_1 J_2} s} \quad (16.59)$$

- (i) Instantane Verstärkung:

Diese lautet:

$$V_0 = \frac{\frac{d}{J_1 J_2}}{1} = \frac{d}{J_1 J_2} > 0 \quad (16.60)$$

- (ii) Relativgrad:

Die Differenz von Nennerordnung und Zählerordnung lautet  $\delta = 3 - 1 = 2$ , daher ist eine essentielle Eigenschaft *nicht* erfüllt.

- (iii) Minimalphasigkeit:

Wegen (16.47) enthält der Zähler ein Hurwitzpolynom, die Nullstelle ist negativ. Dadurch ist Minimalphasigkeit gewährleistet.

Zusammenfassend gilt, dass die Lastdrehzahl von Zweimassensystemen mit Funnel-Control nicht regelbar ist, weil solche Zweimassensysteme nicht high-gain-fähig sind. Eine Möglichkeit, Funnel-Control dennoch anzuwenden, besteht darin, auf die zeitliche Änderung, d.h. auf die Ableitung von  $y(t) = \omega_2(t)$  zurückzugehen. Allerdings entsteht durch das Differenzieren des Messwertes eine in der Regelungstechnik hinlänglich bekannte Problematik.

### 16.9.3 Zustandsregler mit Funnel

Unter dem Einfluss von Messrauschen erweist sich bekanntlich jegliche Art des Differenzierens als nachteilig, da hochfrequentes Rauschen dabei besonders hoch verstärkt wird und damit den Nutzanteil des Messsignales vollständig überlagert. Daher wird der oben angedeutete Ansatz mit Differenzierung des gemessenen Ausgangssignales nicht weiter verfolgt, sondern eine Zustandsrückführung zur Relativgradreduzierung eingeführt, die bezüglich Messrauschen eine verbesserte Robustheit besitzt.

Die Diskussion im Abschnitt 16.3 zeigt, dass für Systeme mit hohen Relativgraden eine Ausgangsrückführung nicht ausreichend ist, um Stabilität für beliebig anwachsende Verstärkungen zu erhalten. Dieser Sachverhalt spiegelt sich ebenfalls in Theorem 16.1 bzw. in der Definition der zugrundeliegenden Systemklasse 16.2 wider. Damit ist in solchen Fällen Funnel-Control ohne weitere Maßnahmen nicht anwendbar.

Es ist jedoch bekannt, dass durch eine Zustandsrückführung zusätzliche Nullstellen einfügt werden, was die gewünschte Relativgradreduzierung erlaubt. Zur Anwendung von Funnel-Control am Zweimassensystem muss daher ein Übergang zu einer Zustandsregelung erfolgen. Aus diesem Grund wird anstelle des physikalischen Ausgangssignales bzw. der anlagentechnisch relevanten Größe

$$\omega_2 = (1 \ 0 \ 0) \underline{x}, \quad \text{wobei} \quad \underline{x} = \begin{pmatrix} \omega_2 \\ \Delta\varphi \\ \omega_1 \end{pmatrix} \quad (16.61)$$

demnach eine (zunächst) frei wählbare Linearkombination der Zustände

$$y_r = \underline{c}^T \underline{x} = k_{\omega_2} \omega_2 + k_{\Delta\varphi} \Delta\varphi + k_{\omega_1} \omega_1 \neq x_d = \omega_2^* - \omega_2 \quad (16.62)$$

zurückgeführt. Im Gegensatz zum üblichen Vorgehen bei der Auslegung einer Zustandsregelung, wobei aus dem gewünschten Verhalten der geregelten Strecke der Rückführvektor

$$\underline{c}^T = (k_{\omega_2} \ k_{\Delta\varphi} \ k_{\omega_1}) \in \mathbb{R}^{1 \times 3} \quad (16.63)$$

bestimmt wird, wird hier wegen der unbekannten Parameter der Regelstrecke ein anderer Weg beschritten. Die Idee ist dabei, das Rückföhrsignal  $y_r$  als neuen Streckenausgang zu betrachten, so dass sich eine neue Übertragungsfunktion

$$G(s) = \frac{y_r(s)}{u(s)} = \underline{c}^T (sI - A)^{-1} \underline{b} \quad (16.64)$$

ergibt, welche bei geeigneter Wahl von  $c$  einen relativen Grad von eins besitzt. Diese „neue“ Regelstrecke mit dem neuen Ausgang  $y_r$  kann dann mit Funnel-Control beherrscht werden. Allerdings ist zu beachten, dass neben der Reduktion des Relativgrades auch Minimalphasigkeit verlangt ist. Dieser Forderung muss bei der Wahl von  $k_{\omega_2}$ ,  $k_{\Delta\varphi}$  und  $k_{\omega_1}$  Rechnung getragen werden. Dadurch erschwert sich die Wahl geeigneter Auskoppelkoeffizienten. Dabei kommt ein ingenieurtechnischer Ansatz zum Tragen, der eine vollständige Unkenntnis der Strecke ausschließt.

Am Beispiel des Zweimassensystems ist leicht ersichtlich, dass in der Antriebstechnik Teilwissen über die Regelstrecke durchaus vorhanden ist. So wird die Struktur des Zweimassensystems als bekannt angesetzt, sowie die Vorzeichen aller relevanten Parameter (Massenträgheiten, Federsteifigkeit und Dämpfung sind positiv, siehe Gleichung (16.47)), die aus physikalischen Gründen nur positiv sein können. Die Beträge der Parameter sollen im Zusammenhang mit adaptiver Regelung als unbekannt angenommen werden dürfen. Es lässt sich zeigen, dass dieses Wissen bereits ausreichend ist, um eine minimalphasige Auskopplung zu bestimmen. Die genaue Lage der Nullstellen ist in diesem Fall zwar nicht bekannt (da abhängig von den unbekannten Werten der Systemparameter), aber sie befinden sich in jedem Falle in der stabilen linken Halbebene, was die erforderliche Minimalphasigkeit bedingt.

Wird der Ausgang durch eine Linearkombination aller drei Systemzustände festgelegt, ergibt sich die entsprechende Übertragungsfunktion durch:

$$\begin{aligned} G(s) &= \frac{\underline{c}^T \underline{x}(s)}{u(s)} = \frac{y_r(s)}{u(s)} = \underline{c}^T (sI - A)^{-1} \underline{b} \\ &= \frac{\frac{k_{\omega_1}}{J_1} s^2 + \frac{(k_{\Delta\varphi} J_2 + (k_{\omega_2} + k_{\omega_1}) d)}{J_1 J_2} s + \frac{(k_{\omega_2} + k_{\omega_1}) c}{J_1 J_2}}{s^3 + d \frac{J_1 + J_2}{J_1 J_2} s^2 + c \frac{J_1 + J_2}{J_1 J_2} s} \quad (16.65) \end{aligned}$$

Die bei der Reglersynthese frei einstellbaren Parameter  $k_{\omega_2}$ ,  $k_{\Delta\varphi}$ ,  $k_{\omega_1} \in \mathbb{R}$  werden genutzt, um die Koeffizienten des Zählerpolynomes und damit die Übertragungsnullstellen zu beeinflussen.

Damit das Zweimassensystem mit Zustandsauskopplung der high-gain-fähigen Systemklasse  $\mathcal{S}$  zugeordnet werden kann, ist nicht nur die Untersuchung der instantanen Verstärkung und des Relativgrades erforderlich, sondern zusätzlich auch die Bestimmung geeigneter Auskoppelkoeffizienten, so dass Minimalphasigkeit sichergestellt ist.

Die erforderlichen Eigenschaften lassen sich anhand der Übertragungsfunktion (16.65) ermitteln:

- (i) Instantane Verstärkung:

Die instantane Verstärkung ist  $V_0 = \frac{k_{\omega_1}}{J_1}$ . Wegen (16.47) ist diese positiv, wenn

$$k_{\omega_1} > 0 \quad (16.66)$$

gewählt wird.

- (ii) Relativgrad:

Wird (16.66) erfüllt, so steht im Zähler der Übertragungsfunktion (16.65) ein Polynom zweiter Ordnung. Daraus resultiert eine Differenzordnung (=Relativgrad) von  $\delta = 3 - 2 = 1$ . Diesem Zweck ist die Zustandsrückführung in Kombination mit Funnel-Control geschuldet. Es ist daher derjenige Zustand in der Linearkombination  $y(t) = cx(t)$  zu gewichten, dessen erste zeitliche Ableitung direkt durch die Stellgröße  $u(t)$  beeinflusst wird. Mit  $k_{\omega_1} \neq 0$  wird sichergestellt, dass die Antriebsmaschinendrehzahl in der Linearkombination in Erscheinung tritt.

- (iii) Minimalphasigkeit:

Zur Einhaltung dieses Kriteriums muss im Zähler von (16.65) ein Hurwitz-Polynom enthalten sein. Nachdem es sich um ein Polynom vom Grad zwei handelt, müssen gemäß dem bekannten Hurwitz-Test alle drei Polynomkoeffizienten sämtlich positiv sein. Folglich sind für Minimalphasigkeit die drei Ungleichungen

$$\frac{k_{\omega_1}}{J_1} > 0 \quad (16.67)$$

$$\frac{(k_{\Delta\varphi} J_2 + (k_{\omega_2} + k_{\omega_1})d)}{J_1 J_2} > 0 \quad (16.68)$$

$$\frac{(k_{\omega_2} + k_{\omega_1})c}{J_1 J_2} > 0 \quad (16.69)$$

simultan einzuhalten. Unter Berücksichtigung von (16.47) ergeben sich diese Aussagen für die frei wählbaren Auskoppelkoeffizienten  $c_i$ :

$$(16.67) \Rightarrow k_{\omega_1} > 0 \quad (16.70)$$

$$(16.69) \Rightarrow k_{\omega_2} + k_{\omega_1} > 0 \Rightarrow k_{\omega_2} > -k_{\omega_1} \quad (16.71)$$

$$(16.68) \Rightarrow k_{\Delta\varphi} > -\frac{d}{J_2}(k_{\omega_2} + k_{\omega_1}) \quad (16.72)$$

Diese Bedingungen sind notwendig und hinreichend, um Minimalphasigkeit zu erhalten. Jedoch beeinflussen die unbekannten Systemparameter die Koeffizienten im Zähler von (16.65) und damit auch Ungleichung (16.72). Aus diesem Grund ist der Ausdruck auf der rechten Seite der Ungleichung in (16.72) parameterabhängig und somit in der praktischen Anwendung nicht als Zahlenwert zu ermitteln. Jedoch ist die Aussage  $-d(k_{\omega_2} + k_{\omega_1})/J_2 < 0$  offenkundig, so dass  $k_{\Delta\varphi}$  durch eine negative Schranke begrenzt wird. Aus diesem Grund wird (16.72) auch dann eingehalten, wenn

$$k_{\Delta\varphi} \geq 0 \quad (16.73)$$

gefordert wird. Dadurch verlieren die obigen Forderungen ihre Abhängigkeit von den unbekannten Systemparametern, sind jedoch nun nicht mehr notwendig, sondern lediglich hinreichend. Man stellt also fest, dass die Übertragungsfunktion (16.65) ein minimalphasiges System beschreibt, wenn die Auskoppelkoeffizienten die Bedingungen

$$k_{\omega_1} > 0, \quad k_{\Delta\varphi} \geq 0, \quad k_{\omega_2} > -k_{\omega_1} \quad (16.74)$$

einhalten.

Sofern die Rückführung  $c$  die Bedingungen (16.74) einhält, die (16.67) – (16.69) implizieren, liegt ein Zweimassensystem mit allen erforderlichen strukturellen Eigenschaften vor, um unter Regelung mit Funnel-Control stabiles Verhalten zu zeigen.

#### Anmerkung:

- Dieses Ergebnis beinhaltet die Untersuchungen in Abschnitt 16.9.1, weil  $k_{\omega_2} = 0$ ,  $k_{\Delta\varphi} = 0$  und  $k_{\omega_1} = 1$  eine gültige Konfiguration darstellt.
- Nachdem  $k_{\Delta\varphi} = 0$  gesetzt werden darf, ist eine Messung des Wellenwinkels nicht notwendig, wenn das Regelziel einzig die Stabilität des Regelkreises ist. Für weitergehende Eigenschaften (wie beispielsweise aktive Bedämpfung von Oszillationen in der Transmissionswelle) kann eine Gewichtung des Verdrehwinkels unter Umständen unumgänglich werden.

Bekanntermaßen wird bei Zustandsreglern allerdings nicht mehr auf die ursprüngliche Größe (in diesem Falle  $\omega_2$ ) geregelt, sondern auf eine Hilfsregelgröße, nämlich auf eine Linearkombination

$$y_r = \underline{c}^T \underline{x} = k_{\omega_2}\omega_2 + k_{\Delta\varphi}\Delta\varphi + k_{\omega_1}\omega_1 \quad (16.75)$$

$$\text{ beachte: } \quad x_d = \omega_2^* - \omega_2 \neq e$$

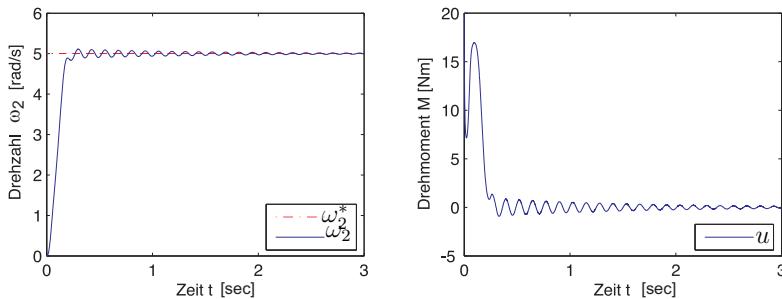
der Zustände. Dieser Tatsache muss Rechnung getragen werden, indem die gewünschte Sollgröße mit einem geeigneten Verstärkungsfaktor multipliziert wird.

Die benötigte Verstärkung ergibt sich aus dem Prozessverhalten am Gleichgewichtspunkt. Im Falle des Zweimassensystems geht bei der Berechnung der Vorverstärkung allerdings der Wellenwinkel ein, der wiederum vom Lastmoment geprägt ist. Um wunschgemäß eine nur kleine stationäre Drehzahlabweichung zu erreichen, muss damit die Sollwertskalierung abhängig vom Lastmoment erfolgen, welches im Allgemeinen nur sehr unzureichend bekannt ist. Eine Vereinfachung besteht darin, den Verdrehwinkel in der Linearkombination nicht zu gewichten, also  $k_{\Delta\varphi} = 0$  zu wählen. In diesem Fall ist die Vorverstärkung unabhängig vom Lastmoment auszulegen und ergibt sich zu

$$V = k_{\omega_2} + k_{\omega_1} > 0 \quad (16.76)$$

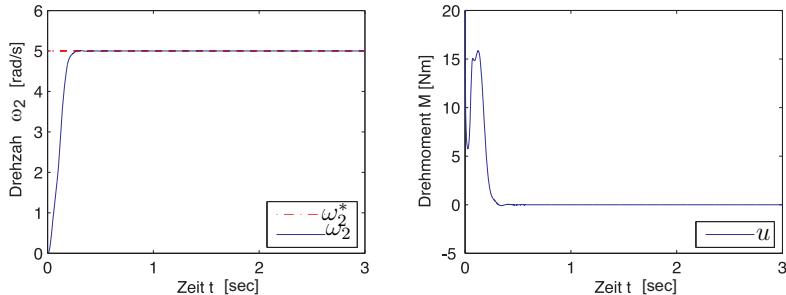
Die bisher dargestellten Ergebnisse sollen nun anhand einer Simulationsstudie verifiziert werden. Dabei wird zunächst ein Zweimassensystem ohne Lasteinwirkung untersucht, darauf folgend soll ein zusätzliches Widerstandsmoment hinzugefügt werden. Bei den Untersuchungen wurde bisher und wird weiterhin angenommen, dass die Dynamik der Stromregelung vernachlässigt werden kann.

In Abbildung 16.15 ist der simulierte Verlauf der Drehzahl  $\omega_2$  für Sprunganregung aufgezeichnet, wobei durch die Wahl  $k_{\Delta\varphi} = 0$  der Wellenverdrehwinkel nicht berücksichtigt wird. Nachdem das Zweimassensystem eine Polstelle im Ursprung besitzt, liegt global integrales Verhalten vor. Solche Strecken bringen demzufolge bereits ein Internes Modell für konstante Sollwertverläufe mit. An dieser Tatsache ändert eine Zustandsauskopplung nichts, weshalb die Regelung statio-när genau arbeitet, d.h. der Regelfehler verschwindet asymptotisch. Wegen der



**Abb. 16.15:** Regelung ohne Verdrehwinkel ( $k_{\Delta\varphi} = 0$ ). Es bleibt keine stationäre Regeldifferenz, da die Strecke globales I-Verhalten besitzt. Die Regelung kann die Schwingungen nicht bedämpfen.

fehlenden Information über die Wellenverdrehung  $\Delta\varphi$  kann jedoch keine aktive Dämpfung der schwingenden Drehzahl  $\omega_2$  erreicht werden. Funnel-Control sorgt lediglich dafür, dass die Linearkombination  $y_r = k_{\omega_2}\omega_2 + k_{\omega_1}\omega_1$  nur mit gerin-ger Amplitude schwingt und den Trichter nicht verlässt. Dies wird erreicht, wenn zur Schwingung von  $\omega_2$  eine gegenphasige Schwingung von  $\omega_1$  erzeugt wird. Auch



**Abb. 16.16:** Regelung mit Verdrehwinkel ( $k_{\Delta\varphi} = 50$ ). Dadurch kann die Regelung die Schwingungen ausreichend gut bedämpfen.

durch einen schmäleren Trichter kann daher die Schwingung des physikalischen Ausganges  $\omega_2$  nicht aktiv bedämpft werden. Nur bedingt durch die natürliche Eigendämpfung der Welle klingt die Schwingung ab. Daraus geht hervor, dass ohne Verdrehwinkel zwar stabiles Verhalten erzielt wird, eine brauchbare Regelung kann so allerdings nicht realisiert werden. Zur Verbesserung der Performance muss der Verdrehwinkel zurückgeführt und dazu auch gemessen werden. Teilweise erfordert dies keine zusätzliche Sensorik: falls die Drehgeschwindigkeiten beider Massen nicht mittels Tachogeneratoren, sondern über Positionsgeber ermittelt werden, kann aus der Differenz der Positionen (Winkel) völlig unproblematisch der Verdrehwinkel der Welle errechnet werden. Die „Winkelmessung“ wird hierbei durch eine einfache Subtraktion ersetzt. Im Falle einer Rückführung, d.h. für  $k_{\Delta\varphi} > 0$ , werden Schwingungen nun auch im Ausgang  $y_r$  sichtbar, so dass von Funnel-Control eine aktive Bedämpfung vorgenommen werden kann. Diese Tatsache zeigt sich auch anhand Abbildung 16.16. Der Unterschied zwischen den Simulationen von Abbildung 16.15 und Abbildung 16.16 ist die Wahl des Rückführkoeffizienten  $k_{\Delta\varphi}$ , der im ersten Fall zu  $k_{\Delta\varphi} = 0$  gesetzt ist, im zweiten Fall zu  $k_{\Delta\varphi} = 50$ .

Die Bedeutung des Verdrehwinkels für das Rückföhrrsignal  $y_r = \underline{c}^T \underline{x}$  wird plausibel, wenn die physikalisch bedingte Eigendynamik des Zweimassensystems betrachtet wird. Ausgehend von einem Anfangszustand

$$\underline{x}(0) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad (16.77)$$

d.h. beide Massen befinden sich in Ruhe, die Welle ist jedoch gespannt, werden beide Massen gleichzeitig losgelassen. Es ist leicht einzusehen, dass in der Folge eine gegenphasige, sinoidale Schwingung zwischen beiden Massen entstehen wird.

Auf mathematischem Wege bestätigt sich dieses Gedankenexperiment, wenn die Lösung der homogenen Differentialgleichung für obigen Anfangswert beispielsweise mittels Variation der Konstanten berechnet wird. Der Ansatz  $\underline{x}(t) =$

$\exp\{\mathbf{A}t\}\underline{x}(0)$  führt mit den Abkürzungen

$$R = (J_1 + J_2)[4kJ_1J_2 - (J_1 + J_2)d^2] > 0 \quad (16.78)$$

$$\omega_e = \frac{1}{2} \frac{1}{J_1 J_2} \sqrt{R} > 0 \quad (16.79)$$

$$\lambda = \frac{1}{2} \frac{J_1 + J_2}{J_1 J_2} d > 0 \quad (16.80)$$

auf die Bewegung der beiden Massen, die sich zu

$$\omega_1(t) = -J_2 \cdot \frac{2c}{\sqrt{R}} \exp\{-\lambda t\} \cdot \sin(\omega_e t) \quad (16.81)$$

und

$$\omega_2(t) = +J_1 \cdot \frac{2c}{\sqrt{R}} \exp\{-\lambda t\} \cdot \sin(\omega_e t) \quad (16.82)$$

ergeben. Diese beiden Signale sind exponentiell abklingende Sinusschwingungen. Sie unterscheiden sich in ihrer Amplitude, wobei durch das unterschiedliche Vorzeichen eine Phasenverschiebung von  $180^\circ$  besteht.

Dies bedeutet, dass durch Interferenz eine Rekuktion bzw. sogar vollständige Auslöschung der Schwingung möglich ist, wenn der Verdrehwinkel nicht gewichtet wird. Falls im Grenzfall

$$k_{\omega_2} = k_{\omega_1} \frac{J_2}{J_1} \quad (16.83)$$

gilt, sind die tatsächlich vorhandenen Schwingungen in der Linearkombination

$$y_r(t) = k_{\omega_2}\omega_2(t) + k_{\omega_1}\omega_1(t) = k_{\omega_1} \left( \frac{J_2}{J_1} \omega_2(t) + \omega_1(t) \right) = 0 \quad (16.84)$$

nicht sichtbar. Der Regler kann daher nicht unterscheiden, ob sich das Zweimasensystem in der Ruhelage

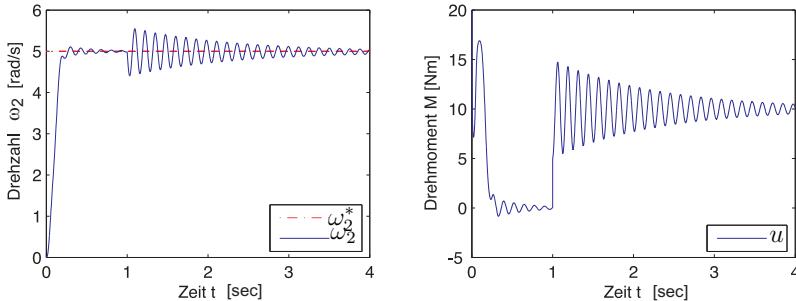
$$\underline{x} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (16.85)$$

befindet, oder ob unerwünschte Oszillationen vorliegen.

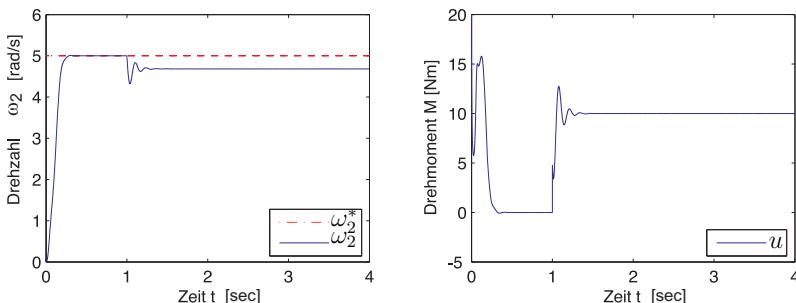
Aus einem Vergleich der beiden Abbildungen 16.15 und 16.16 wird deutlich, dass eine Steigerung von  $k_{\Delta\varphi}$  die Dämpfung erhöht, im Bild 16.16 sind keinerlei Schwingungen mehr zu erkennen.

Genauso wie im lastfreien Fall, entstehen auch unter Lasteinwirkung (zum Zeitpunkt  $t = 1$  sec wird ein Lastmoment von 10 Nm aufgeschaltet) Schwingungen, welche nicht aktiv bedämpft werden, wenn der Verdrehwinkel nicht zurückgekoppelt wird. In Abbildung 16.17 wird zwar trotz Last asymptotisch eine gute Genauigkeit erzielt, die entstehenden Schwingungen machen diese Regelung jedoch unbrauchbar. Es ist zu bemerken, dass der Sollwert nicht exakt erreicht

wird, stationäre Genauigkeit im strengen Sinne liegt nicht vor. Bedingt durch die Wahl eines sehr schmalen Trichters bleibt der stationäre Regelfehler jedoch äußerst klein, so dass aus praktischer Sicht von stationärer Genauigkeit gesprochen werden kann. Analog zur Simulation ohne Widerstandsmoment bewirkt



**Abb. 16.17:** Regelung ohne Verdrehwinkel ( $k_{\Delta\varphi} = 0$ ). Es bleibt zwar keine stationäre Regeldifferenz, aber die Regelung kann die Schwingungen nicht bedämpfen.



**Abb. 16.18:** Regelung mit Verdrehwinkel ( $k_{\Delta\varphi} = 50$ ). Damit werden zwar die Schwingungen gut bedämpft, aber es entsteht eine nicht ausregelbare stationäre Regeldifferenz.

auch unter Lasteinfluss das Zurückführen aller drei Systemzustände eine Verbesserung der Dämpfung. Da nun auch der Verdrehwinkel einen Einfluss auf das Fehlersignal hat, entstehen unter Last bemerkenswerte stationäre Abweichungen, wie aus Abbildung 16.18 erkennbar ist. Diese lassen sich selbst durch Verengen des Trichters nicht beseitigen.

Aus der Untersuchung des Fehlers kann eine solche Abweichung einfach erklärt werden. Durch die Berücksichtigung von  $\Delta\varphi$  ergibt sich ein Regelfehler von:

$$e = \underbrace{(k_{\omega_2} + k_{\omega_1})\omega_2^*}_{\text{Vorfilter}} - \underbrace{(k_{\omega_2}\omega_2 + k_{\Delta\varphi}\Delta\varphi + k_{\omega_1}\omega_1)}_{= y_r} \quad (16.86)$$

Im stationären Zustand sind beide Drehzahlen gleich, d.h.  $\omega_1 = \omega_2$ , woraus folgt:

$$e = (k_{\omega_2} + k_{\omega_1})(\omega_2^* - \omega_2) - k_{\Delta\varphi}\Delta\varphi \quad (16.87)$$

Wenn  $k_{\Delta\varphi} = 0$  gilt, so ist im stationären Zustand der Fehler  $e$  proportional zur Differenz  $x_d = \omega_2^* - \omega_2$ . Durch die Regelung kann  $e$  beliebig klein gemacht werden, wodurch demzufolge auch die Abweichung von  $\omega_2^*$  und  $\omega_2$ , also  $x_d$ , wunschgemäß klein wird. Für den Fall  $k_{\Delta\varphi} \neq 0$  ändern sich die Verhältnisse unter Last jedoch. Da hier von der Welle Moment übertragen werden muss, gilt  $\Delta\varphi \neq 0$ . Deswegen folgt aus  $e = 0$  nicht die Gleichheit von  $\omega_2^*$  und  $\omega_2$ , es kann aus einem kleinen Fehler  $e$  nun nicht mehr auf eine kleine Abweichung zwischen  $\omega_2^*$  und  $\omega_2$  geschlossen werden. Wie Abbildung 16.18 zeigt, bricht unter Lasteinwirkung die Drehzahl ein und behält eine beträchtliche stationäre Abweichung. Durch die verbesserte Dämpfung führen jedoch Lastsprünge nicht mehr zu langanhaltenden Schwingungen. Da für stationäre Genauigkeit unter Last der Verdrehwinkel nicht berücksichtigt werden darf ( $k_{\Delta\varphi}$  klein), für eine brauchbare Dämpfung aber berücksichtigt werden muss ( $k_{\Delta\varphi}$  groß), entsteht hier ein problematischer Kompromiss zwischen Dämpfung und Genauigkeit. Durch den Regler wird der Fehler  $e = (k_{\omega_2} + k_{\omega_1})\omega_2^* - y_r = V\omega_2^* - y_r$  (zwischen Sollwert und zurückgeführter Linearkombination) im Trichter gehalten, aber durch  $k_{\Delta\varphi} \cdot \Delta\varphi$  in  $y_r$  nicht die prozesstechnisch relevante Drehzahlabweichung  $x_d = \omega_2^* - \omega_2$ . Deshalb kann es durchaus zu Drehzahlabweichungen kommen, auch wenn der Trichter den Fehler  $e$  klein hält. An dieser Stelle wird ersichtlich, dass eine Gewichtung des Verdrehwinkels mit  $k_{\Delta\varphi}$  stets eine Abweichung der Prozessgröße nach sich zieht. Daraus erwächst bei Verwendung einer statischen Zustandsrückführung

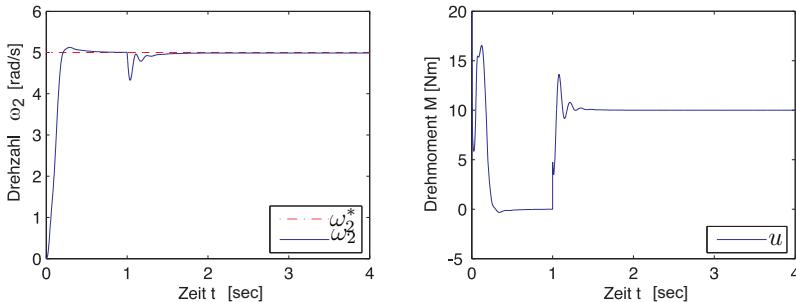
$$y_r(t) = (k_{\omega_2}, \ k_{\Delta\varphi}, \ k_{\omega_1}) \underline{x}(t) \quad (16.88)$$

ein Zielkonflikt zwischen stationärer Genauigkeit und aktiver Bedämpfung von Oszillationen.

Aus Gleichung (16.87) geht hervor, dass der Regelfehler  $e(t)$  gegen den Wert  $-k_{\Delta\varphi}\Delta\varphi$  konvergieren muss, damit die relevante Differenz  $x_d(t) = \omega_2^*(t) - \omega_2(t)$  verschwindet. Anstelle der Forderung  $e \rightarrow -k_{\Delta\varphi}\Delta\varphi$  kann allerdings das Regelziel  $e \rightarrow 0$  beibehalten werden, wenn zum Sollwert  $\omega_2^*(t)$  der zusätzliche Offset  $k_{\Delta\varphi}\Delta\varphi/V$  addiert wird. Obwohl die Sollwertanpassung formal zu bewerkstelligen ist, gestaltet sich die Realisierung schwierig.

Der notwendige Offset hängt vom Verdrehwinkel der Welle ab und ist dadurch eine Funktion des Lastmomentes und der Wellensteifigkeit. Die Korrektur des Sollwertes muss also abhängig vom Lastmoment erfolgen, was aus Sicht der praktischen Anwendung nicht tragbar ist. Das stationär einwirkende Lastmoment ist meist unbekannt, was selbst in einfachen Beispielen wie Fahrstuhl oder Rolltreppe zutrifft. Hier ist nicht bekannt, welches Personengewicht tatsächlich befördert werden muss, damit bleibt auch der zu erwartende Torsionswinkel im Antrieb unbekannt. Des Weiteren besteht eine Abhängigkeit von der Wellensteifigkeit, die ein schwer zu bestimmender Systemparameter ist und im Zusammenhang mit adaptiver Regelung als unbekannt angenommen werden muss. Aus diesen Gründen wird die Sollwertanpassung nicht weiter verfolgt, weil in nahezu allen praktischen Anwendungen ein Lastmoment auftritt.

Eine in der Regelungstechnik weit verbreitete Möglichkeit zur Lösung dieses Problems ist die Verwendung eines Führungsintegrators. Dieser verändert den Sollwert des Regelkreises solange bis stationär  $\omega_2^* = \omega_2$  gilt. Zweifellos führt ein solcher Ansatz zum Ziel (siehe Abbildung 16.19), benötigt jedoch ein hohes Maß an Systemkenntnis. Weil dem Führungsintegrator die Differenz  $x_d = \omega_2^* - \omega_2$  am Eingang zur Verfügung gestellt wird, entsteht eine zusätzliche Rückkoppelungsschleife. Dadurch könnte die Stabilität des Gesamtsystems verloren gehen. Der Beweis, dass stabiles Verhalten vorliegt, ist hier jedoch nicht möglich, weil die Regelstrecke unbekannte Parameter enthält.



**Abb. 16.19:** Regelung mit Verdrehwinkel ( $k_{\Delta\varphi} = 50$ ) und Sollwertanpassung durch einen Führungsintegrator. Damit werden nun sowohl die Schwingungen gut gedämpft, als auch die stationäre Regeldifferenz ausgeregelt.

Damit muss ein intelligenter Ansatz entwickelt werden, der bei auftretenden Schwingungen den Verdrehwinkel zur besseren Bedämpfung stark gewichtet ( $k_{\Delta\varphi}$  groß) und nach Abklingen der Schwingungen die Gewichtung zur Erzielung stationärer Genauigkeit zurücknimmt ( $k_{\Delta\varphi}$  klein). Im Grundsatz lässt sich diese Aufgabe durch einen variablen Gewichtungsfaktor, also durch eine geeignete Funktion  $k_{\Delta\varphi}(t)$  lösen. Durch die Zeitvarianz wird ein notwendiger Freiheitsgrad bereitgestellt, so dass – je nach aktueller Situation – die beiden konkurrierenden Ziele (Dämpfung und Genauigkeit) erreicht werden können.

Es ist unschwer einzusehen, dass eine geeignete Funktion  $k_{\Delta\varphi}(t)$  vor Einschalten der Regelung nicht festgelegt werden kann: hierzu müsste vorab bekannt sein, zu welchem Zeitpunkt Schwingungen entstehen werden, also wann ein Lastschlag auftritt. Aus diesem Grund wird eine dynamische Zustandsauskopplung

$$y_r(t) = k_{\omega_2}\omega_2(t) + k_{\Delta\varphi}q(t) + k_{\omega_1}\omega_1(t) \quad (16.89)$$

eingeführt [208], [108], in der die Größe  $q(t)$  aus einer Hochpassfilterung des Verdrehwinkels  $\Delta\varphi(t)$  hervorgeht. Ein solches Hochpassfilter (Abk. HPF) lässt sich beispielsweise durch die Übertragungsfunktion

$$G_F(s) = \frac{q(s)}{\Delta\varphi(s)} = \frac{Ts}{Ts + 1}, \quad T > 0 \quad (16.90)$$

darstellen. Damit gilt  $q(s) = G_F(s)\Delta\varphi(s)$ . Wird dieser Zusammenhang in die Laplacetransformierte von Gleichung (16.89) eingesetzt, ergibt sich:

$$\begin{aligned} y_r(s) &= k_{\omega_2}\omega_2(s) + k_{\Delta\varphi}G_F(s)\Delta\varphi(s) + k_{\omega_1}\omega_1(s) = \\ &= k_{\omega_2}\omega_2(s) + k_{\Delta\varphi}\frac{Ts}{Ts+1}\Delta\varphi(s) + k_{\omega_1}\omega_1(s) \end{aligned} \quad (16.91)$$

Weil darin die Übertragungsfunktion des Filters enthalten ist, wird diese Rückführung als dynamische Rückführung bezeichnet.

Im stationären Zustand ist  $s = 0$  zu setzen, so dass die Übertragungsfunktion des Filters den Ausdruck

$$G_F(s = 0) = \frac{Ts}{Ts+1} = 0 \quad (16.92)$$

liefert. Für Anregungen mit hohen Frequenzen muss der Betragsgang der Übertragungsfunktion für  $\omega \neq 0$  ausgewertet werden, was

$$|F_F(j\omega)| = \frac{|jT\omega|}{|jT\omega + 1|} = \frac{T\omega}{\sqrt{(T\omega)^2 + 1}} \quad \Rightarrow \quad \lim_{\omega \rightarrow \infty} |F_F(j\omega)| = 1 \quad (16.93)$$

und damit die Aussage

$$|F_F(j\omega)| \approx 1, \quad \forall \omega \gg 1/T \quad (16.94)$$

ergibt. Aus diesen Überlegungen lässt sich das Verhalten des Filters und dessen Auswirkung auf die Rückkopplung ableiten:

Stationär ist in der Übertragungsfunktion des Filters  $s = 0$  zu setzen. Daran ist erkennbar, dass ein konstanter Verdrehwinkel vom Filter ausgelöscht wird und daher, wie gewünscht, in der Linearkombination nicht mehr in Erscheinung tritt. Dies hat die gleiche Wirkung wie die Wahl  $k_{\Delta\varphi} = 0$ .

Sobald jedoch hochfrequente Schwingungen angeregt werden, treten diese wegen (16.94) auch in der Linearkombination auf. Durch den Hochpasscharakter der Filterung gilt :

$$k_{\Delta\varphi}q(t) \approx k_{\Delta\varphi}\Delta\varphi(t) \quad (16.95)$$

Daher sind Oszillationen auch im gefilterten Signal vorhanden, was analog zu einer Gewichtung vom Verdrehwinkel mit einer Verstärkung  $k_{\Delta\varphi} > 0$  ist. Aus diesem Grund beeinflussen Schwingungen in  $\Delta\varphi$  das Fehlersignal und können deshalb bedämpft werden.

Die Filterung des Verdrehwinkels  $\Delta\varphi$  entspricht damit genau einer geeigneten, zeitvarianten Wahl des Verstärkungsfaktors  $k_{\Delta\varphi}$ , besitzt jedoch den bedeutenden Vorteil, dass vorab keine einschränkende Festlegung getroffen werden muss.

Nachdem durch das Hochpassfilter zusätzliche Dynamik in den Regelkreis eingebbracht worden ist, wird eine erneute Stabilitätsanalyse notwendig. Es ist zu beweisen, dass die Eigenschaften des Zweimassensystems (instantane Verstärkung,

Relativgrad und Minimalphasigkeit) durch das Filter nicht berührt werden. Hierzu werden Zweimassensystem und Hochpassfilter als Single-Input-Multi-Output- bzw. als Multi-Input-Single-Output-System betrachtet, die in Serienschaltung das Gesamtsystem bilden, dessen Übertragungsfunktion zu überprüfen ist.

Mit den Eingängen  $x_1(\cdot) = \omega_2(\cdot)$ ,  $x_2(\cdot) = \Delta\varphi(\cdot)$  und  $x_3(\cdot) = \omega_1(\cdot)$  lässt sich das Übertragungsverhalten der dynamischen Zustandsrückführung im Laplacebereich durch

$$y_r(s) = G_F(s) \cdot \begin{pmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \end{pmatrix}, \quad G_F(s) = \begin{pmatrix} k_{\omega_2} & k_{\Delta\varphi} \frac{T_s}{Ts+1} & k_{\omega_1} \end{pmatrix} \quad (16.96)$$

beschreiben. Wird der Zustandsvektor als Ausgangsgröße definiert, lautet die Übertragungsfunktion des Zweimassensystems:

$$\begin{pmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \end{pmatrix} = G_Z(s) \cdot u(s), \quad G_Z(s) = (sI - A)^{-1}b \quad (16.97)$$

Die Reihenschaltung ergibt sich aus einer Multiplikation beider Übertragungsfunktionen und ist durch

$$G(s) = \frac{y_r(s)}{u(s)} = G_F(s)G_Z(s) = \begin{pmatrix} k_{\omega_2} & k_{\Delta\varphi} \frac{T_s}{Ts+1} & k_{\omega_1} \end{pmatrix} (sI - A)^{-1}b \\ = \frac{k_{\omega_1} J_2 T s^3 + [(k_{\omega_2} + k_{\omega_1}) d T + k_{\Delta\varphi} T J_2 + k_{\omega_1} J_2] s^2 + (k_{\omega_2} + k_{\omega_1})(c T + d) s + (k_{\omega_2} + k_{\omega_1}) c}{s[J_1 J_2 s^2 + d(J_1 + J_2)s + c(J_1 + J_2)](Ts + 1)} \quad (16.98)$$

festgelegt.

Daraus wird entnommen:

(i) Instantane Verstärkung:

Die instantane Verstärkung ist:

$$V_0 = \frac{k_{\omega_1} J_2 T}{J_1 J_2 T} = \frac{k_{\omega_1}}{J_1} > 0 \quad (16.99)$$

(ii) Relativgrad:

Die Differenz die Nenner- und Zählerordnung ist  $\delta = 4 - 3 = 1$ .

(iii) Minimalphasigkeit:

Wie schon in Kap. 16.9.1 ausgeführt, streben bei wachsender Reglerverstärkung  $k$  die Pole des Nennerpolynoms zu den Nullstellen des Zählerpolynoms. Die Minimalphasigkeit des Zählerpolynoms ist daher zwingend. Durch den Filter wächst der Grad des Zählers von zwei auf drei an. Im Gegensatz zur Übertragungsfunktion (16.55) (Kombination aus Zweimassensystem und PI-Baustein), die ebenfalls ein Polynom dritter Ordnung

enthält, liegt hier keine faktorisierte Darstellung vor. Daher erhöht sich hier der Aufwand für die Hurwitz-Prüfung.

Neben den Forderungen

$$\begin{aligned}
 k_{\omega_1} J_2 T > 0 &\Rightarrow k_{\omega_1} > 0 \\
 (k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2 + k_{\omega_1} J_2 > 0 &\Rightarrow k_{\Delta\varphi} > -\frac{(k_{\omega_2} + k_{\omega_1})dT + k_{\omega_1} J_2}{(TJ_2)} \\
 (k_{\omega_2} + k_{\omega_1})(cT + d) > 0 &\Rightarrow k_{\omega_2} > -k_{\omega_1} \\
 (k_{\omega_2} + k_{\omega_1})c > 0 &\Rightarrow k_{\omega_2} > -k_{\omega_1}
 \end{aligned} \tag{16.100}$$

muss gleichzeitig auch

$$(k_{\omega_2} + k_{\omega_1})(cT + d) \cdot [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2 + k_{\omega_1} J_2] - (k_{\omega_2} + k_{\omega_1})c \cdot k_{\omega_1} J_2 T > 0 \tag{16.101}$$

geprüft werden:

$$\begin{aligned}
 &(k_{\omega_2} + k_{\omega_1})(cT + d) \cdot [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2 + k_{\omega_1} J_2] \\
 &\quad - (k_{\omega_2} + k_{\omega_1})c \cdot k_{\omega_1} J_2 T \\
 &= (k_{\omega_2} + k_{\omega_1}) [cT [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2]] \\
 &\quad + d [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2 + k_{\omega_1} J_2]] \\
 &\quad + (k_{\omega_2} + k_{\omega_1})ck_{\omega_1} J_2 T - (k_{\omega_2} + k_{\omega_1})ck_{\omega_1} J_2 T \\
 &= (k_{\omega_2} + k_{\omega_1}) [cT [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2]] \\
 &\quad + d [(k_{\omega_2} + k_{\omega_1})dT + k_{\Delta\varphi} TJ_2 + k_{\omega_1} J_2]] > 0
 \end{aligned} \tag{16.102}$$

Wird also weiterhin die aus Gleichung (16.74) bekannte Forderung

$$k_{\omega_1} > 0, \quad k_{\omega_2} > -k_{\omega_1}, \quad k_{\Delta\varphi} \geq 0 \tag{16.103}$$

beibehalten, liegt ein minimalphasiges System vor.

Die Kombination aus Zweimassensystem mit PI-Baustein und die Kombination aus Zweimassensystem mit Hochpassfilter sind jeweils – wie gezeigt – high-gain-fähig. Es bleibt nun die Frage, ob beide Erweiterungen gleichzeitig am Zweimassensystem vorgenommen werden dürfen, d.h. ob die Kombination aus Zweimassensystem mit PI-Baustein und Hochpassfilter durch Funnel-Control regelbar ist.

Hierauf lässt sich auf obigen Ergebnissen basierend eine sehr einfache Antwort geben. Wird die Kombination aus Zweimassensystem mit Hochpassfilter (die bekanntermaßen high-gain-fähig ist) um den PI-Baustein (Instantane Verstärkung

$k_p > 0$ , Relativgrad null, minimalphasig) ergänzt, kann dies als Produkt der zugehörigen Übertragungsfunktionen dargestellt werden. Nachdem die Gesamtübertragungsfunktion aus einem Produkt der Zählerpolynome, sowie aus einem Produkt der Nennerpolynome besteht lässt sich ablesen, dass die

- Pole der Gesamtübertragungsfunktion durch die Vereinigungsmenge aller Pole der beiden Einzelübertragungsfunktionen gegeben sind und dass die
- Nullstellen der Gesamtübertragungsfunktion durch die Vereinigungsmenge aller Nullstellen der beiden Einzelübertragungsfunktionen gegeben sind.

Daher ist die Gesamtübertragungsfunktion genau dann minimalphasig, wenn die beiden Einzelübertragungsfunktionen minimalphasig sind. Die Multiplikation zweier Polynome ergibt ein Produktpolynom vom Grad, der der Summe der beiden Einzelordnungen entspricht. Damit ist der Relativgrad des Gesamtsystems exakt die Summe der beiden Relativgrade der Einzelsysteme. Weil der PI-Baustein den Relativgrad null hat, bleibt der Relativgrad eins des Zweimassensystems mit Filter erhalten. Ebenfalls bleibt auch das Vorzeichen der instanten Verstärkung unverändert.

Nachdem die Kombination aus Zweimassensystem mit PI-Baustein und Hochpassfilter high-gain-fähig ist, wird hier eine Simulation vorgestellt, die das Verhalten des Regelkreises eruiert und den positiven Effekt der Erweiterungsmaßnahmen hervorhebt. Weitere Simulationsstudien zu diesem Thema finden sich beispielsweise in [211].

In Abbildung 16.20 wird das Zweimassensystem geregelt, wobei der Funnel-Regler entweder durch den PI-Regler oder durch den Hochpassfilter ergänzt wird. Die Strecke befindet sich in einem konstanten Arbeitspunkt bei 10 rad/s. Zum Zeitpunkt  $t = 2$  sec wird ein Lastmoment sprungförmig auf der Abtriebsseite appliziert.

Wird der Funnel-Regler allein verwendet, bricht die Drehzahl bei Belastung erheblich ein. Der stationäre Fehler wird verringert, wenn der PI-Regler eingesetzt wird. Es verbleibt allerdings ein Regelfehler, trotz integralem Anteil. Dieser beruht auf der Tatsache, dass der Zustandsregler eine Linearkombination aller drei Zustände als Regelgröße betrachtet, und nicht die Prozessgröße  $\omega_2$ . Ebenfalls lässt sich eine Verringerung des Fehlers durch den Einsatz des Hochpassfilters erreichen, der den Term  $k_{\Delta\varphi}\Delta\varphi$  in Gleichung (16.87) auslöscht. Keine der beiden Erweiterungen ist jedoch in der Lage, für stationäre Genauigkeit zu sorgen.

Dagegen wird in Abbildung 16.21 der Funnel-Regler mit beiden Erweiterungsmaßnahmen gleichzeitig angewendet. Dadurch können beide Summanden in Gleichung (16.87) eliminiert werden, so dass erwartungsgemäß der Lastschlag zwar einen temporären Drehzahleinbruch nach sich zieht, stationär aber kein Regelfehler auftritt.

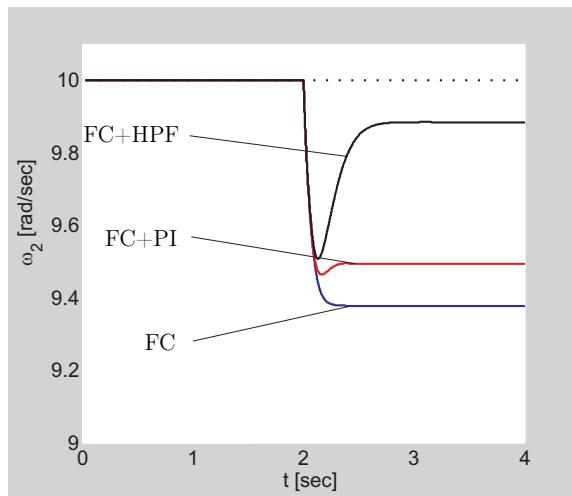


Abb. 16.20: Erweiterung des Funnel-Controllers durch internes Modell bzw. Hochpassfilter.

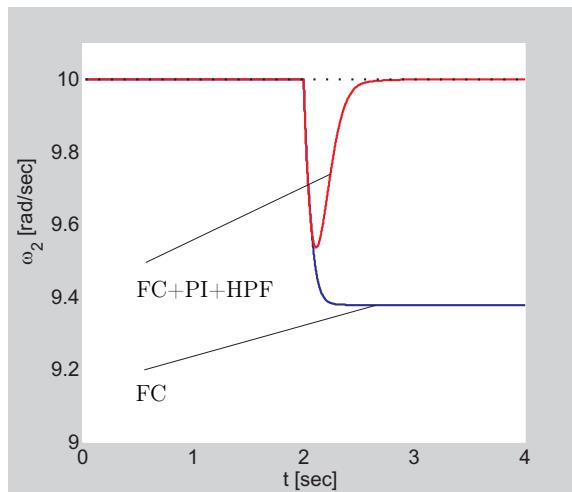


Abb. 16.21: Gleichzeitige Erweiterung des Funnel-Controllers mit internem Modell und Hochpassfilter.

## 16.10 Funnel-Regelung für das nichtlineare Zweimassensystem

Wird für eine hinreichend gute Modellierung die Berücksichtigung von Nichtlinearitäten erforderlich, muss das lineare Modell (16.44), (16.45) um die entspre-

chenden Terme ergänzt werden. Gewöhnlicherweise ist Reibung der dominante unter den nichtlinearen Effekten. Deshalb wird das lineare Modell durch einen nichtlinearen Reibterm erweitert. Dabei soll die Charakteristik der Reibkennlinie keine Rolle spielen, so dass weder eine aufwändige Identifikation, noch Modellierung notwendig wird. Um einer sehr weit gefassten Klasse von Reibkennlinien (selbst Hystereseffekte, wie der in der Tribologie bekannte time-lag effect oder frictional memory [9] sind in der Beschreibung eingeschlossen) zugänglich zu sein, wird die Reibung als dynamischer Operator dargestellt. Jener bildet eine Funktion (in diesem Falle den Drehzahlverlauf der Lastmasse) auf eine weitere Funktion ab, welche das zugehörige Reibmoment beschreibt.

Ein solcher Operator ist mathematisch darstellbar durch:

$$\mathbf{N} : \mathcal{C}([0, \infty[; \mathbb{R}) \rightarrow \mathcal{L}^\infty([0, \infty[; \mathbb{R}) \quad (16.104)$$

In Worten ausgedrückt bedeutet dies, dass eine stetige reelle Funktion (die auf der positiven reellen Zeitachse definiert ist) in eine beschränkte reelle Funktion abgebildet wird, die ebenfalls auf der positiven reellen Zeitachse definiert ist. Zusätzlich wird an den Operator eine globale Beschränktheitsbedingung gestellt [108]:

$$\sup \left\{ |(\mathbf{N}\zeta)(t)| \mid t \geq 0, \zeta \in \mathcal{C}([0, \infty[; \mathbb{R}) \right\} < \infty \quad (16.105)$$

Jene besagt, dass jede stetige Testfunktion  $\zeta(t)$ , die auf der positiven reellen Zeitachse definiert ist, in eine Funktion  $(\mathbf{N}\zeta)(t)$  abgebildet wird, deren höchster Betrag (das Supremum) beschränkt ist. Dabei sei explizit betont, dass die Testfunktion  $\zeta(t)$  selbst nicht beschränkt sein muss. Beispielsweise die unbeschränkt anwachsende Exponentialfunktion  $\zeta(t) := \exp\{t\}$  wäre ein möglicher Kandidat.

Diese Bedingung verbietet offensichtlich, mit dem Operator  $\mathbf{N}$  viskose Reibanteile zu modellieren. Daher wird viskose Reibung durch  $\nu_V \omega_2(t)$  mit einem nicht-negativen Gleitreibungskoeffizienten  $\nu_V \geq 0$  beschrieben und zum linearen Teil der Differentialgleichung hinzugefügt [208].

Schlussendlich entsteht durch diese Modifikation das nichtlineare Modell des Zweimassensystems :

$$\begin{aligned} \dot{\underline{x}}(t) &= \mathbf{A} \underline{x}(t) + \underline{b}_L [\nu_V \omega_2(t) + (\mathbf{N}\omega_2)(t)] + \underline{b}[u(t) + z(t)], & \underline{x}(0) \in \mathbb{R}^3 \\ y_r(t) &= \underline{c}^T \underline{x}(t) \end{aligned} \quad (16.106)$$

Weiter verbessert wird die Realitätsnähe des Modells, indem am Stelleingang  $u(t)$  eine additive Störgröße  $z(t)$  berücksichtigt wird [108].

Damit die dynamische Zustandsrückführung mittels Hochpassfilter auch im nichtlinearen Zweimassensystem angesetzt werden kann, muss die Beschreibung mit Hilfe einer Übertragungsfunktion vermieden werden und stattdessen eine Zustandsbeschreibung gewählt werden. Eine mögliche Realisierung für (16.90) lautet:

$$\begin{aligned} \dot{x}_F(t) &= -\frac{1}{T_F} x_F(t) + \Delta\varphi(t), & x_F(0) = 0 \\ q(t) &= -\frac{1}{T_F} x_F(t) + \Delta\varphi(t), & T_F > 0 \end{aligned} \quad (16.107)$$

Damit kann das Zweimassensystem zusammen mit dem Filter durch folgende Differentialgleichung vierter Ordnung beschrieben werden [208]:

$$\begin{aligned} \begin{pmatrix} \dot{\omega}_2(t) \\ \Delta\dot{\varphi}(t) \\ \dot{\omega}_1(t) \\ \dot{x}_F(t) \end{pmatrix} &= \underbrace{\begin{bmatrix} -(d + \nu_V)/J_2 & c/J_2 & d/J_2 & 0 \\ -1 & 0 & 1 & 0 \\ d/J_1 & -c/J_1 & -d/J_1 & 0 \\ 0 & 1 & 0 & -1/T_F \end{bmatrix}}_{:= \mathbf{A}_F} \begin{pmatrix} \omega_2(t) \\ \Delta\varphi(t) \\ \omega_1(t) \\ x_F(t) \end{pmatrix} \\ &+ \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} [(\mathbf{N}\omega_2)(t) + M_2(t)] + \begin{pmatrix} 0 \\ 0 \\ 1/J_1 \\ 0 \end{pmatrix} [u(t) + z(t)] \\ y_r(t) &= (k_{\omega_2} \quad k_{\Delta\varphi} \quad k_{\omega_1} \quad -k_{\Delta\varphi}/T_F) \begin{pmatrix} \omega_2(t) \\ \Delta\varphi(t) \\ \omega_1(t) \\ x_F(t) \end{pmatrix} \end{aligned} \quad (16.108)$$

Der nichtlineare Reiboperator  $\mathbf{N}$  enthält als Argument weiterhin den physikalischen Systemausgang, der identisch mit der Drehzahl  $\omega_2$  ist.

Für die Betrachtung nichtlinearer Systeme können Übertragungsfunktionen nicht herangezogen und mittels Hurwitztest auf Minimalphasigkeit hin geprüft werden. Insbesondere tritt bei nichtlinearen Systemen an die Stelle der Minimalphasigkeit die Stabilität der Nulldynamik. Zur Bestimmung der Nulldynamik ist eine Transformation in BINF (siehe Kapitel 12) zweckmäßig. Anschließend wird deren dynamisches Verhalten, d.h. die Stabilitätseigenschaft der Nulldynamik untersucht. Ebenso muss durch den Wegfall der Übertragungsfunktion die instantane Verstärkung und der Relativgrad auf anderem Wege berechnet werden. An dieser Stelle werden die Ausführungen in Kapitel 12 genutzt, aus welchen die Berechnungsvorschrift unmittelbar hervorgeht.

### Relativgrad und Instantane Verstärkung

Für die Anwendung von Funnel-Control muss die Regelstrecke einen Relativen Grad von Eins besitzen. Ohne Hochpassfilter ist dies erfüllt, es muss nun überprüft werden, ob der Filter den Relativgrad erhöht. Aus der Definition des Relativgrades für nichtlineare Systeme folgt, dass das Eingangssignal  $u$  die erste Ableitung von  $y_r$  beeinflussen muss, damit der Relativgrad gleich eins ist. Da der Ausgang  $y_r = \underline{c}^T \underline{x}$  ist, gilt für dessen Ableitung:

$$\begin{aligned}
y_r &= (k_{\omega_2} \ k_{\Delta\varphi} \ k_{\omega_1} \ -k_{\Delta\varphi}/T_F) (\dot{\omega}_2(t) \ \Delta\dot{\varphi}(t) \ \dot{\omega}_1(t) \ \dot{x}_F(t))^T \\
&= \begin{pmatrix} k_{\omega_2} \\ k_{\Delta\varphi} \\ k_{\omega_1} \\ -k_{\Delta\varphi}/T_F \end{pmatrix}^T \begin{bmatrix} -(d + \nu_V)/J_2 & c/J_2 & d/J_2 & 0 \\ -1 & 0 & 1 & 0 \\ d/J_1 & -c/J_1 & -d/J_1 & 0 \\ 0 & 1 & 0 & -1/T_F \end{bmatrix} \begin{pmatrix} \omega_2(t) \\ \Delta\varphi(t) \\ \omega_1(t) \\ x_F(t) \end{pmatrix} \\
&+ \begin{pmatrix} k_{\omega_2} \\ k_{\Delta\varphi} \\ k_{\omega_1} \\ -k_{\Delta\varphi}/T_F \end{pmatrix}^T \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} [(\mathbf{N}\omega_2)(t) + M_2(t)] \\
&+ \begin{pmatrix} k_{\omega_2} \\ k_{\Delta\varphi} \\ k_{\omega_1} \\ -k_{\Delta\varphi}/T_F \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ 1/J_1 \\ 0 \end{pmatrix} [u(t) + z(t)]
\end{aligned}$$

Nachdem die Stellgröße  $u(t)$  mit dem Faktor  $k_{\omega_1}/J_1$  gewichtet wird, erhält der Eingang Einfluss auf die erste Ableitung, wenn  $k_{\omega_1} \neq 0$  gewählt wird. Der Faktor  $k_{\omega_1}/J_1$  ist gleichzeitig auch die Instantane Verstärkung des Systems und positiv, wenn  $k_{\omega_1} > 0$  gewählt wird. Damit ist gezeigt, dass der Relativgrad auch im nichtlinearen Fall durch den Filter nicht verändert wird und weiterhin bei eins bleibt.

### Minimalphasigkeit bzw. Stabilität der Nulldynamik

Hier wird die Frage, ob der Filter zulässig ist, deutlich schwieriger. Da es sich um ein nichtlineares System handelt, muss dazu die Stabilität der Nulldynamik nachgewiesen werden. Um die Nulldynamik isolieren zu können, erfolgt zunächst eine Transformation des Gesamtsystems in Byrnes-Isidori-Normalform. Anschließend wird das Verhalten der Nulldynamik untersucht.

Um in Byrnes-Isidori-Normalform zu gelangen, wird eine invertierbare Transformationsmatrix  $\Phi$  bestimmt, die eine Koordinatentransformation der Art

$$\underline{z} = \Phi \cdot \underline{x} \quad \Leftrightarrow \quad \underline{x} = \Phi^{-1} \cdot \underline{z} \quad (16.109)$$

vornimmt. Die Invertierbarkeit ist notwendig um eine ein-eindeutige Abbildungsvorschrift zu erhalten, also um eine Rücktransformation in  $\underline{x}$ -Koordinaten zu ermöglichen.

Nach Isidori [115] kann

$$\Phi = \begin{bmatrix} k_{\omega_2} & k_{\Delta\varphi} & k_{\omega_1} & -k_{\Delta\varphi}/T \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16.110)$$

als Transformationsmatrix gewählt werden. Damit besitzt das transformierte System in  $\underline{z}$ -Koordinaten die Struktur,

$$\dot{z}_1 = \beta(\underline{z}) + \alpha(\underline{z}) \cdot u \quad (16.111)$$

$$\dot{z}_2 = q_2(\underline{z})$$

$$\dot{z}_3 = q_3(\underline{z})$$

$$\dot{z}_4 = q_4(\underline{z})$$

$$y = (1 \ 0 \ 0 \ 0) z = z_1 \quad (16.112)$$

wobei sich die Größen  $\alpha$ ,  $\beta$  und  $q_i$  aus

$$\alpha(z) = \frac{k_{\omega_1}}{J_1} \quad (16.113)$$

$$\beta(z) = \begin{pmatrix} k_{\omega_2} \\ k_{\Delta\varphi} \\ k_{\omega_1} \\ -k_{\Delta\varphi}/T_F \end{pmatrix}^T \mathbf{A}_F \Phi^{-1} z + \begin{pmatrix} k_{\omega_2} \\ k_{\Delta\varphi} \\ k_{\omega_1} \\ -k_{\Delta\varphi}/T_F \end{pmatrix}^T \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} (\mathbf{N}z_2)(t) \quad (16.114)$$

$$q_i(z) = \phi_i \mathbf{A}_F \Phi^{-1} z + \phi_i \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \\ 0 \end{pmatrix} (\mathbf{N}z_2)(t) \quad (16.115)$$

$$\phi_1 = (1 \ 0 \ 0 \ 0) \quad (16.116)$$

$$\phi_2 = (0 \ 1 \ 0 \ 0) \quad (16.117)$$

$$\phi_3 = (0 \ 0 \ 0 \ 1) \quad (16.118)$$

errechnen. Das Argument der Reibung  $(\mathbf{N}z_2)(t)$  ist die Drehzahl der Lastmasse  $\omega_2$ , in  $\underline{x}$ -Koordinaten also  $x_1$ . Die Umrechnung von  $x_1$  in  $\underline{z}$ -Koordinaten ergibt gemäß der gewählten Transformationsvorschrift:

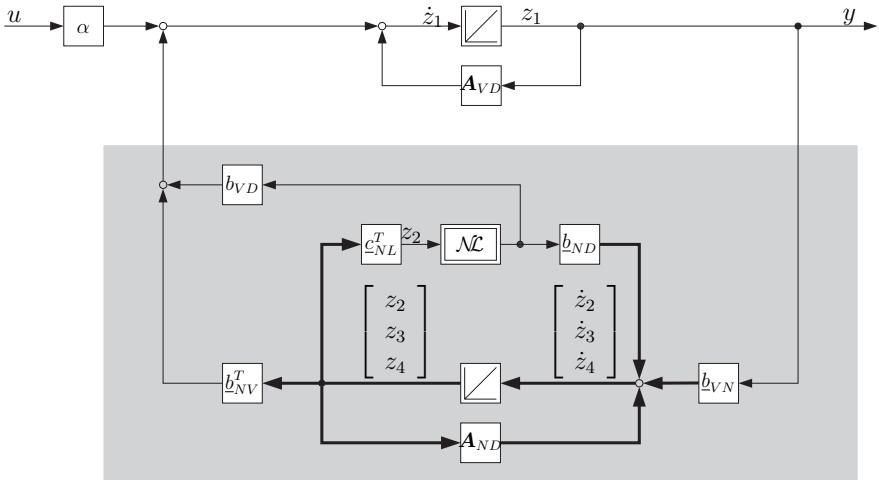
$$x_1 = [1 \ 0 \ 0 \ 0] \cdot \underline{x} = [1 \ 0 \ 0 \ 0] \Phi^{-1} \underline{z} = z_2 \quad (16.119)$$

Daher muss im  $\underline{z}$ -Koordinatensystem der Zustand  $z_2$  als Argument in die Nichtlinearität eingesetzt werden.

Das transformierte System setzt sich aus einem linearen Teil und einem nichtlinearen Anteil zusammen. Um die Nulldynamik zu isolieren, muss dieses System in eine "Integratorkette" (hier bestehend aus *einem* Integrator im Vorwärtszweig) und Nulldynamik aufgeteilt werden. Der Relativgrad gibt die Ordnung der Vorwärtsdynamik an, die deshalb nur den einen Zustand  $z_1$  besitzt. Die Nulldynamik enthält die restlichen drei Zustände  $z_2$  bis  $z_4$ :

$$\begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{VD} & \vdots & \underline{\mathbf{b}}_{NV}^T \\ \cdots & \cdots & \cdots \\ \underline{\mathbf{b}}_{VN} & \vdots & \mathbf{A}_{ND} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} b_{VD} \\ \vdots \\ b_{ND} \end{bmatrix} (\mathbf{N}z_2)(t) + \begin{bmatrix} \alpha \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (16.120)$$

Der Einkoppelvektor für die Nichtlinearität besteht aus zwei Anteilen. Zum einen



**Abb. 16.22:** Signalflussplan des Gesamtsystems, bestehend aus nichtlinearem Zweimassensystem und  $DT_1$ -Filter, in Byrnes-Isidori-Normalform.

die Einwirkung auf die Vorwärtsdynamik  $b_{VD} = -k_{\omega_2}/J_2$ , zum anderen aus dem Vektor

$$b_{ND} = \begin{bmatrix} -1/J_2 \\ 0 \\ 0 \end{bmatrix} \quad (16.121)$$

der den Einfluss der Nichtlinearität auf die Nulldynamik beschreibt. Da die Nichtlinearität den Zustand  $z_2$  als Argument enthält, muss mit dem Vektor

$$c_{NL}^T = [1 \quad 0 \quad 0] \quad (16.122)$$

dieser eine Zustand aus dem gesamten Zustandsvektor der Nulldynamik selektiert werden. Der lineare Anteil der Zustandsbeschreibung wird von einer Matrix bestimmt, die sich aus dem Eigenwert der Vorwärtsdynamik  $\mathbf{A}_{VD}$ , der Zustandsmatrix der Nulldynamik  $\mathbf{A}_{ND}$ , und den beiden Verkopplungen zwischen Vorwärts- und Nulldynamik zusammensetzt. Dabei beschreibt  $b_{NV}$  den Einfluss der Nulldynamik auf die Vorwärtsdynamik, wohingegen der Einfluss vom Systemausgang

$y = z_1$  auf die Nulldynamik durch  $\underline{b}_{VN}$  berücksichtigt wird. Die Einträge der Zustandsmatrix

$$\begin{bmatrix} \mathbf{A}_{VD} & \vdots & \underline{b}_{NV}^T \\ \cdots & \cdot & \cdots \\ \underline{b}_{VN} & \vdots & \mathbf{A}_{ND} \end{bmatrix} = \begin{bmatrix} \beta_{lin,1} & \vdots & \beta_{lin,2} & \beta_{lin,3} & \beta_{lin,4} \\ \cdots & \cdot & \cdots & \cdots & \cdots \\ q_{2,lin,1} & \vdots & q_{2,lin,2} & q_{2,lin,3} & q_{2,lin,4} \\ q_{3,lin,1} & \vdots & q_{3,lin,2} & q_{3,lin,3} & q_{3,lin,4} \\ q_{4,lin,1} & \vdots & q_{4,lin,2} & q_{4,lin,3} & q_{4,lin,4} \end{bmatrix} \quad (16.123)$$

können zeilenweise zu den Vektoren

$$\begin{aligned} \beta_{lin} &= [\beta_{lin,1} \quad \beta_{lin,2} \quad \beta_{lin,3} \quad \beta_{lin,4}] = [k_{\omega_2} \quad k_{\Delta\varphi} \quad k_{\omega_1} \quad -k_{\Delta\varphi}/T] \mathbf{A}_F \Phi^{-1} \\ &\quad (16.124) \end{aligned}$$

$$q_{2,lin} = [q_{2,lin,1} \quad q_{2,lin,2} \quad q_{2,lin,3} \quad q_{2,lin,4}] = [1 \quad 0 \quad 0 \quad 0] \mathbf{A}_F \Phi^{-1}$$

$$q_{3,lin} = [q_{3,lin,1} \quad q_{3,lin,2} \quad q_{3,lin,3} \quad q_{3,lin,4}] = [0 \quad 1 \quad 0 \quad 0] \mathbf{A}_F \Phi^{-1}$$

$$q_{4,lin} = [q_{4,lin,1} \quad q_{4,lin,2} \quad q_{4,lin,3} \quad q_{4,lin,4}] = [0 \quad 0 \quad 0 \quad 1] \mathbf{A}_F \Phi^{-1}$$

zusammengefasst werden. Damit ergibt sich für die vier Blöcke der obenstehenden Matrix:

Vorwärtsgleichungen:

$$\mathbf{A}_{VD} = \beta_{lin,1} = \frac{1}{k_{\omega_1}} \left( \frac{k_{\omega_2} d}{J_2} - \frac{k_{\omega_1} d}{J_1} + k_{\Delta\varphi} \right) \quad (16.125)$$

Nulldynamik:

$$\begin{aligned} \mathbf{A}_{ND} &= \begin{bmatrix} q_{2,lin,2} & q_{2,lin,3} & q_{2,lin,4} \\ q_{3,lin,2} & q_{3,lin,3} & q_{3,lin,4} \\ q_{4,lin,2} & q_{4,lin,3} & q_{4,lin,4} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{d}{J_2} \left( 1 + \frac{k_{\omega_2}}{k_{\omega_1}} \right) & \frac{1}{J_2} \left( c - d \frac{k_{\Delta\varphi}}{k_{\omega_1}} \right) & \frac{d}{J_2} \frac{1}{T} \frac{k_{\Delta\varphi}}{k_{\omega_1}} \\ -1 - \frac{k_{\omega_2}}{k_{\omega_1}} & -\frac{k_{\Delta\varphi}}{k_{\omega_1}} & \frac{1}{T} \frac{k_{\Delta\varphi}}{k_{\omega_1}} \\ 0 & 1 & -\frac{1}{T} \end{bmatrix} \quad (16.126) \end{aligned}$$

Auskoppelvektor der Nulldynamik:

$$\underline{b}_{NV}^T = [\beta_{lin,2} \quad \beta_{lin,3} \quad \beta_{lin,4}] \quad (16.127)$$

Einkoppelvektor der Nulldynamik:

$$\underline{b}_{VN} = \begin{bmatrix} q_{2,lin,1} \\ q_{3,lin,1} \\ q_{4,lin,1} \end{bmatrix} = \frac{1}{k_{\omega_1}} \begin{bmatrix} \frac{d}{J_2} \\ 1 \\ 0 \end{bmatrix} \quad (16.128)$$

Um eine komprimierte Darstellung der Nulldynamik zu erhalten, werden die Substitutionen

$$x := k_{\omega_2}/k_{\omega_1} \quad y := k_{\Delta\varphi}/k_{\omega_1} \quad a := d/J_2 \quad b := c/J_2 \quad \tau := 1/T \quad (16.129)$$

eingesetzt. Damit verkürzt sich die zu untersuchende Nulldynamik zu:

$$\begin{bmatrix} \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \underbrace{\begin{bmatrix} -a(1+x) - c & b - ya & ya\tau \\ -(1+x) & -y & y\tau \\ 0 & 1 & -\tau \end{bmatrix}}_{\mathbf{A}_{ND}} \cdot \begin{bmatrix} z_2 \\ z_3 \\ z_4 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{-1}{J_2} \\ 0 \\ 0 \end{bmatrix}}_{b_{ND}} \cdot (\mathbf{N} z_2) + \underbrace{\frac{1}{k_{\omega_1}} \begin{bmatrix} a \\ 1 \\ 0 \end{bmatrix}}_{b_{VN}} \underbrace{y}_{z_1} \quad (16.130)$$

Die Physik fordert positive Werte für die unbekannten Systemparameter  $J_1$ ,  $J_2$ ,  $c$  und  $d$ . Das ist das einzige Wissen, das für die Auslegung der Rückführungen  $c_i$  notwendig ist. Unter der Annahme, dass die frei wählbaren Design-Parameter  $k_{\omega_2}$ ,  $k_{\Delta\varphi}$ ,  $k_{\omega_1}$  und  $T$  auch positiv angesetzt werden, sind sämtliche Variable in der Systemmatrix  $\mathbf{A}_{ND}$  positiv.

Das Gesamtsystem (Zweimassensystem mit Filter) kann durch Funnel-Control geregelt werden, wenn sich sämtliche Eigenwerte der Matrix  $\mathbf{A}_{ND}$  in der linken Halbebene befinden. Da eine direkte Untersuchung der Eigenwertrealteile sehr aufwändig ist, und in diesem Fall in unübersichtlichen Gleichungen endet, wird stattdessen das charakteristische Polynom mittels Hurwitz-Test überprüft. Das charakteristische Polynom lautet:

$$\begin{aligned} P(s) = s^3 &+ \underbrace{[\tau + (a + ax + y + c)]s^2}_{=:k_1>0} \\ &+ \underbrace{[(xb + b) + (cy + ax\tau + a\tau + ct)]s + xb\tau + b\tau}_{=:k_2>0} \end{aligned} \quad (16.131)$$

Dabei werden nur positive Größen addiert, somit sind alle Koeffizienten positiv. Daher ist für Stabilität nach Hurwitz notwendig und hinreichend, wenn:

$$[\tau + k_1] \cdot [(xb + b) + k_2] - (xb\tau + b\tau) > 0 \quad (16.132)$$

Da die negativen Summanden  $-xb\tau$  und  $-b\tau$  auch mit positivem Vorzeichen enthalten sind, bleiben nach Ausmultiplizieren und Kürzen lediglich positive Terme übrig:

$$[\tau + k_1] \cdot [(xb + b) + k_2] - (xb\tau + b\tau) = \tau k_2 + k_1(xb + b) + k_1 k_2 > 0 \quad (16.133)$$

Damit ist die Stabilität des linearen Anteiles der Nulldynamik bewiesen, d.h.:

$$\operatorname{Re}\{\lambda_i\} < 0 \quad \text{für alle Eigenwerte } \lambda_i \text{ von } A_{ND} \quad (16.134)$$

Somit ist klar, dass der lineare Anteil der Nulldynamik asymptotisch stabil ist. Zusätzlich wird die Dynamik aber noch von der Nichtlinearität  $(Nz_2)(t)$  beeinflusst, die über den Einkoppelvektor

$$b_{ND} = \begin{pmatrix} \frac{-1}{J_2} & 0 & 0 \end{pmatrix}^T \quad (16.135)$$

einwirkt. Nachdem der Operator  $N$  per Voraussetzung (16.105) die globale Beschränktheit erfüllt, darf die Größe  $(Nz_2)(t)$  als beschränktes Eingangssignal auf ein lineares, asymptotisch stabiles System verstanden werden. Bekanntermaßen folgt bei linearen Systemen aus asymptotischer Stabilität unmittelbar BIBO-Stabilität. Damit kann das beschränkte Eingangssignal nicht destabilisierend wirken und die drei Zustandsgrößen  $z_2$ ,  $z_3$  und  $z_4$  verlaufen auf beschränkten Trajektorien.

Aus diesem Grund erfüllt die vorgeschlagene Zustandsauskopplung mit HPF aus dem Zweimassensystem mit Reibung die Bedingungen, die Funnel-Control an die Regelstrecke stellt. Diese Form der dynamischen Zustandsrückführung kann deshalb mit Funnel-Control kombiniert werden, ohne dabei Stabilitätsprobleme zu verursachen.

Die Zustandsregelung mit Funnel-Control soll nun am nichtlinearen Zweimasensystem erprobt werden. Das Simulationsmodell ist nichtlinear und enthält die Reibkennlinie

$$N : \omega_2(\cdot) \mapsto \frac{2}{\pi} \cdot \operatorname{atan}(10 \omega_2(\cdot)). \quad (16.136)$$

Um stationäre Genauigkeit<sup>3)</sup> zu erhalten, wird zunächst  $k_{\Delta\varphi} = 0$  gewählt. Es ist aus der Diskussion des linearen Systems bekannt, dass ohne Berücksichtigung des Torsionswinkels, d.h. mit einer Zustandsauskopplung

$$y_r(t) = (1 \ 0 \ 1) \underline{x}(t) \quad (16.137)$$

die Möglichkeit besteht, nicht nur den Regelfehler beliebig zu verkleinern, sondern ebenso auch die Abweichung  $|\omega_2^*(t) - \omega_2(t)|$  zwischen Sollwert

$$\omega_2^*(t) = \begin{cases} 0 & \forall t < 0 \\ 1 & \forall t \geq 0 \end{cases} \quad (16.138)$$

und Arbeitsmaschinendrehzahl  $\omega_2(t)$ . Dem Funnel-Regler

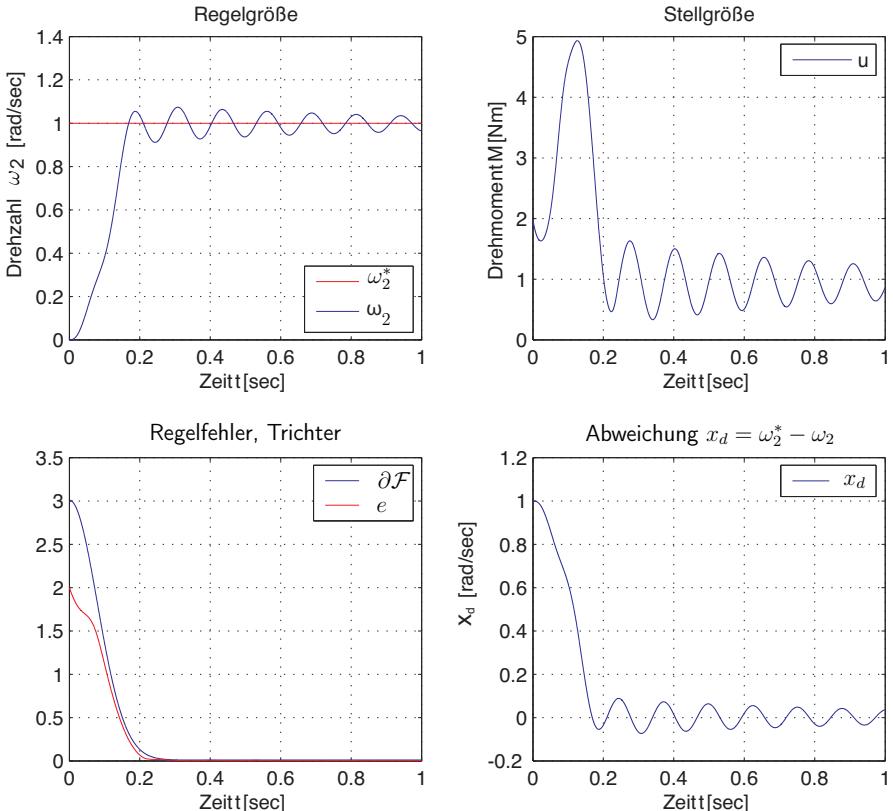
<sup>3)</sup> Im Gegensatz zur in der Literatur gebräuchlichen Verwendung bezeichnet hierbei der Begriff „stationäre Genauigkeit“ nicht einen verschwindenden Regelfehler, sondern einen bleibenden Regelfehler, der allerdings durch eine entsprechende Wahl des Trichters beliebig verkleinert werden kann. Insofern meint „genau“ in diesem Zusammenhang nicht „genau Null“, sondern „unsichtbar klein“ oder „zufriedenstellend klein“.

$$u(t) = k(t)e(t) \text{ mit } k(t) = \frac{1}{\partial\mathcal{F}(t) - |e(t)|} \text{ und } e(t) = V\omega_2^*(t) - y_r(t) \neq x_d(t) \quad (16.139)$$

wird der Trichter

$$\partial\mathcal{F}(t) = 3 \cdot e^{-80t^2} + 0.01 \quad (16.140)$$

vorgegeben. Für „große“ Zeiten ist der Regelfehler durch den Wert 0.01 begrenzt. Mit den Reglerparametern  $k_{\Delta\varphi} = 0$  und  $k_{\omega_2} = k_{\omega_1} = 1$  ergibt sich:

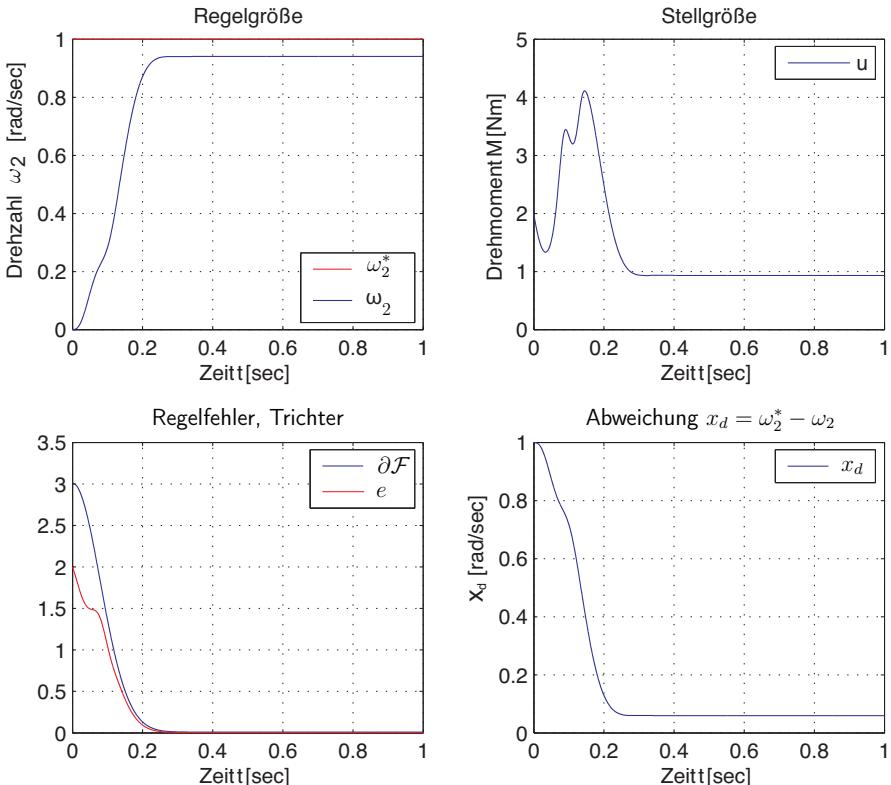


**Abb. 16.23:** Regelung des Zweimassensystems ohne Berücksichtigung des Verdrehwinkels mittels Zustandsauskopplung  $\underline{c}^T = (1 \ 0 \ 1)$ .

$$|e(t)| < 0.01 \Rightarrow |x_d(t)| = |\omega_2^*(t) - \omega_2(t)| = \frac{|e(t)|}{k_{\omega_2} + k_{\omega_1}} < \frac{0.01}{2} = 0.005 \quad (16.141)$$

Somit kann maximal ein Unterschied zwischen Sollwert und Lastdrehzahl von 0.5% des Sollwertes auftreten. Eine derart geringe Drehzahlabweichung ist im

Diagramm 16.23 wegen der Zeichengenauigkeit nicht erkennbar und wird als tolerierbar angenommen. Als Nachteil führt  $k_{\Delta\varphi} = 0$  auf einen Regler, der Oszillationen nicht aktiv bedämpfen kann.



**Abb. 16.24:** Regelung des Zweimassensystems mit Berücksichtigung des Verdrehwinkels mittels Zustandsauskopplung  $c = (1 \ 50 \ 1)$ ;  $k_{\Delta\varphi} = 50$ , konstant.

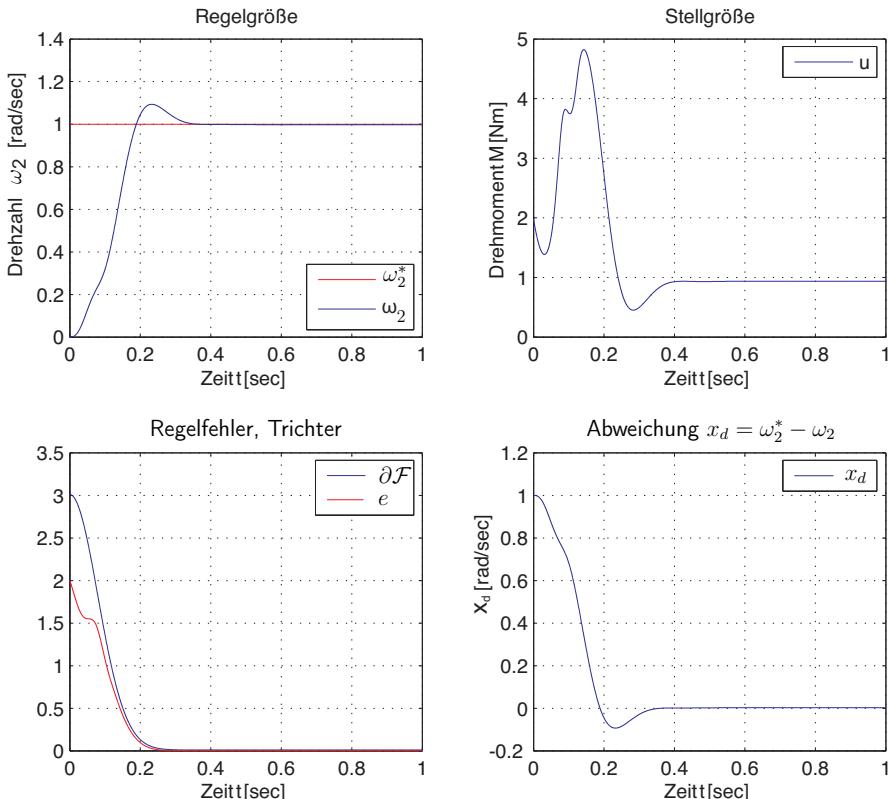
Analog zum Fall des linearen Zweimassensystems ohne Reibeinfluss kann eine ausreichend hohe Dämpfung erzielt werden, wenn beispielsweise  $k_{\Delta\varphi} = 50$  konstant gesetzt wird. Mit dieser Änderung ergibt sich die Auskopplung:

$$y_r(t) = (1 \ 50 \ 1) \underline{x}(t) \quad (16.142)$$

In Abbildung 16.24 sind die störenden Schwingungen verschwunden, allerdings tritt nun eine deutliche Drehzahlabweichung zu Tage. Weil ein proportionaler Zusammenhang zwischen  $e(t)$  und  $x_d(t) = \omega_2^*(t) - \omega_2(t)$  ausschließlich für den Sonderfall  $k_{\Delta\varphi} = 0$  vorliegt, lässt sich durch Änderung des Trichters keine Verbesserung erreichen.

Durch die Wahl  $k_{\Delta\varphi} = 50$  werden die mechanischen Schwingungen der Welle zwar bedämpft. Allerdings kann der Regelfehler nicht beliebig klein gehalten werden. Dadurch ist stationäre Genauigkeit nicht mehr erreichbar.

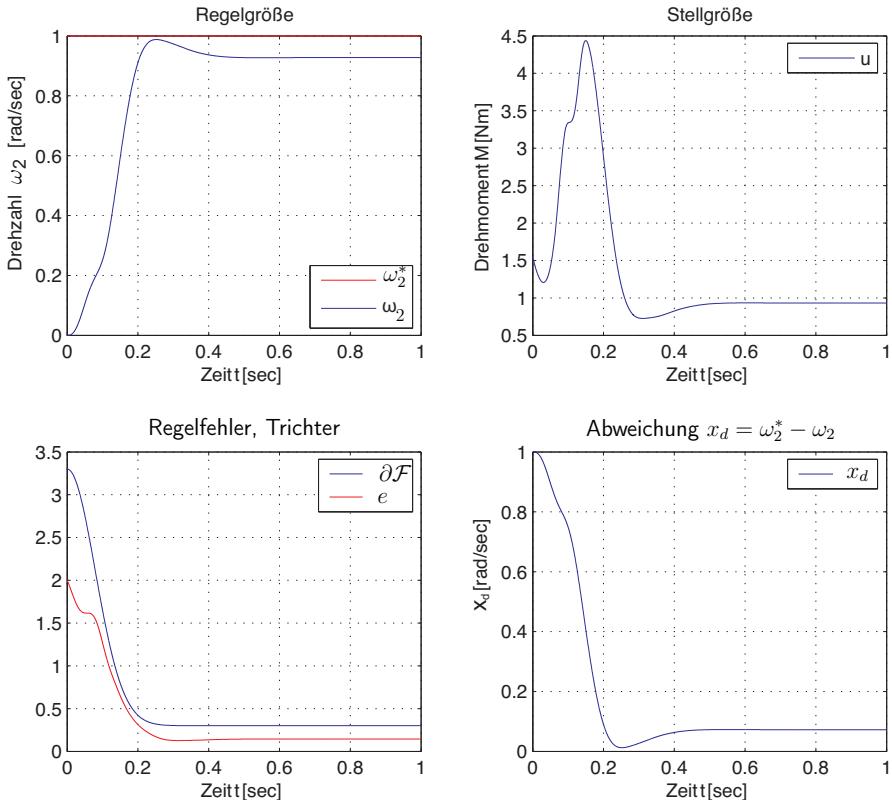
Nun wird der Hochpassfilter eingesetzt, der den Widerspruch zwischen Dämpfen und kleinem Fehler beseitigt. In Abbildung 16.25 ist zu sehen, dass keine Schwingungen in der Regelgröße vorhanden sind. Der Filter gewährleistet, dass im stationären Zustand ein proportionaler Zusammenhang zwischen  $e$  und  $x_d = \omega_2^* - \omega_2$  besteht. Daher besteht zwar ein bleibender Regelfehler, dieser kann jedoch durch einen sehr schmal ausgelegten Trichter beliebig klein gehalten werden.



**Abb. 16.25:** Auflösung des Zielkonfliktes: Ausgeprägte Bedämpfung des schwingungsfähigen Systems und beliebig kleiner Regelfehler durch den Einsatz des Hochpassfilters.

Nachdem bislang lediglich ein proportionaler Regler eingesetzt wurde, kann die Regeldifferenz stationär bei Belastung nicht verschwinden. Durch den ge-

wählten Ansatz kann jedoch durch einen engen Trichter die Größe  $e$  und damit auch  $x_d(t) = \omega_2^*(t) - \omega_2(t)$  klein gehalten werden. Dies ist allerdings nur möglich, solange kein Messrauschen wirksam ist. Ansonsten muss der Trichter das Rauschsignal umschließen, und muss hierfür eine minimale Breite aufweisen. Dies zieht eine erhöhte Abweichung in  $e$  und damit auch in  $x_d(t) = \omega_2^*(t) - \omega_2(t)$  nach sich. In Abbildung 16.26 verbleibt eine große Regeldifferenz, da sich der Trichter nicht weit genug verengt.



**Abb. 16.26:** Die Verwendung eines breiten Trichters erlaubt naturgemäß einen großen Regelfehler.

Weil der Simulation in Abbildung 16.26 der Wert  $\Delta = 0.3$  zugrundegelegt ist, verjüngt sich der Trichter nicht in ausreichendem Maße und erlaubt einen beträchtlichen Regelfehler.

Um solche Situationen zu vermeiden wird wiederum ein PI-Regler

$$\begin{aligned} \dot{\xi}(t) &= v(t), & \xi(0) &= 0 \\ u(t) &= k_I \xi(t) + k_P v(t), & k_I, k_P > 0 \end{aligned} \quad (16.143)$$

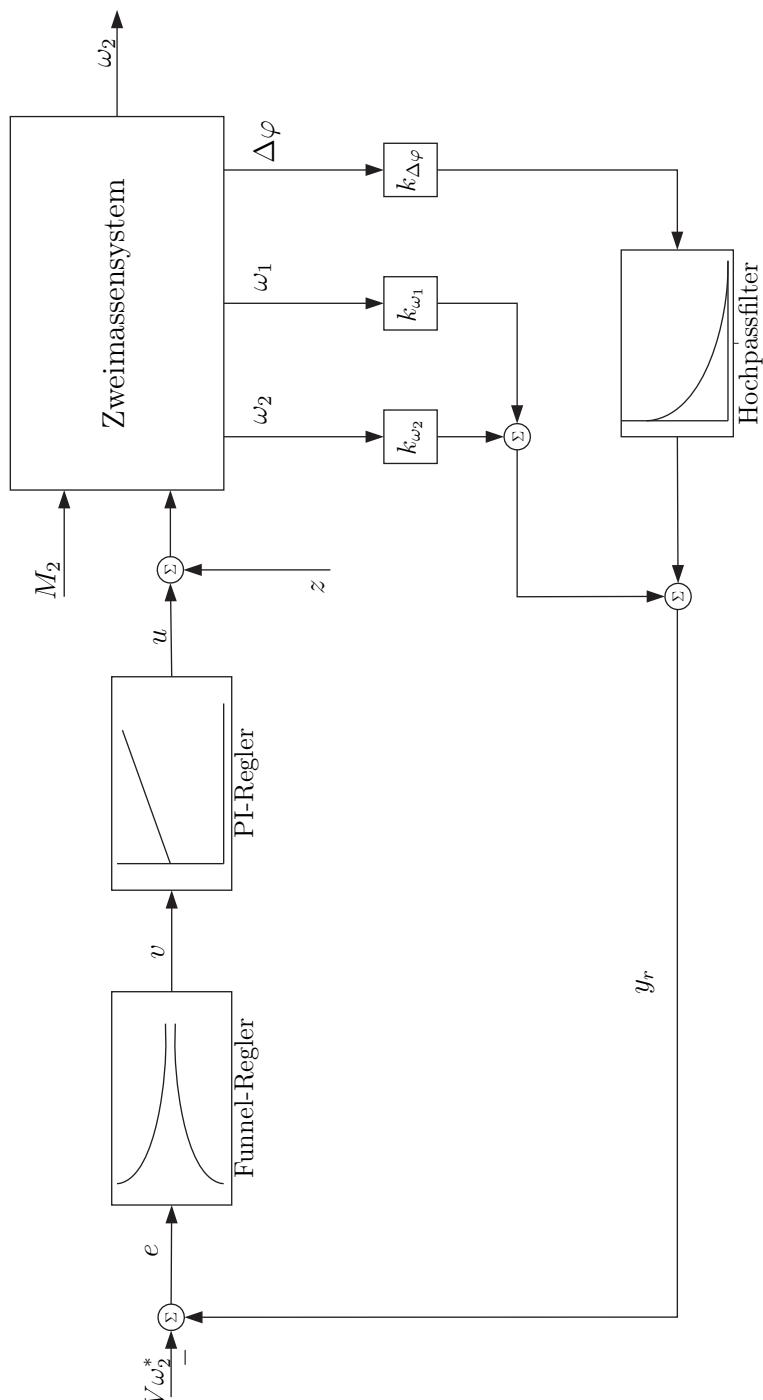


Abb. 16.27: Signalflussplan des gesamten Regelkreises.

als Internes Modell hinzugefügt. Die daraus resultierende Struktur der Regelung ist in Abbildung 16.27 gezeigt. Damit dieser minimalphasig ist, müssen  $k_i$  und  $k_p$  gleiches Vorzeichen besitzen.

Analog zur Vorgehensweise im linearen Fall werden PI-Regler, Zweimassensystem und Filter zu einem Gesamtsystem fünfter Ordnung zusammengefasst und anschließend einer Transformation auf Byrnes-Isidori-Normalform unterzogen. Das Gesamtsystem fünfter Ordnung wird durch folgendes Modell beschrieben:

$$\begin{aligned} \begin{pmatrix} \dot{\omega}_2(t) \\ \Delta\dot{\varphi}(t) \\ \dot{\omega}_1(t) \\ \dot{\eta}(t) \\ \dot{\xi}(t) \end{pmatrix} &= \begin{bmatrix} -d/J_2 - \nu_V/J_2 & c/J_2 & d/J_2 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ d/J_1 & -c/J_1 & -d/J_1 & 0 & c_I/J_1 \\ 0 & 1 & 0 & -1/T & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \omega_2(t) \\ \Delta\varphi(t) \\ \omega_1(t) \\ \eta(t) \\ \xi(t) \end{pmatrix} \\ &+ \begin{pmatrix} -1/J_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} (\mathbf{N}\omega_2)(t) + \begin{pmatrix} 0 \\ 0 \\ k_P/J_1 \\ 0 \\ 1 \end{pmatrix} v(t) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1/J_1 \\ 0 \end{pmatrix} z(t) \\ y_r(t) &= \begin{pmatrix} k_{\omega_2} & k_{\Delta\varphi} & k_{\omega_1} & -\frac{k_{\Delta\varphi}}{T} & 0 \end{pmatrix} \\ &\cdot (\omega_2(t) \quad \Delta\varphi(t) \quad \omega_1(t) \quad \eta(t) \quad \xi(t))^T \end{aligned} \quad (16.144)$$

Es muss gezeigt werden, dass das Gesamtsystem, bestehend aus Zweimassen- system, DT<sub>1</sub>-Filter und PI-Regler eine stabile Nulldynamik und einen Relativgrad  $\delta$  von Eins besitzt und daher zur Systemklasse P gehört.

Für ein lineares Zweimassensystem mit DT<sub>1</sub>-Filter ist die Minimalphasigkeit des Gesamtsystems offensichtlich: wenn anstelle einer nichtlinearen Reibkennlinie  $\mathcal{NL}(\cdot)$  eine lineare Funktion eingesetzt wird, folgt aus den in Abschnitt 16.9.3 durchgeföhrten Überlegungen die Minimalphasigkeit der Kombination Zweimassensystem mit Filter. Der PI-Regler seinerseits besitzt eine reelle Nullstelle bei

$$s = -k_i/k_p = -v < 0 \quad (16.145)$$

und ist daher minimalphasig. Eine Reihenschaltung aus zwei minimalphasigen linearen Übertragungsgliedern behält die Minimalphasigkeit im Gesamtsystem bei, da lediglich die Zählerpolynome multipliziert werden. Dies entspricht einer Vereinigung der Nullstellen von beiden Einzelblöcken. Aus diesem Grund zeigt sich die Minimalphasigkeit im linearen Fall auf trivialem Wege.

Anders gestaltet sich dagegen die Situation, wenn ein nichtlineares Zweimassensystem betrachtet wird. In diesem Fall muss zunächst aus der BINF die Nulldynamik ermittelt und deren Stabilität untersucht werden.

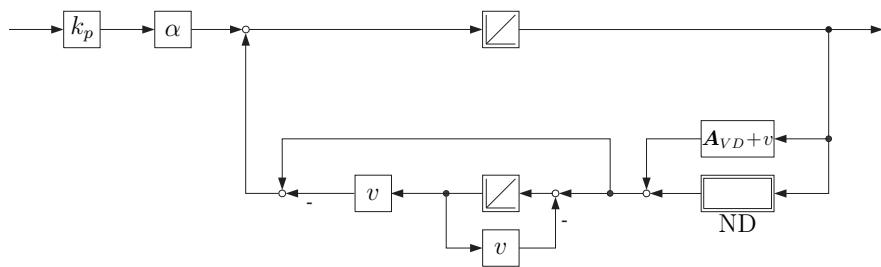
Die Verwendung der Transformation

$$\underline{z} = \Phi \cdot \underline{x} \quad (16.146)$$

mit der Transformationsmatrix

$$\Phi = \begin{bmatrix} k_{\omega_2} & k_{\Delta\varphi} & k_{\omega_1} & -\frac{k_{\Delta\varphi}}{T} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ k_{\omega_2} & k_{\Delta\varphi} & k_{\omega_1} & -\frac{k_{\Delta\varphi}}{T} & -\frac{k_{\omega_1} k_p}{J_2} \end{bmatrix} \quad (16.147)$$

führt auf eine geeignete Darstellung des Gesamtsystems in Byrnes-Isidori-Normalform.



**Abb. 16.28:** Signalflussplan des Gesamtsystems in Byrnes-Isidori-Normalform.

Dieses ist gegeben durch:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{VD} + v & \underline{b}_{NV}^T & -v \\ & 0 & \\ \underline{b}_{NV} & \mathbf{A}_{ND} & 0 \\ & 0 & \\ \mathbf{A}_{VD} + v & \underline{b}_{NV}^T & -v \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} + \begin{bmatrix} b_{VD} \\ b_{ND} \\ b_{VD} \end{bmatrix} \mathcal{N}(z_2) + \begin{bmatrix} \alpha k_p \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad (16.148)$$

Der Ausgang ist:

$$y = z_1 \quad (16.149)$$

Da sich durch die Zusammenfassung ein Gesamtsystem fünfter Ordnung ergibt, das einen Relativgrad  $\delta$  von Eins besitzt, folgt eine Nulldynamik vierter Ordnung. Die Stabilitätsuntersuchung für ein nichtlineares System vierter Ordnung gestaltet sich erfahrungsgemäß recht aufwändig. Eine geeignete Transformationsvorschrift führt jedoch auf eine derartige Darstellung der Nulldynamik, so dass eine Ausnutzung der bereits erhaltenen Ergebnisse ermöglicht wird.

Die Nulldynamik wird nun durch nachfolgende Differentialgleichung vierter Ordnung beschrieben:

$$\begin{bmatrix} \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \end{bmatrix} = \begin{bmatrix} & & 0 & \\ \mathbf{A}_{ND} & & 0 & \\ & & 0 & \\ & b_{NV}^T & -v & \end{bmatrix} \cdot \begin{bmatrix} z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} + \begin{bmatrix} b_{ND} \\ b_{VD} \end{bmatrix} \mathcal{NL}(z_2) + \begin{bmatrix} b_{VN} \\ \mathbf{A}_{VD} + v \end{bmatrix} z_1 \quad (16.150)$$

Dabei sind aufgrund der Wahl der Transformationsmatrix  $\Phi$  die Einträge  $\mathbf{A}_{ND}$ ,  $b_{NV}$ , usw. unverändert geblieben, die Werte beibehalten werden.

Eine Untersuchung der Stabilitätseigenschaften des linearen Subsystems ist ausreichend. Das führt auf die Zustandsmatrix

$$\mathbf{A}_{Lin,PI} = \begin{bmatrix} & & 0 & \\ & \mathbf{A}_{Lin} & 0 & \\ & & 0 & \\ \star & \star & \star & -v \end{bmatrix} \quad (16.151)$$

die auf Stabilität hin untersucht werden muss. Dabei bedeuten die Sterneinträge beliebige reelle Werte, welche hier keine Relevanz besitzen. Weil in der vierten Spalte neben dem Block  $\mathbf{A}_{Lin}$  nur Nulleinträge stehen, behält die Matrix  $\mathbf{A}_{Lin,PI}$  genau die selben Eigenwerte wie die Matrix  $\mathbf{A}_{Lin}$ . Es kommt lediglich ein zusätzlicher Eigenwert  $-v$  hinzu.

Wenn ein Eigenwert  $\lambda_i$  von  $\mathbf{A}_{Lin}$  in der Diagonalen der Matrix  $\mathbf{A}_{Lin,PI}$  abgezogen wird, so entsteht Rangabfall in  $\mathbf{A}_{Lin}$  und damit auch in der Gesamtmatrix, weil die Determinante nach der 4ten Spalte entwickelt werden kann. Dann gilt:

$$\det(\mathbf{A}_{Lin,PI} - \lambda_i \cdot \mathbf{I}) = (-v - \lambda_i) \cdot \underbrace{\det(\mathbf{A}_{Lin} - \lambda_i \cdot \mathbf{I})}_{=0} = 0 \quad (16.152)$$

Daraus folgt, dass alle Eigenwerte  $\lambda_i$  der Matrix  $\mathbf{A}_{Lin}$  auch gleichzeitig Eigenwerte der Matrix  $\mathbf{A}_{Lin,PI}$  sind. Wenn der Wert  $-v$  in der Diagonalen subtrahiert wird, besitzt  $\mathbf{A}_{Lin}$  i.A. zwar vollen Rang, die letzte Spalte von  $\mathbf{A}_{Lin,PI}$  enthält jedoch nur Nulleinträge. Dies führt zum Rangabfall der Gesamtmatrix  $\mathbf{A}_{Lin,PI}$ , weshalb  $-v$  als weiterer Eigenwert hinzukommt. Weil die Matrix  $\mathbf{A}_{Lin}$  auf geeignete Weise in der Matrix  $\mathbf{A}_{Lin,PI}$  enthalten ist, können die erzielten Ergebnisse hier direkt übernommen werden und machen eine Stabilitätsuntersuchung für ein System vierter Ordnung überflüssig.

Die Nulldynamik ist also auch bei Einsatz des PI-Bausteins stabil, wenn  $v > 0$  gewählt wird. Wegen der Festlegung

$$v = \frac{k_i}{k_p} \quad (16.153)$$

ist dies erfüllt, wenn  $k_i$  und  $k_p$  gleiches Vorzeichen besitzen. Das bedeutet allerdings keine Einschränkung, sondern ist bereits dadurch erfüllt, weil der PI-Block selbst als minimalphasig angesetzt worden ist.

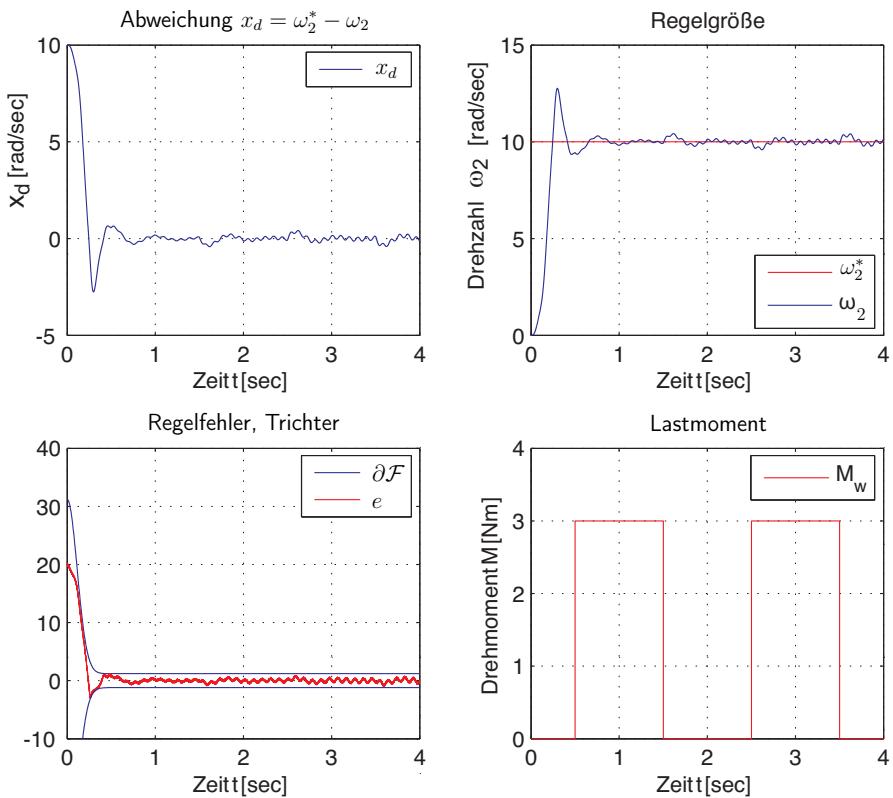
## 16.11 Ergebnisse mit Filter und Integralanteil

Durch Verwendung des DT<sub>1</sub>-Filters zusammen mit dem Integralen Anteil, kann nun ein breiter Trichter eingesetzt werden, und dennoch eine stationär genaue Regelung realisiert werden. In Abbildung 16.29 ist ein Simulationsergebnis für das nichtlineare Zweimassensystem dargestellt. Die Zustände des Zweimassensystems werden mit einem Rauschsignal beaufschlagt, bevor diese der Regelung zur Verfügung gestellt werden. Dadurch wird der Einfluss von Messrauschen nachgebildet, dem die reale Anlage unterliegt. Trotz Einfluss eines Lastmomentes verschwindet der Regelfehler, angeregte Schwingungen im System klingen rasch ab. Die Lastsprünge führen dabei nur zu geringen Abweichungen in der Drehzahl der Arbeitsmaschine. Damit ist dieses Konzept vielversprechend hinsichtlich einer Implementierung an der realen Anlage.

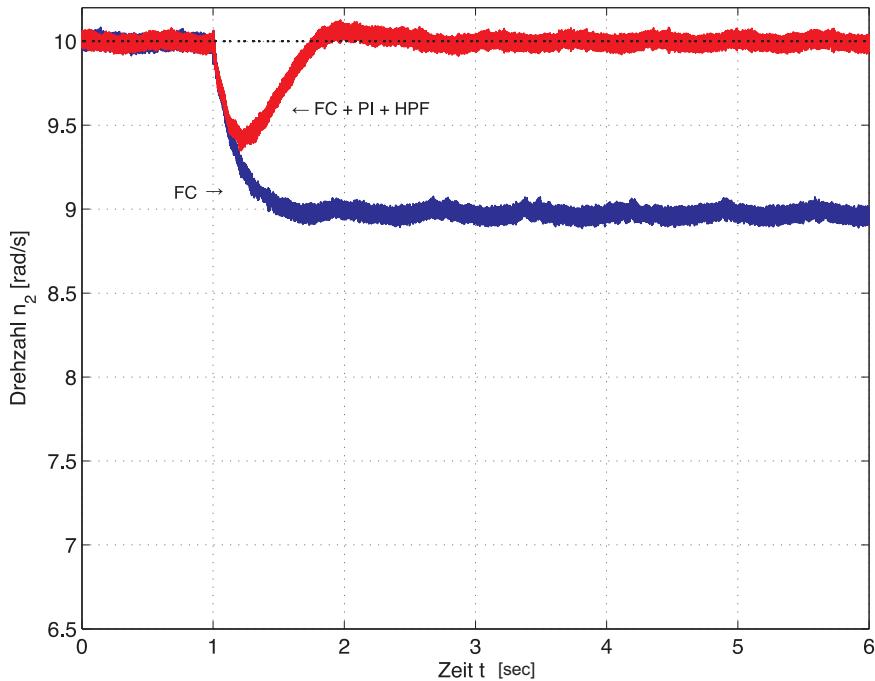
Die beschriebenen Regler wurden am Laborprüfstand evaluiert, der einer nichtlinearen Reibkennlinie unterworfen ist. Um zunächst den linearen Fall zu simulieren wurde für die Messung in Abbildung 16.30 durch ein zusätzliches Drehmoment auf der Lastseite der Einfluss der Reibung eliminiert. Aus diesem Grund erzielt der Funnel-Regler (ohne Integralanteil) stationär genaues Verhalten. Wird zusätzlich im stationären Arbeitspunkt (hier bei 10 rad/s) ein weiteres Lastmoment aufgebracht, ist ein deutlicher Drehzahleinbruch zu verzeichnen. Der proportionale Funnel-Regler ist nicht in der Lage, darauf adäquat zu reagieren.

Werden dagegen die beschriebenen Erweiterungsmaßnahmen (Hochpassfilter und internes Modell) eingesetzt, kann der Drehzahleinbruch ausgeregelt werden und die ursprüngliche Drehzahl von 10 rad/s wird wieder erreicht.

Ein ähnliches Bild zeigt sich auch am nichtlinearen Zweimassensystem. In Abbildung 16.31 offenbart der Prüfstand seine nichtlineare Charakteristik, weil keine Kompensation der Reibung erfolgt. Dadurch kann der Funnel-Regler selbst ohne zusätzlichem Lastmoment keine stationäre Genauigkeit erreichen. Allein die Reibung ist ausreichend, um eine deutliche Abweichung zu verursachen. Durch eine Zusatzlast wird die Drehzahlabweichung nochmals erhöht und beträgt im Falle der Messung ca. 15%. Auch am reibungsbehafteten System zeigen die vorgeschlagenen Erweiterungsmaßnahmen ihre Wirkung und regeln den Lastschlag aus. Dadurch wird trotz permanent einwirkender Last die Solldrehzahl beibehalten. Weitere Messergebnisse finden sich in [208] und [108].



**Abb. 16.29:** Simulationsergebnis für das nichtlineare Zweimassensystem unter Einfluss von Rauschen und Lastsprüngen. Dabei wurde sowohl die dynamische Zustandsrückführung als auch der Integrale Anteil verwendet.



**Abb. 16.30:** Messergebnis an der realen Anlage im Idealfall (Reibung ist durch Zusatzmoment kompensiert).

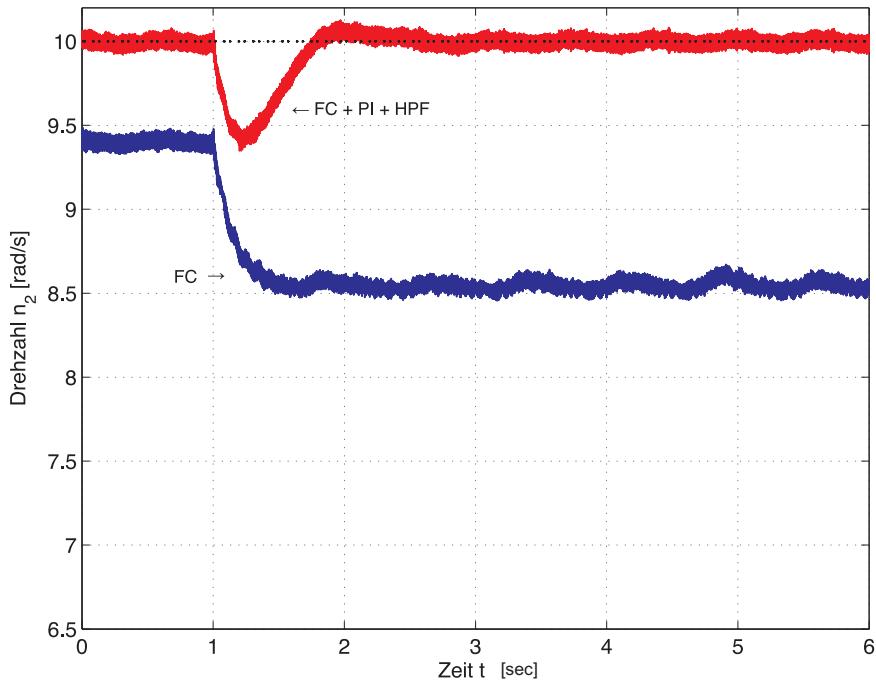


Abb. 16.31: Messergebnis an der Anlage im realen Falle mit Reibung.

# 17 Funnel-Control: Implementierung, Erweiterung und Anwendung

Christoph M. Hackl

In Kap. 16 wurden ausführlich Motivation und theoretische Grundlagen der hochverstärkungsbasierten nicht-identifizierenden adaptiven Regelung besprochen, unter anderem wurde einführend Funnel-Control behandelt und an einem schwingungsfähigen nichtlinearen Zwei-Massen-System implementiert.

Dieses Kapitel baut auf diesen Grundlagen auf und beschreibt Funnel-Control (FC) in größerem Detail. Es werden zusätzlich folgende Aspekte behandelt:

- Trichterentwurf mit Hilfe einer sogenannten ‘beschreibenden Trichterfunktion’ und anschauliche Erläuterung anhand von fünf Trichterbeispielen
- Vorstellung der zulässigen Klasse der Referenz- bzw. Sollwertsignale und eines realitätsnahen beispielhaften Referenzverlaufs (Überlagerung von konstanten, rampen- und sinusförmigen Sollwerten)
- Vorstellung der allgemeinen Systemklasse<sup>1)</sup>  $\mathcal{S}$  und Veranschaulichung anhand von drei linearen Beispielsystemen mit Simulationsergebnissen
- Berücksichtigung von Kundenanforderungen im Regler- bzw. Trichterdesign
- Vorstellung zweier Möglichkeiten der Skalierung der Reglerverstärkung zur Beschleunigung und/oder Bedämpfung des transienten Verhaltens
- Anpassung der Reglerverstärkung abhängig von der vertikalen *als auch der zukünftigen (minimalen) Distanz* (hierdurch wird ein ‘effektiverer’ Stellgrößenverlauf erzielt, der zu einer beschleunigten Systemantwort führt)
- Vorstellung dreier Ansätze zur Auswertung der minimalen zukünftigen Distanz

Es soll eine breite Grundlage geschaffen werden, die es dem Leser ermöglicht, Funnel-Control eigenständig zu implementieren, — sofern nötig — zu erweitern und anzuwenden.

---

<sup>1)</sup> Es wird auf die Darstellung des multiple-input multiple-output (MIMO) Falls verzichtet und nur der single-input single-output (SISO) Fall behandelt.

Der Funnel-Regler ist ein einfacher proportionaler aber nichtlinearer Regler mit *nicht notwendigerweise monoton steigender* Verstärkung (im Gegensatz zu klassischen ‘high-gain’ Ansätzen). Die Verstärkung wird nicht dynamisch sondern instantan an die aktuelle Regelungssituation angepasst bzw. adaptiert<sup>2)</sup>. Sie ist somit zeitvariant und speicherlos. Messrauschen ist zulässig. Die Genauigkeit der Folgewertregelung lässt sich *a priori* über z.B. eine fallende Funktion der Zeit — dem sogenannten Trichterrand (engl. funnel boundary) — vorgeben. Mithilfe des Trichterverlaufs können Kundenanforderungen direkt beim Reglerentwurf berücksichtigt werden.

Die Reglerverstärkung ist indirekt proportional zur gewählten Distanz zwischen Fehler und Trichterrand. Eine Skalierung der Distanz und somit der Verstärkung ist möglich und bietet zusätzliche Freiheitsgrade. Funnel-Control benötigt lediglich eine (sehr) grobe Streckenkenntnis, nur die Strecken- bzw. Modellstruktur müssen bekannt sein. Alle Prozesse oder Strecken einer Klasse  $\mathcal{S}$  — der Systeme mit Relativgrad  $\delta = 1$ , stabiler Nulldynamik und bekanntem Vorzeichen der instantanen Systemverstärkung — können mit Funnel-Control geregelt werden [103, 104, 105, 102]. Die Reglerauslegung bzw. die Trichterwahl sind unabhängig von den Systemparametern und somit inhärent robust. Langsame Parameterschwankungen oder -unsicherheiten, die nicht die Systemstruktur ändern, beeinflussen das Regelergebnis kaum. Der Regelfehler wird weiterhin innerhalb der Trichtergrenzen verlaufen.

Innerhalb des Trichters kann jedoch ein beliebiger Fehlerverlauf mit z.B. Überschwingen und/oder Oszillationen auftreten — insbesondere bei ‘weiten’ Trichtern. Um dieses nachteilige und oft unerwünschte Verhalten zu verbessern, wird eine ‘Dämpfungsskalierung’ der Reglerverstärkung und *Error Reference Control (ERC)* vorgestellt. Durch die zusätzliche Skalierung können ruhigere (bedämpfte) Fehlerverläufe erzielt werden. Dagegen erlaubt Error Reference Control die Führung des Regelfehlers entlang eines vorgegebenen ‘Wunschfehlerverlaufs’ (z.B. ohne Überschwingen). Error Reference Control ist direkt von Funnel-Control abgeleitet und somit für die gleiche Systemklasse  $\mathcal{S}$  anwendbar.

Wie bereits im Kapitel 16 dargestellt, ist die Einschränkung auf Relativgrad eins restriktiv: z.B. besitzt das schwingungsfähige Zwei-Massen-System (2MS) bei Regelung der Lastdrehzahl bereits einen Relativgrad von zwei. Daher wird der Relativgrad durch eine Zustandsrückführung reduziert [75, 209]. Das entstandene (erweiterte) Hilfssystem ist wieder Element der Klasse  $\mathcal{S}$  und erlaubt die Anwendung der zeitvarianten nicht-identifizierenden Regelungsstrategien FC und ERC.

Zusätzlich werden in diesem Kapitel Getriebe mit Übersetzung  $g_r \in \mathbb{R} \setminus \{0\}$  zugelassen. Da das Übersetzungsverhältnis  $g_r$  das Vorzeichen der instantanen Verstärkung und die interne Dynamik des 2MS direkt beeinflusst, muss  $g_r$  bei

---

<sup>2)</sup> Bei klassischen hochverstärkungsbasierten Regelungsverfahren wird die Reglerverstärkung  $k$  *dynamisch* z.B. durch  $\dot{k}(t) = |e(t)|^2$  adaptiert. In diesem Beitrag wird die Anpassung bei Funnel-Control als *nicht-dynamische* sondern zeitvariante Adaption angesehen.

der Relativgradreduktion und bei der Wahrung einer stabilen Nulldynamik berücksichtigt werden.

Des Weiteren wird eine statische Zustandsrückführung vorgeschlagen, die eine aktive Dämpfung des Zwei-Massen-Systems zulässt, *ohne* des in Kap. 16 vorgeschlagenen Hochpassfilter implementieren zu müssen.

Die inhärente proportionale Eigenschaft des zeitvarianten Reglers führt zu einer stationären Reglerabweichung — sofern z.B. Störungen (hier durch Lastmomente und/oder Reibung) auf die Regelstrecke wirken. Um nun eine asymptotisch genaue Festwertregelung und gutes Störverhalten zu erzielen, kann das Regelgesetz durch einen *PI*-ähnlichen Anteil erweitert werden ohne dabei Relativgrad, Stabilität der Nulldynamik oder Vorzeichen der instantanen Verstärkung zu ändern (siehe Kap. 16 oder auch [74]).

Abschließend werden Messergebnisse präsentiert, die einen direkten Vergleich der Reglerperformanz von Funnel-Control (FC), Error Reference Control (ERC) und einem *LQR*-Zustandsregler (SF) am Zwei-Massen-System ermöglichen. Beide adaptive (zeitvariante) Regelverfahren zeigen ein besseres Folgeverhalten als der Zustandsregler. Zusätzlich ist mithilfe der Dämpfungsskalierung bei ERC ein ruhigerer Verlauf des Regelfehlers als bei FC zu erreichen.

## 17.1 Funnel-Control (FC)

Funnel-Control entwickelt und erstmals 2002 vorgestellt von Ilchmann et al. [103], ist eine nicht-identifizierende adaptive (zeitvariante) Regelungsstrategie. Sie gilt als Weiterentwicklung der klassischen hochverstärkungsbasierten Regelungskonzepte (wie z.B. high-gain feedback,  $\lambda$ -tracking mit oder ohne  $\sigma$ -Modifikation [29, 100]). Das Regelgesetz ist einfach. Der gemessene Regelfehler wird mit einer zeitvarianten Reglerverstärkung  $k(t, e(t))$  gewichtet. Funnel-Control erlaubt die robuste Regelung von nichtlinearen Strecken der Klasse  $\mathcal{S}^3)$  mit Relativgrad  $\delta = 1$ , stabiler Nulldynamik [114] (minimalphasig im linearen Fall) und bekanntem (z.B. positiven) Vorzeichen der instantanen Verstärkung (high-frequency gain). Die instantane Verstärkung einer Regelstrecke mit Relativgrad  $\delta = 1$  beschreibt den Einfluss ('Wirkrichtung') der Stellgröße  $u$  auf die (erste) zeitliche Ableitung  $\dot{y}$  des Systemausgangs  $y$ . In Abb. 17.1 ist der geschlossene Regelkreis — bestehend aus System der Klasse  $\mathcal{S}$  und Funnel-Regler — dargestellt. Der zeitvariante (adaptive) proportionale Regler generiert die Stellgröße

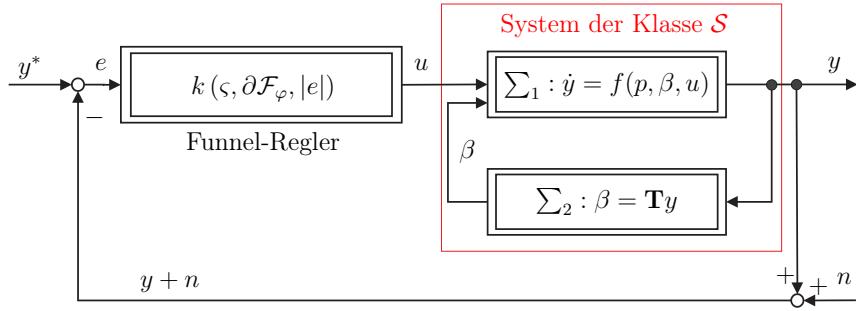
$$u(t) = k(t, e(t)) \cdot e(t) \quad (17.1)$$

aus zeitvarianter Verstärkung<sup>4)</sup>  $k(t, e(t)) := k(\varsigma(t), \partial\mathcal{F}_\varphi(t), e(t))$  und gemessenem Regelfehler  $e(t)$ .

---

<sup>3)</sup> für detaillierte Informationen über die Systemklasse  $\mathcal{S}$ , siehe [103, 104, 105]. Hier werden die einzelnen Subsysteme  $\Sigma_1$  und  $\Sigma_2$  sowie der Operator  $\mathbf{T}$  definiert und anhand von Beispielsystemen ausführlich besprochen.

<sup>4)</sup> Skalierung  $\varsigma(\cdot)$  und Trichterrand  $\partial\mathcal{F}_\varphi(\cdot)$  sind beides lediglich Funktionen der Zeit  $t \geq 0$ , daher ist die verkürzte Schreibweise  $k(t, e(t))$  zulässig und sinnvoll.



**Abb. 17.1:** Block-Diagramm eines Regelkreises mit Funnel-Control und einem System der Klasse  $\mathcal{S}$

Der Regelfehler

$$e(t) = y^*(t) - y(t) - n(t) \quad (17.2)$$

ist wie gewöhnlich definiert als Differenz zwischen Sollwert  $y^*(t)$  (Referenz) und Systemausgang  $y(t)$ . Hochfrequentes Messrauschen  $n(t)$  ist zulässig.

Die Verstärkung  $k(t, e(t))$  ist indirekt proportional zur gewählten Distanz zwischen Trichterrand  $\partial\mathcal{F}_\varphi(t) = 1/\varphi(t)$  (beschrieben durch die Funktion  $\varphi(\cdot)$ , siehe auch Abschnitt 17.1.1) und Fehler  $e(t)$ . Die vertikale Distanz

$$d_V(t, e(t)) = \partial\mathcal{F}_\varphi(t) - |e(t)| \quad (17.3)$$

wertet beispielsweise zum aktuellen Zeitpunkt  $t \geq 0$  (siehe Abb. 17.2) die Differenz zwischen Trichterrand  $\partial\mathcal{F}_\varphi(t)$  und Absolutwert<sup>5)</sup>  $|e(t)|$  des Fehlers aus. Neben der vertikalen Distanz  $d_V(t, e(t))$  ist auch die zukünftige (minimale) Distanz

$$d_F(t, e(t)) = \min_{t_F \geq t} \sqrt{(t_F - t)^2 + (\partial\mathcal{F}_\varphi(t_F) - |e(t)|)^2} \quad (17.4)$$

zu einem zukünftigen Zeitpunkt  $t_F \geq t$  zulässig (siehe [71, 76, 104] bzw. Abschnitt 17.4). Die Reglerverstärkung (17.1) kann mit (17.3) oder (17.4) explizit entweder zu

$$k(t, e(t)) = k_V(t, e(t)) = \frac{\varsigma(t)}{d_V(t, e(t))} = \frac{\varsigma(t)}{\partial\mathcal{F}_\varphi(t) - |e(t)|} = \frac{\varsigma(t) \varphi(t)}{1 - \varphi(t) |e(t)|} \quad (17.5)$$

oder

$$k(t, e(t)) = k_F(t, e(t)) = \frac{\varsigma(t)}{d_F(t, e(t))} \quad (17.6)$$

<sup>5)</sup> Im Falle eines Mehrgrößensystems (multiple-input, multiple-output: *MIMO*) muss der Absolutbetrag für jedes  $1 < m \in \mathbb{N}$  durch z.B. die euklidische Norm  $\|\underline{e}(t)\| = \sqrt{\langle \underline{e}(t), \underline{e}(t) \rangle}$  des vektoriellen Fehlers  $\underline{e}(t) \in \mathbb{R}^M$  ersetzt werden.

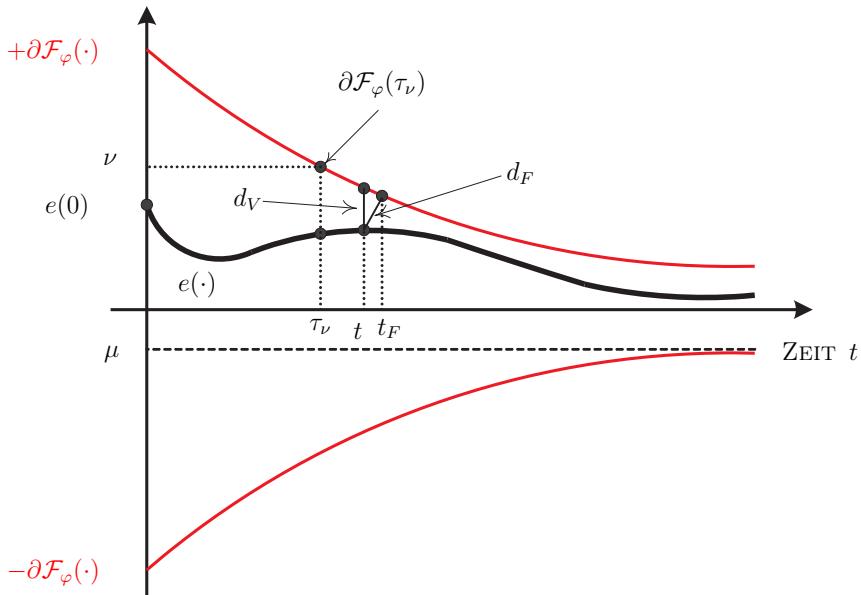


Abb. 17.2: Grundlegende Idee von Funnel-Control

angegeben werden. Als weiterer Freiheitsgrad kann die Reglerverstärkung mit einer beschränkten sogenannten ‘Skalierungsfunktion’  $\varsigma(\cdot) > 0$  multipliziert werden. Die Skalierung erlaubt z.B. die Festlegung einer minimalen Verstärkung (siehe [71, 103, 104] und Abschnitt 17.3). Die Adaption in (17.5) oder (17.6) führt dazu, dass sich die Verstärkung  $k_V(t)$  oder  $k_F(t)$  vergrößern wird, falls sich der Regelfehler  $e(t)$  dem Trichterrand  $\partial\mathcal{F}_\varphi(t)$  nähert (kritische Situation) und verringern ‘darf’, falls sich  $e(t)$  wieder vom Trichter weg bewegt (unkritische Situation). In [103, 104] wird gezeigt, dass sowohl Verstärkung  $k_V(\cdot)$  oder  $k_F(\cdot)$  als auch Fehler  $e(\cdot)$  beschränkt bleiben, sofern beliebig aber endliche Stellgrößen  $u(\cdot)$  zur Verfügung stehen.

Die internen Systemgrößen in Abb. 17.1 stellen eine beschränkte Störung  $p(\cdot)$  und mit  $\beta(\cdot) = (\mathbf{T}y)(\cdot)$  die interne Dynamik der Strecke dar. Die interne Dynamik wird hierbei durch einen kausalen und lokal Lipschitz-stetigen Operator  $\mathbf{T}$  beschrieben (für die exakten Eigenschaften siehe Definition 17.1 oder Kap. 16). Der Trichterrand

$$\partial\mathcal{F}_\varphi(t) = \frac{1}{\varphi(t)} \quad (17.7)$$

wird für alle  $t > 0$  durch die Inverse einer entsprechend gewählten beschränkten, stetigen und strikt positiven ‘beschreibenden Trichterfunktion’  $\varphi(\cdot)$  mit  $\sup_{t \geq 0} \varphi(t) < \infty$  festgelegt (siehe Abschnitt 17.1.1 oder [103, 104]). Der Wert  $\varphi(0) = 0$  ist erlaubt.

Durch die Wahl des Trichterrandes  $\partial\mathcal{F}_\varphi(t) = 1/\varphi(t) > 0$  für alle  $t \geq 0$  kann ein beliebig kleiner Regelfehler erzwungen werden, jedoch muss der Fall  $\limsup_{t \rightarrow \infty} |e(t)| \neq 0$  zugelassen werden. Dieser Nachteil ist inhärent für proportionale Regelkreise (z.B. ohne integralen Bestandteil) und somit kein spezifischer von Funnel-Control. Wie bereits im vorangegangenen Kapitel 16 (durch interne Modelle) oder in [74] gezeigt wurde, kann dieser Nachteil durch eine (auch nichtlineare [74]) PI-ähnliche Erweiterung des Funnel-Konzeptes umgegangen werden. Hiermit kann stationäre Genauigkeit (bei konstanter Sollwertvorgabe) und/oder gutes Störverhalten des Regelkreises erreicht werden. Die PI-ähnliche Erweiterung gefährdet die Zugehörigkeit der erweiterten Strecke zur relevanten Systemklasse  $\mathcal{S}$  nicht [74].

### 17.1.1 Trichterentwurf: Beschreibende Trichterfunktion $\varphi(\cdot)$ und Trichterrand $\partial\mathcal{F}_\varphi(\cdot)$

Um eine große Entwurfsfreiheit zu bieten, soll nun die zulässige Funktionenklasse  $\Phi$  vorgestellt werden. Die beschreibende Trichterfunktion  $\varphi(\cdot)$  muss

$$\Phi := \left\{ \varphi \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R}_{\geq 0}) \mid \begin{array}{l} \varphi(s) > 0 \quad \forall s > 0 \\ \liminf_{s \rightarrow \infty} \varphi(s) > 0 \end{array} \right\} \quad (17.8)$$

entstammen. Somit sind alle Funktionen  $\varphi(\cdot) \in \Phi$  positiv auf dem Intervall  $(0, \infty)$  und für  $t \rightarrow \infty$  von Null weg beschränkt. Aufgrund  $\varphi(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R}_{\geq 0})$  ist  $\varphi(\cdot)$  lokal absolut stetig auf  $\mathbb{R}_{\geq 0}$  und besitzt eine essentiell beschränkte erste Ableitung  $\dot{\varphi}(\cdot) \in \mathcal{L}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$  [103, 104]. Zu jedem  $\varphi(\cdot) \in \Phi$  definieren wir die Trichterpunktmenge

$$\mathcal{F}_\varphi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}, \quad t \mapsto \{e \in \mathbb{R} \mid \varphi(t)|e| < 1\} \quad (17.9)$$

in der für jeden Zeitpunkt  $t \geq 0$  der Fehler  $e(\cdot)$  ‘verlaufen’ darf. Schließlich kann der Trichter  $\mathbb{F}_\varphi$  durch den Graph von  $\mathcal{F}_\varphi$  angegeben werden mit

$$\mathbb{F}_\varphi := \text{graph}(\mathcal{F}_\varphi) := \{(t, e) \in (\mathbb{R}_{\geq 0} \times \mathbb{R}) \mid e \in \mathcal{F}_\varphi(t)\}. \quad (17.10)$$

Der Trichter umschließt also alle Punktpaare  $(t, e) \in \mathbb{R}_{\geq 0} \times \mathbb{R}$  bei denen der Regelfehler durch die beschreibende Funktion  $\varphi(\cdot)$  begrenzt wird und ist daher eine Untermenge (‘Teilebene’) des  $\mathbb{R}^2$ .

Allgemein kann der Trichterrand  $\partial\mathcal{F}_\varphi$  durch folgende Funktion

$$\partial\mathcal{F}_\varphi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}, \quad t \mapsto \{x \in \mathbb{R}_{>0} \mid \varphi(t)|x| = 1\} \quad (17.11)$$

beschrieben werden. Für alle  $t > 0$  ist der Trichterrand durch

$$\partial\mathcal{F}_\varphi(t) = \frac{1}{\varphi(t)} \quad (17.12)$$

festgelegt (für alle  $t > 0$  ist  $\varphi(t) > 0$ ).

Theoretisch kann durch  $\limsup_{t \rightarrow \infty} \partial\mathcal{F}_\varphi(t) = (\liminf_{t \rightarrow \infty} \varphi(t))^{-1} > 0$  die Endgenauigkeit beliebig klein gewählt werden. Bei verrauschten Messsignalen muss jedoch die maximale Rauschamplitude  $\|n\|_\infty$  immer vom Trichterrand umschlossen werden, d.h.  $\partial\mathcal{F}_\varphi(t) \geq \liminf_{t \rightarrow \infty} \partial\mathcal{F}_\varphi(t) > \|n\|_\infty$  muss für alle  $t \geq 0$  gelten. Zusätzlich hängt die erreichbare Endgenauigkeit von der zur Verfügung stehenden Stellgröße ab.

### Trichterbeispiele

Im folgenden Abschnitt werden unterschiedliche beschreibende Trichterfunktionen und die zugehörigen Trichterränder vorgestellt. Die dargestellten Beispiele sollen ein Gefühl für die Möglichkeiten bei der Wahl der beschreibenden Trichterfunktion vermitteln und insbesondere nachweisen, dass sie der Funktionenmenge  $\Phi$  entstammen und somit überhaupt zulässige Trichterränder beschreiben. Auch wenn die Funktionenklasse  $\Phi$  nicht nur monoton wachsende Funktionen  $\varphi(\cdot)$  enthält, und somit auch wieder steigende Trichterränder  $\partial\mathcal{F}_\varphi(\cdot)$  erlaubt wären, ist oft in der Anwendung das Wiederanwachsen des Fehlerbetrages unerwünscht. Daher werden im Folgenden ausschließlich monoton wachsende beschreibende Trichterfunktionen und daher monoton fallende Trichterränder eingeführt. Für alle folgenden Beispiele sind in Abb. 17.3 die Verläufe der beschreibenden Trichterfunktionen und der zugehörigen Trichterränder dargestellt.

**Unendlicher Standardtrichter  $\partial\mathcal{F}_\infty$ :** Das folgende Beispiel wurde bereits in [103, 104] eingeführt. Der ‘Unendlicher Standardtrichter’ ist aus theoretischer Sicht besonders sinnvoll, da er von ‘unendlich’ kommend jeden endlichen Anfangsfehler  $e(0)$  umschließen kann. Man wählt die beschreibende Trichterfunktion

$$\begin{aligned}\varphi_\infty : \quad \mathbb{R}_{\geq 0} &\rightarrow \mathbb{R}_{\geq 0}, \\ t &\mapsto \varphi_\infty(t) := \frac{t}{[(1-\epsilon) \cdot t + \epsilon\tau]\lambda} = \frac{t}{at + b}\end{aligned}\tag{17.13}$$

mit den reellen und positiven Konstanten  $\lambda, \tau > 0$ ,  $\epsilon \in (0, 1)$ ,  $a = (1 - \epsilon)\lambda > 0$  und  $b = \epsilon\lambda\tau > 0$ . Diese beschreibende Trichterfunktion startet bei  $\varphi_\infty(0) = 0$ , ist monoton steigend und stetig. Mit

$$\forall t \geq 0 : \quad \dot{\varphi}_\infty(t) = \frac{\epsilon\tau\lambda}{([(1-\epsilon) \cdot t + \epsilon\cdot\tau]\lambda)^2} = \frac{b}{(a \cdot t + b)^2} > 0\tag{17.14}$$

ist ihre zeitliche Ableitung  $\dot{\varphi}_\infty(\cdot)$  strikt positiv und beschränkt. Das Maximum der Ableitung existiert für  $t = 0$  und entspricht

$$\dot{\varphi}_\infty(0) = \frac{1}{\epsilon\lambda\tau} = \frac{1}{b}.\tag{17.15}$$

Dagegen ergibt sich für  $t \rightarrow \infty$

$$\lim_{t \rightarrow \infty} \dot{\varphi}_\infty(t) = 0. \quad (17.16)$$

Die beschreibende Trichterfunktion  $\varphi_\infty(\cdot)$  ist von Null weg beschränkt für  $t \rightarrow \infty$ , da

$$\liminf_{t \rightarrow \infty} \varphi_\infty(t) = \lim_{t \rightarrow \infty} \varphi_\infty(t) = \frac{1}{(1-\varepsilon)\lambda} = \frac{1}{a} > 0. \quad (17.17)$$

Somit ist  $\varphi_\infty(\cdot)$  Element der Funktionenmenge  $\Phi$  und zulässig als beschreibende Trichterfunktion des ‘Unendlichen Standardtrichterrandes’

$$\forall t > 0 : \quad \partial\mathcal{F}_\infty(t) := \frac{1}{\varphi_\infty(t)} = \frac{[(1-\epsilon) \cdot t + \epsilon \cdot \tau]\lambda}{t} = \frac{at + b}{t} \quad (17.18)$$

Der Fall  $t = 0$  ist in der inversen Darstellung nicht eingeschlossen, hier strebt der Trichteranfangswert

$$\lim_{t \rightarrow 0+} \partial\mathcal{F}_\infty(t) = \infty. \quad (17.19)$$

Jeder endliche Anfangsfehler  $|e(0)| < \infty$  wird also umschlossen. Für die Ableitung des unendlichen Standardtrichters erhält man

$$\forall t > 0 : \quad \partial\dot{\mathcal{F}}_\infty(t) = -\frac{\dot{\varphi}_\infty(t)}{\varphi_\infty(t)^2} = -\frac{\epsilon\lambda\tau}{t^2} = -\frac{b}{t^2} \quad (17.20)$$

und für  $t \rightarrow 0+$  strebt auch die Ableitung gegen  $-\infty$ .

Die asymptotische Endgenauigkeit des Trichterrandes kann abgelesen werden zu

$$\lim_{t \rightarrow \infty} \partial\mathcal{F}_\infty(t) = (1-\varepsilon)\lambda = a \quad (17.21)$$

Für jeden Zeitpunkt  $\tau > 0$  nimmt der unendliche Standardtrichterrand folgenden Wert an

$$\partial\mathcal{F}_\infty(\tau) = \frac{1}{\varphi_\infty(\tau)} = \frac{[\tau - \epsilon \cdot \tau + \epsilon \cdot \tau]\lambda}{\tau} = \lambda. \quad (17.22)$$

Da  $\partial\mathcal{F}_\infty(\cdot)$  monoton fällt, kann direkt daraus geschlossen werden, dass für jeden Zeitpunkt  $t \geq \tau$  der Absolutwert des Fehlers kleiner als der Trichterrand zum Zeitpunkt  $\tau$  sein wird, d.h.

$$|e(t)| < \partial\mathcal{F}_\infty(\tau) = \lambda. \quad (17.23)$$

Des Weiteren gilt wegen (17.17)

$$\lim_{t \rightarrow \infty} \partial\mathcal{F}_\infty(t) = (1-\epsilon)\lambda = a > 0. \quad (17.24)$$

Die meisten realen Anwendungen erlauben die Bestimmung des Anfangsfehlers. Daher ist es meist ausreichend einen endlichen Anfangswert  $\partial\mathcal{F}_\varphi(0) < \infty$  für den Trichterrand zu wählen.

**Anangepasster Standardtrichter  $\partial\mathcal{F}_\infty^*$ :** Der angepasste Trichterrand  $\partial\mathcal{F}_\infty^*$  lässt sich direkt vom unendlichen Standardtrichter  $\partial\mathcal{F}_\infty$  ableiten. Es wird lediglich eine Konstante  $\varphi_0^* > 0$  addiert, die es ermöglicht einen endlichen Anfangswert zu fixieren.

Die beschreibende Trichterfunktion des angepassten Standardtrichters ist gegeben durch

$$\varphi_\infty^* : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0},$$

$$t \mapsto \varphi_\infty^*(t) := \frac{t}{[(1 - \varepsilon^*) \cdot t + \varepsilon^* \tau^*] \lambda^*} + \varphi_0^* = \frac{t}{a^* t + b^*} + \varphi_0^* \quad (17.25)$$

mit den positiven Konstanten  $\tau^*, \lambda^* > 0$ ,  $\varepsilon^* \in (0, 1)$ ,  $a^* := (1 - \varepsilon^*) \lambda^* > 0$  und  $b^* := \varepsilon^* \tau^* \lambda^*$ . Mit

$$\forall t \geq 0 : \dot{\varphi}_\infty^*(t) = \frac{\varepsilon^* \tau^* \lambda^*}{[((1 - \varepsilon^*) \cdot t + \varepsilon^* \tau^*) \lambda^*]^2} = \frac{b^*}{[a^* t + b^*]^2} > 0, \quad (17.26)$$

erhält man die positive und beschränkte zeitliche Ableitung  $\dot{\varphi}_\infty^*(\cdot)$ . D.h. die beschreibende Trichterfunktion  $\varphi_\infty^*(\cdot)$  ist streng monoton wachsend und es gilt  $\varphi_\infty^*(\cdot) \in \Phi$ . Der angepasste Standardtrichterrand ist nun für alle  $t \geq 0$  definiert durch

$$\partial\mathcal{F}_\infty^*(t) := \frac{1}{\varphi_\infty^*(t)} = \frac{(1 - \varepsilon^*) \lambda^* t + \varepsilon^* \tau^* \lambda^*}{t + \varphi_0^* \lambda^* ((1 - \varepsilon^*) t + \varepsilon^* \tau^*)} = \frac{a^* t + b^*}{t + \varphi_0^* (a^* t + b^*)} \quad (17.27)$$

Dieser Trichter strebt für große Zeiten gegen

$$\lim_{t \rightarrow \infty} \partial\mathcal{F}_\infty^*(t) = \frac{a^*}{1 + \varphi_0^* a^*} = \frac{(1 - \varepsilon^*) \lambda^*}{1 + \varphi_0^* (1 - \varepsilon^*) \lambda^*} \quad (17.28)$$

und beginnt beim endlichen Wert

$$\partial\mathcal{F}_\infty^*(0) = \frac{1}{\varphi_0^*}. \quad (17.29)$$

Der Anfangswert  $\frac{1}{\varphi_0^*}$  kann nun bei gegebenem Anfangsfehler  $e(0) \neq 0$ , so angepasst werden, dass  $\varphi_0^* = \frac{1}{|e(0)|} > 0$  und somit  $\partial\mathcal{F}_\infty^*(0) > |e(0)|$ .

**Exponentieller Trichter  $\partial\mathcal{F}_E$ :** Stabile lineare Systeme besitzen ein exponentiell abklingendes transientes Verhalten (siehe z.B. [56, 203]). Insbesondere für solche Systeme ist es daher sinnvoll einen ‘Exponentiellen Trichter’ einzuführen. Solch ein exponentieller Verlauf wird durch die beschreibende Trichterfunktion invers beschrieben

$$\varphi_E : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0},$$

$$\begin{aligned} t \mapsto \varphi_E(t) &:= \frac{\varphi_{E,0} \varphi_{E,\infty}}{(\varphi_{E,\infty} - \varphi_{E,0}) \exp\left(-\frac{t}{T_E}\right) + \varphi_{E,0}} \\ &= \frac{1}{\left(\frac{1}{\varphi_{E,0}} - \frac{1}{\varphi_{E,\infty}}\right) \cdot \exp\left(-\frac{t}{T_E}\right) + \frac{1}{\varphi_{E,\infty}}}. \quad (17.30) \end{aligned}$$

Um das transiente Verhalten vorgeben zu können, stehen als Parameter die positiven Konstanten  $\varphi_{E,\infty} > \varphi_{E,0} > 0$  und die Zeitkonstante  $T_E > 0$  zur Verfügung. Mit

$$\forall t \geq 0 : \quad \dot{\varphi}_E(t) = \frac{\varphi_{E,0}\varphi_{E,\infty}(\varphi_{E,\infty} - \varphi_{E,0}) \exp\left(-\frac{t}{T_E}\right)}{T_E \left[(\varphi_{E,\infty} - \varphi_{E,0}) \exp\left(-\frac{t}{T_E}\right) + \varphi_{E,0}\right]^2} > 0, \quad (17.31)$$

erhält man die strikt positive und beschränkte zeitliche Ableitung  $\dot{\varphi}_E(\cdot)$ . Somit ist  $\varphi_E(\cdot) \in \Phi$ . Der Exponentielle Trichterrand ist streng monoton fallend und kann explizit angegeben werden mit

$$\partial\mathcal{F}_E(t) := \frac{1}{\varphi_E(t)} = \left( \frac{1}{\varphi_{E,0}} - \frac{1}{\varphi_{E,\infty}} \right) \cdot \exp\left(-\frac{t}{T_E}\right) + \frac{1}{\varphi_{E,\infty}}. \quad (17.32)$$

Man erhält die asymptotische Endgenauigkeit

$$\lim_{t \rightarrow \infty} \partial\mathcal{F}_E(t) = \frac{1}{\varphi_{E,\infty}} > 0 \quad (17.33)$$

unabhängig vom frei wählbaren Anfangswert  $\partial\mathcal{F}_E(0)$ , so dass die Bedingung

$$\partial\mathcal{F}_E(0) = \frac{1}{\varphi_{E,0}} \stackrel{!}{>} |e(0)| \quad (17.34)$$

auf jeden Fall erfüllt werden kann.

**Gaußglocken-Trichter  $\partial\mathcal{F}_G$ :** Manchmal kann es abhängig von der vorliegenden Regelstrecke von Vorteil sein, wenn anfangs der Regelkreis nicht zu stark gefordert wird. Dies ist beispielsweise durch die Vorgabe eines Gaußglocken-Trichters möglich, der durch

$$\begin{aligned} \varphi_G : \quad \mathbb{R}_{\geq 0} &\rightarrow \mathbb{R}_{>0}, \\ t \mapsto \varphi_G(t) &:= \frac{\varphi_{G,0}\varphi_{G,\infty}}{(\varphi_{G,\infty} - \varphi_{G,0}) \exp\left(-\left(\frac{t}{T_G}\right)^2\right) + \varphi_{G,0}} \\ &= \frac{1}{\left(\frac{1}{\varphi_{G,0}} - \frac{1}{\varphi_{G,\infty}}\right) \cdot \exp\left(-\left(\frac{t}{T_G}\right)^2\right) + \frac{1}{\varphi_{G,\infty}}} \end{aligned} \quad (17.35)$$

invers beschrieben wird. Durch die Parameter  $\varphi_{G,\infty} > \varphi_{G,0} > 0$  und die Zeitkonstante  $T_G$  kann wieder das gewünschte dynamische Verhalten eingestellt werden. Mit

$$\forall t \geq 0 : \quad \dot{\varphi}_G(t) = \frac{2 \cdot t \cdot \varphi_{G,0}\varphi_{G,\infty}(\varphi_{G,\infty} - \varphi_{G,0}) \exp\left(-\left(\frac{t}{T_G}\right)^2\right)}{T_G^2 \left[(\varphi_{G,\infty} - \varphi_{G,0}) \exp\left(-\frac{t}{T_G}\right) + \varphi_{G,0}\right]^2} \geq 0 \quad (17.36)$$

ist die zeitliche Ableitung  $\dot{\varphi}_G(\cdot)$  stetig und wegen  $\lim_{t \rightarrow \infty} t \exp\left(-\left(\frac{t}{T_G}\right)^2\right) = 0$  beschränkt. Es gilt wieder  $\varphi_G(\cdot) \in \Phi$ . Insbesondere  $\dot{\varphi}_G(0) = 0$  lässt anfänglich dem Regelkreis mehr ‘Spielraum’, erst für spätere Zeitpunkte  $t > 0$  zeigt der Gaußglocken-Trichterrand

$$\partial\mathcal{F}_G(t) := \frac{1}{\varphi_G(t)} = \left( \frac{1}{\varphi_{G,0}} - \frac{1}{\varphi_{G,\infty}} \right) \cdot \exp\left(-\left(\frac{t}{T_G}\right)^2\right) + \frac{1}{\varphi_{G,\infty}} \quad (17.37)$$

einen steilen Abfall. Die Endgenauigkeit

$$\lim_{t \rightarrow \infty} \partial\mathcal{F}_G(t) = \frac{1}{\varphi_{G,\infty}} > 0 \quad (17.38)$$

kann über die Variable  $\varphi_{G,\infty} > 0$  entsprechend fixiert werden. Der Anfangswert ist variabel und muss lediglich die Anfangsbedingung

$$\partial\mathcal{F}_G(0) = \frac{1}{\varphi_{G,0}} > |e(0)| \quad (17.39)$$

erfüllen.

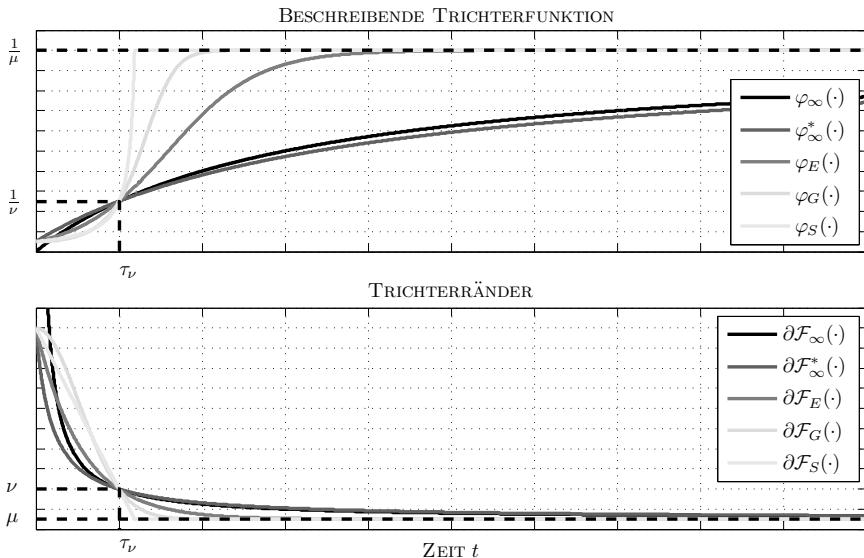
**Simpler Trichter  $\partial\mathcal{F}_S$ :** Später wird gezeigt werden, dass die Auswertung der zukünftigen Distanz (17.4) Vorteile haben kann. Insbesondere für ‘einfache’ Trichterränder ist die Bestimmung der zukünftigen Distanz sehr einfach möglich. Daher soll abschließend ein simpler Trichteraufbau vorgestellt werden. Beispielsweise erlaubt die beschreibende Trichterfunktion

$$\varphi_S : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}, \quad t \mapsto \varphi_S(t) := \begin{cases} \frac{\varphi_{S,0}\varphi_{S,\infty}\tau_\mu}{-(\varphi_{S,\infty} - \varphi_{S,0})t + \varphi_{S,\infty}\tau_\mu} & , t \leq \tau_\mu \\ \varphi_{S,\infty} & , t > \tau_\mu \end{cases} \quad (17.40)$$

mit ihren Designparametern  $\varphi_{S,\infty} > \varphi_{S,0} > 0$  und dem Knickzeitpunkt  $\tau_\mu > 0$  die Darstellung eines besonders einfachen Trichters, der lediglich aus Geraden zusammengesetzt ist. Die zeitliche Ableitung lässt sich wie folgt definieren (man beachte die nicht differenzierbare Stelle bei  $\tau_\mu$ )

$$\dot{\varphi}_S(t) = \begin{cases} \frac{\varphi_{S,0}\varphi_{S,\infty}\tau_\mu(\varphi_{S,\infty} - \varphi_{S,0})}{[-(\varphi_{S,\infty} - \varphi_{S,0}) \cdot t + \varphi_{S,\infty}\tau_\mu]^2} > 0 & , t \leq \tau_\mu \\ 0 & , t > \tau_\mu \end{cases} \quad (17.41)$$

Diese ist echt größer Null für alle  $0 \leq t \leq \tau_\mu$  und damit auf diesem Intervall streng monoton steigend. Die Ableitung selbst wächst monoton, da  $\dot{\varphi}_S(t_1) < \dot{\varphi}_S(t_2)$  für alle  $\tau_\mu \geq t_2 > t_1 > 0$ . Die Ableitung hat den Wertebereich  $\{0\} \cup [\dot{\varphi}_S(0), \dot{\varphi}_S(\tau_\mu)] = \{0\} \cup [\frac{\varphi_{S,0}(\varphi_{S,\infty} - \varphi_{S,0})}{\varphi_{S,\infty}\tau_\mu}, \frac{\varphi_{S,\infty}(\varphi_{S,\infty} - \varphi_{S,0})}{\varphi_{S,0}\tau_\mu}]$ . Somit ist  $\varphi_S(\cdot) \in \Phi$ .



**Abb. 17.3:** Beispiele für nicht-fallende beschreibende Trichterfunktionen und entsprechend nicht-wachsende Trichterränder

Die Darstellung des ‘Simplen Trichterrandes’ ergibt sich zu

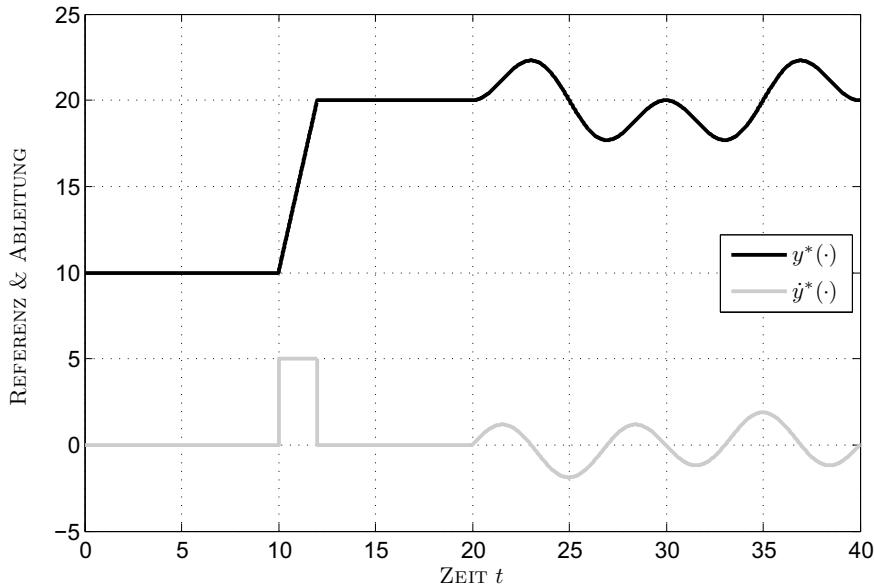
$$\partial\mathcal{F}_S(t) := \begin{cases} -\frac{1}{\varphi_{S,0}} - \frac{1}{\varphi_{S,\infty}} \\ \quad - \frac{\varphi_{S,\infty}}{\tau_\mu} \cdot t + \frac{1}{\varphi_{S,0}} & , t \leq \tau_\mu \\ \frac{1}{\varphi_{S,\infty}} & , t > \tau_\mu \end{cases} \quad (17.42)$$

mit dem Anfangswert  $\frac{1}{\varphi_{S,0}} > |e(0)| > 0$  und der asymptotischen Endgenauigkeit  $\frac{1}{\varphi_{S,\infty}} > 0$ . Ab dem Zeitpunkt  $\tau_\mu > 0$  knickt die fallende Gerade ab und der Trichterrand geht in eine horizontale Linie über (siehe Abb. 17.3).

### 17.1.2 Referenz- bzw. Sollwertsignale

Die zugelassene Funktionenklasse der Referenz- bzw. Sollwertsignale umfasst alle lokal absolut stetigen Funktionen mit essentiell beschränkter Ableitung, d.h.  $y^*(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  mit der zugehörigen Norm  $\|y^*\|_\infty := \|y^*\|_\infty + \|\dot{y}^*\|_\infty$  [103, 104]. Dieser Funktionenraum umfasst nahezu alle typischen Referenzverläufe in der Industrie. Mit einer wichtigen Ausnahme: dem (gewichteten) Einheitssprung

$$y^*(t) = \bar{y}^* \cdot \sigma(t - t_0) = \begin{cases} \bar{y}^* & , t \geq t_0 \\ 0 & , t < t_0 \end{cases} \quad (17.43)$$



**Abb. 17.4:** Beispielhafter Sollwertverlauf  $y^*(\cdot)$  (SISO): Zusammengesetzt aus konstanten, rampen- und sinusförmigen Referenzsignalen

mit konstanter Höhe (Wichtung)  $\bar{y}^* \in \mathbb{R}$ . Für alle  $t > 0$  führt eine plötzliche und sprunghafte Änderung im Referenzsignal zu einer sprunghaften (unstetigen) Änderung des Fehlers, was u.U. das Verlassen (Herausspringen) des Fehlers aus dem Trichter zur Folge haben kann. *Lediglich für  $t = 0$  ist ein Sollwertsprung zulässig*, da dieser in der Anfangsbedingung  $\varphi(0)|e(0)| < 1$  berücksichtigt werden kann. Somit muss für jedes  $t_0 > 0$  der Sollwertsprung  $y^*(t) = \bar{y}^* \sigma(t - t_0)$  durch eine (steile) und saturierte Rampe der Art

$$y^*(t) = \text{sat}[m(t - t_0)]_{0}^{\bar{y}^*} := \begin{cases} \bar{y}^* & , t > \frac{\bar{y}^*}{m} + t_0 \\ m(t - t_0) & , t_0 \leq t \leq \frac{\bar{y}^*}{m} + t_0 \\ 0 & , t < t_0 \end{cases} \quad (17.44)$$

mit hinreichend großer Steigung  $m \in \mathbb{R}$  approximiert werden. Zur Funktionenklasse  $\mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  gehören z.B. für  $L \geq 0$  auch alle Polynome

$$P(t) = \sum_{k=0}^L a_k (t - \tau_i)^k, \quad (17.45)$$

alle Überlagerungen von sinusförmigen Signalen

$$S(t) = \sum_{k=0}^L b_{S,k} (\sin(\omega_{S,k}(t - \tau_i) + \phi_{S,k})^k b_{C,k} \cos(\omega_{C,k}(t - \tau_i) + \phi_{C,k}))^k, \quad (17.46)$$

und auch Kompositionen von (17.45) und (17.46) auf jedem kompakten<sup>6)</sup> Intervall  $D_i = [t_i, t_{i+1}] \subset \mathbb{R}_{\geq 0}$ . Hierbei repräsentieren die Parameter in (17.45) und (17.46) konstante und reelle Koeffizienten  $a_k, b_{S,k}, b_{C,k} \in \mathbb{R}$ , zeitliche Verschiebungen  $\tau_i > 0$ , Kreisfrequenzen  $\omega_{S,k}, \omega_{C,k} > 0$  und Phasenwinkel  $\phi_{S,k}, \phi_{C,k} \in \mathbb{R}$  für alle  $i, k \in \mathbb{N}$ . Die zeitlichen Ableitungen  $\dot{P}(\cdot)$  und  $\dot{S}(\cdot)$  existieren und sind beschränkt auf jedem  $D_i$ . Diese skalaren Beispiele können direkt auf vektorielle Verläufe erweitert werden, z.B. durch  $\underline{y}^*(t) := [y_1^*(t), \dots, y_M^*(t)]^\top \in \mathbb{R}^M$  mit  $y_{i,\text{ref}}(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  für alle  $i \in \{1, \dots, M\}$ .

In Abb. 17.4 ist ein exemplarischer Sollwertverlauf über der Zeit  $[0s, 40s]$  dargestellt. Der Verlauf entsteht durch die Überlagerung

$$y^*(t) = \begin{cases} 10, & t \in [0s, 10s] \\ \text{sat}[5(t - 10s)]_{10}^{20}, & t \in [10s, 12s] \\ 20, & t \in [12s, 20s] \\ 3 \sin(0.05 \frac{\text{rad}}{s} (t - 20s)) \cos(0.1 \frac{\text{rad}}{s} (t - 20s) + \frac{\pi}{2}) + 20, & t \in [20s, 40s] \end{cases} \quad (17.47)$$

Der Referenzverlauf (17.47) kann als Prototyp zur Bewertung der Performanz von Folgewertregelungen (engl.: tracking) angesehen werden. Die Referenz (17.47) — dargestellt in Abb. 17.4 — konfrontiert den Regelkreis mit einem konstanten Sollwert (Festwertregelung) und einem sich ändernden rampen- bzw. sinusförmigen Verlauf (Folgewertregelung).

### 17.1.3 Systemklasse $\mathcal{S}$

In [103] werden die zulässige Systemklasse  $\mathcal{S}$  ausführlich beschrieben und allgemeine Prototypen für z.B. endlich-dimensionale lineare und auch nichtlineare *MIMO*<sup>7)</sup>-Systeme angeführt. Für diesen Beitrag soll vereinfachend und der Anschaulichkeit dienlich nur der *SISO*<sup>8)</sup> Fall betrachtet werden. Hierzu wird nochmals die nichtlineare und allgemeinste Darstellung der Systemklasse  $\mathcal{S}$  wiederholt. Im Prinzip müssen die drei bekannten Anforderungen für ‘high-gain’ stabilisierbare Systeme (siehe auch Kap. 16) erfüllt sein, d.h. falls der zu regelnde Prozess:

---

<sup>6)</sup> Eine kompakte Menge ist abgeschlossen und beschränkt.

<sup>7)</sup> multiple-input, multiple-output

<sup>8)</sup> single-input, single-output

- (A1) einen Relativgrad  $\delta$  von eins,
- (A2) eine (asymptotisch) stabile Nulldynamik (Minimalphasigkeit) und
- (A3) ein bekanntes (z.B. positives) Vorzeichen der instantanen Verstärkung

besitzt, kann Funnel-Control (bzw. jeder hochverstärkungsisierte Ansatz) zur Regelung des Prozesses (Systems) angewendet werden. Abbildung 17.1 zeigte bereits exemplarisch den geschlossenen Regelkreis. Systeme der Klasse  $\mathcal{S}$  bestehen aus zwei miteinander verkoppelten Subsystemen: der Vorwärtssdynamik  $\Sigma_1$  (Eigendynamik) und der internen Dynamik  $\Sigma_2$  (Nulldynamik nach BINF für  $y(\cdot) = 0$ ). Die Strecke wird von der Stellgröße  $u(t) \in \mathbb{R}$  getrieben. Die Stellgröße wirkt aufgrund des Relativgrades von eins direkt auf die zeitliche Ableitung des Ausgangs  $\dot{y}(t) = f(p(t), \beta(t), u(t))$ , wobei  $p(t) \in \mathbb{R}$  als Systemstörung und  $\beta(t) = (\mathbf{T}y)(t) \in \mathbb{R}$  als interne Systemgröße angesehen werden können. Entsprechend der Ableitung wird sich der Ausgang  $y(t) \in \mathbb{R}$  der Strecke verändern. Mit diesem Vorwissen lässt sich die Systemklasse  $\mathcal{S}$  mathematisch präzise definieren. Zuerst soll die Operatorabbildung  $\mathbf{T} : \mathcal{C}([-h; \infty); \mathbb{R}) \rightarrow \mathcal{L}_{loc}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$  genauer ausgeführt werden. Der Operator erlaubt eine Beschreibung der (asymptotischen) Nulldynamik in einem allgemeineren Sinn (z.B.  $BIBO^9$ -Stabilität):

**Definition 17.1 Operatorklasse  $\mathcal{T}$  [103, 104]:** Ein Operator  $\mathbf{T}$  ist Element der Klasse  $\mathcal{T}$ , genau dann, wenn für ein  $h \geq 0$  die folgenden Eigenschaften erfüllt werden:

1.  $\mathbf{T} : \mathcal{C}([-h; \infty); \mathbb{R}) \rightarrow \mathcal{L}_{loc}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$

2. Für jedes  $\delta > 0$  existiert ein  $\Delta > 0$ , sodass für alle  $\zeta(\cdot) \in \mathcal{C}([-h; \infty); \mathbb{R})$ ,

$$\sup_{t \in [-h, \infty)} |\zeta(t)| < \delta \Rightarrow |(\mathbf{T}\zeta)(t)| \leq \Delta \quad \text{für fast alle } t \geq 0 \quad (17.48)$$

3. Für alle  $t \in \mathbb{R}_{\geq 0}$  gilt:

- (a) für alle  $\zeta(\cdot), \xi(\cdot) \in \mathcal{C}([-h; \infty); \mathbb{R})$

$$\zeta(\cdot) \equiv \xi(\cdot) \quad \text{auf } [-h, t]$$

$$\Rightarrow (\mathbf{T}\zeta)(s) = (\mathbf{T}\xi)(s) \quad \text{für fast alle } s \in [0, t] \quad (17.49)$$

(b) für alle stetigen  $\beta : [-h, t] \rightarrow \mathbb{R}$  existieren  $\tau, \delta, c > 0$ , so dass für alle  $\zeta(\cdot), \xi(\cdot) \in \mathcal{C}([-h; \infty), \mathbb{R})$  mit  $\zeta|_{[-h, t]} = \beta = \xi|_{[-h, t]}$  und  $\zeta(s), \xi(s) \in \mathbb{B}_\delta(\beta(t))$  für alle Zeitpunkte  $s \in [t, t + \tau]$  folgendes gilt:

$$\text{ess-sup}_{s \in [t, t + \tau]} |(\mathbf{T}\zeta)(s) - (\mathbf{T}\xi)(s)| \leq c \sup_{s \in [t, t + \tau]} |\zeta(s) - \xi(s)| \quad (17.50)$$

---

<sup>9)</sup> bounded-input, bounded output

Der Operator (oder das Funktional)  $\mathbf{T}$  bildet also eine stetige Funktion in eine stückweise stetige und lokal essentiell beschränkte Funktion (Eigenschaft 1 in Definition 17.1) ab. Dadurch wird das zulässige Verhalten der internen Dynamik  $\beta$  (bzw. Nulldynamik) verallgemeinert. Die Minimalphasigkeitsforderung bei linearen Systemen oder die Forderung nach asymptotisch stabiler Nulldynamik bei nichtlinearen Systemen werden zu einer ‘bounded-input, bounded-output’ Forderung abgeschwächt (Eigenschaft 2 in Definition 17.1). Die Operatorabbildung soll nur von aktuellen bzw. vergangenen Eingängen abhängen, somit wird die Forderung nach Kausalität erfüllt (Eigenschaft 3a in Definition 17.1). Abschließend, entspricht Eigenschaft 3b in Definition 17.1 einer Forderung nach lokaler Lipschitz-Stetigkeit der Abbildung auf jedem kompakten Intervall. Hiermit wird die Lösbarkeit der Differentialgleichung im Sinne des Existenz- und Eindeutigkeitssatzes (siehe z.B. [235, 87]) gewahrt.

Mit der Definition des Operators  $\mathbf{T}$  im Hinterkopf lässt sich die Systemklasse  $\mathcal{S}$  präzise angeben:

**Definition 17.2 Systemklasse  $\mathcal{S}$  [103, 104]:**  $\mathcal{S}$  ist die Klasse der nichtlinearen single-input  $u$ , single-output  $y$  Systeme, beschrieben durch das Triple  $(p, f, \mathbf{T})$  und die nichtlineare Differentialgleichung der Form

$$\dot{y}(t) = f(p(t), (\mathbf{T}y)(t), u(t)), \quad y|_{[-h, 0]} = y^0 \in \mathcal{C}([-h, 0]; \mathbb{R}) \quad (17.51)$$

wobei für  $h \geq 0$ , folgende Eigenschaft erfüllt sein müssen:

1.  $p(\cdot) \in \mathcal{L}^\infty(\mathbb{R}; \mathbb{R})$
2.  $f(\cdot, \cdot, \cdot) \in \mathcal{C}(\mathbb{R} \times \mathbb{R} \times \mathbb{R}; \mathbb{R})$
3.  $\mathbf{T} : \mathcal{C}([-h, \infty); \mathbb{R}) \rightarrow \mathcal{L}_{loc}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$  ist Element der Operatorklasse  $\mathcal{T}$
4. Für jede die nicht leere, kompakte Menge  $\mathcal{C} \subset \mathbb{R} \times \mathbb{R}$  und jede Stellgrößenfolge  $(u_n)_{n \in \mathbb{N}} \subset \mathbb{R} \setminus \{0\}$  gilt:

$$|u_n| \rightarrow \infty \text{ wenn } n \rightarrow \infty$$

$$\implies \min_{(p, \beta) \in \mathcal{C}} \text{sign}(u_n) f(p, \beta, u_n) \rightarrow \infty \text{ wenn } n \rightarrow \infty \quad (17.52)$$

Die nicht negative Konstante  $h \geq 0$  beschreibt das ‘Gedächtnis’ des Systems (17.51). Der Term  $p(\cdot)$  stört die Vorwärtsdynamik und somit den Ausgang. Er kann als essentiell-beschränkte (interne und/oder externe) *Eingangsstörung* angesehen werden (Eigenschaft 1 in Definition 17.2).

Die Funktion  $f(\cdot, \cdot, \cdot)$  wird als stetig auf dem gesamten Definitionsbereich angenommen (Eigenschaft 2 in Definition 17.2), sodass *mindestens eine* Lösung des Anfangswertproblems (17.51) existiert (Peano Existenzsatz, siehe z.B. [235]).

Der Operator  $\mathbf{T}$  ist Element der Klasse  $\mathcal{T}$  (Eigenschaft 3 in Definition 17.2; siehe auch Definition 17.1) und bildet somit jeden beschränkten Ausgang  $y(\cdot)$  des Systems auf eine beschränkte interne Systemgröße  $\beta(\cdot)$  der Rückwärtsdynamik  $\Sigma_2$  ab. Man beachte hierbei, dass für eine Systemordnung  $n \in \mathbb{N}$  und für einen Relativgrad  $\delta = 1$ , die interne Dynamik die Ordnung  $q = n - 1$  besitzt. Die Systemgröße  $\beta(t) \in \mathbb{R}$  kann als Skalarprodukt des Zustandsvektors der internen Dynamik und eines systembedingten Einkoppelvektors angesehen werden (siehe (17.63) unter LTI SISO Systeme).

Abschließend verallgemeinert Eigenschaft 4 in Definition 17.2 das Konzept des bekannten (hier: positiven) Vorzeichens der instantanen Verstärkung für *lineare* Systeme auf nichtlineare Systeme, d.h. jede hinreichend große und vorzeichenrichtige Stellgröße beschleunigt die Strecke hinreichend schnell in die gewünschte Richtung (z.B. der Referenz entgegen).

### 17.1.3.1 LTI SISO Systeme der Klasse $\mathcal{S}$

Die Unterklasse der linearen zeit-invarianten (LTI) und single-input, single-output (SISO) Systeme ist von besonderem Interesse in der klassischen linearen Regelungstheorie und hinreichend gut bekannt in der Industrie. Daher soll nun die abstrakte allgemeine Klasse  $\mathcal{S}$  möglichst anschaulich anhand von LTI SISO Systemen ohne Durchgriff  $d = 0$  erläutert werden.

Ein  $n$ -dimensionales minimalphasiges System mit Eingang  $u(t) \in \mathbb{R}$ , Ausgang  $y(t) \in \mathbb{R}$  und dem Zustandsvektor  $\underline{x}(t) \in \mathbb{R}^n$  soll in Zustandsdarstellung

$$\left. \begin{aligned} \dot{\underline{x}}(t) &= \mathbf{A}\underline{x}(t) + \underline{b}u(t) & \text{mit Anfangswert} & \underline{x}(0) = \underline{x}^0 \in \mathbb{R}^n \\ y(t) &= \underline{c}^\top \underline{x}(t) \end{aligned} \right\} \quad (17.53)$$

betrachtet werden. Hierbei entsprechen  $\mathbf{A} \in \mathbb{R}^{n \times n}$  der Systemmatrix,  $\underline{b} \in \mathbb{R}^n$  dem Einkoppelvektor und  $\underline{c} \in \mathbb{R}^n$  dem Auskoppelvektor. Alle Einträge der Matrix und der Vektoren seien konstant und reell.

Ein LTI SISO System (17.53) ist Element der Systemklasse  $\mathcal{S}$ , falls folgende Anforderungen erfüllt sind:

(L1) einen Relativgrad von eins, d.h.  $\underline{c}^\top \underline{b} \neq 0$ ,

(L2) minimalphasiges System mit

$$\det \begin{bmatrix} s\mathbf{I}_n - \mathbf{A} & \underline{b} \\ \underline{c}^\top & 0 \end{bmatrix} \neq 0 \quad \text{für alle } s \in \overline{\mathbb{C}}_+ \quad (17.54)$$

(L3) ein bekanntes (z.B. positives) Vorzeichen der instantanen Verstärkung, d.h.  $\underline{c}^\top \underline{b} > 0$

Hierbei entsprechen (L1)-(L3) den oben genannten Anforderungen (A1)-(A3) für den linearen Fall. Dies soll nun sukzessive nachgewiesen werden.

Der Relativgrad lässt sich für beliebige Systeme am einfachsten durch sukzessives Ableiten des Ausgangs ermitteln (siehe z.B. [114] oder Kap. 12). Sobald die Stellgröße  $u(t)$  auf die  $\delta$ -te Ableitung  $y^{(\delta)}(t)$  direkt einwirkt, ist mit  $\delta \leq n$  der Relativgrad des Systems gefunden.

Somit muss  $\underline{c}^\top \underline{b} \neq 0$  für ein LTI SISO System (17.53) der Systemklasse  $\mathcal{S}$  gelten, da mit

$$\dot{y}(t) = \underline{c}^\top \dot{\underline{x}}(t) = \underline{c}^\top \underline{A} \underline{x}(t) + \underline{c}^\top \underline{b} u(t) \quad (17.55)$$

nur dann die Stellgröße auf die erste Ableitung wirken kann.

Mithilfe der Forderung (L2) kann System (17.53) auf Minimalphasigkeit geprüft werden. In diesem Sinne bedeutet Minimalphasigkeit, dass keine Nullstellen des Systems in der geschlossenen rechten komplexen Halbebene liegen. In anderen Worten besagt Forderung (L3), dass  $(\underline{A}, \underline{b})$  stabilisierbar und  $(\underline{A}, \underline{c})$  detektierbar<sup>10)</sup> sein müssen [103, 104, 144, 139].

Abschließend muss die instantane Systemverstärkung von (17.53) ein bekanntes (z.B. positives) Vorzeichen haben, d.h.  $\text{sign}(\underline{c}^\top \underline{b}) = 1$  und somit

$$\underline{c}^\top \underline{b} > 0. \quad (17.56)$$

Um nun den Zusammenhang zwischen dem allgemeinen nichtlinearen System (17.51) und dem LTI SISO System (17.53) herausarbeiten zu können, wird (17.53) zunächst in die sogenannte *Byrnes-Isidori Normalform* (BINF) transformiert.

Da  $\underline{c}^\top \underline{b} > 0$  existiert die Inverse  $(\underline{c}^\top \underline{b})^{-1}$ . Nun wähle eine Basismatrix  $\mathbf{V}$  des Kerns von  $\underline{c}$  mit

$$\mathbf{V} \in \mathbb{R}^{n \times (n-1)} \quad \text{so dass} \quad \text{Bild} \mathbf{V} = \text{Kern}(\underline{c}) \quad (17.57)$$

und

$$\mathbf{N} := (\mathbf{V}^\top \mathbf{V})^{-1} \mathbf{V}^\top (\mathbf{I}_n - \underline{b}(\underline{c}^\top \underline{b})^{-1} \underline{c}^\top) \in \mathbb{R}^{(n-1) \times n}. \quad (17.58)$$

Damit definiert man die lineare Abbildung (Koordinatentransformation)

$$\mathbf{S} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \underline{x} \mapsto \begin{pmatrix} \underline{y} \\ \underline{z} \end{pmatrix} := \mathbf{S}\underline{x} \quad (17.59)$$

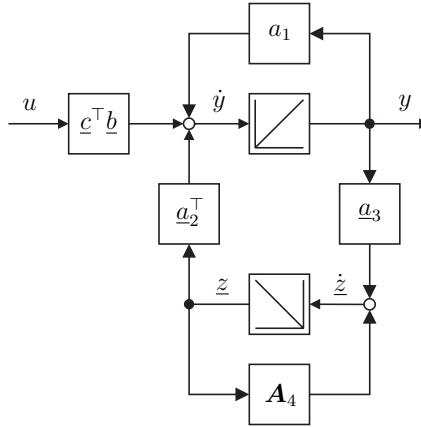
mit

$$\mathbf{S} := \begin{bmatrix} \underline{c}^\top \\ \mathbf{N} \end{bmatrix} \quad \text{und} \quad \mathbf{S}^{-1} = [\underline{b}(\underline{c}^\top \underline{b})^{-1}, \quad \mathbf{V}] \quad (17.60)$$

Die Koordinaten- bzw. Ähnlichkeitstransformation (17.59) überführt (17.53) in die gewünschte Byrnes-Isidori Normalform

---

<sup>10)</sup> Stabilisierbarkeit und Detektierbarkeit sind abgeschwächte Formen von Steuerbarkeit und Beobachtbarkeit: nicht-steuerbare bzw. nicht-beobachtbare Zustände müssen stabil sein.



**Abb. 17.5:** LTI SISO System in Byrnes-Isidori Normalform

$$\begin{aligned} \dot{y}(t) &= a_1 y(t) + \underline{a}_2^\top \underline{z}(t) + \underline{c}^\top \underline{b} u(t) , \quad y(0) = y^0 \\ \dot{\underline{z}}(t) &= \underline{a}_3 y(t) + \underline{A}_4 \underline{z}(t) , \quad \underline{z}(0) = \underline{z}^0 \end{aligned} \quad \left. \right\} \quad (17.61)$$

mit dem Systemausgang  $y(t) \in \mathbb{R}$ , den internen Systemzuständen  $\underline{z}(t) \in \mathbb{R}^{n-1}$  (der Nulldynamik für  $y(t) = 0$  für alle  $t \geq 0$ ) und der Konstanten  $a_1 = \underline{c}^\top \underline{A} \underline{b} (\underline{c}^\top \underline{b})^{-1} \in \mathbb{R}$ , den Vektoren  $\underline{a}_2^\top = \underline{c}^\top \underline{A} \mathbf{V} \in \mathbb{R}^{1 \times (n-1)}$ ,  $\underline{a}_3 = \underline{N} \underline{A} \underline{b} (\underline{c}^\top \underline{b})^{-1} \in \mathbb{R}^{(n-1) \times 1}$  und der Matrix  $\underline{A}_4 = \underline{N} \underline{A} \mathbf{V} \in \mathbb{R}^{(n-1) \times (n-1)}$ . Der Signalflussplan in Byrnes-Isidori Normalform ist in Abb. 17.5 dargestellt. Aufgrund  $\underline{c}^\top \underline{b} \neq 0$ , der Minimalphasigkeitsbedingung (17.54) und

$$\begin{aligned} \forall s \in \overline{\mathbb{C}}_+ : \quad 0 \neq \det \begin{bmatrix} s \mathbf{I}_n - \underline{A} & \underline{b} \\ \underline{c}^\top & 0 \end{bmatrix} &= \det \begin{bmatrix} s - a_1 & \underline{a}_2^\top & \underline{c}^\top \underline{b} \\ \underline{a}_3 & s \mathbf{I}_{n-1} - \underline{A}_4 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ &= (-1)^{n+2} \det \begin{bmatrix} \underline{a}_2^\top & \underline{c}^\top \underline{b} \\ s \mathbf{I}_{n-1} - \underline{A}_4 & 0 \end{bmatrix} \\ &= (-1)^{2N+3} \underline{c}^\top \underline{b} \det [s \mathbf{I}_{n-1} - \underline{A}_4] \end{aligned} \quad (17.62)$$

(Verwendung des Laplace'schen Entwicklungssatzes [50]) besitzt die Matrix  $\underline{A}_4$  ein Spektrum  $\sigma(\underline{A}_4) \subset \mathbb{C}_-$  in der offenen linken komplexen Halbebene, d.h. der homogenen Anteil der internen Dynamik ist exponentiell stabil.  $\underline{A}_4$  ist eine Hurwitz-Matrix. Definiert man nun

$$(\mathbf{T}y)(t) := a_1 y(t) + \underline{a}_2^\top \int_0^t \exp(\underline{A}_4(t-\tau)) \underline{a}_3 y(\tau) d\tau \quad (17.63)$$

und

$$p(t) := \begin{cases} \underline{a}_2^\top \exp(\underline{A}_4 t) \underline{z}^0 & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (17.64)$$

sind der (lineare) Operator  $\mathbf{T} : \mathcal{C}([0; \infty), \mathbb{R}) \rightarrow \mathcal{L}_{loc}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$  Element der Operatorklasse  $\mathcal{T}$  und die Störung  $p(\cdot) \in \mathcal{L}_{loc}^\infty(\mathbb{R}_{\geq 0}; \mathbb{R})$ . Nun lässt sich für  $h = 0$  mithilfe von

$$f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}, \quad (p, \beta, u) \mapsto p + \beta + \underline{c}^\top \underline{b} u \quad (17.65)$$

das LTI SISO System (17.53) mit der zusammengefassten internen Dynamik  $\beta(\cdot) = (\mathbf{T}y)(\cdot)$  in einer Variable ausdrücken

$$\begin{aligned} \dot{y}(t) &= p(t) + (\mathbf{T}y)(t) + \underline{c}^\top \underline{b} u(t), & y(0) &= y^0 \in \mathbb{R} \\ &= f(p(t), \beta(t), u(t)), \end{aligned} \quad (17.66)$$

was sogleich der allgemeinen Form (17.51) entspricht. Der Vergleich mit Abb. 17.1 zeigt, dass die Anteile (17.63) und (17.66) jeweils mit den verkoppelten Teilsystemen  $\Sigma_1$  und  $\Sigma_2$  übereinstimmen.

*Bemerkung.* Für jede nicht leere, kompakte Menge  $\mathcal{C} \subset \mathbb{R} \times \mathbb{R}$  kann folgende Abschätzung für die Summe  $p + \beta$  in (17.65) gefunden werden

$$|\min_{(p,\beta) \in \mathcal{C}} \text{sign}(u) (p + \beta)| \leq \max_{(p,\beta) \in \mathcal{C}} |p + \beta| \quad (17.67)$$

wobei  $\infty > \max_{(p,\beta) \in \mathcal{C}} |p + \beta| := c > 0$  auf jedem Kompaktum  $\mathcal{C}$ . Damit lässt sich Forderung (17.52) überprüfen, indem die Ergebnisse (17.67) und  $\text{sign}(u) (\underline{c}^\top \underline{b} u) = \underline{c}^\top \underline{b} |u| > 0$  kombiniert und für (17.65) ausgewertet werden:

$$\begin{aligned} \max_{(p,\beta) \in \mathcal{C}} |p + \beta| + \underline{c}^\top \underline{b} |u| &\geq \min_{(p,\beta) \in \mathcal{C}} \text{sign}(u) f(p, \beta, u) \\ &\geq -\max_{(p,\beta) \in \mathcal{C}} |p + \beta| + \underline{c}^\top \underline{b} |u| \quad \text{für alle } u \in \mathbb{R} \end{aligned} \quad (17.68)$$

Nun lässt sich schlussfolgern, dass

$$\begin{aligned} \min_{(p,\beta) \in \mathcal{C}} \text{sign}(u) f(p, \beta, u) &\geq -\max_{(p,\beta) \in \mathcal{C}} |p + \beta| + \underline{c}^\top \underline{b} |u| \\ &\geq -c + \underline{c}^\top \underline{b} |u| \quad \text{für alle } u \in \mathbb{R} \setminus \{0\} \end{aligned} \quad (17.69)$$

und somit Eigenschaft 4 in Definition 17.2 bzw. (17.52) mit der Forderung  $\underline{c}^\top \underline{b} > 0$  für LTI SISO Systeme gleichbedeutend ist.

*Bemerkung.* Kann für das zu untersuchende LTI SISO System (17.53) direkt eine Übertragungsfunktion der Form

$$G_S(s) = \frac{Y(s)}{U(s)} = \underline{c}^\top (s\mathbf{I}_n - \mathbf{A})^{-1} \underline{b} \quad (17.70)$$

$$= V_S \frac{1 + c_1 s + \dots + c_M s^M}{1 + a_1 s + \dots + a_{n-1} s^{n-1} + a_n s^n} =: V_S \frac{N_S(s)}{D_S(s)} \quad (17.71)$$

Simulationsparameter			
Systemdaten	Werte		
	$S_1$	$S_3$	$S_3$
Streckenverstärkung $V_S$	2	50	3
1. Zählerzeitkonstante $T_{N1}$	–	$0.5s$	–
1. Zählerzeitkonstante $T_{N2}$	–	$0.01s$	–
Zählerkreisfrequenz $\omega_{N0}$	–	–	$12\pi \frac{rad}{s}$
Zählerdämpfung $D_N$	–	–	0.05
Nennerzeitkonstante $T_{D1}$	$2s$	$1s$	$0.1s$
Nennerkreisfrequenz $\omega_{D0}$	–	$10 \frac{rad}{s}$	$8\pi \frac{rad}{s}$
Nennerdämpfung $D_D$	–	-0.3	0.07
Anfangswert $y(0)$	0	0	0

**Tabelle 17.1:** Simulationsdaten für die Beispieldsysteme  $S_1$ ,  $S_2$  und  $S_3$

angegeben werden, vereinfacht sich die Untersuchung deutlich. Der Relativgrad  $\delta = n - m$  bildet sich aus der Differenz des Nennergrades und des Zählergrades. Für Relativgrad  $\delta = 1$  gilt demzufolge  $m = n - 1$ . Für Minimalphasigkeit müssen die Wurzeln des Zählerpolynoms  $N_S(s)$  negativen Realteil aufweisen. Zur Überprüfung müssen also die Koeffizienten  $c_1, \dots, c_M$  grob bekannt sein, um im Sinne Ackermanns [2] robuste Stabilität der Zählerpolynomwurzeln nachprüfen zu können. Die instantane Verstärkung erhält man über den Grenzübergang

$$V_0 := \lim_{s \rightarrow \infty} \{s^{n-m} G_S(s)\} = V_S \frac{c_M}{a_N}. \quad (17.72)$$

### 17.1.3.2 Beispieldsysteme

Im Folgenden sollen die allgemein hergeleiteten Ergebnisse für drei einfache lineare Systeme veranschaulicht werden. Alle drei Beispieldsysteme werden innerhalb des Beitrages immer wieder auftauchen und für Simulationen oder Illustrationen herangezogen. In Tab. 17.1 sind die Systemparameter der drei Beispiele aufgelistet. Es sei jedoch betont, dass diese für Regler- bzw. Trichterdesign *nicht* benötigt werden, sondern lediglich für den interessierten Leser (z.B. für eigene Simulationen) explizit benannt sind.

#### Beispiel $S_1$ ( $PT_1$ )

Als erstes und einfachstes Beispiel soll ein  $PT_1$  System mit der Übertragungsfunktion

$$G_{S_1}(s) = \frac{Y(s)}{U(s)} = \frac{V_S}{1 + sT_{D1}} \quad (17.73)$$

betrachtet werden. Die Systemverstärkung  $V_S \in \mathbb{R} \setminus \{0\}$  und die Zeitkonstante  $T_{D1} \in \mathbb{R}$  seien konstant. Es lässt sich die einfache lineare Differentialgleichung mit Anfangswert  $y^0$  herleiten

$$\dot{y}(t) = -\frac{1}{T_{D1}}y(t) + \frac{V_S}{T_{D1}}u(t), \quad y(0) = y^0 \in \mathbb{R} \quad (17.74)$$

Für  $T_{D1} \neq 0$  und  $\text{sign}(V_S) = \text{sign}(T_{D1})$  ist das Beispielsystem  $S_1$  Element der Klasse  $\mathcal{S}$ , da alle drei Bedingungen erfüllt sind mit

- Relativgrad

$$\delta = 1 - 0 = 1, \quad (17.75)$$

- positiver instantaner Verstärkung

$$V_0 = \lim_{s \rightarrow \infty} s F_{S_1}(s) = \lim_{s \rightarrow \infty} \frac{V_S s}{1 + s T_{D1}} = \frac{V_S}{T_{D1}} > 0 \quad (17.76)$$

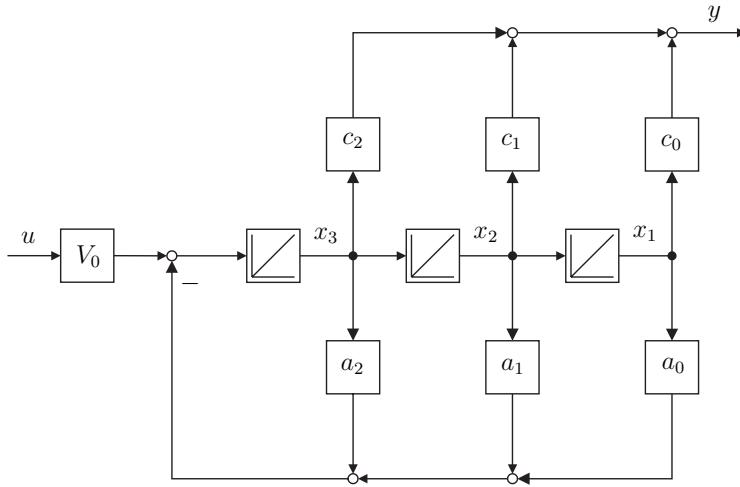
- und da keine interne Dynamik vorhanden ist, muss nicht auf Minimalphasigkeit geprüft werden (der einzige Pol wandert für steigende Verstärkungen des Reglers ins negativ Unendliche entsprechend der Wurzelortskurve in Abb. 17.7)

### Beispiel $S_2$ ( $PD_2T_3$ )

Das zweite Beispiel liege in Regelungsnormalform (siehe Abb. 17.6) vor und habe die Übertragungsfunktion

$$\begin{aligned} G_{S_2}(s) &= V_S \frac{(1 + sT_{N1})(1 + sT_{N2})}{(1 + sT_{D1}) \left( 1 + \frac{2D_D}{\omega_{D0}}s + \frac{1}{\omega_{D0}^2}s^2 \right)} \\ &= V_S \frac{1 + s(T_{N1} + T_{N2}) + s^2T_{N1}T_{N2}}{1 + s \left( T_{D1} + \frac{2D_D}{\omega_{D0}} \right) + s^2 \left( \frac{2D_D T_{D1}}{\omega_{D0}} + \frac{1}{\omega_{D0}^2} \right) + s^3 \frac{T_{D1}}{\omega_{D0}^2}} \\ &= V_S \frac{\omega_{D0}^2 T_{N1} T_{N2}}{T_{D1}} \cdot \frac{\frac{1}{T_{N1} T_{N2}} + s \frac{T_{N1} + T_{N2}}{T_{N1} T_{N2}} + s^2}{\frac{\omega_{D0}^2}{T_{D1}} + s \left( \frac{2D_D \omega_{D0}}{T_{D1}} + \omega_{D0}^2 \right) + s^2 \left( 2D_D \omega_{D0} + \frac{1}{T_{D1}} \right) + s^3} \\ &= V_0 \cdot \frac{c_0 + c_1 s + c_2 s^2}{a_0 + a_1 s + a_2 s^2 + s^3} \end{aligned} \quad (17.77)$$

mit den Nullstellen  $s_{01} = -\frac{1}{T_{N1}} < 0$  und  $s_{02} = -\frac{1}{T_{N2}} < 0$  (Wurzeln des Zählers) in der offenen linken komplexen Halbebene und den Polstellen  $p_1 = -\frac{1}{T_{D1}} < 0$  und  $p_{2,3} = -\omega_{D0} D_D \pm \omega_{D0} \sqrt{D_D^2 - 1}$  (komplex und instabil für  $D_D < 0$ ). Das Beispielsystem habe die Systemparameter wie in Tab. 17.1 angegeben. Es soll nun



**Abb. 17.6:** Beispielsysteme  $G_{S_2}(s)$  und  $G_{S_3}(s)$  in Regelungsnormalform (RNF)

nicht der vereinfachte Weg über die Analyse der Übertragungsfunktion (17.77) zur Überprüfung der drei Systembedingungen (A1)-(A3) gegangen werden, sondern der allgemeine Weg über die Zustandsdarstellung und deren Transformation in Byrnes-Isidori Normalform. Der Umweg wird deshalb beschritten, um später in Abschnitt 17.6 (nicht-identifizierende Lastdrehzahlregelung des nichtlinearen Zwei-Massen-Systems) mit dieser Art der Untersuchung vertraut zu sein. Durch Koeffizientenvergleich lässt sich aus der Übertragungsfunktion (17.77) direkt die Zustandsdarstellung in Regelungsnormalform ablesen

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\underline{x}(t) + \underline{b}u(t) \\ y(t) &= \underline{c}^\top \underline{x}(t) \end{aligned}, \quad \underline{x}(0) = \underline{x}^0 \in \mathbb{R}^3 \quad \left. \right\} \quad (17.78)$$

mit der Systemmatrix

$$\begin{aligned} \underline{A} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{\omega_{D0}^2}{T_{D1}} & -\left(\frac{2D_D\omega_{D0}}{T_{D1}} + \omega_{D0}^2\right) & -\left(2D_D\omega_{D0} + \frac{1}{T_{D1}}\right) \end{bmatrix}, \end{aligned} \quad (17.79)$$

dem Einkoppelvektor

$$\underline{b} = \begin{pmatrix} 0 \\ 0 \\ V_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ V_S \frac{\omega_{D0}^2 T_{N1} T_{N2}}{T_{D1}} \end{pmatrix} \quad (17.80)$$

und dem Auskoppelvektor

$$\underline{c}^\top = (c_0 \ c_1 \ c_2)^\top = \left( \frac{1}{T_{N1}T_{N2}} \ \frac{T_{N1}+T_{N2}}{T_{N1}T_{N2}} \ 1 \right)^\top. \quad (17.81)$$

Die Zustandsdarstellung (17.78) lässt sich nun analog der allgemeinen Herleitung in Abschnitt 17.1.3.1 durch Koordinatentransformation in BINF überführen, um dann die Zugehörigkeit zur Klasse  $\mathcal{S}$  zu überprüfen. Hierzu wählt man

$$\mathbf{V} = \begin{bmatrix} \frac{1}{c_0} & -\frac{1}{c_0} \\ -\frac{1}{c_1} & 0 \\ 0 & \frac{1}{c_2} \end{bmatrix} \quad (17.82)$$

als Basismatrix des  $\text{Kern}(\underline{c})$  entsprechend (17.57) und berechnet  $\mathbf{N}$  mithilfe von (17.58), man erhält die Transformationsmatrix

$$\mathbf{S} = \begin{bmatrix} \underline{c}^\top \\ \mathbf{N} \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 \\ 0 & -c_1 & 0 \\ -c_0 & -c_1 & 0 \end{bmatrix} \quad (17.83)$$

und dessen Inverse

$$\mathbf{S}^{-1} = [\underline{b}(cb)^{-1}, \ \mathbf{V}] = \begin{bmatrix} 0 & \frac{1}{c_0} & -\frac{1}{c_0} \\ 0 & -\frac{1}{c_1} & 0 \\ \frac{1}{c_2} & 0 & \frac{1}{c_2} \end{bmatrix}. \quad (17.84)$$

Die Koordinatentransformation  $(y, \underline{z}) = \mathbf{S}\underline{x}$  führt zur ‘neuen’ Systemdarstellung

$$\left. \begin{array}{l} \dot{y}(t) = \underbrace{\frac{c_1 - c_2 a_2}{c_2} y(t)}_{a_1} + \underbrace{\left( -\frac{c_2 a_0}{c_0} - \frac{c_0 - c_2 a_1}{c_1} \right)}_{a_2^\top} \underline{z}(t) + \underbrace{V_0 c_2}_{\underline{c}^\top \underline{b}} u(t) \\ \dot{\underline{z}}(t) = \underbrace{\left( c - \frac{c_1}{c_2} \right)}_{a_3} y(t) + \underbrace{\begin{bmatrix} 0 & -\frac{c_1}{c_2} \\ \frac{c_0}{c_1} & -\frac{c_1}{c_2} \end{bmatrix}}_{\mathbf{A}_4} \underline{z}(t), \quad \begin{pmatrix} y(0) \\ \underline{z}(0) \end{pmatrix} = \mathbf{S}\underline{x}^0 \end{array} \right\} \quad (17.85)$$

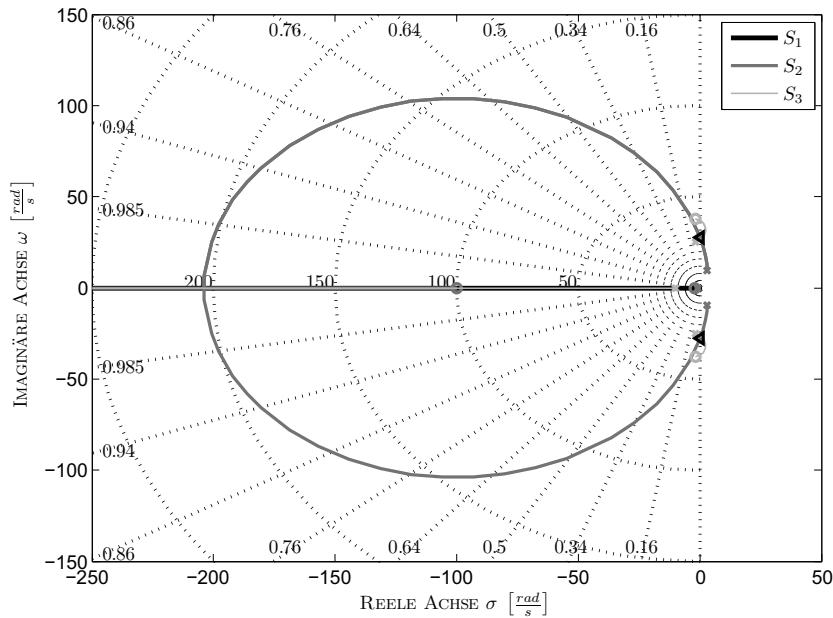
Das transformierte System lässt sich wie in Abb. 17.5 darstellen und ist für  $T_{N1}, T_{N2} > 0$  und  $V_S/T_{D1} > 0$  Element der Klasse  $\mathcal{S}$ , denn Überprüfung von (A1)-(A3) ergibt:

- Relativgrad (auch über (17.77) möglich)

$$\delta = n - m = 3 - 2 = 1, \quad (17.86)$$

- positive instantane Verstärkung

$$\underline{c}^\top \underline{b} = V_0 c_2 = V_S \frac{\omega_{D0}^2 T_{N1} T_{N2}}{T_{D1}} > 0 \quad (17.87)$$



**Abb. 17.7:** Wurzelortskurven der Beispielsysteme  $G_{S_1}(s)$ ,  $G_{S_2}(s)$  und  $G_{S_3}(s)$ : Die nötige Mindestverstärkung  $k^* > 0$  ist durch die beiden Dreiecke auf der imaginären Achse gekennzeichnet

- minimalphasiges System, da

$$\sigma(\mathbf{A}_4) = \left\{ \frac{-c_1 \pm \sqrt{c_1^2 - 4c_0c_2}}{2c_2} \right\} = \left\{ -\frac{1}{T_{N1}}, -\frac{1}{T_{N2}} \right\} \subset \mathbb{C}_- \quad (17.88)$$

Zur Veranschaulichung ist in Abb. 17.7 die Wurzelortskurve von  $S_2$  (neben  $S_1$  und  $S_3$ ) gezeigt. Für steigende Verstärkungen nähern sich die Pole des geschlossenen Regelkreises den Nullstellen der minimalphasigen Strecke (17.77) und dem negativ Unendlichen. Für alle konstanten Verstärkungen  $k|_{const.} > k^*$  (gekennzeichnet durch die Dreiecke auf der imaginären Achse) ist der geschlossene Regelkreis stabil, für kleinere Verstärkungen verbleiben zwei Pole in der rechten komplexen Halbebene.

### Beispiel $S_3$ ( $PD_2T_3$ mit konjugiert-komplexen Nullstellen)

Als drittes Beispielsystem betrachte man die Übertragungsfunktion

$$\begin{aligned}
 G_{S_3}(s) &= \frac{Y(s)}{U(s)} = \frac{N_S(s)}{D_S(s)} \\
 &= V_S \cdot \frac{1 + \frac{2D_N}{\omega_{N0}}s + \frac{1}{\omega_{N0}^2}s^2}{(1 + sT_{D1}) \left( 1 + \frac{2D_D}{\omega_{D0}}s + \frac{1}{\omega_{D0}^2}s^2 \right)} \\
 &= V_S \frac{\omega_{D0}^2}{\omega_{N0}^2 T_{D1}} \frac{\omega_{N0}^2 + 2D_N \omega_{N0} s + s^2}{\frac{\omega_{D0}^2}{T_{D1}} + s \left( \frac{2D_D \omega_{D0}}{T_{D1}} + \omega_{D0}^2 \right) + s^2 \left( 2D \omega_{D0} + \frac{1}{T_{D1}} \right) + s^3} \\
 &= V_0 \cdot \frac{c_0 + c_1 s + c_2 s^2}{a_0 + a_1 s + a_2 s^2 + s^3} \tag{17.89}
 \end{aligned}$$

mit den Nullstellen  $s_{1,2} = -\omega_{N0}D_N \pm \omega_{N0}\sqrt{D_N^2 - 1}$  in der linken offenen komplexen Ebene und den Polstellen  $p_1 = -\frac{1}{T_{D1}} < 0$  und  $p_{2,3} = -\omega_{D0}D_D \pm \omega_{D0}\sqrt{D_D^2 - 1}$ . Für die durchgeführten Simulationen sind die Systemdaten in Tab. 17.1 zusammengefasst.

Für alle Parameterwerte  $\omega_{N0} > 0$ ,  $D_N > 0$  und  $V_S/T_{D1} > 0$  sind die drei notwendigen Bedingungen (A1)-(A3) für Zugehörigkeit zur Systemklasse  $\mathcal{S}$  erfüllt:

- Relativgrad

$$\delta = n - m = 3 - 2 = 1 \tag{17.90}$$

- positive instantane Verstärkung

$$\underline{c}^\top \underline{b} = 2V_0 c_2 = 2 \cdot \frac{V_S \omega_{D0}^2}{T_{D1} \omega_{N0}^2} > 0 \tag{17.91}$$

- Minimalphasigkeit

$$\sigma(\mathbf{A}_4) = \left\{ \frac{-c_1 \pm \sqrt{c_1^2 - 4c_0c_2}}{2c_2} \right\} = \left\{ -\omega_{N0}D_N \pm \omega_{N0}\sqrt{D_N^2 - 1} \right\} \subset \mathbb{C}_- \tag{17.92}$$

In Abb. 17.7 ist auch die Wurzelortskurve des Beispielsystems  $S_3$  aufgetragen. Für stetig steigende Werte der Rückführungsverstärkung nähern sich die Pole den Nullstellen des ungeregelten Systems (17.89) und dem negativ Unendlichen. Aus Abbildung 17.7 kann die Mindestverstärkung  $k^*$  abgelesen werden, für jeden (konstanten) Verstärkungswert  $k|_{const.} > k^*$  ergibt sich somit ein stabiler Regelkreis. Der geschlossene Regelkreis aus Beispielsystem  $S_3$  und Funnel-Regler wird aufgrund der konjugiert-komplexen Nullstellen  $s_{1,2}$  schwingungsanfällig sein (siehe auch Abschnitt 17.3).

### 17.1.4 Regelziel

Für jede vorgegebene beschreibende Trichterfunktion  $\varphi(\cdot) \in \Phi$ , jede Referenz  $y^*(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  und jede Strecke der Klasse  $\mathcal{S}$  sichert das einfache proportionale und zeitvariante Regelgesetz (17.1) die Regelziele:

- Folgewertregelung mit vorgegebenem transienten Verhalten innerhalb des *a priori* festgelegten Trichters
- Folgewertregelung mit vorgegebener asymptotischer Endgenauigkeit

Der Regelfehler  $e(\cdot)$  ‘entwickelt’ sich also nur innerhalb der Trichtergrenzen  $\pm \partial \mathcal{F}_\varphi(\cdot)$ . Zusätzlich bleiben alle weiteren Größen des Regelkreises (z.B. Reglerverstärkung  $k(\cdot)$ , Zustandsgrößen  $x(\cdot)$  und Stellgröße  $u(\cdot)$ ) beschränkt. Auf die Formulierung der Sätze und Beweise wird verzichtet und dem interessierten Leser Satz 7 (mit Beweis) in [103] oder Satz 2 mit Proposition 3 und 5 (jeweils mit Beweisen) in [104] empfohlen.

## 17.2 Kundenanforderungen

In der Industrie wird die gewünschte Reglerperformanz eines Regelkreises häufig für einen Sollwertsprung  $y^*(t) = \bar{y}^* \sigma(t - t_0)$  der Amplitude  $\bar{y}^* \in \mathbb{R} \setminus \{0\}$  auf einem Intervall  $I := [t_0, t_{end}]$  anhand der folgenden drei Kundenanforderungen bewertet bzw. festgelegt (siehe auch Abb. 17.8) [203, 184, 139]:

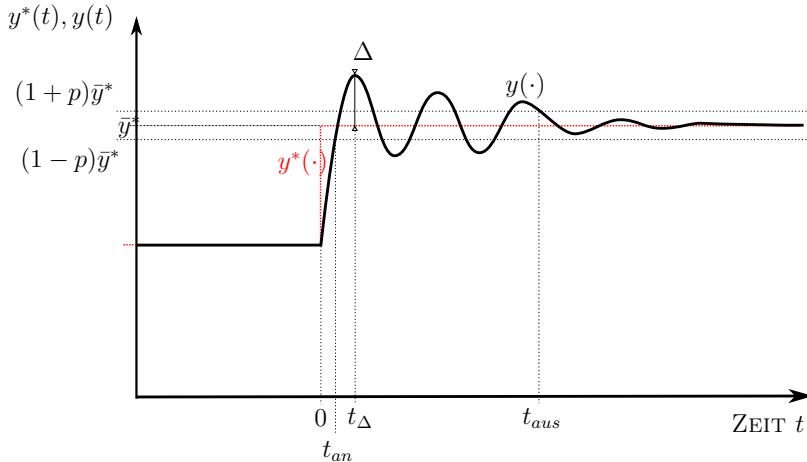
- (KA1) Anregelzeit  $t_{an}$  [s], d.h. hier schneidet der Ausgang  $y(t_{an})$  zum ersten mal das Toleranzband bei  $(1 - p)\bar{y}^*$  für  $1 > p > 0$  (z.B.  $p \in [3\%, 5\%]$ )
- (KA2) Ausregelzeit  $t_{aus}$  [s], d.h. der Ausgang  $y(t)$  verbleibt für alle  $t \geq t_{aus}$  innerhalb des Toleranzbandes  $(1 \pm p)\bar{y}^*$
- (KA3) Überschwingweite  $\Delta = \max_{t \in [t_{an}, t_{end}]} |y(t) - \bar{y}^*| / \bar{y}^*$

Jeder nicht steigende Trichterrand (d.h.  $\partial \dot{\mathcal{F}}_\varphi(t) \leq 0$  für alle  $t \geq 0$  und  $\partial \mathcal{F}_\varphi(t_1) \geq \partial \mathcal{F}_\varphi(t_2)$  für alle  $t_2 > t_1 \geq 0$ ) ist in der Lage Kundenanforderungen im Sinne von Abb. 17.8 direkt im Regler- bzw. Trichterdesign zu berücksichtigen. Für die Wahl  $\nu = (1 - p)\bar{y}^* > 0$  und  $\tau_\nu = t_{an} > 0$  garantiert ein Trichterdesign  $\partial \mathcal{F}_\varphi(\tau_\nu) = \nu$  gleichzeitig die Einhaltung von (KA1)-(KA3) in Abb. 17.8 beschriebenen Kundenanforderungen. Somit ist gesichert, dass der (absolute) Regelfehler zum gewünschten Zeitpunkt  $\tau_\nu$  die gewünschte Schranke  $\nu$  unterschreitet und

$$\forall t \geq \tau_\nu : \quad |e(t)| < \nu = \partial \mathcal{F}_\varphi(\tau_\nu). \quad (17.93)$$

Zusätzlich kann für hinreichend große Zeiten eine gewünschte Endgenauigkeit  $\mu < \nu$  festgelegt werden, d.h.

$$\limsup_{t \rightarrow \infty} |e(t)| < \lim_{t \rightarrow \infty} \partial \mathcal{F}_\varphi(t) = \mu. \quad (17.94)$$

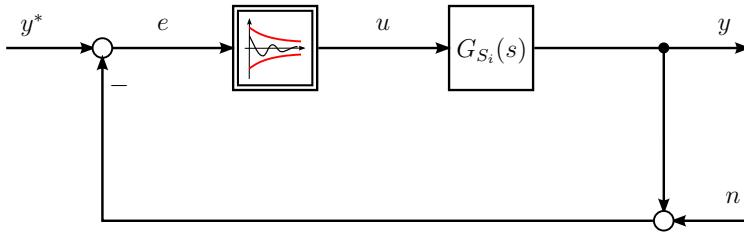


**Abb. 17.8:** Festlegung der Regelgüte eines Regelkreises bei Sollsprung  $y^*(t) = \bar{y}^* \sigma(t - t_0)$  zum Zeitpunkt  $t_0 = 0$  und  $\bar{y}^* > 0$  durch Anregzeit  $t_{an}$ , Ausregelzeit  $t_{aus}$  und Überschwingweite  $\Delta$  zum Zeitpunkt  $t_\Delta > 0$

Kundenanforderungen ( $\tau_\nu, \nu, \mu$ ) und Design der Beispieltrichter					
$\partial\mathcal{F}_\infty(t)$	$\partial\mathcal{F}_E^*(t)$	$\partial\mathcal{F}_E(t)$	$\partial\mathcal{F}_G(t)$	$\partial\mathcal{F}_S(t)$	
–	$\frac{1}{\varphi_0^*} >  e(0) $	$\frac{1}{\varphi_{E,0}} >  e(0) $	$\frac{1}{\varphi_{G,0}} >  e(0) $	$\frac{1}{\varphi_{S,0}} >  e(0) $	
$\lambda = \nu$	$\lambda^* = \frac{\nu}{1-\varphi_0^*\nu}$	$\frac{1}{\varphi_{E,\infty}} = \mu$	$\frac{1}{\varphi_{G,\infty}} = \mu$	$\frac{1}{\varphi_{S,\infty}} = \mu$	
$\tau = \tau_\nu$	$\tau^* = \tau_\nu$	$T_E = \frac{\tau_\nu}{\ln\left(\frac{1/\varphi_{E,0}-\mu}{(\nu-\mu)}\right)}$	$T_G = \frac{\tau_\nu}{\sqrt{\ln\left(\frac{1/\varphi_{G,0}-\mu}{(\nu-\mu)}\right)}}$	$\tau_\mu = \frac{1-\varphi_{S,0}\mu}{1-\varphi_{S,0}\nu}\tau_\nu$	
$\varepsilon = \frac{\nu-\mu}{\nu}$	$\varepsilon^* = \frac{1-\mu/\nu}{1-\varphi_0^*\mu}$	–	–	–	

**Tabelle 17.2:** Berechnungsvorschriften für Beispieltrichterränder bei Vorgabe der Kundenanforderungen ( $\tau_\nu, \nu, \mu$ )

Die Kundenanforderungen (KA1)-(KA3) in Abb. 17.8 können also durch das Triple  $(\tau_\nu, \nu, \mu)$  im Trichterdesign (siehe Abb. 17.2) umgesetzt werden. Lediglich für  $t < t_{an}$  kann z.B. ein Überschwingen mit  $|y(t)| > \Delta + \bar{y}^*$  unter Umständen nicht vermieden werden. Hierzu ist ein ‘asymmetrischer Trichterverlauf’ notwendig, solch ein Trichterentwurf wird in Abschnitt 17.5 (Error Reference Control) vorgestellt. In Tab. 17.2 sind für alle Beispieltrichter (siehe S. 703ff.) die Berechnungsvorschriften für die einzelnen Parameter der beschreibenden Trichterfunktion zur Einhaltung der Kundenanforderungen  $(\tau_\nu, \nu, \mu)$  zusammengefasst.



**Abb. 17.9:** Beispielhafter Regelkreis aus System  $F_{S_i}$  und Funnel-Regler (FC) für alle  $i = 1, 2, 3$

---

### Simulationsparameter für Regelkreis FC + $S_2$

---

Kundenanforderungen ( $\tau_\nu, \nu, \mu$ )	(0.1s, 1, 0.1)
Anfangswert $\frac{1}{\varphi_{E,0}}$	12
Skalierungsfunktion $\varsigma(\cdot)$	1
Referenz $y^*(t)$	$10 \sigma(t)$

---

**Tabelle 17.3:** Kundenanforderungen, Anfangswert des exponentiellen Trichters, Skalierungsparameter und Referenz für Regelkreis FC und  $S_2$  für alle Trichterränder

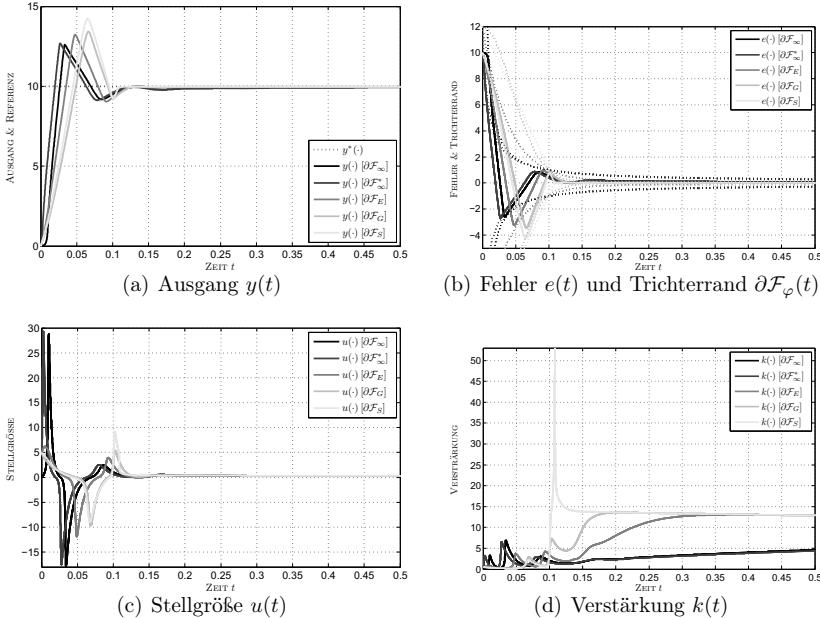
### Simulationsbeispiele

In diesem Abschnitt soll die Reglerperformanz des Funnel-Reglers (FC) am Beispielsystem  $S_2$  simulativ untersucht werden. Der Regelkreis ist in Abb. 17.9 skizziert. Messrauschen  $n(\cdot) = 0$  wird vernachlässigt. Es sollen alle fünf vorgestellten Beispieltrichter  $\partial\mathcal{F}_\infty, \partial\mathcal{F}_\infty^*, \partial\mathcal{F}_E, \partial\mathcal{F}_G$  und  $\partial\mathcal{F}_S$  (siehe Abschnitt 17.1.1) zum Einsatz kommen und deren Auswirkung auf den geschlossenen Regelkreis bewertet werden. Um die Ergebnisse vergleichen zu können, sind alle Beispieltrichterränder entsprechend der Kundenanforderung  $(\tau_\nu, \nu, \mu) = (0.1s, 1, 0.1)$  ausgelegt. Die Trichterränder mit endlichem Anfangswert starten einheitlich bei  $\partial\mathcal{F}_\varphi(0) = 12$ . Als Regelgesetz wird (17.1) mit (17.5) und  $\varsigma(\cdot) = 1$ , also

$$u(t) = \frac{1}{d_V(t, e(t))} e(t) = \frac{1}{\partial\mathcal{F}_\varphi(t) - |e(t)|} e(t) = \frac{\varphi(t)}{1 - \varphi(t)|e(t)|} \quad (17.95)$$

für alle fünf Trichterränder implementiert, wobei  $\varphi$  durch die entsprechende beschreibende Trichterfunktion ersetzt werden muss. D.h. die Reglerperformanz unterscheidet sich nur aufgrund der unterschiedlichen beschreibenden Trichterfunktionen  $\varphi_\infty, \varphi_\infty^*, \varphi_E, \varphi_G$  und  $\varphi_S$ . In Tab. 17.3 sind die Simulationsparameter für den Vergleich zusammengestellt. Die Simulationsergebnisse sind vergleichend in Abbildung 17.10 zusammengefasst. In Abb. 17.10a sind die Verläufe der Refe-

renz  $y^*(\cdot)$  und der entsprechenden Systemausgänge  $y(\cdot)$  dargestellt. Die Verläufe der Beispieltrichterränder und der zugehörigen Fehler  $e(\cdot)$  zeigt Abb. 17.10b. In Abb. 17.10c & d sind jeweils die Stellgrößen  $u(\cdot)$  und die Verstärkungen  $k(\cdot)$  geplottet. Das Regelziel (Festwertregelung mit Vorgabe des transienten Verhal-



**Abb. 17.10:** Simulativer Vergleich der geschlossenen Regelkreise aus FC und Beispielsystem  $S_2$  für die Beispieltrichterränder  $\partial\mathcal{F}_\infty$ ,  $\partial\mathcal{F}_\infty^*$ ,  $\partial\mathcal{F}_E$ ,  $\partial\mathcal{F}_G$  und  $\partial\mathcal{F}_S$

tens) kann für alle Trichterränder eingehalten werden. Der Fehler verbleibt innerhalb des vorgegebenen Trichters. Die Verstärkung passt sich in Abhängigkeit der vertikalen Distanz zwischen Fehler und Trichterrand entsprechend an. Alle Regelkreise zeigen ein deutliches Überschwingen, was zwar durch eine geänderte Kundenanforderung ( $\tau_\nu, \hat{\nu}, \hat{\mu}$ ) u.U. verkleinert, aber niemals durch die Wahl eines symmetrischen Trichterrandes beseitigt werden kann. Diese Beobachtung motiviert den erweiterten Ansatz Error Reference Control (ERC) in Kap. 17.5. Insbesondere bei Einsatz des ‘Simplen Trichterrandes’ kommt es aufgrund des ‘Knickes’ im Trichterverlauf (bei  $\tau_\mu \approx 1.1\text{s}$ ) zu einem drastischen Peak in der Stellgröße (siehe Abb. 17.10d).

### 17.3 Skalierung der Reglerverstärkung

Eine Skalierung

$$\varsigma: \mathbb{R}_{\geq 0} \rightarrow [m, \infty) \quad \text{und} \quad 1/\varsigma(\cdot) \in \Phi \quad (17.96)$$

der Reglerverstärkung (17.5) und (17.6) (oder der gewählten Distanz) ist zulässig, da für  $\varsigma(\cdot) \geq m > 0$  der nötige Eingriff durch das Regelgesetz (17.1) gewahrt bleibt [104]. Die Skalierung entspricht einem zusätzlichen Freiheitsgrad beim Reglerdesign. Insbesondere kann eine minimale Verstärkung fixiert werden, da z.B. bei Auswertung der vertikalen Distanz gilt:

$$\forall t \geq 0 \forall e \in \mathbb{R}: \quad k(t, e) = \frac{\varsigma(t)}{\partial \mathcal{F}_\varphi(t) - |e|} \geq \frac{m}{\|\partial \mathcal{F}_\varphi\|_\infty} > 0. \quad (17.97)$$

Die erhöhte Mindestverstärkung hat eine beschleunigte Antwort und/oder eine erhöhte Endgenauigkeit (bei Strecken ohne integralen Anteil) des Regelkreises zur Folge. Im einfachsten Fall ist es häufig vorteilhaft  $k(t, e) \geq 1$  für alle  $t \geq 0$  und  $e \in \mathbb{R}$  zu wählen, d.h.  $\varsigma(t) = \partial \mathcal{F}_\varphi(t)$ . In der Anwendung sind häufig kleine Endgenauigkeiten  $1 \gg \mu > 0$  wünschenswert, um dann aber  $k(t, e(t)) = \varsigma(t)/(\partial \mathcal{F}_\varphi(t) - |e(t)|) < 1/(\partial \mathcal{F}_\varphi(t) - |e(t)|)$  für  $t > \tau \in \{\tau \in \mathbb{R}_{\geq 0} \mid \partial \mathcal{F}_\varphi(\tau) < 1\}$  zu vermeiden, ist folgende Wahl der Skalierung

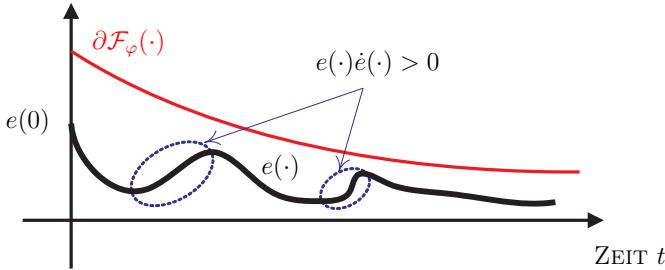
$$\forall t \geq 0: \quad \varsigma(t) = \partial \mathcal{F}_\varphi(t) + (1 - \mu) \quad (17.98)$$

zu bevorzugen.

Zusätzlich ermöglicht der gewonne Freiheitsgrad das transiente Verhalten in der Art positiv zu beeinflussen, dass z.B. Schwingungen im Fehlersignal  $e(t)$  reduziert werden können (siehe Abb. 17.11 als Motivation). Hierzu muss lediglich die Ableitung des Fehlers  $\dot{e}(t)$  vorliegen bzw. rekonstruierbar sein. Eventuell ist eine Glättung (Filterung) des differenzierten Signals nötig, um Rauscheinflüsse zu reduzieren. Die inhärente Verzögerung einer Signalfilterung verringert jedoch die positive dämpfende Eigenschaft der Skalierung. Der folgende Abschnitt schlägt eine einfache Erweiterung vor, um z.B. überschwingende oder oszillierende Fehlerverläufe zu bedämpfen.

#### ‘Bedämpfende’ Skalierung

Funnel-Control garantiert zwar einen Fehlerverlauf innerhalb des gewählten Trichters, jedoch können insbesondere für kleine Zeiten (v.a. dann wenn der Trichter noch ‘weit’ ist) Überschwingen und/oder Oszillationen im Regelfehler auftreten. In vielen Industrieanwendungen ist solch eine Regelkreisantwort unerwünscht. Zudem ist im vorhinein eine Aussage über das Auftreten von Oszillationen oder Überschwingen nicht ohne weiteres möglich — das System ist nur strukturell bekannt.



**Abb. 17.11:** Mögliche Schwingungen des Fehlers  $e(\cdot)$  innerhalb der Trichtergrenzen und grundlegende Motivation für die Dämpfungsskalierung  $\varsigma_D(\cdot)$  in (17.99)

Die Dämpfungsskalierung

$$\varsigma_D : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R},$$

$$t \mapsto \varsigma_D(t) := \text{sat} \left[ \varsigma_G e(t) \dot{e}(t) \right]_0^{\varsigma_D^{max}} + \varsigma_0 \quad (17.99)$$

kann sofern  $\dot{e}(\cdot)$  zur Verfügung steht direkt implementiert werden, hierbei sind  $\varsigma_0 \geq 1$ ,  $\varsigma_G \geq 0$  und  $\varsigma_D^{max} > 0$  frei wählbar. Der maximale Wert der Sättigung  $\varsigma_D^{max} > 0$  limitiert z.B. die Verstärkung von Messrauschen. Somit gilt

$$\forall t \geq 0 : \quad \varsigma_D(t) \in [\varsigma_0, \varsigma_D^{max} + \varsigma_0], \quad (17.100)$$

und  $1/\varsigma_D(\cdot) \in \Phi$ , da  $e(\cdot)$  und  $\dot{e}(\cdot)$  stetig und beschränkt sind [104].

In Abb. 17.11 sind Motivation und grundlegende Idee für die bedämpfende Skalierung (17.99) illustriert. Für alle Zeitpunkte  $t \geq 0$  bei denen  $|e(t)|$  abnimmt (also  $e(t)\dot{e}(t) < 0$ ) ist die Dämpfungsskalierung (17.99) nicht aktiv, da der Regelfehler sich in die gewünschte Richtung bewegt (er entfernt sich vom Trichterrand). Sobald der Absolutbetrag des Regelfehlers aber steigt (also  $e(t)\dot{e}(t) > 0$ ), erhöht für  $\varsigma_G, \varsigma_D^{max} > 0$  die Dämpfungsskalierung (17.99) die Reglerverstärkung  $k(t, e(t))$  zusätzlich. Aufgrund der ‘high-gain’ Fähigkeit der Systemklasse  $\mathcal{S}$  bewirkt die erhöhte Verstärkung eine Beschleunigung des Prozesses. Die größere Stellgröße  $u(t)$  zwingt den Reglerfehler ‘schneller’ zurück auf Null.

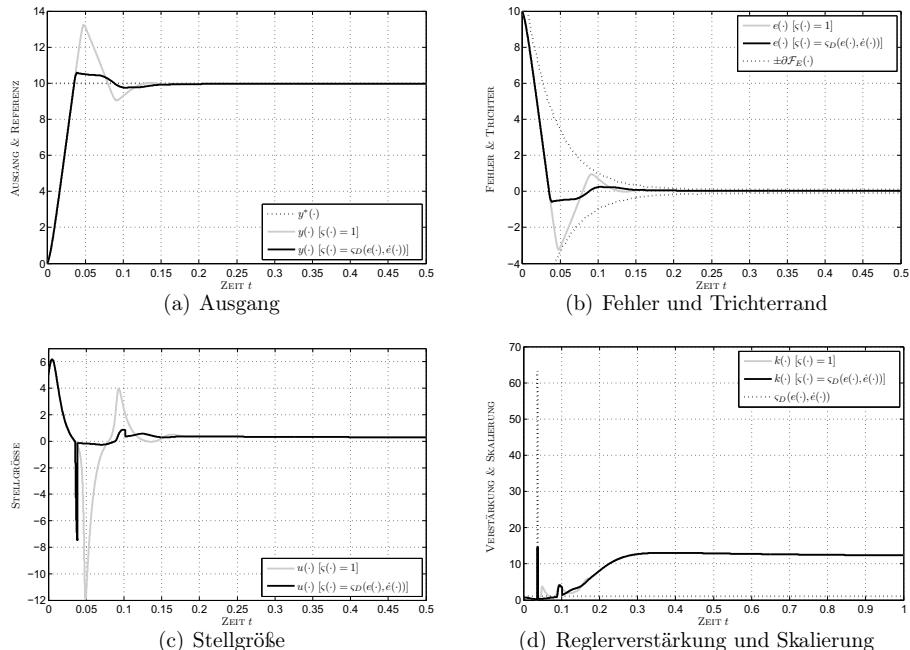
Der vorgeschlagene Weg zur Dämpfung der Systemantwort über die Skalierungsfunktion (17.99) ist nicht physikalisch motiviert (z.B. im Sinne einer Reduktion der Energie im System). Es wird lediglich die inhärente Eigenschaft hochverstärkungsstabilisierbarer Systeme (also Systeme für die (A1)-A(3) zutreffen) ausgenutzt, dass hohe Verstärkungen den geschlossenen Regelkreis stabilisieren und zusätzlich beschleunigen.

Bei Stellgrößenbeschränkung (z.B.  $|u(t)| \leq u_{max}$  für alle  $t \geq 0$ ) und zu groß gewählter Verstärkungswerte (z.B.  $\varsigma_G \gg 1$ ) arbeitet der Regelkreis eventuell

schon für sehr kleine Regelfehler  $|e(t)| \ll 1$  als schaltender Zwei-Punkt-Regler (bang-bang regulator) mit  $u(t) \in \{-u_{max}, +u_{max}\}$  für alle  $t \geq 0$  (ähnlich dem Schnattern bei ‘sliding-mode control’). Abhängig von der Anwendung ist dies durch eine Verringerung von  $\varsigma_G$  zu vermeiden.

### Simulationsbeispiele

Die folgenden Simulationsbeispiele zeigen den gewünschten Effekt der Dämpfungsskalierung (17.99) auf das Systemverhalten des geschlossenen Regelkreises: i) Überschwingen (siehe Abb. 17.12b) und ii) auftretende Oszillationen (siehe Abb. 17.13b) im Fehlerverlauf  $e(\cdot)$  können reduziert bzw. unterdrückt werden. Es werden die vertikale Distanzauswertung (17.3) und der Exponentielle Trichterrand  $\partial\mathcal{F}_E$  mit (17.32) implementiert. Die geschlossenen Regelkreise entsprechen jeweils dem in Abb. 17.9: für Beispielsystem  $S_2$  und  $S_3$ . Messrauschen  $n(\cdot)$  ist nicht berücksichtigt. In Tab. 17.4 sind alle Simulationsparameter zusammengefasst.

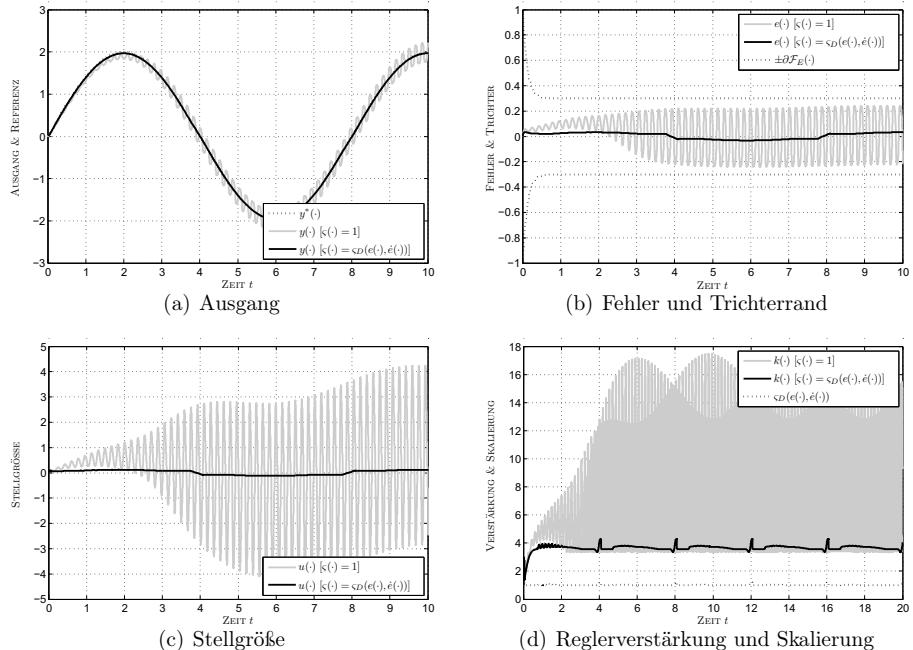


**Abb. 17.12:** System  $S_2$ : Simulationsergebnisse für Regelkreis  $FC + S_2$  ohne und mit Dämpfungsskalierung  $\xi_D(\cdot)$

### Simulationsparameter für Regelkreise mit $S_2$ und $S_3$

	$S_2$	$S_3$
Kundenanforderungen $(\tau_\nu, \nu, \mu)$	$(0.1s, 1, 0.1)$	$(0.1s, 0.6, 0.3)$
Anfangswert $\frac{1}{\varphi_{E,0}}$	12	1
Skalierungsfunktion $\varsigma(\cdot)$	$\varsigma(\cdot) = 1$ oder $\varsigma(\cdot) = (17.99)$	
Minimaler Wert $\varsigma_0$	1	1
Verstärkung $\varsigma_G$	100	100
Obere Schranke $\varsigma_D^{max}$	199	9
Referenz $y^*(t)$	$10\sigma(t)$	$2\sin(\frac{\pi}{4}t)$

**Tabelle 17.4:** Kundenanforderungen, Anfangswert des exponentiellen Trichters, Skalierungsparameter und Referenzsignale für Regelkreise aus FC und  $S_2$  und  $S_3$



**Abb. 17.13:** System  $S_3$ : Simulationsergebnisse für Regelkreis FC +  $S_3$  ohne und mit Dämpfungsskalierung  $s_D(\cdot)$

**Regelkreis aus Funnel-Control FC und Beispielsystem  $S_2$ :** In Abb. 17.12 sind die Simulationsergebnisse für den Regelkreis FC und  $S_2$  gezeigt. Der Regelkreis soll dem Sollwertsprung  $y^*(t) = 10\sigma(t)$  folgen. Der anfängliche Fehler

beträgt somit  $e(0) = 10$ . In Abb. 17.12a und 17.12b sind jeweils der Verlauf des Systemausgangs  $y(\cdot)$  bzw. des Fehlers  $e(\cdot)$  ohne und mit Dämpfungsskalierung dargestellt. Aufgrund der Skalierung mit (17.99) (siehe Abb. 17.12c) und der sich daraus ergebenden Stellgröße (siehe Abb. 17.12d) kann das anfängliche Überschwingen sichtlich reduziert werden. In Abb. 17.12d ist der Verlauf von  $\varsigma_D(\cdot)$  aufgetragen, wie erwartet steigt für alle  $t \geq 0$  mit  $e(t)\dot{e}(t) > 0$  die Skalierung  $\varsigma_D(t)$  deutlich an.

**Regelkreis aus Funnel-Control FC und Beispielsystem  $S_3$ :** In Abb. 17.13 sind Simulationsergebnisse für FC und  $S_3$  zusammengefasst. Hier soll einer sinusförmigen Referenz  $y^*(t) = 2\sin(\frac{\pi}{4}t)$  gefolgt werden. Abbildungen 17.13a und 17.13b zeigen wieder vergleichend die Verläufe der Systemausgänge  $y(\cdot)$  und der Fehler  $e(\cdot)$  ohne und mit Dämpfungsskalierung. Für alle  $t \geq 0$  mit  $e(t)\dot{e}(t) > 0$  steigen die Skalierung (siehe Abb. 17.13d) als auch die Reglerverstärkung  $k(\cdot)$  (siehe Abb. 17.13c) minimal aber ausreichend stark an. Die generierten Stellgrößen (siehe Abb. 17.13d) bedämpfen die Schwingungen nahezu vollkommen, wogegen im unskalierten Fall mit  $\varsigma(\cdot) = 1$  turbulente Schwingungen in Ausgang, Fehler, Stellgröße und Verstärkung dauerhaft auftreten.

## 17.4 Minimale zukünftige Distanz (MD)

Die vertikale Distanz garantiert, wie schon beispielhaft gezeigt wurde, einen Fehlerverlauf innerhalb des Trichters. Doch für jeden monoton fallenden Trichterrand ist offensichtlich, dass es eine zukünftige (minimale) Distanz  $d_F(t, e(t)) \leq d_V(t, e(t))$  zwischen aktuellem Absolutbetrag des Fehlers  $|e(t)|$  und einem zukünftigen Punkt des Trichterrandes  $\partial\mathcal{F}_\varphi(t_F)$  für  $t_F \geq t$  geben muss (siehe Abb. 17.2 und [104]). Bei der Auswertung der zukünftigen Distanz wird der bereits bekannte — weil vorgegebene — Verlauf des Trichterrandes sinnvoll genutzt. Der Regler schaut ‘quasi in die Zukunft’ und beachtet den zukünftigen Wert des Trichterrandes  $\partial\mathcal{F}(t_F)$  bei der Verstärkungsanpassung. Es kommt zu einem beschleunigten Systemverhalten des geschlossenen Regelkreises, da deutlich größere Stellgrößen mit  $k_F(t, e(t)) \geq k_V(t, e(t))$  für fast alle  $t \geq 0$  generiert werden.

### Proposition 17.1

*Minimale (zukünftige) Distanz (MD) [71, 76]*

Für jede stetig-differenzierbare Trichterrandfunktion  $\partial\mathcal{F}_\varphi(\cdot)$  mit  $-\infty < \partial\dot{\mathcal{F}}_\varphi(t) \leq \partial\dot{\mathcal{F}}_\varphi(\tau) \leq 0$  für alle  $\tau \geq t \in \mathbb{R}_{\geq 0}$  und  $\lim_{t \rightarrow \infty} \partial\dot{\mathcal{F}}_\varphi(t) = 0$  existiert eine minimale (zukünftige) Distanz

$$d_F(t, e(t)) = \min_{t_F \geq t} \sqrt{(\partial\mathcal{F}_\varphi(t_F) - |e(t)|)^2 + (t_F - t)^2} \quad (17.101)$$

mit der Eigenschaft, dass

$$\forall t \in \mathbb{R}_{\geq 0} : \quad d_F(t, e(t)) \leq d_V(t, e(t)) \quad \wedge \quad t \leq t_F < t + d_V(t, e(t)) \quad (17.102)$$

gilt, wobei  $d_V(t, e(t))$  wie in (17.3) ausgewertet wird. Falls  $t_H > t$  existiert mit  $\partial\mathcal{F}_\varphi(t_H) = |e(t)|$ , gilt zusätzlich

$$\begin{aligned} \forall t \in \mathbb{R}_{\geq 0} : \quad d_F(t, e(t)) &\leq \min \{d_V(t, e(t)), (t_H - t)\} \\ \wedge \quad t < t_F < \min \{t_H, t + d_V(t, e(t))\} \end{aligned} \quad (17.103)$$

*Beweis.* Die kürzeste Strecke zwischen zwei Punkten  $(t, x_1(t))$  und  $(\tau, x_2(\tau))$  in der Ebene  $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$  ist eine Gerade. Diese hat die Länge

$$R(t, \tau) = \sqrt{(x_2(\tau) - x_1(t))^2 + (\tau - t)^2} \geq 0.$$

Nun setze  $x_1(t) = |e(t)|$ , zeichne einen Kreis  $\mathcal{B}_{R(t, \tau)}(|e(t)|)$  um  $|e(t)|$  mit Radius  $R(t, \tau)$  und vergrößere den Radius  $R(t, \tau)$  bis die Kreislinie  $\bar{\mathcal{B}}_{R(t, \tau)}(|e(t)|)$  zum ersten Mal den Trichterrand an mindestens einer Stelle  $\partial\mathcal{F}_\varphi(\tau)$  mit  $\tau \geq t$  berührt. Solche Berührpunkte existieren, da  $\partial\dot{\mathcal{F}}_\varphi(t) \leq 0$  für alle  $t \geq 0$ . Es gilt dann  $R(t, \tau) = \sqrt{(\partial\mathcal{F}_\varphi(\tau) - |e(t)|)^2 + (\tau - t)^2}$  für alle  $\tau \geq t$ .

Nun wähle die Funktion  $x_2 : [t, t + d_V(t, e(t))] \rightarrow \mathbb{R}_{\geq 0}$ , so dass

$$\forall \tau \in [t, t + d_V(t, e(t))] : (x_2(\tau) - |e(t)|)^2 + (\tau - t)^2 = d_V(t, e(t))^2 \quad (17.104)$$

Dann ergeben sich die beiden Fälle

- (a)  $\forall \tau \in [t, t + d_V(t, e(t))] : x_2(\tau) \leq \partial\mathcal{F}_\varphi(\tau) \Rightarrow t_F = t \wedge d_F(t, e(t)) = d_V(t, e(t))$
- (b)  $\exists \tau \in (t, t + d_V(t, e(t))) : x_2(\tau) > \partial\mathcal{F}_\varphi(\tau) \Rightarrow \tau > t$  und

$$\begin{aligned} d_V(t, e(t))^2 &= (x_2(\tau) - |e(t)|)^2 + (\tau - t)^2 \geq (\partial\mathcal{F}_\varphi(\tau) - |e(t)|)^2 + (\tau - t)^2 \\ &= R(t, \tau)^2 \end{aligned}$$

Für den Fall (b) lässt sich ein minimales  $t_F \in (t, t + d_V(t, e(t)))$  finden für das  $R(t, \tau) \geq R(t, t_F) =: d_F(t, e(t))$  gilt. Aus den Fällen (a) und (b) folgt (17.102). Falls es Zeitpunkte  $t \geq 0$  mit  $|e(t)| \geq \lim_{s \rightarrow \infty} \partial\mathcal{F}_\varphi(s) > 0$  gibt, existiert

$$t_H := \min_{\tau > t} \{\tau \in \mathbb{R}_{\geq 0} \mid \partial\mathcal{F}_\varphi(\tau) = |e(t)|\}.$$

Der Fall  $d_V(t, e(t)) \leq t_H - t$  ist trivial und folgt aus obiger Betrachtung. Daher sei nun  $d_V(t, e(t)) > t_H - t$ . Man wählt wieder  $x_2 : [t, t_H] \rightarrow \mathbb{R}_{\geq 0}$ , so dass

$$\forall \tau \in [t, t_H] : (x_2(\tau) - |e(t)|)^2 + (\tau - t)^2 = (t_H - t)^2. \quad (17.105)$$

Aufgrund von  $-\infty < \partial\dot{\mathcal{F}}_\varphi(t) \leq \partial\dot{\mathcal{F}}_\varphi(\tau) \leq 0$  für alle  $\tau \geq t \in \mathbb{R}_{\geq 0}$ ,  $\lim_{t \rightarrow \infty} \partial\dot{\mathcal{F}}_\varphi(t) = 0$  und  $d_V(t, e(t)) > t_H - t$ ,

$$\exists \hat{t} \in (t, t_H) : \quad x_2(\hat{t}) = \partial \mathcal{F}_\varphi(\hat{t}).$$

Somit gilt

$$\forall \tau \in (\hat{t}, t_H) : \quad x_2(\tau) > \partial \mathcal{F}_\varphi(t)$$

und

$$\begin{aligned} d_V(t, e(t))^2 &> (t_H - t)^2 = (x_2(\tau) - |e(t)|)^2 + (\tau - t)^2 \\ &> (\partial \mathcal{F}_\varphi(\tau) - |e(t)|)^2 + (\tau - t)^2 = R(t, \tau)^2 \end{aligned}$$

Durch die Wahl eines minimalen  $t_F \in (\hat{t}, t_H)$  bei dem  $R(t, \tau) \geq R(t, t_F) = d_F(t, e(t))$  zeigt man (17.103).

**Korollar 17.4.1** Für jede Skalierungsfunktion  $\varsigma(\cdot) > 0$  gilt:

$$\forall t \geq 0 : \quad k_F(t, e(t)) = \frac{\varsigma(t)}{d_F(t, e(t))} \geq \frac{\varsigma(t)}{d_V(t, e(t))} = k_V(t, e(t)).$$

*Beweis.* Mit (17.101) wird die minimale (zukünftige) Distanz gewählt, die aufgrund Proposition 17.1 existiert. Somit folgt direkt aus (17.5), dass

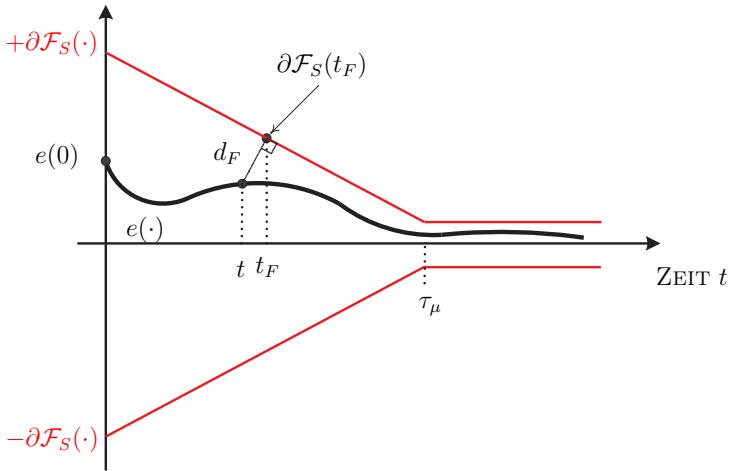
$$\forall t \geq 0 : \quad k_F(t) = \frac{\varsigma(t)}{d_F(t, e(t))} \geq \frac{\varsigma(t)}{d_V(t, e(t))} = k_V(t, e(t)).$$

Mit Proposition 17.1 und Korollar 17.4.1 ergibt sich insbesondere für die anfängliche Stellgröße

$$u_F(0) = \underbrace{\frac{\varsigma(0)}{d_F(0, e(0))}}_{=k_F(0, e(0))} e(0) \geq u_V(0) = \underbrace{\frac{\varsigma(0)}{d_V(0, e(0))}}_{=k_V(0, e(0))} e(0) \quad (17.106)$$

Der Regelkreis wird also bei Auswertung der zukünftigen Distanz insbesondere zu Beginn stärker in Richtung der Referenz  $y^*$  beschleunigt als bei Verwendung der vertikalen Distanz. Das Regelgesetz (17.1) ist generell effektiver und bedingt eine beschleunigte Systemantwort des Regelkreises.

Die Schwierigkeit verbirgt sich im Auffinden des zukünftigen Zeitpunktes  $t_F \geq t$ , an dem die minimale Distanz  $d_F(t, e(t))$  auftritt. Für einfache Trichterränder — z.B. für den ‘Simplen Trichterrand’  $\partial \mathcal{F}_S(t)$  — lässt sich die minimale (zukünftige) Distanz noch analytisch herleiten und auswerten (siehe Abschnitt 17.4.1). Jedoch kann dies nicht im Allgemeinen angenommen werden. Daher werden in den Abschnitten 17.4.2 und 17.4.3 numerische und differenzierende Verfahren zum Auffinden der minimalen Distanz vorgestellt.



**Abb. 17.14:** Analytische Methode zur Berechnung der minimalen (zukünftigen) Distanz

#### 17.4.1 Analytischer Ansatz (aMD)

Für besonders einfache Trichterränder kann der zukünftige Zeitpunkt  $t_F$  analytisch berechnet und somit die minimale Distanz ausgewertet werden. Die Grundidee ist Folgende: Zur Berechnung von  $t_F$  muss eine verbindende Gerade zwischen aktuellem Absolutwert des Fehlers  $|e(t)|$  und Trichterrand  $\partial\mathcal{F}_S(t_F)$  gefunden werden, die auf der Tangente (Ableitung des Trichters im Punkt  $t_F$ ) senkrecht steht. Allgemein lässt sich die Bedingung für aufeinander senkrecht stehende Geraden durch

$$\frac{\partial\mathcal{F}_\varphi(t_F) - |e(t)|}{t_F - t} \cdot \frac{d}{dt}\partial\mathcal{F}_\varphi(t)\Big|_{t=t_F} = -1 \quad (17.107)$$

formulieren. Damit lässt sich explizit ein  $t_F \geq t$  berechnen, welches direkt die (analytische) Auswertung von (17.101) ermöglicht [71, 76].

Der Beispieltrichterrand  $\partial\mathcal{F}_S$  lässt diese Berechnung und Auswertung der minimalen Distanz zu, d.h. hier lässt sich auf analytischem Wege der zukünftige Zeitpunkt  $t_F \geq t$  bestimmen. Hierzu nutzt man (17.107), ersetzt  $\partial\mathcal{F}_\varphi(t)$  durch (17.42) und löst die Gleichung entsprechend nach  $t_F$  auf. Man erhält

$$\tilde{t}_F = \frac{\varphi_{S,\infty}\tau_\mu(\varphi_{S,\infty} - \varphi_{S,0})(1 - \varphi_{S,0}|e(t)|) + (\varphi_{S,\infty}\varphi_{S,0}\tau_\mu)^2 t}{(\varphi_{S,\infty} - \varphi_{S,0})^2 + (\varphi_{S,\infty}\varphi_{S,0}\tau_\mu)^2} \quad (17.108)$$

für alle  $t \geq 0$ . Die Fallunterscheidung

$$t_F = \begin{cases} \tilde{t}_F & , \tilde{t}_F \leq \tau_\mu \wedge t \leq \tau_\mu \\ \tau_\mu & , \tilde{t}_F \geq \tau_\mu \wedge t \leq \tau_\mu \\ t & , t > \tau_\mu \end{cases} \quad (17.109)$$

berücksichtigt den Knick bei  $\tau_\mu$  und gibt jeweils den korrekten zukünftigen Zeitpunkt  $t_F \geq t$  an. Der gewonnene Zeitpunkt  $t_F$  aus (17.109) ermöglicht die analytischen Berechnung der minimalen zukünftigen Distanz

$$d_F(t, e(t)) = \sqrt{(\partial\mathcal{F}_S(t_F) - |e(t)|)^2 + (t_F - t)^2} \quad (17.110)$$

zwischen ‘Simplen Trichterrand’  $\partial\mathcal{F}_S(t_F)$  und aktuellem Fehlerbetrag  $|e(t)|$ . Für alle Zeitpunkte  $t > \tau_\mu$  gilt  $t_F = t$  und daher geht für (17.42) die minimale (zukünftige) Distanz  $d_F(t, e(t))$  in die vertikale Distanz  $d_V(t, e(t))$  über.

Die analytische Methode erlaubt exakte und direkte Berechnung der minimalen Distanz. Es muss kein iteratives oder rekursives Verfahren implementiert werden. Aber es liegt auf der Hand, dass Bedingung (17.107) nicht immer auf ein analytisch lösbares Gleichungssystem führt. Abhängig von der Wahl des Trichterrandes müssen Wurzeln von Polynomen höherer Ordnung (z.B.  $> 4$ ) oder von transzendenten Gleichungen gefunden werden, was eine analytische Auswertung von (17.101) unmöglich macht. Da aber für beliebige Trichterränder die Verwendung der minimalen (zukünftigen) Distanz wünschenswert ist, soll im folgenden Abschnitt ein numerischer Ansatz vorgestellt werden, der iterativ die minimale (zukünftige) Distanz approximiert.

### 17.4.2 Numerischer Ansatz (nMD)

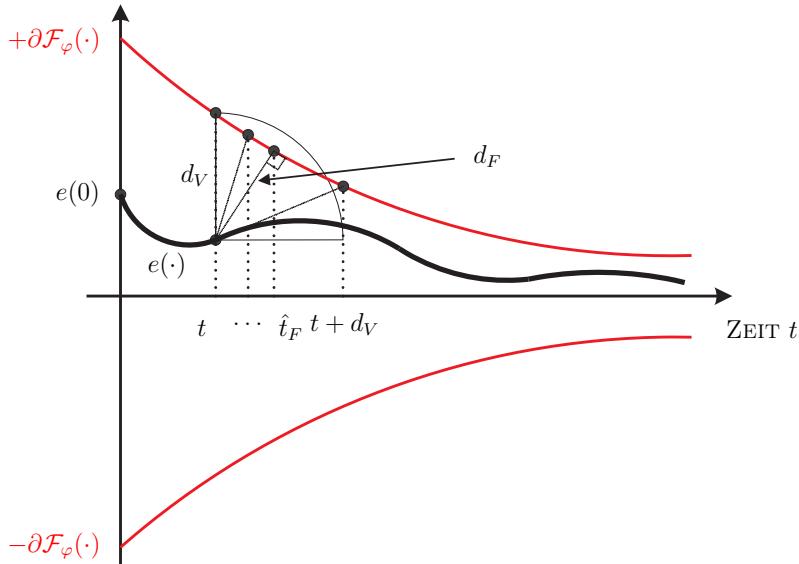
Da das Minimum in (17.101) abhängig von der Wahl des Trichters eventuell schwer analytisch zu finden ist, kann bei der Implementierung auf eine numerische Approximation (Suche) zurückgegriffen werden (ebenfalls vorgestellt in [104, 71, 76]). Ausgehend von der vertikalen Distanz (17.3) kann allgemein davon ausgegangen werden, dass die zukünftige (minimale) Distanz innerhalb des Zeithorizontes  $t \leq t_F \leq d_V(t, e(t))$  auftreten muss. Daher reicht es aus, das Zeintervall  $I := [t, d_V(t, e(t))]$  für jeden Zeitpunkt  $t \geq 0$  entsprechend nach der zukünftigen Zeit  $t_F \geq t$  abzusuchen, um die minimale zukünftige Distanz zu finden. Dazu bietet es sich an, dieses Intervall in  $N \in \mathbb{N}$  (nicht notwendigerweise äquidistante) Teilstücke zu partitionieren, hierzu wählt man beliebige

$$0 = h_1 < h_2 < \dots < h_N \leq 1. \quad (17.111)$$

Dann lässt sich die zukünftige (minimale) Distanz durch

$$d_{F,N}(t, e(t)) := \min_{n \in 1, \dots, N} \sqrt{(h_n d_V(t, e(t)))^2 + (\partial\mathcal{F}_\varphi(t + h_n d_V(t, e(t)) - |e(t)|)^2)} \quad (17.112)$$

numerisch approximieren. Man nähert sich pro Iterationsschritt dem wirklichen Zeitpunkt  $t_F$  mit der Schrittweite  $h_i d_V(t, e(t))$ . Es muss im Allgemeinen davon ausgegangen werden, dass der iterativ erreichte Zeitpunkt  $\hat{t}_F \approx t_F$  sich vom wirklichen unterscheidet. In Abb. 17.16 ist der Programmablaufplan der numerischen (iterativen) Methode dargestellt. Der Algorithmus kann mit einer einfach



**Abb. 17.15:** Numerische Methode zur Approximation der zukünftigen (minimalen) Distanz

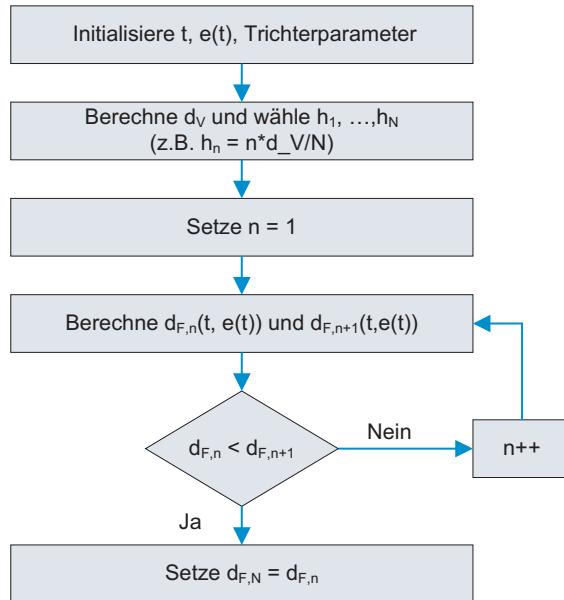
FOR-Schleife implementiert werden und wird maximal  $N$  Mal durchlaufen. Die numerische Methode lässt große Freiheit bei der Wahl des Trichterrandes zu. Die minimale Distanz kann abhängig von der Partitionierung (17.111) sehr genau approximiert werden. Ein Nachteil dieses iterativen Verfahrens ist die eventuelle Gefährdung der Realzeitfähigkeit: für große  $N$  (hohe Genauigkeit) muss in Abhängigkeit der wirklichen Lage des Zeitpunktes  $t_F$  die Iterationsschleife sehr häufig durchlaufen werden.

#### 17.4.3 Differenzierender Ansatz (dMD)

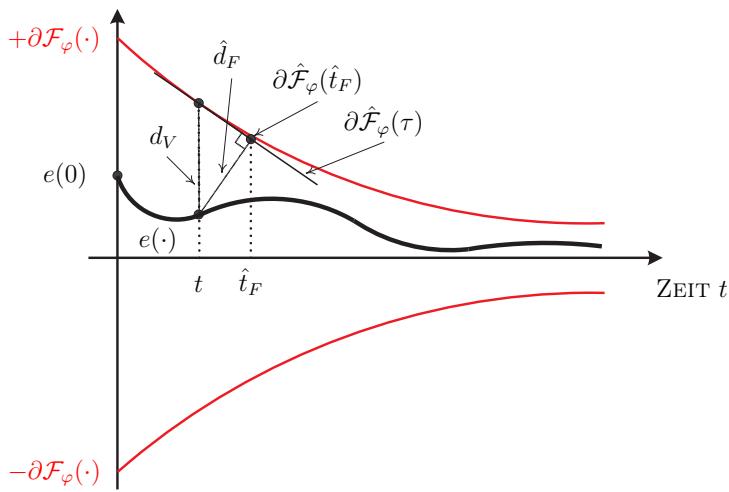
Um eine eventuelle Gefährdung der Realzeitfähigkeit von Beginn an auszuschließen, soll abschließend ein letztes Verfahren vorgestellt werden. Der differenzierende Ansatz nutzt die Eigenschaften von stetig-differenzierbaren und monoton fallenden Trichterrändern in der Art aus, dass der Trichterrand  $\partial\mathcal{F}_\varphi(t)$  um  $t \geq 0$  durch eine Gerade (Tangente)

$$\forall \tau \geq t : \quad \partial\hat{\mathcal{F}}_\varphi(\tau) := \dot{\partial\mathcal{F}}_\varphi(t)(\tau - t) + \partial\mathcal{F}_\varphi(t) \quad (17.113)$$

approximiert wird. Mithilfe von (17.107) und (17.113) kann die approximierte minimale (zukünftige) Distanz analytisch berechnet werden. Durch die expliziten Rechenvorschriften ist ein *nicht-iterativer* Programmablauf gesichert. Hierbei



**Abb. 17.16:** Programmablaufplan der numerischen Methode (Iteratives Verfahren)



**Abb. 17.17:** Differenzierender Ansatz zur Approximation der minimalen (zukünftigen) Distanz

zeigt sich dass die analytische Näherung der zukünftigen Distanz  $\hat{d}_F(t, e(t)) \leq d_F(t, e(t))$  zu noch schnellerem transienten Verhalten führen kann. Die Idee des differenzierenden Ansatzes ist in Abb. 17.17 dargestellt. Die Eigenschaften dieses Ansatzes sind zusammengefasst in

### Proposition 17.2

*Differenzierend-approximierte minimale Distanz (dMD) [71, 76]*

Für jede stetig-differenzierbare Trichterrandfunktion  $\partial\mathcal{F}_\varphi(\cdot)$  mit  $-\infty < \partial\dot{\mathcal{F}}_\varphi(t) \leq \partial\dot{\mathcal{F}}_\varphi(\tau) \leq 0$  für alle  $\tau \geq t \in \mathbb{R}_{\geq 0}$  und  $\lim_{t \rightarrow \infty} \partial\dot{\mathcal{F}}_\varphi(t) = 0$  existiert die tangentiale Approximation (17.113) des Trichters  $\partial\mathcal{F}_\varphi(\cdot)$ . Mit (17.113) und (17.107) kann der approximierte zukünftige Zeitpunkt

$$\hat{t}_F = \frac{t - \partial\dot{\mathcal{F}}_\varphi(t) \cdot (d_V(t, e(t)) - \partial\dot{\mathcal{F}}_\varphi(t) \cdot t)}{1 + \partial\dot{\mathcal{F}}_\varphi^2(t)} = t - \frac{\partial\dot{\mathcal{F}}_\varphi(t) d_V(t, e(t))}{1 + \partial\dot{\mathcal{F}}_\varphi^2(t)} \quad (17.114)$$

analytisch berechnet werden. Dann lässt sich mit

$$\hat{d}_F(t, e(t)) = \sqrt{(\partial\dot{\mathcal{F}}_\varphi(\hat{t}_F) - |e(t)|)^2 + (\hat{t}_F - t)^2} \quad (17.115)$$

die differenzierend-approximierte minimale (zukünftige) Distanz auswerten. Es gilt  $t \leq \hat{t}_F \leq t + d_V(t, e(t))$ . Die approximierte Distanz hat folgende Eigenschaft

$$\forall t \in \mathbb{R}_{\geq 0} : \quad \hat{d}_F(t, e(t)) \leq d_F(t, e(t)) \leq d_V(t, e(t)) \quad (17.116)$$

*Beweis.* Da  $\partial\mathcal{F}_\varphi(\cdot)$  differenzierbar ist, existiert  $\partial\dot{\mathcal{F}}_\varphi(\tau)$  in (17.113) für alle  $\tau \geq t \geq 0$ . Sie repräsentiert eine Gerade mit Steigung  $\partial\dot{\mathcal{F}}_\varphi(t)$  und da  $\partial\dot{\mathcal{F}}_\varphi(\tau) \geq \partial\dot{\mathcal{F}}_\varphi(t)$  gilt  $\partial\dot{\mathcal{F}}_\varphi(\tau) \leq \partial\mathcal{F}_\varphi(\tau)$  für alle  $\tau \geq t$ . Für die Senkrechte auf diese Tangente muss (17.107) gelten. Durch Auflösen nach  $\hat{t}_F$  erhält man direkt (17.114). Mit bekanntem  $\hat{t}_F$  kann (17.115) ausgewertet werden. Da  $\partial\dot{\mathcal{F}}_\varphi(t) \leq 0$  für alle  $t \geq 0$ , folgt aus (17.114) dass  $\hat{t}_F \geq t$ . Proposition 17.1 gibt die Existenz einer minimalen Distanz  $d_F(t, e(t)) \leq d_V(t, e(t))$ . Nun wähle die Funktion  $x_2 : [t, t + d_V(t, e(t))] \rightarrow \mathbb{R}_{\geq 0}$ , so dass

$$\forall \tau \in [t, t + d_V(t, e(t))] : (x_2(\tau) - |e(t)|)^2 + (\tau - t)^2 = \hat{d}_F(t, e(t))^2 \quad (17.117)$$

Der Verlauf von  $x_2(\cdot)$  beschreibt einen Viertelkreis, der im Punkt  $(\hat{t}_F, \partial\dot{\mathcal{F}}(\hat{t}_F))$  von der Tangente (17.113) berührt wird, d.h. für alle  $\tau \in [t, t + d_V(t, e(t))]$  mit  $\hat{t}_F \neq \tau$  gilt  $x_2(\tau) < \partial\dot{\mathcal{F}}(\tau)$ . Für  $t_F \in [t, t + d_V(t, e(t))]$  lässt sich nun ein  $x_2(t_F)$  finden, so dass  $x_2(t_F) \leq \partial\dot{\mathcal{F}}(\hat{t}_F) < \partial\mathcal{F}_\varphi(t_F)$  und damit

$$\begin{aligned} \hat{d}_F(t, e(t))^2 &= (x_2(t_F) - |e(t)|)^2 + (t_F - t)^2 \\ &< (\partial\mathcal{F}_\varphi(t_F) - |e(t)|)^2 + (t_F - t)^2 = d_F(t, e(t))^2 \end{aligned} \quad (17.118)$$

**Korollar 17.4.2** Für jede Skalierungsfunktion  $\varsigma(\cdot) > 0$  gilt:

$$\forall t \geq 0 : \quad k_{\hat{F}}(t, e(t)) := \frac{\varsigma(t)}{\hat{d}_F(t, e(t))} \geq k_F(t, e(t)).$$

*Beweis.* Mit (17.115) erhält man die differenzierend-approximierte minimale zukünftige Distanz  $\hat{d}_F(t, e(t))$ . Dann folgt mit (17.116) direkt

$$k_{\hat{F}}(t, e(t)) = \frac{\varsigma(t)}{\hat{d}_F(t, e(t))} \geq \frac{\varsigma(t)}{d_F(t, e(t))} = k_F(t, e(t)).$$

Zusammenfassend kann festgehalten werden, dass der differenzierende Ansatz mit der approximierten Distanz  $\hat{d}_F(t, e(t))$  zu der aggressivsten Reglerverstärkung  $k_{\hat{F}}(t, e(t)) \geq k_F(t, e(t)) \geq k_V(t, e(t))$  führt [71, 76].

#### 17.4.4 Simulationsbeispiele

Es sollen nun die Auswirkungen von

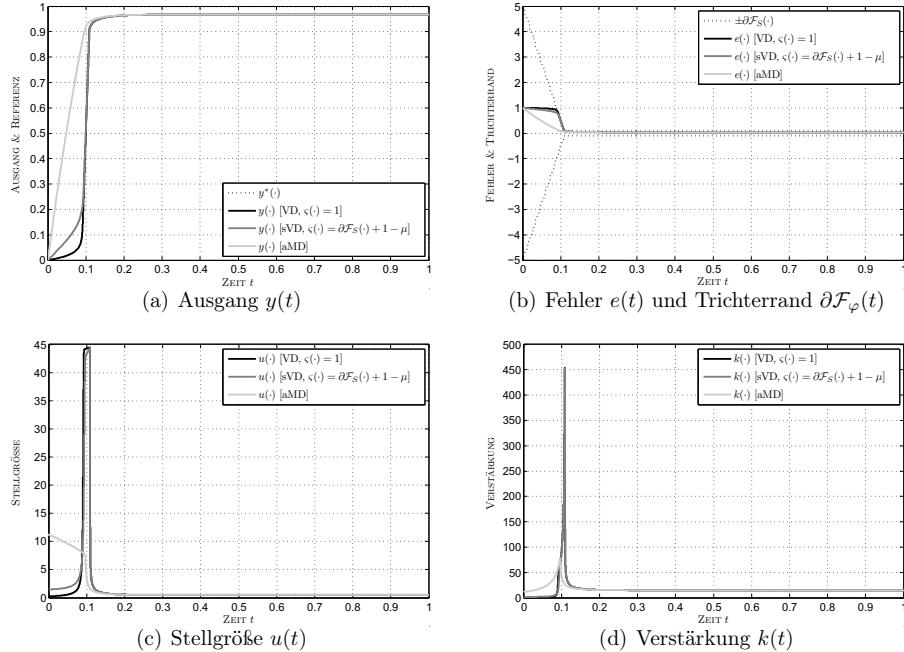
- einer Skalierung der Reglerverstärkung (z.B. mit  $\varsigma(t) = 1$  und  $\varsigma(t) = \partial\mathcal{F}_\varphi(t) + 1 - \mu$ )
- und unterschiedlicher Distanzauswertungen (z.B. vertikale Distanz (17.3), analytische (17.110), numerische (17.112) und differenzierende (17.115) minimale Distanz)

auf die Reglerperformanz des geschlossenen Regelkreises aus Funnel Regler (FC) und Beispielsystem  $S_1$  (siehe S. 717 und Tab. 17.1) veranschaulicht werden. Hierzu werden die Regelkreise entsprechend Abb. 17.9 in Matlab/Simulink für oben genannte Skalierungen und Distanzauswertungen implementiert. Die Simulationsergebnisse in Abbildungen 17.18 und 17.19 erlauben einen qualitativen Vergleich der Reglerperformanz. Die exakten Simulationsparameter sind in Tab. 17.5 zusammengefasst. Die analytische minimale Distanz (aMD) wird für den ‘Simplen Trichterrand’  $\partial\mathcal{F}_S$  entsprechend (17.110) ausgewertet. Hingegen werden die numerische (nMD) und differenzierende (dMD) minimale Distanz für den Exponentiellen Trichterrand  $\partial\mathcal{F}_E$  entsprechend (17.112) und (17.115) umgesetzt.

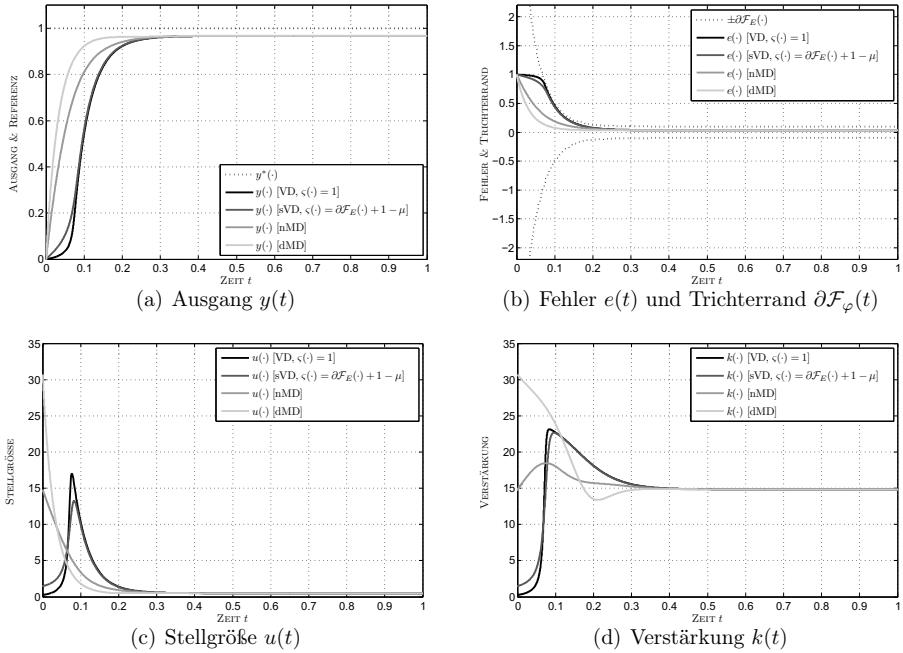
Abbildungen 17.18a,b und 17.19a,b zeigen, dass alle Regelkreise dem Einheits-sollsprung innerhalb der erlaubten Trichtergrenzen folgen können. Offensichtlich ist eine asymptotisch genaue Festwertregelung aufgrund des proportionalen Verhaltens der Strecke  $S_1$  nicht möglich. Ein effektiverer Stellgrößen- und Verstärkungsverlauf führt zu einem beschleunigten Einschwingen der Regelkreise mit

	VD	sVD	aMD	nMD	dMD
Kundenanforderungen		$(\tau_\nu, \nu, \mu) = (0.1s, 0.5, 0.1)$			
Trichteranfangswert		$\partial\mathcal{F}_S(0) = \partial\mathcal{F}_E(0) = 5$			
Skalierung $\varsigma(t)$	1	$\partial\mathcal{F}_\varphi(t) + \mu - 1$	1	1	1
max. Iterationszahl $N$	—	—	—	10000	—
Anfangswert $y(0)$	0	0	0	0	0
Referenz $y^*(t)$	$\sigma(t)$	$\sigma(t)$	$\sigma(t)$	$\sigma(t)$	$\sigma(t)$

**Tabelle 17.5:** Simulationsdaten für Vergleich der vertikalen (VD), skaliert-vertikalen (sVD), analytischen (aMD), numerischen (nMD) und differenzierenden minimalen Distanz (nMD) am Beispielsystem S1 (PT1)



**Abb. 17.18:** Simulativer Vergleich der Reglerperformanz von unterschiedlichen Distanz-Auswertungen und Skalierung bei ‘Simplen Trichterrand’  $\partial\mathcal{F}_S$ : vertikale (VD), skaliert-vertikale (sVD) und analytische minimale Distanz (aMD)

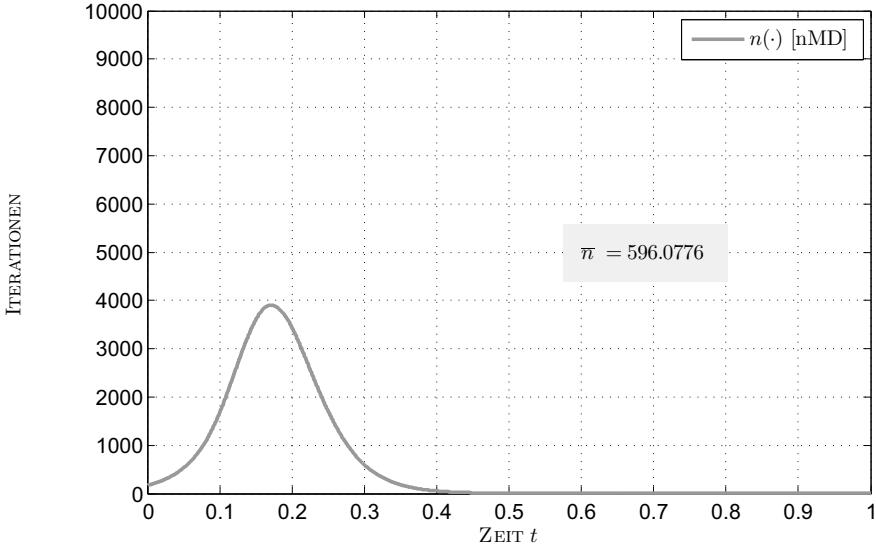


**Abb. 17.19:** Simulativer Vergleich der Reglerperformanz von unterschiedlichen Distanz-Auswertungen und Skalierung bei Exponentiellem Trichterrand  $\partial\mathcal{F}_E$ : vertikale (VD), skaliert-vertikale (sVD), numerischer (nMD) und differenzierender (dMD) minimaler Distanz

Auswertung der minimalen (zukünftigen) Distanz aMD, nMD und dMD (siehe Abb. 17.18 und 17.19 c,d). Insbesondere zu Beginn führen die minimalen Distanzauswertungen für höhere Verstärkungen und Stellgrößen, hierdurch können die steilen Peaks in Verstärkung und Stellgröße bei VD und sVD vermieden werden. Die Skalierung der vertikalen Distanz (sVD) führt zu einem leicht verbesserten (beschleunigten) Transienten (siehe 17.18 und 17.19a,b). In Abb. 17.20 sind die benötigten Iterationen des numerischen Ansatzes (nMD) pro Zeitschritt gezeigt. Im Mittel werden für das simulierte Beispiel knapp 600 Iterationen pro Zeitschritt bei  $N = 10000$  (hier: hohe Genauigkeit gewünscht) benötigt.

## 17.5 Error Reference Control (ERC)

Error Reference Control (ERC) [69, 70, 78] ist direkt von Funnel-Control abgeleitet. Entsprechend deckt der Beweis in [103] die folgenden Ausführungen bereits ab. Es muss kein gesonderter mathematischer Beweis geführt werden. Lediglich



**Abb. 17.20:** Anzahl der Iterationen  $n$  und Mittelwert  $\bar{n}$  des numerischen Ansatzes [ $nMD$ ]

die Vorstellung über den Trichterrand muss angepasst werden (siehe Abb. 17.21). Dieser geht in eine Art Schlauch um die Referenz  $y^*(\cdot)$  über, und kann als asymmetrisch ausgelegter Trichterrand angesehen werden.

Aufgrund dieser speziellen asymmetrischen Auslegung des Trichterrandes (im Folgenden mit ‘virtueller Schlauch’ bezeichnet) und einer ‘erweiterten’ Referenz

$$\forall t \geq 0 : \quad y_{ERC}^*(t) := y^*(t) - e^*(t) \quad (17.119)$$

mit  $y^*(\cdot), e^*(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  erlaubt ERC nicht nur *a priori* festgelegtes transientes Verhalten innerhalb des virtuellen Schlauchs (siehe Abb. 17.21), sondern zusätzlich einen Fehlerverlauf  $e(\cdot)$  nahe eines *a priori* festgelegten ‘Wunschfehlerverlaufs’  $e^*(\cdot)$  (siehe Abb. 17.21). Der Regelkreis entspricht prinzipiell dem in Abb. 17.1. Es wird nun lediglich der Funnel Regler (FC) durch den Error Reference Regler (ERC) ersetzt. Es kann die gleiche Systemklasse  $\mathcal{S}$  beherrscht und eine Folgewertregelung innerhalb des virtuellen Schlauchs

$$\mathbb{E}_{(e_-, e_+)} := \{(t, e) \in \mathbb{R}_{\geq 0} \times \mathbb{R} \mid e_-(t) < e < e^+(t)\} \quad (17.120)$$

garantiert werden, sofern der Anfangsfehler  $e(0)$  innerhalb  $\mathbb{E}_{(e_-, e_+)}$  startet. In Abb. 17.21 ist ein möglicher Fehlerverlauf  $e(\cdot)$  innerhalb eines Beispialschlauchs dargestellt. Der Fehlerverlauf  $e(\cdot)$  ist durch den virtuellen Schlauch noch stärker eingeschränkt (siehe Abb. 17.21) und entwickelt sich nahe des festgelegten Wunschfehlerverlaufs — der Fehlerreferenz  $e^*(\cdot) \in \mathcal{C}^1(\mathbb{R}_{\geq 0}; \mathbb{R})$  — (siehe

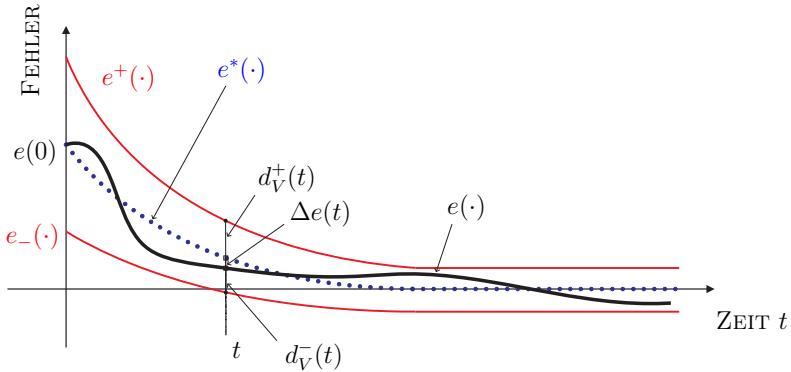


Abb. 17.21: Grundlegende Idee von *Error Reference Control (ERC)*

Abb. 17.21 und 17.22). Diese ‘Fehlerreferenz’ kann mithilfe des Anfangsfehlers  $e(0) = y^*(0) - y(0)$  festgelegt werden, im Folgenden wählen wir vereinfachend

$$\forall t \geq 0 : \quad e^*(t) = \underbrace{[y^*(0) - y(0)]}_{=e(0)} \cdot \exp\left(-\frac{t}{T_{ERC}}\right) \quad (17.121)$$

wobei mit  $T_{ERC} > 0$  das exponentielle Abklingen beliebig schnell vorgegeben werden kann. Der Wunschfehlerverlauf  $e^*(\cdot)$  konvergiert zu Null, d.h.  $\lim_{t \rightarrow \infty} e^*(t) = 0$ .

Zur Beschreibung des virtuellen Schlauchs (17.120), legt man für  $\mu^+ > 0$  und  $\kappa^+(\cdot) \in \mathcal{C}(\mathbb{R}_{\geq 0}; \mathbb{R}_{\geq 0})$  mit  $\lim_{t \rightarrow \infty} \kappa^+(t) = 0$  den oberen Schlauchrand

$$e^+(t) = e^*(t) + \mu^+ (1 + \kappa^+(t)) =: \partial \mathcal{F}_\varphi^+(t) = \frac{1}{\varphi^+(t)} \quad (17.122)$$

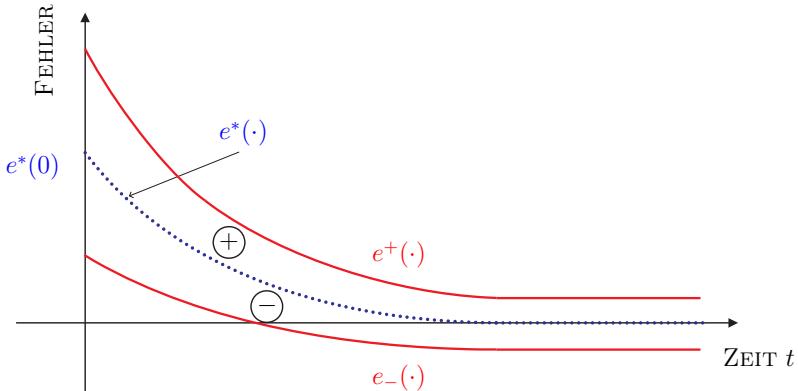
und für  $\mu_- > 0$  und  $\kappa_-(\cdot) \in \mathcal{C}(\mathbb{R}_{\geq 0}; \mathbb{R}_{\geq 0})$  mit  $\lim_{t \rightarrow \infty} \kappa_-(t) = 0$  den untere Schlauchrand

$$e_-(t) = e^*(t) - \mu_- (1 + \kappa_-(t)) =: \partial \mathcal{F}_\varphi^-(t) = \frac{1}{\varphi_-(t)} \quad (17.123)$$

fest. Der Schlauch  $\mathbb{E}_{(e_-, e^+)}$  entspricht somit einem speziell asymmetrisch ausgelegten Trichterrand mit oberem  $\partial \mathcal{F}_\varphi^+(\cdot)$  und unterem  $\partial \mathcal{F}_\varphi^-(\cdot)$  Verlauf. Die Festlegung  $e_-(0) < e(0) = e^*(0) < e^+(0)$  garantiert, dass der Anfangsfehler  $e(0)$  vom Schlauch umschlossen wird. Häufig ist es ausreichend  $\mu^+ = \mu_-$  und für alle  $t \geq 0$   $\kappa^+(t) = \kappa_-(t)$  zu wählen (Reduktion der freien Parameter).

Oberer (17.122) und unterer (17.123) Schlauchrand konvergieren jeweils gegen die obere  $\lim_{t \rightarrow \infty} e^+(t) = \mu^+$  und die untere Endgenauigkeit  $\lim_{t \rightarrow \infty} e_-(t) = -\mu_-$ .

Wie kann erreicht werden, dass der Fehler  $e(\cdot)$  der Fehlerreferenz  $e^*(\cdot)$  folgt und innerhalb des Schlauchs verbleibt? Hierzu bestimmt man für jeden Zeitpunkt  $t \geq 0$  den ‘oberen’ Abstand



**Abb. 17.22:** Vorzeichen der Stellgröße bei **Error Reference Control (ERC)**

$$d_V^+(t, e(t)) = e^+(t) - e(t) \quad (17.124)$$

zwischen Regelfehler  $e(t)$  und oberer Schlauchgrenze  $e^+(t)$  bzw. den ‘unteren’ Abstand

$$d_V^-(t, e(t)) = e(t) - e_-(t). \quad (17.125)$$

zwischen Regelfehler  $e(t)$  und unterer Schlauchgrenze  $e_-(t)$ . Man wählt für ERC die zeitvariante Verstärkung

$$k(t, e(t)) = \frac{\varsigma(t)}{\min(d_V^+(t), d_V^-(t))}. \quad (17.126)$$

Die zeitvariante Verstärkung (17.126) — ähnlich zu Funnel Control, siehe Gl. (17.5) — setzt sich zusammen aus der Skalierungsfunktion  $\varsigma(\cdot)$  in (17.96) und des Kehrwertes der Minimumauswahl zwischen  $d_V^+(t, e(t))$  und  $d_V^-(t, e(t))$ .

Die Dämpfungsskalierung  $\varsigma_D(\cdot)$  kann ebenfalls bei ERC sinnvoll eingesetzt werden. Es müssen lediglich der Fehler und dessen Ableitung durch  $\Delta e(t)$  und  $\Delta \dot{e}(t)$  ersetzt werden, man erhält

$$\begin{aligned} \varsigma_{D,ERC}(t) &= \text{sat} \left[ \varsigma_G \Delta e(t) \Delta \dot{e}(t) \right]_0^{\varsigma_D^{max}} + \varsigma_0 \\ &= \text{sat} \left[ \varsigma_G (e(t) - e^*(t)) (\dot{e}(t) - \dot{e}^*(t)) \right]_0^{\varsigma_D^{max}} + \varsigma_0 \end{aligned} \quad (17.127)$$

analog zu (17.99).

Das proportionale zeitvariante Regelgesetz (analog zu (17.1))

$$u(t) = k(t, e(t)) \cdot \underbrace{(e(t) - e^*(t))}_{=: \Delta e(t)} \quad (17.128)$$

$$= k(t, e(t)) \cdot \underbrace{(y^*(t) - e^*(t) - y(t) - n(t))}_{=: y_{ERC}^*(t)} \quad (17.129)$$

Simulationsparameter für FC und ERC		
	FC	ERC
Kundenanforderungen $(\tau_\nu, \nu, \mu)$	$(0.1s, 1, 0.1)$	$(0.1s, 1, 0.1)$
Zeitkonstanten $T_E$ und $T_{ERC}$	$0.0387 s$	$0.0387 s$
Anfangswert $\frac{1}{\varphi_{E,0}}$ und $e^+(0)$	12	12
Skalierungsfunktion $\varsigma(\cdot)$	1	1
Referenz $y^*(t)$	$10\sigma(t)$	$10\sigma(t)$
Anfangsfehler $e(0)$	10	10
Genauigkeit $\mu_- = \mu^+$	—	0.1
Funktion $\kappa_-(\cdot) = \kappa^+(\cdot)$	—	$\left(\frac{1}{\varphi_{E,0}} -  e(0) \right) \exp\left(-\frac{t}{T_E}\right)$

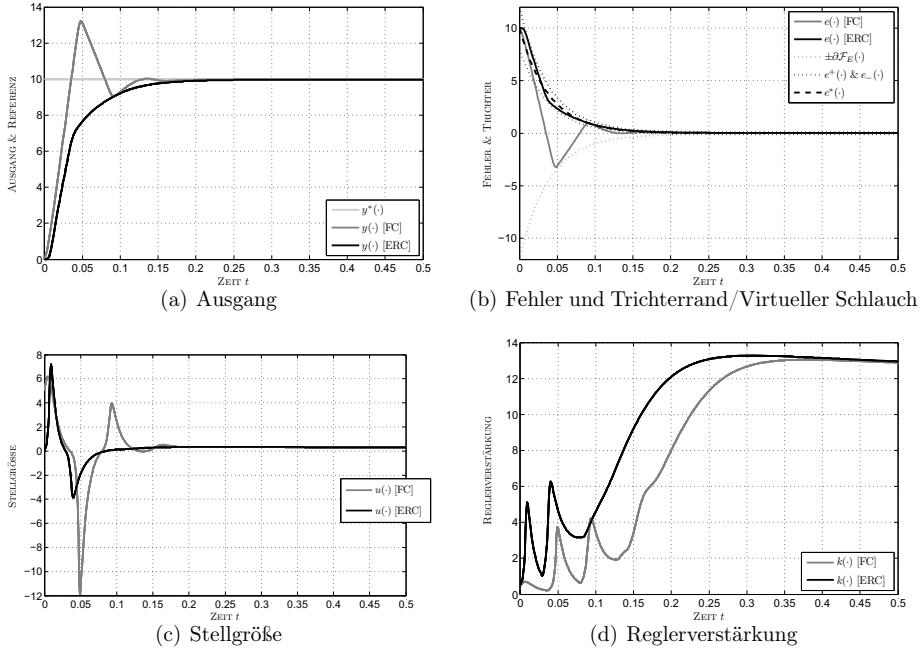
**Tabelle 17.6:** Trichter- und Schlauchentwurf für FC und ERC

garantiert schließlich, dass der Fehler  $e(\cdot)$  innerhalb des virtuellen Schlauchs (17.120) nahe  $e^*(\cdot)$  verläuft. Hierbei entspricht  $y^*_{ERC}(\cdot) := y^*(\cdot) - e^*(\cdot) \in \mathcal{W}^{1,\infty}(\mathbb{R}_{\geq 0}; \mathbb{R})$  dem erweiterten Referenzsignal.

Abb. 17.22 zeigt das Vorzeichen der durch (17.129) generierten Stellgrößen. Abhängig von der Lage des Fehlers  $e(t)$  — ‘oberhalb’ oder ‘unterhalb’ des Wunschfehlerverlaufs  $e^*(t)$  — ändert die Stellgröße  $u(t)$  ihr Vorzeichen und sichert dadurch die nötige *Beschleunigung* oder *Verlangsamung* des Regelkreises. Diese Eigenschaft erlaubt nicht nur einen Fehlerverlauf innerhalb des virtuellen Schlauchs, sondern auch eine Fehlerentwicklung  $e(\cdot)$  möglichst nahe um den festgelegten Wunschfehlerverlauf  $e^*(\cdot)$ .

## Simulation

Es werden die Simulationen für Beispielsystem  $S_2$  (siehe Tab. 17.1) aus Abschnitt 17.3 wiederholt, damit können die Reglerperformanz von FC und ERC miteinander verglichen werden. Die Auslegung des Exponentiellen Trichters und des Virtuellen Schlauchs erfolgt entsprechend der Vorgaben in Tab. 17.6. Beide Regelkreise sollen dem Sollwertsprung  $y^*(t) = 10\sigma(t)$  folgen. Um bestmögliche Vergleichbarkeit der Simulationsergebnisse zu erzielen wird der obere Schlauchrand so ausgelegt, dass für alle  $t \geq 0$  der obere Schlauchrand  $e^+(t) = \partial\mathcal{F}_E(t)$  mit dem Exponentiellen Trichterrand übereinstimmt. Beide zeitvarianten Reglerverstärkungen (17.5) und (17.126) werden *nicht* skaliert. Alle Simulationsparameter sind in Tab. 17.6 aufgelistet. Abb. 17.23 zeigt die Simulationsergebnisse für FC und ERC. Beide zeitvarianten Ansätze erfüllen ihre Regelziele. Es kann der Sollvorgabe innerhalb der erlaubten Grenzen gefolgt werden (siehe Abb. 17.23a,b). Dabei zeigt FC das schon bekannte Überschwingen, wogegen ERC aperiodisches transientes Verhalten erzielt (auch eine Skalierung bei FC würde das Überschwin-



**Abb. 17.23:** Gegenüberstellung der Simulationsergebnisse für Funnel-Control (FC) und Error Reference Control (ERC) am Beispielsystem  $S_2$  (jeweils ohne Skalierung)

gen nur Reduzieren aber nicht beseitigen!). Durch die engere Begrenzung mithilfe des Schlauchs  $\mathbb{E}_{e^+, e_-}$  wird mit ERC ein nahezu identischer Verlauf von Fehler  $e(\cdot)$  und Wunschfehler  $e^*(\cdot)$  erreicht.

## 17.6 Anwendung

Abschließend werden nun einige der vorgestellten Ergebnisse an einem mechatronischen Standardsystem getestet und implementiert.

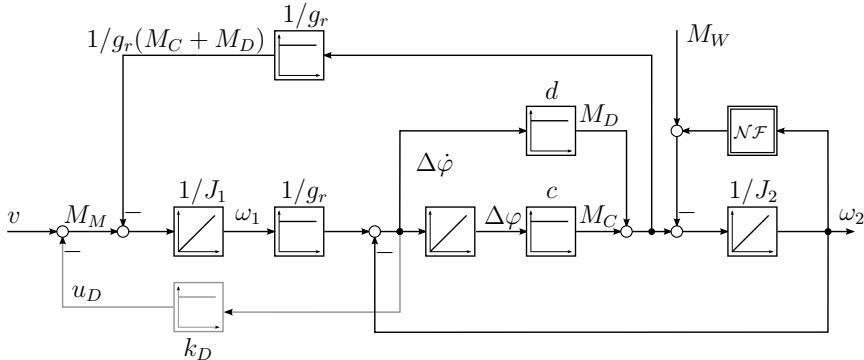
### 17.6.1 Nichtlineares Zwei-Massen-System (2MS)

Für die Anwendung wird wieder das nichtlineare Zwei-Massen-System betrachtet (siehe Abb. 17.24). Hierbei sind zwei drehend-gelagerte Massen  $J_1 > 0$  (Motorträgheitsmoment) und  $J_2 > 0$  [ $kg\ m^2$ ] (Arbeitsmaschinenträgheitsmoment) über eine Welle mit Steifigkeit  $c > 0$  und Dämpfung  $d > 0$  und ein Getriebe mit der Übersetzung  $g_r \in \mathbb{R} \setminus \{0\}$  miteinander mechanisch verbunden. Das schwungsfähige Zwei-Massen-System wird durch das Motormoment  $M_M(t)$  [ $Nm$ ]

beschleunigt oder gebremst, durch das als Störung wirkende arbeitsmaschinenseitige Last- oder Widerstandsmoment  $M_W(t)$  [ $Nm$ ] beansprucht und besitzt die internen Zustände: Winkelgeschwindigkeit  $\omega_1(t)$  [ $\frac{rad}{s}$ ] des Motors, Verdrehwinkel der Welle  $\Delta\varphi(t)$  [ $rad$ ] und Winkelgeschwindigkeit  $\omega_2(t)$  [ $\frac{rad}{s}$ ] der Arbeitsmaschine (Last). Diese lassen sich im Zustandsvektor

$$\underline{x}(t)^\top = (\omega_1(t) \quad \Delta\varphi(t) \quad \omega_2(t)) \quad (17.130)$$

zusammenfassen. Es soll die Lastwinkelgeschwindigkeit als Ausgang  $y(t) = \omega_2(t)$  [ $\frac{rad}{s}$ ] des 2MS geregelt werden.



**Abb. 17.24:** Signalflussplan des aktiven gedämpften Zwei-Massen-Systems (2MS)

Mithilfe des Signalflussplans in Abb. 17.24 kann die Zustandsdarstellung des nichtlinearen Zwei-Massen-Systems abgeleitet werden. Die motorseitig-wirkenden Reibmomente können als auf die Lastseite umgerechnet betrachtet werden und sind somit in der lastseitig-wirkenden nichtlinearen Reibung  $\mathcal{NF}$  berücksichtigt. Man erhält die nichtlineare Differentialgleichung

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\underline{x}(t) + \underline{b}M_M(t) + \underline{b}_L(M_W(t) + (\mathcal{NF}\omega_2)(t)) \quad , \underline{x}(0) = \underline{x}^0 \in \mathbb{R}^n \\ y(t) &= \underline{c}^\top \underline{x}(t) \end{aligned} \quad \left. \right\} \quad (17.131)$$

wobei  $M_M(t) = v(t) - u_D(t)$ . Es ergeben sich die Systemmatrix

$$\underline{A} = \begin{bmatrix} -\frac{d}{g_r^2 J_1} & -\frac{c}{g_r J_1} & \frac{d}{g_r J_1} \\ 1/g_r & 0 & -1 \\ \frac{d}{g_r J_2} & \frac{c}{J_2} & -\frac{d}{J_2} \end{bmatrix}, \quad (17.132)$$

der Einkoppelvektor

$$\underline{b}^\top = (1/J_1 \quad 0 \quad 0) \quad (17.133)$$

und der Auskoppelvektor

$$\underline{c}^\top = (0 \ 0 \ 1) \quad (17.134)$$

Das mechanische System wird durch (externe) Lastmomente  $M_W(t)$  [ $Nm$ ] gestört, die durch den Störgrößeneinkoppelvektor

$$\underline{b}_L^\top = (0 \ 0 \ -\frac{1}{J_2}) \quad (17.135)$$

lastseitig wirken. Zusätzlich wirkt Lagerreibung  $(\mathcal{N}\mathcal{F}\omega_2)(t)$  der Beschleunigung entgegen.

Folgt man der Argumentation in [106] oder Kap. 16, kann der Reibungseinfluss  $(\mathcal{N}\mathcal{F}\omega_2)(t)$  in einen unbeschränkten viskosen Anteil  $\nu_V \omega_2(\cdot)$  (mit dem viskosen Reibungskoeffizienten  $\nu_V > 0$ ) und einen beschränkten Anteil  $(\mathbf{N}\omega_2)(\cdot)$  aufgespalten werden. Der viskose Anteil lässt sich der Zustandsmatrix  $\mathbf{A}$  zuschlagen, man erhält die erweiterte Matrix

$$\mathbf{A}_V := \mathbf{A} + \nu_V \underline{b}_L (0 \ 0 \ 1) = \begin{bmatrix} -\frac{d}{g_r^2 J_1} & -\frac{c}{g_r J_1} & \frac{d}{g_r J_1} \\ 1/g_r & 0 & -1 \\ \frac{d}{g_r J_2} & \frac{c}{J_2} & -\frac{d + \nu_V}{J_2} \end{bmatrix} \quad (17.136)$$

Die geschätzten Parameter des Zwei-Massen-Systems sind in Tab. 17.7 zusammengefasst. Die Parameter wurden mithilfe von strukturierten rekurrenten Netzen identifiziert [89, 7]. Auch wenn für dieses Experiment alle Parameter zur Verfügung stünden, werden für die nicht-identifizierenden Regelungsverfahren Funnel-Control (FC) und Error Reference Control (ERC) diese *nicht* benötigt. Aufgrund der physikalischen Eigenschaften des Zwei-Massen-Systems (z.B. alle Parameter größer Null) können ohne exakte Parameterkenntnis die Anforderungen (A1)-(A3) für Zugehörigkeit zur Klasse  $\mathcal{S}$  nachgeprüft werden. Im Hinblick auf eine Regelung der lastseitigen Winkelgeschwindigkeit entsteht hier schon ein Widerspruch, der Relativgrad ist mit

$$\delta_{2MS} = 3 - 1 = 2 \quad (17.137)$$

(siehe [78] oder auch Kap. 16) zu hoch. Es muss also der Relativgrad auf eins reduziert werden, um überhaupt Funnel-Control oder Error Reference Control anwenden zu können. Dies erfolgt im Abschnitt 17.6.3 explizit unter Berücksichtigung des Einflusses der Getriebeübersetzung  $g_r$ .

### 17.6.2 Aktive Dämpfung durch statische Zustandsrückführung

In diesem Abschnitt wollen wir einen physikalisch motivierten Weg zur Dämpfung des 2MS beschreiben, also einen anderen als in Kap. 16 vorgeschlagen. Kein Hochpassfilter wird entworfen, sondern unabhängig von der verwendeten Zustandsrückführung (zur Relativgradreduktion) eine *statische* Dämpfung implementiert. Diese kann empirisch an der (z.B. ungeregelten) Anlage getestet und

Daten der Implementierung		
2MS	Symbol & Wert	Dimension
Rotorträgheitsmoment	$J_1 = 0.166$	$[kg\ m^2]$
Lastträgheitsmoment	$J_2 = 0.333$	$[kg\ m^2]$
Wellensteifigkeit	$c = 410$	$\left[\frac{Nm}{rad}\right]$
Wellendämpfung	$d = 0.025$	$\left[\frac{Nm}{rad}\right]$
Übersetzung	$g_r = 1$	[1]
Viskoser Reibungskoeffizient	$\nu_V = 0.0018$	$\left[\frac{Nm}{rad}\right]$
Stellgrößenbeschränkung	$M_{M,max} = 22$	[Nm]
Lastmoment	$M_W = 5\sigma(t - 5s)$	[Nm]
Einstellungen in Matlab/Simulink (xPC-Target)		
Fixed-step solver	Runge-Kutta (ode4)	–
Abtastperiode	$h = 1 \cdot 10^{-3}$	[s]

Tabelle 17.7: Daten des Versuchsstandes (Zwei-Massen-System)

nach den Wünschen des Ingenieurs ausgelegt werden. Die statische Zustandsrückführung

$$u_D(t) = k_D \underbrace{\begin{bmatrix} \frac{1}{g_r} & 0 & -1 \end{bmatrix}}_{:= b_D^\top} \underline{x}(t) = k_D \left( \frac{1}{g_r} \omega_1(t) - \omega_2(t) \right) = k_D \Delta \dot{\varphi}(t) \quad (17.138)$$

generiert für Rückführkoeffizienten  $k_D \neq 0$  mit  $\text{sign}(k_D) = \text{sign}(g_r)$  ein ‘Dämpfungsmoment’  $u_D$  [Nm] [78]. Der konstante Dämpfungskoeffizient  $k_D \left[ \frac{Nm}{rad} \right]$  gewichtet die Änderung (zeitliche Ableitung) des Verdrehwinkels  $\Delta \dot{\varphi}$ . Die Zustandsrückführung (17.138) führt zu einer erweiterten Systemmatrix

$$\mathbf{A}_D := \mathbf{A}_V - k_D \mathbf{b} \mathbf{b}^\top = \begin{bmatrix} -\frac{d + g_r k_D}{g_r^2 J_1} & -\frac{c}{g_r J_1} & \frac{d + g_r k_D}{g_r J_1} \\ 1/g_r & 0 & -1 \\ \frac{d}{g_r J_2} & \frac{c}{J_2} & -\frac{d + \nu_V}{J_2} \end{bmatrix} \quad (17.139)$$

Man erhält die nichtlineare Zustandsdarstellung des *gedämpften* 2MS

$$\begin{aligned} \dot{\underline{x}}(t) &= \mathbf{A}_D \underline{x}(t) + \mathbf{b} v(t) + \mathbf{b}_L (M_W(t) + (\mathbf{N} \omega_2)(t)) \quad , \underline{x}(0) = \underline{x}^0 \in \mathbb{R}^n \\ y(t) &= \underline{c}^\top \underline{x}(t) \end{aligned} \quad \left. \right\} \quad (17.140)$$

Als Nächstes soll der Einfluss der statischen Rückführung untersucht werden, hierzu vernachlässigen wir die nichtlinearen Effekte (hier: Reibung) und betrachten den linearen Fall. Man erhält die Übertragungsfunktion

$$\begin{aligned}
G_{\omega_2}(s) &:= \frac{\omega_2(s)}{V(s)} = \underline{c}^\top (s\mathbf{I}_3 - \mathbf{A}_D)^{-1} \underline{b} \\
&= \frac{\frac{1}{g_r} \left( s \frac{d}{c} + 1 \right)}{s \left( J_1 + \frac{J_2}{g_r^2} \right) \left[ s^2 \frac{J_1 J_2}{\left( J_1 + \frac{J_2}{g_r^2} \right) c} + s \frac{1}{c} \left( d + \frac{J_2}{g_r (J_1 + \frac{J_2}{g_r^2})} k_D \right) + 1 \right]}.
\end{aligned} \tag{17.141}$$

Vergleicht man nun Zähler und Nenner mit einem schwingungsfähigen

$$F_{PT_2}(s) = \frac{1}{1 + 2D/\omega_0 s + 1/\omega_0^2 s^2} \tag{17.142}$$

lässt sich die Eigenfrequenz

$$\omega_0 = \sqrt{\frac{\left( J_1 + \frac{J_2}{g_r^2} \right) c}{J_1 J_2}} \tag{17.143}$$

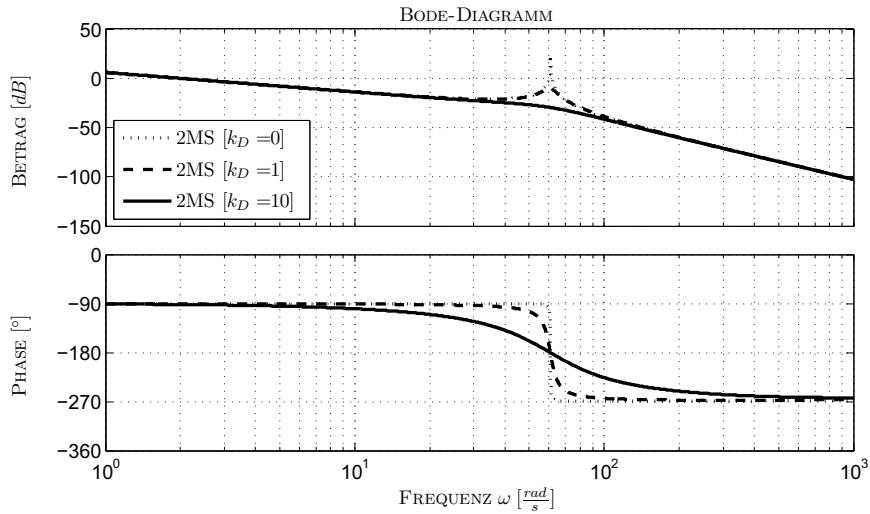
bestimmen. Sie stimmt mit der bekannten Frequenz des nicht aktiv gedämpften Zwei-Massen-Systems überein [202, 203]. Dagegen erhält man als Dämpfungskoeffizient

$$D_D(k_D) = \frac{\omega_0}{2c} \underbrace{\left( d + \frac{J_2}{g_r \left( J_1 + \frac{J_2}{g_r^2} \right)} k_D \right)}_{\geq d > 0} \geq \frac{\omega_0}{2c} d = D \tag{17.144}$$

welcher durch die Rückführung  $k_D$  (beliebig) eingestellt werden kann. Für jeden Wert  $k_D \text{ sign}(g_r) > 0$  kann also die Dämpfung des mechanischen Systems vergrößert werden. In der Realität sind drei Einschränkungen zu beachten:

1. die zur Verfügung stehende Stellgröße  $M_M(t)$  (Motormoment),
2. das transiente Verhalten der Leistungselektronik und des Antriebes (wie schnell kann das Dämpfungsmoment  $u_D$  erzeugt werden?) und
3. je größer die Wahl der Rückführung  $|k_D|$ , desto langsamer die Systemantwort bzw. -dynamik, da jede Änderung  $\Delta\dot{\varphi}$  des Verdrehwinkels durch das Dämpfungsmoment  $u_D$  bestraft, und somit lastmaschinen-seitig dem Aufbau eines Beschleunigungsmoments entgegengewirkt wird.

Alle Einschränkungen sind im Allgemeinen nicht zu restriktiv, da angenommen werden kann, dass bei richtiger und anwendungsbezogener Spezifikation des Antriebes eine aktive Bedämpfung des schwingungsfähigen mechanischen Systems berücksichtigt worden und somit möglich ist.



**Abb. 17.25:** Bode-Diagramm des Zwei-Massen-Systems für unterschiedliche Dämpfungs faktoren  $k_D \in \{0, 1, 10\}$

*Bemerkung.* Es ist wichtig eine Vorzeichen-korrekte Rückführung  $k_D$  mit  $\text{sign}(k_D) = \text{sign}(g_r)$  zu wählen, denn ein falsches Vorzeichen wird die Be dämpfung des mechanischen Systems verringern und für  $\frac{k_D}{g_r} \leq -\frac{d}{J_2} \left( J_1 + \frac{J_2}{g_r^2} \right)$  sogar destabilisieren.

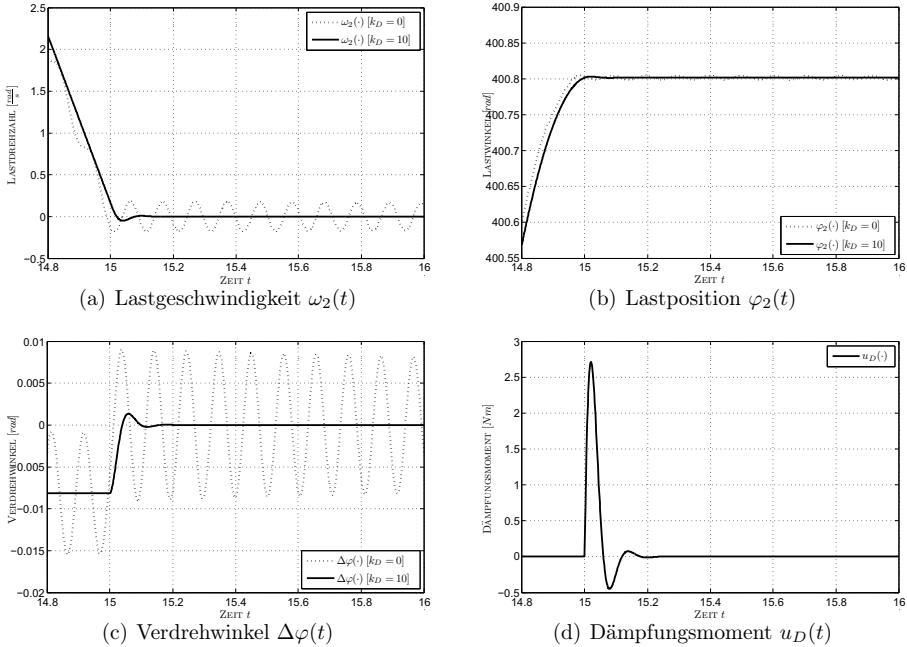
In Abb. 17.25 sind die Bode-Diagramme unterschiedlich gedämpfter Zwei Massen-Systeme dargestellt. Für steigende Werte von  $k_D$  nimmt der Peak im Betragsverlauf ab. Auch der in Abb. 17.26 gezeigte Ausschnitt eines An- und Ablaufversuchs des 2MS bestätigt die gewünschte Wirkung des Dämpfungs moments  $u_D(\cdot)$  (siehe Abb. 17.26d) für eine Verstärkung  $k_D = 10$ . Für  $k_D = 0$  treten dagegen deutliche Schwingungen v.a. im Verdrehwinkel  $\Delta\varphi(\cdot)$  auf (siehe Abb. 17.26c).

### 17.6.3 Erweiterung des 2MS für Zugehörigkeit in $\mathcal{S}$

Entsprechend den Vorüberlegungen in [210, 209, 75, 77, 69, 78, 106] oder Kap. 16 kann der Relativgrad des gedämpften Zwei-Massen-Systems (17.140) durch eine Neu-Definition des Ausgangs mithilfe des erweiterten Auskoppelvektors

$$\underline{c}_r^\top := (k_{\omega_1} \quad k_{\Delta\varphi} \quad k_{\omega_2}) \quad (17.145)$$

für alle  $k_{\omega_1} \neq 0$  reduziert werden. Das entstandene ‘Hilfssystem’



**Abb. 17.26:** Gegenüberstellung der Schwingungsneigung des ZMS ohne und mit Dämpfung [ZOOM]

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\underline{x}(t) + \underline{b}v(t) + \underline{b}_L(M_W(t) + (\mathbf{N}\omega_2)(t)) \quad , \underline{x}(0) = \underline{x}_0 \in \mathbb{R}^n \\ y_r(t) &= \underline{c}_r^\top \underline{x}(t) \end{aligned} \quad \left. \right\} \quad (17.146)$$

mit dem Hilfsausgang  $y_r(t)$  muss nun auf Einhaltung der Bedingungen (A1)-(A3) überprüft werden. Da (17.146) ein nichtlineares System beschreibt, soll (17.146) in Byrnes-Isidori Normalform überführt werden. Hierzu können die Kenntnisse aus Abschnitt 17.1.3.1 sinnvoll genutzt werden.

Obwohl das erweiterte Hilfssystem (17.146) nichtlineare Bestandteile aufweist, führt die Wahl

$$\mathbf{V} = \begin{bmatrix} -\frac{k_{\Delta\varphi}}{k_{\omega_1}} & -\frac{k_{\omega_2}}{k_{\omega_1}} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (17.147)$$

und die Berechnung von  $\mathbf{N}$  entsprechend (17.58) auf die Transformationsmatrix

$$\mathbf{S} = \begin{bmatrix} \underline{c}_r \\ \mathbf{N} \end{bmatrix} = \begin{bmatrix} k_{\omega_1} & k_{\Delta\varphi} & k_{\omega_2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17.148)$$

mit der Inversen

$$\mathbf{S}^{-1} = [\underline{b}(\underline{\mathcal{C}}_r^\top \underline{b})^{-1}, \quad \mathbf{V}] = \begin{bmatrix} 1/k_{\omega_1} & -k_{\Delta\varphi}/k_{\omega_1} & -k_{\omega_2}/k_{\omega_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (17.149)$$

Für die Koordinatentransformation

$$\begin{pmatrix} y_r \\ z \end{pmatrix} := \mathbf{S} \underline{x} \quad (17.150)$$

erhält man also die Konstante

$$a_1 = \underline{\mathcal{C}}_r^\top \mathbf{A}_D \underline{b} (\underline{\mathcal{C}}_r^\top \underline{b})^{-1} = -\frac{d + g_r k_D}{g_r^2 J_1} + \frac{k_{\Delta\varphi}}{k_{\omega_1}} \frac{1}{g_r} + \frac{k_{\omega_2}}{k_{\omega_1}} \frac{d}{g_r J_2} =: \Xi(\mathbf{A}_D, \underline{b}, \underline{\mathcal{C}}_r) \quad (17.151)$$

mit dem Wurzelschwerpunkt<sup>11)</sup>  $\Xi(\mathbf{A}_D, \underline{b}, \underline{\mathcal{C}}_r)$  des erweiterten Systems (17.146), die Vektoren

$$\underline{a}_2 = (\underline{\mathcal{C}}_r^\top \mathbf{A}_D \mathbf{V})^\top = \begin{bmatrix} -k_{\omega_1} \frac{c}{g_r J_1} - k_{\Delta\varphi} \Xi(\mathbf{A}_D, \underline{b}, \underline{\mathcal{C}}_r) + k_{\omega_2} \frac{c}{J_2} \\ k_{\omega_1} \frac{d + g_r k_D}{g_r J_1} - k_{\Delta\varphi} - k_{\omega_2} \left( \Xi(\mathbf{A}_D, \underline{b}, \underline{\mathcal{C}}_r) + \frac{d + \nu_V}{J_2} \right) \end{bmatrix}, \quad (17.152)$$

und

$$\underline{a}_3 = \mathbf{N} \mathbf{A}_D \underline{b} (\underline{\mathcal{C}}_r^\top \underline{b})^{-1} = \frac{1}{k_{\omega_1}} \begin{bmatrix} \frac{1}{g_r} \\ \frac{g_r}{d} \\ \frac{g_r}{g_r J_2} \end{bmatrix}, \quad (17.153)$$

die Systemmatrix der internen Dynamik

$$\mathbf{A}_4 = \mathbf{N} \mathbf{A} \mathbf{V} = \begin{bmatrix} -\frac{k_{\Delta\varphi}}{k_{\omega_1}} \frac{1}{g_r} & -\frac{k_{\omega_2}}{k_{\omega_1}} \frac{1}{g_r} - 1 \\ -\frac{k_{\Delta\varphi}}{k_{\omega_1}} \frac{d}{g_r J_2} + \frac{c}{J_2} & -\frac{k_{\omega_2}}{k_{\omega_1}} \frac{d}{g_r J_2} - \frac{d + \nu_V}{J_2} \end{bmatrix} \quad (17.154)$$

und dem transformierten Störgrößeneinkoppelvektor

$$\underline{b}_{r,L} := \mathbf{S} \underline{d}_L = \begin{pmatrix} -k_{\omega_2}/J_2 \\ 0 \\ -1/J_1 \end{pmatrix}. \quad (17.155)$$

---

<sup>11)</sup> im Wurzelschwerpunkt kreuzen sich die Asymptoten der Wurzelortskurve (siehe auch [56] oder [144])

Mit (17.151), (17.152), (17.153), (17.154) und  $z_2(t) = \omega_2(t)$  lässt sich nun die kompakte Darstellung des erweiterten nichtlinearen Systems (17.146) in Byrnes-Isidori Normalform angeben

$$\begin{aligned} \dot{y}_r(t) &= a_1 y_r(t) + \underline{a}_2^\top \underline{z}(t) + \underline{c}_r^\top \underline{b} M_M(t) - \frac{k_{\omega_2}}{J_2} (M_W(t) + (\mathbf{N} z_2)(t)) \\ \dot{\underline{z}}(t) &= \underline{a}_3 y_r(t) + \mathbf{A}_4 \underline{z}(t) + \begin{pmatrix} 0 \\ -\frac{1}{J_1} \end{pmatrix} (M_W(t) + (\mathbf{N} z_2)(t)) \end{aligned} \quad (17.156)$$

mit  $y_r(0) = y_r^0$  und  $\underline{z}(0) = \underline{z}^0$ . Der Signalflussplan von (17.156) entspricht dem in Abb. 17.5 mit zusätzlicher nichtlinearer Störung.

Außer der Getriebeübersetzung  $g_r \neq 0$  sind alle physikalischen Parameter positiv bzw. nicht negativ mit  $J_1 > 0$ ,  $J_2 > 0$ ,  $c > 0$ ,  $d > 0$  und  $\nu_V \geq 0$  [200]. Somit besitzt das erweiterte transformierte System (17.156) für  $k_{\omega_1} > 0$  eine positive instantane Verstärkung

$$V_{0,r \text{ 2MS}} := \underline{c}_r^\top \underline{b} = \frac{k_{\omega_1}}{J_1} > 0 \quad (17.157)$$

und einen Relativgrad

$$\delta_{r, \text{2MS}} = 1. \quad (17.158)$$

Zur Überprüfung der Stabilität der Nulldynamik wird die charakteristische Gleichung

$$\begin{aligned} \chi_{\mathbf{A}_4}(\lambda) &= \det(\lambda \mathbf{I}_n - \mathbf{A}_4) \\ &= \underbrace{(k_{\omega_1} J_2)}_{=: b_2} \lambda^2 + \underbrace{\left( d \left( k_{\omega_1} + \frac{k_{\omega_2}}{g_r} \right) + \nu_V k_{\omega_1} + k_{\Delta\varphi} \frac{J_2}{g_r} \right)}_{=: b_1} \lambda + \\ &\quad \underbrace{c \left( k_{\omega_1} + \frac{k_{\omega_2}}{g_r} \right) + k_{\Delta\varphi} \frac{\nu_V}{g_r}}_{=: b_0} \end{aligned} \quad (17.159)$$

der Matrix  $\mathbf{A}_4$  näher untersucht. Für folgende Wahl der Rückführkoeffizienten

$$k_{\omega_1} > 0 \iff b_2 > 0 \quad (17.160)$$

$$k_{\omega_1} + \frac{k_{\omega_2}}{g_r} \geq 0 \wedge k_{\omega_1} > 0 \wedge \frac{k_{\Delta\varphi}}{g_r} \geq 0 \implies b_1 > 0 \quad (17.161)$$

$$\frac{k_{\Delta\varphi}}{g_r} \geq 0 \wedge k_{\omega_1} + \frac{k_{\omega_2}}{g_r} > 0 \implies b_0 > 0 \quad (17.162)$$

ist das Polynom (17.159) Hurwitz und somit die interne Dynamik BIBO stabil, da  $M_W(\cdot)$  und  $(\mathbf{N} z_2)(\cdot)$  beschränkt angenommen wurden. Falls keine Störungen

---

**Zulässige Wahl der Rückführkoeffizienten**

	$g_r > 0$	$g_r < 0$
Rückführung für $\omega_1$	$k_{\omega_1} > 0$	$k_{\omega_1} > 0$
Rückführung für $\Delta\varphi$	$k_{\Delta\varphi} \geq 0$	$k_{\Delta\varphi} \leq 0$
Rückführung für $\omega_2$	$k_{\omega_2} > -g_r k_{\omega_1}$	$k_{\omega_2} < -g_r k_{\omega_1}$

---

**Tabelle 17.8:** Wahl der Rückführungscoeiffizienten  $k_{\omega_1}, k_{\Delta\varphi}, k_{\omega_2}$  in Abhängigkeit der Getriebeübersetzung  $g_r \in \mathbb{R} \setminus \{0\}$  zur Reduktion des Relativgrades und zum Erhalt der stabilen Nulldynamik

herrschen — also  $M_W(\cdot) = (\mathbf{N}z_2)(\cdot) = 0$  — ist (17.156) linear, daher minimalphasig mit exponentiell stabiler Nulldynamik.

Zusammenfassend gefährdet also die dämpfende Zustandsrückführung (17.138) die Einhaltung der Anforderungen (A1)-(A3) nicht. In Tab. 17.8 sind die erlaubten Konstellationen der Rückführkoeffizienten  $k_{\omega_1}$ ,  $k_{\Delta\varphi}$  und  $k_{\omega_2}$  aufgelistet, um ein erweitertes System (17.156) der Klasse  $\mathcal{S}$  zu garantieren. Aufgrund der Verwendung des statischen Dämpfungsansatzes (17.138) kann die Rückführung des Verdrehwinkels ( $k_{\Delta\varphi} = 0$ ) und die Implementierung des Hochpassfilters (siehe Kap. 16) vermieden werden.

#### 17.6.4 Messung und Bewertung der Ergebnisse

Abbildung 17.27 zeigt den geschlossenen Regelkreis — bestehend aus 2MS, Zustandsrückführung, PI-Erweiterung, FC oder ERC. Die im folgenden verwendete PI-Erweiterung entspricht einem simplen PI-Regler mit der Übertragungsfunktion

$$G_{PI}(s) = \frac{V(s)}{U(s)} = 1 + \frac{1}{sT_I} = \frac{1 + sT_I}{sT_I} \quad (17.163)$$

und ist notwendig, um stationäre Genauigkeit als auch gutes Störverhalten zu erreichen. Der PI-Regler in (17.163) ist für alle  $T_I > 0$  minimalphasig, hat einen Relativgrad  $r_{PI} = 0$  und eine positive instantane Verstärkung  $V_{0,PI} = 1$ . Daher ändert sich nichts an der Struktur des erweiterten Systems aus PI und 2MS mit Eingang  $u$  und Ausgang  $y_r$ :

- Relativgrad des Hilfssystems  $\delta_r = \delta_{r,2MS} + \delta_{PI} = 1$ ,
- positive instantane Verstärkung  $V_{0,r} = V_{0,r2MS} \cdot V_{0,PI} > 0$
- und stabile Nulldynamik für  $T_I > 0$  und eine Wahl der Rückführkoeffizienten  $k_{\omega_1}$ ,  $k_{\Delta\varphi}$  und  $k_{\omega_2}$  entsprechend Tab. 17.8

Reglerdesign				
Reglerparameter	SF	FC	ERC	Dim.
$k_{\omega_1}$	3.95	1	1	$[\frac{Nm}{rad}]$
$k_{\Delta\varphi}$	35.45	0	0	$[\frac{Nm}{rad}]$
$k_{\omega_2}$	2.25	-0.1	-0.1	$[\frac{Nm}{rad}]$
$PI$ -Zeitkonstante $T_I$	—	0.0667	0.0667	$[s]$
Vorverstärkung $k_V$	18.97 <sup>11)</sup>	0.9	0.9	$[1]$
Verstärkung $k_D$	—	5	5	$[\frac{Nm}{rad}]$

Trichterdesign und Wahl der Skalierungsfunktion				
Kundenanford. $(\tau_\nu, \nu, \mu)$	—	(1, 2, 0.8)	(1, 2, 0.8)	$[s, \frac{rad}{s}]$
Anfangsw. $\partial\mathcal{F}_E(0) = e^+(0)$	—	25.1	25.1	$[\frac{rad}{s}]$
Zeitkonstante $T_E = T_{ERC}$	—	0.329	0.329	$[s]$
Funktion $\kappa^+(t) = \kappa_-(t)$	—	—	$\left(\frac{1}{\varphi_{E,0}} -  e(0) \right)e^{-\frac{t}{T_E}}$	$[\frac{rad}{s}]$
Endgenauigkeit $\mu^+ = \mu_-$	—	—	0.8	$[\frac{rad}{s}]$
Skalierung $\varsigma(t)$	—	1	(17.127)	$[Nm]$
Verstärkung $\varsigma_G$	—	—	2	$[Nm]$
Begrenzung $\varsigma_D^{max}$	—	—	49	$[Nm]$
Minimum $\varsigma_0$	—	—	1	$[Nm]$

**Tabelle 17.9:** Reglerparameter für SF, FC und ERC mit entsprechendem Design des Exponentiellen Trichters (FC) und des virtuellen Schlauchs (ERC)

Zugehörigkeit zur Klasse  $\mathcal{S}$  wird also durch (17.163) nicht gefährdet (siehe auch Kap. 16 oder [74]). Die Vorverstärkung

$$k_V = k_{\omega_1} + k_{\omega_2} \quad (17.164)$$

ist zur Anpassung des Referenzsignals  $y^*(t)$  notwendig, um den Hilfssollwertverlauf

$$y_r^*(t) = k_V y^*(t) = k_V \omega_2^*(t) \quad (17.165)$$

zu generieren [209, 75]. Damit entsteht der erweiterte Hilfsfehler

$$e_r(t) = y_r^*(t) - y_r(t) \quad (17.166)$$

der innerhalb des Trichters (FC) bzw. des Schlauchs (ERC) geführt werden kann. Nur das erweiterte Hilfssystem (17.146) bzw. (17.156) erfüllt die Bedingungen (A1)-(A3), was erst die Anwendung von FC und ERC erlaubt. Für die Messungen am Versuchsstand werden FC mit (17.1) und ERC mit (17.129) jeweils

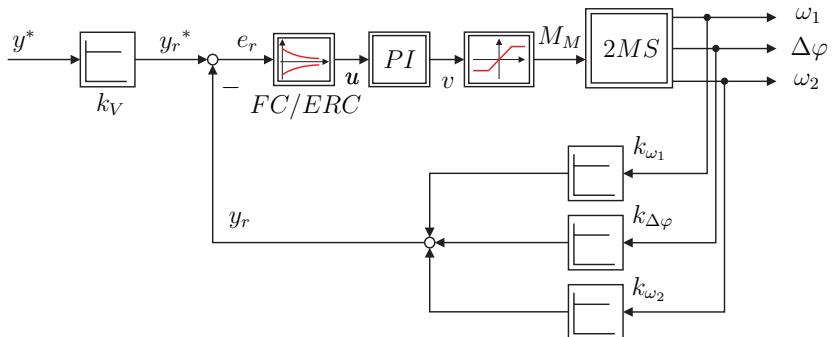
<sup>11)</sup> Dieser Wert entspricht der Verstärkung des Integrators der äußeren Kaskade der Zustandsregelung mit integralem Anteil

bei Auswertung der vertikalen Distanz implementiert. Die generierte Stellgröße  $u(t) = k(t, e_r(t)) e_r(t)$  dient jeweils als Eingang von (17.163). Zusätzlich wird bei beiden Verfahren die dämpfende Zustandsrückführung (17.138) eingesetzt. FC und ERC generieren jeweils das Motormoment

$$M_M(t) = v(t) - u_D(t) = k(t, e_r(t)) e_r(t) + \int_0^t k(\tau, e_r(\tau)) e_r(\tau) d\tau - u_D(t). \quad (17.167)$$

Die dämpfende Skalierungsfunktionen (17.99) und (17.127) sind entsprechend für Funnel-Control (FC) und Error Reference Control (ERC) anwendbar. Vor allem zu Beginn — bei noch ‘weitem’ Trichter — ist der positive Effekt bei Funnel-Control (FC) erheblich geringer als bei Error Reference Control (ERC), da die Stellgröße (17.1) nur mit dem Regelfehler  $e(t)$  (bzw. hier mit dem Hilfsfehler  $e_r(t)$ ) sein Vorzeichen ändern kann. Daher wird die Skalierung für Funnel-Control *nicht* implementiert.

Stellgrößenbeschränkungen werden in diesem Beitrag vernachlässigt. Die Regler sind entsprechend so ausgelegt, dass sie nicht in Sättigung fahren, d.h. für alle  $t \geq 0$  gilt  $|M_M(t)| \leq M_{M,\max} = 22 \text{ Nm}$ . Sofern Stellgrößenbeschränkungen zu Problemen führen, können erste Ergebnisse und Ideen aus [72, 73] übernommen werden. Beide nicht-identifizierenden Regelungskonzepte werden mit einem



**Abb. 17.27:** Regelkreis aus 2MS, Zustandsrückführung und FC + PI-Regler (bzw. ERC + PI-Regler)

LQR Zustandsregler<sup>12)</sup> (SF) mit integralem Anteil — als Standardansatz — verglichen.

<sup>12)</sup> Optimaler Reglerentwurf mit Gütefunktional

$$\min_{(k_{\omega_1}, k_{\Delta\varphi}, k_{\omega_2})} J = \int_0^\infty \exp(2k) (\underline{x}^\top Q \underline{x} + r \cdot u^2) d\tau$$

und Stabilitätsreserve  $k = 3$  (Robuster Entwurf) [145]

Die Winkelgeschwindigkeit der Lastmaschine wird als Ausgang  $y(t) = \omega_2(t)$  erachtet. Das Regelziel ist die Folgewertregelung eines vorgegebenen (zeitvarian-ten) Referenzverlaufs  $y^*(t) = \omega_2^*(t)$ . Alle durch Sensoren erfassten Systemgrößen sind durch hochfrequentes Messrauschen gestört, z.B.  $\omega_2(t) = \omega_{2,\text{real}}(t) + n(t)$ . Die relevanten System- und Reglerparameter sind jeweils in den Tabellen 17.7 und 17.9 aufgelistet.

Der Regelkreis soll dem Referenzverlauf

$$\begin{aligned} y^*(t) &= \omega_2^*(t) \\ &= 15 \frac{\text{rad}}{\text{s}} + 5 \frac{\text{rad}}{\text{s}^2} (t - 10\text{s}) \cdot (\sigma(t - 10\text{s}) - \sigma(t - 16\text{s})) + \dots \\ &\quad + 3 \frac{\text{rad}}{\text{s}} \sin\left(\frac{\pi}{10}t\right) \sin\left(\frac{\pi}{5}t\right) \cdot \sigma(t - 20\text{s}) \end{aligned} \quad (17.168)$$

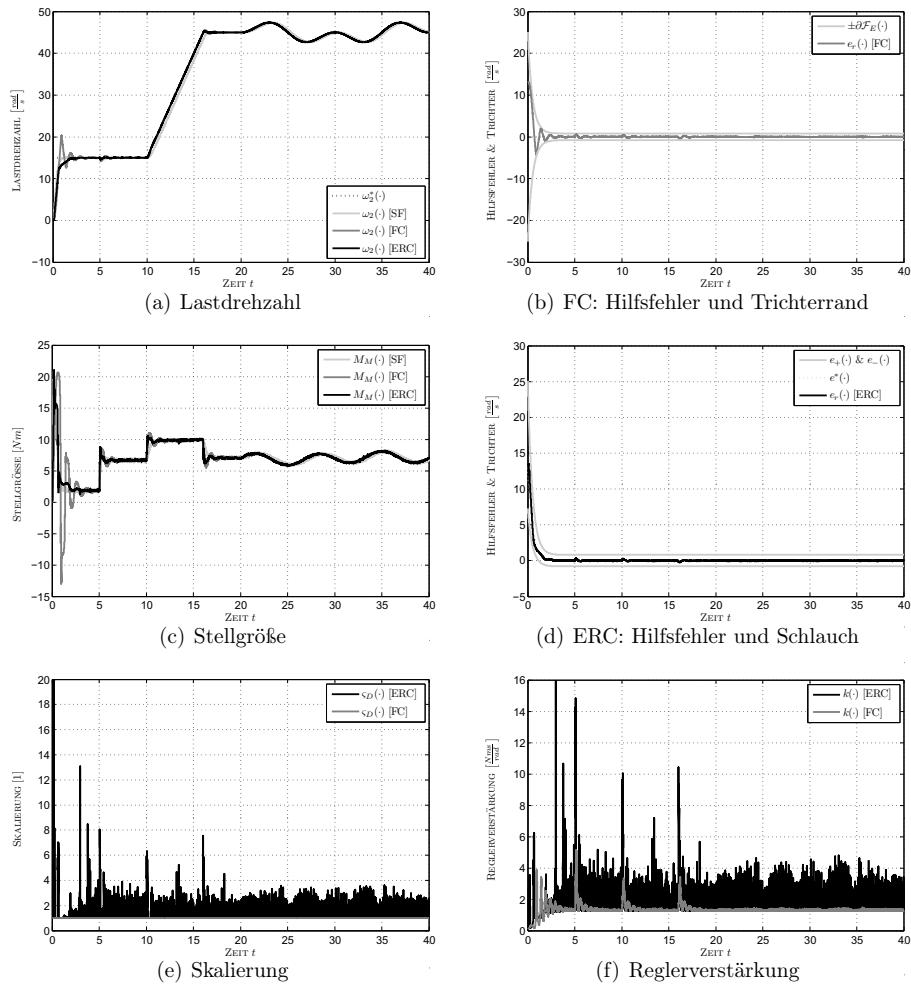
mit der Winkelgeschwindigkeit der Arbeitsmaschine  $\omega_2(t)$  bestmöglich folgen, d.h. der eigentliche Regelfehler ist durch  $e(t) = \omega_2^*(t) - \omega_2(t)$  gegeben. Der Sollwertverlauf besteht aus einem konstantem Anteil, einer Rampe und einer Überlagerung von sinusförmigen Signalen (siehe gepunktete Linie in Abb. 17.28a). Das Referenzsignal ist dem Prototyp in Abb. 17.4 qualitativ ähnlich und soll anhand der Zeitvarianz<sup>13)</sup> eine Herausforderung an die Folgewertregelung darstellen. Die Messergebnisse sind in den Abbildungen 17.28 und 17.29 dargestellt. Die Ergebnisse untermauern für beide adaptive (zeitvariante) Regelverfahren die theoretisch hergeleitete Fähigkeit der Folgewertregelung mit *a priori festgelegtem transienten Verhalten*. Hingegen zeigt der LQR Zustandsregler deutliche Schleppfehler, insbesondere beim Folgeverhalten der Rampe (siehe Abb. 17.29a) bzw. der sinusförmigen Referenzverläufe (siehe Abb. 17.29b). Alle drei Regelungsstrategien erzielen eine gute Störgrößenkompensation, dabei reagiert ERC auf den Lastsprung  $M_W(t) = 5\sigma(t - 5\text{s}) \text{Nm}$  mit der kleinsten Abweichung (siehe Abb. 17.29a). Obwohl ERC ohne jede Kenntnis der Systemparameter entworfen wurde, kann zu Beginn ein annähernd so gut gedämpftes (aperiodisches) Einschwingen wie mit SF erreicht werden (siehe Abb. 17.28a). Während der Zeintervalle  $[0, 5\text{s}]$  und  $[15\text{s}, 20\text{s}]$  wird das Überschwingen von FC deutlich (siehe Abb. 17.28a und 17.29b). Die Hilfsfehler  $e_r(t)$  für FC und ERC verbleiben für alle  $t \geq 0$  innerhalb der vordefinierten Gebiete (siehe Abb. 17.28b,d):

- bei FC innerhalb des exponentiellen Trichters mit Rand  $\partial\mathcal{F}_E(\cdot)$  und
- bei ERC innerhalb des virtuellen Schlauches  $\mathbb{E}_{(e_-, e_+)}$ .

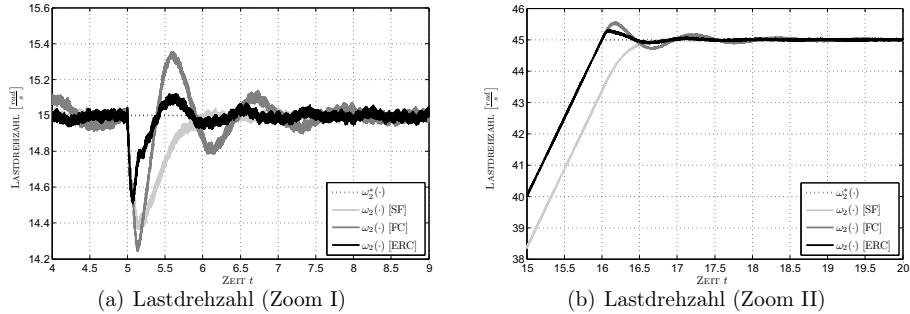
In Abb. 17.30 ist das Verhalten des gedämpften Systems für  $k_D = 5$  dem unge-dämpften System mit  $k_D = 0$  gegenübergestellt. Der Verdrehwinkel entwickelt sich für  $k_D = 5$  deutlich ruhiger als für  $k_D = 0$ . Natürlich muss aufgrund der Vorgabe des Trichters bzw. des Schlauchs die Welle stärker beansprucht werden

---

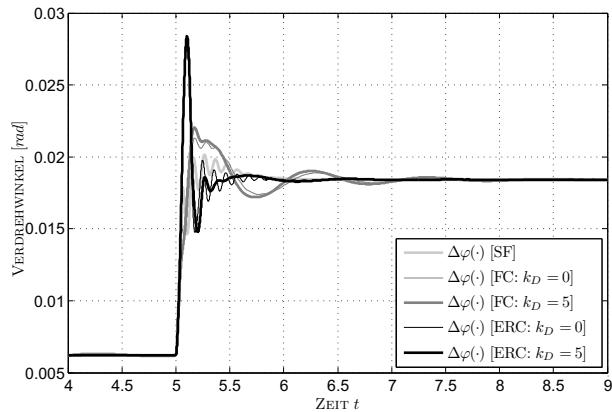
<sup>13)</sup> Die Referenz ist nicht speziell durch einen realen Prozess bedingt, sondern willkürlich gewählt.



**Abb. 17.28:** Gegenüberstellung der Messergebnisse von SF, FC und ERC am nichtlinearen 2MS unter Last



**Abb. 17.29:** Gegenüberstellung der Messergebnisse von SF, FC und ERC am nichtlinearen 2MS unter Last (Zoom I+II)



**Abb. 17.30:** Gegenüberstellung der Verdrehwinkel von SF, FC und ERC am nichtlinearen 2MS für  $k_D = 0$  und  $k_D = 5$  (Simulation)

als bei SF. Hierdurch kann überhaupt erst ein Verlauf innerhalb der festgelegten Grenzen erfolgen. Die Welligkeit im erweiterten Fehler  $e_r(\cdot)$  bzw. erweiterten Systemausgang  $y_r(\cdot)$  kann durch eine kleinere Wahl der Integratorzeitkonstante  $T_I$  reduziert werden. Infolgedessen muss jedoch mit einem verschlechterten (verlangsamten) Störverhalten gerechnet werden.

## 18 Einführung in die Fuzzy–Regelung

In den letzten Jahren gewinnt in der Regelungstechnik ein Verfahren an Bedeutung, das sich von den bisher bekannten Regelungsverfahren grundlegend unterscheidet: die *Fuzzy–Regelung* (*fuzzy control*). Bei der Fuzzy–Regelung handelt es sich um ein *regelbasiertes* (*rule-based*) Regelungsverfahren; das heißt, daß das Verhalten eines Fuzzy–Reglers nicht durch ein mathematisches Regelgesetz, sondern durch verbale Regeln beschrieben wird, wie z. B. bei einer Lageregelung

„Wenn der Abstand klein ist und die Geschwindigkeit mittelgroß ist, dann muß der Drehzahlsollwert klein sein.“

Gegenüber konventionellen Regelungskonzepten besitzt die Fuzzy–Regelung eine Reihe von Vorteilen:

- Zum Entwurf wird kein mathematisches Streckenmodell benötigt.
- Die Fuzzy–Regelung stellt einen einfachen Weg zum Entwurf von nichtlinearen Reglern dar.
- Die Fuzzy–Regelung ist sehr flexibel, d.h. es können Regler mit nahezu beliebigem Verhalten realisiert werden; beim Reglerentwurf besteht eine Vielzahl von Einflußmöglichkeiten.
- Die Fuzzy–Regelung ist sehr anschaulich.
- Man kann auch qualitatives Wissen über die Strecke verwerten, das sich nicht exakt mathematisch formulieren läßt.

Diesen Vorteilen steht aber auch eine Reihe von Nachteilen gegenüber:

- Es existieren keine standardisierten Entwurfsverfahren.
- Die Optimierung von Fuzzy–Reglern erfolgt durch Probieren und ist wegen der Vielzahl von Einflußmöglichkeiten zeitaufwendig.
- Da Fuzzy–Regler nichtlinear sind, ist ihre mathematische Behandlung (z.B. Stabilitätsuntersuchung) schwierig.
- Fuzzy–Regler benötigen einen relativ hohen Rechenaufwand.

Der Fuzzy–Regelung liegt die Theorie der *unscharfen Logik* (*fuzzy logic*) zugrunde, die auf der *Theorie der unscharfen Mengen* (*fuzzy set theory*) aufbaut. Die Grundlagen der unscharfen Mengenlehre wurden im Jahre 1965 von L. Zadeh<sup>1)</sup> formuliert [246]. Den Anstoß dazu erhielt Zadeh durch seine Arbeiten auf dem Gebiet der Mustererkennung. Zunächst wurde die neue Theorie nur wenig beachtet. Mitte der siebziger Jahre griffen dann jedoch einige Wissenschaftler die Idee der unscharfen Logik auf und entwickelten die Grundlagen der Fuzzy–Regelung [147]. Der erste industrielle Einsatz der Fuzzy–Regelung erfolgte im Jahr 1980; es handelte sich um die Regelung eines Zementdrehrohrofens, die von einer dänischen Firma entwickelt wurde [135]. Seitdem wurde die Fuzzy–Regelung für eine Vielzahl von Aufgaben eingesetzt, besonders von japanischen Firmen.

Diese Einführung soll einen Überblick über die Theorie der unscharfen Mengen, die unscharfe Logik und die Fuzzy–Regelung vermitteln. Abschnitt 18.1 gibt eine kurze Einführung in die Theorie der unscharfen Mengen; Abschnitt 18.2 befasst sich mit der unscharfen Logik und Abschnitt 18.3 mit ihrer Anwendung in der Fuzzy–Regelung.

## 18.1 Grundlagen der Theorie der unscharfen Mengen

In seinem ersten Artikel über unscharfe Mengenlehre begründete L. Zadeh das Konzept der unscharfen Menge folgendermaßen [246]:

*„Die Klassen von Objekten, die wir in der realen physikalischen Welt antreffen, haben in den meisten Fällen keine präzise definierten Zugehörigkeitskriterien. Zum Beispiel schließt die Klasse der Tiere sicherlich Hunde, Pferde, Vögel etc. als Mitglieder ein und solche Objekte wie Steine, Flüssigkeiten, Pflanzen etc. aus. Objekte wie Sterne, Bakterien etc. haben jedoch in Bezug auf die Klasse der Tiere einen zweideutigen Status. . . . Sicherlich stellen die ‚Klasse aller reellen Zahlen, die viel größer als Eins sind‘ oder die ‚Menge der schönen Frauen‘ oder die ‚Menge der großen Männer‘ keine Klassen oder Mengen im üblichen mathematischen Sinn dieser Begriffe dar. Dennoch bleibt die Tatsache bestehen, daß solche ungenau definierten ‚Klassen‘ eine wichtige Rolle im menschlichen Denken spielen.“*

### 18.1.1 Definition der unscharfen Menge

Im Gegensatz zum klassischen Mengenbegriff (*scharfe Menge; crisp set*), bei dem ein Objekt entweder Element einer bestimmten Menge ist oder nicht, wird bei

---

<sup>1)</sup> Lotfi A. Zadeh, \*1921 in Baku (UdSSR), Prof. der Elektrotechnik an der University of California, Berkeley.

einer *unscharfen Menge* (*fuzzy set*) der *Zugehörigkeitsgrad* (*degree of membership*<sup>2)</sup>) eines Objekts zu dieser Menge als reelle Zahl angegeben. (Üblicherweise wird als Wertebereich für den Zugehörigkeitsgrad das Intervall  $[0, 1]$  verwendet.) Man kann also bei einer unscharfen Menge  $A$  nicht sagen „ $x$  ist Element von  $A$ “, sondern z. B. „ $x$  besitzt einen Zugehörigkeitsgrad von 0,8 zu  $A$ “. Mathematisch wird dies wie folgt formuliert:

**Definition 18.1** Es sei  $X$  die *Grundmenge*<sup>3)</sup> der betrachteten Objekte (*universe of discourse*). Dann ist eine **unscharfe Menge**  $A$  auf  $X$ <sup>4)</sup> definiert als eine Menge<sup>5)</sup> von geordneten Paaren

$$A = \{(x, \mu_A(x)) | x \in X\}$$

Dabei ordnet die Funktion  $\mu_A : X \mapsto M$  jedem  $x \in X$  seinen Zugehörigkeitsgrad zu  $A$  zu.  $\mu_A$  wird als *Zugehörigkeitsfunktion* (*membership function*) bezeichnet. Der Wertebereich  $M$  von  $\mu_A$  (*membership space*) ist üblicherweise das Intervall  $[0, 1]$ .

**Beispiel 18.1** Es sei  $X = \{\text{Gauss, Kolmogoroff, Laplace, Mozart}\}$ . Dann kann man z. B. die unscharfe „Menge der berühmten Mathematiker“ definieren als  $A = \{(\text{Gauss}, 1), (\text{Kolmogoroff}, 0,4), (\text{Laplace}, 0,8), (\text{Mozart}, 0)\}$ .

**Beispiel 18.2** Es sei  $X$  die Menge der reellen Zahlen. Die unscharfe Menge  $B$  der „Zahlen, die ungefähr gleich 10 sind“, könnte etwa durch eine Zugehörigkeitsfunktion  $\mu_B$  nach Abb. 18.1 beschrieben werden. Die Zahl 9,0 hätte z. B. einen Zugehörigkeitsgrad zur Menge  $B$  von etwa 0,72.

### 18.1.2 Weitere Definitionen

Viele Definitionen der klassischen Mengenlehre können auf unscharfe Mengen übertragen werden:

**Definition 18.2** Die leere Menge  $L$  ist eine unscharfe Menge mit

$$\mu_L(x) = 0 \quad \text{für alle } x \in X$$

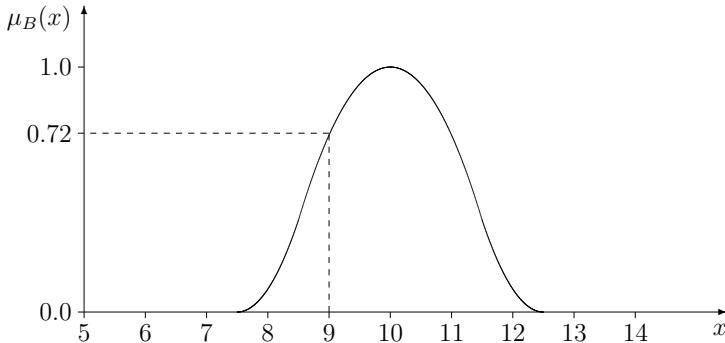
---

<sup>2)</sup> Auch: *grade of membership, grade of compatibility*

<sup>3)</sup>  $X$  ist eine scharfe Menge.

<sup>4)</sup> Manchmal wird  $A$  auch als *unscharfe Teilmenge* (*fuzzy subset*) von  $X$  bezeichnet.

<sup>5)</sup> Die unscharfe Menge wird hier unter Verwendung des klassischen scharfen Mengenbegriffs definiert.



**Abb. 18.1:** Menge der Zahlen, die ungefähr gleich 10 sind

**Definition 18.3** Die Menge  $\mathcal{P}(X)$  aller unscharfen Mengen auf einer Grundmenge  $X$  heißt **Fuzzy-Potenzmenge** (fuzzy power set) der Menge  $X$ .

$\mathcal{P}(X)$  ist also eine (unendlich große) scharfe Menge, deren Elemente unscharfe Mengen sind.

**Definition 18.4** Zwei unscharfe Mengen  $A \in \mathcal{P}(X)$  und  $B \in \mathcal{P}(X)$  heißen **gleich**, wenn gilt

$$\mu_A(x) = \mu_B(x) \quad \text{für alle } x \in X$$

**Definition 18.5** Eine unscharfe Menge  $A \in \mathcal{P}(X)$  heißt **enthalten in** einer unscharfen Menge  $B \in \mathcal{P}(X)$ , wenn gilt

$$\mu_A(x) \leq \mu_B(x) \quad \text{für alle } x \in X$$

Man schreibt dafür „ $A \subseteq B$ “.

Gilt sogar

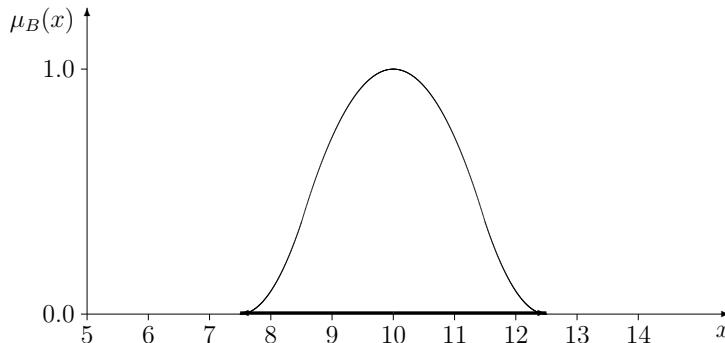
$$\mu_A(x) < \mu_B(x) \quad \text{für alle } x \in X$$

so heißt  $A$  **echt enthalten in**  $B$ . Man schreibt dafür „ $A \subset B$ “.

In der Theorie der unscharfen Mengen sind außerdem folgende Begriffe definiert:

**Definition 18.6** Die **stützende Menge** (support)  $\mathcal{S}(A)$  einer unscharfen Menge  $A \in \mathcal{P}(X)$  ist definiert als die (scharfe) Menge aller  $x \in X$ , für die die Zugehörigkeitsfunktion  $\mu_A(x)$  größer Null ist:

$$\mathcal{S}(A) := \{x \in X | \mu_A(x) > 0\}$$



**Abb. 18.2:** Stützende Menge

Zum Beispiel ist die stützende Menge der „Menge der berühmten Mathematiker“ aus Beispiel 18.1  $S(A) = \{\text{Gauss, Kolmogoroff, Laplace}\}$ ; die stützende Menge der Menge  $B$  aus Beispiel 18.2 ist das Intervall  $(7,5, 12,5)$  (siehe Abb. 18.2).

**Definition 18.7** Die  $\alpha$ -Niveau-Menge ( $\alpha$ -level-set,  $\alpha$ -cut)  $A_\alpha$  einer unscharfen Menge  $A \in \mathcal{P}(X)$  ist definiert als die (scharfe) Menge aller  $x \in X$ , für die die Zugehörigkeitsfunktion  $\mu_A(x)$  größer oder gleich einem bestimmten Wert  $\alpha$  ist:

$$A_\alpha := \{x \in X | \mu_A(x) \geq \alpha\}$$

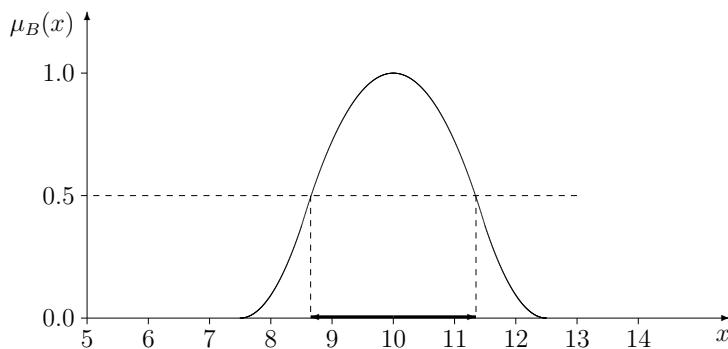
Die Menge aller  $x \in X$ , für die  $\mu_A(x)$  echt größer  $\alpha$  ist, heißt **strenge  $\alpha$ -Niveau-Menge** (strong  $\alpha$ -level-set).

Zum Beispiel ist für die „Menge der berühmten Mathematiker“  $A_{0,7} = \{\text{Gauss, Laplace}\}$ ; für die Menge  $B$  aus Beispiel 18.2 ist  $B_{0,5} = (8,5, 11,5)$  (siehe Abb. 18.3).

**Definition 18.8** Die **Höhe** (height) einer unscharfen Menge  $A$  ist die kleinste obere Schranke<sup>6)</sup> der Zugehörigkeitsfunktion  $\mu_A$  auf  $X$ :

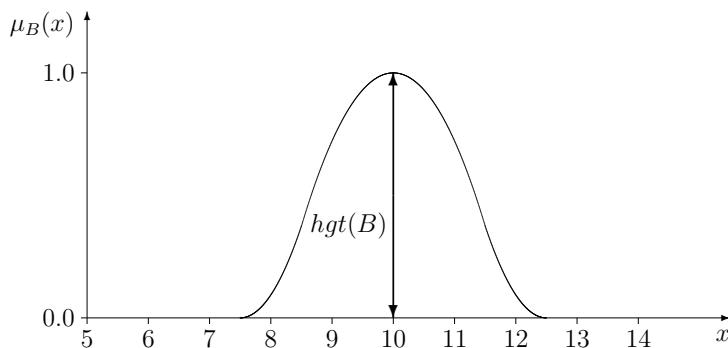
$$\text{hgt}(A) := \sup_{x \in X} \mu_A(x)$$

<sup>6)</sup> Die Bildung der kleinsten oberen Schranke (Supremum) einer Menge von Zahlen ist eine häufig benötigte Operation in der unscharfen Mengenlehre und ihren Anwendungen. Das Supremum ist eine Verallgemeinerung des Maximums. Der Unterschied zwischen Supremum und Maximum wird z. B. an der Funktion  $f(t) = 1 - e^{-t/T}$  (Sprungantwort des PT<sub>1</sub>-Gliedes) deutlich: diese Funktion hat kein Maximum, aber ihr Supremum ist  $\sup_t f(t) = 1$ .



**Abb. 18.3:** Alpha-Niveau Menge  $B_{0,5}$  der Menge  $B$

Bei den üblicherweise verwendeten Zugehörigkeitsfunktionen ist die kleinste obere Schranke gleich dem Maximum der Funktion. Z. B. ist für die Menge  $B$  aus Beispiel 18.2  $\text{hgt}(B) = 1,0$  (siehe Abb. 18.4).



**Abb. 18.4:** Höhe einer unscharfen Menge

**Definition 18.9** Eine unscharfe Menge  $A$  heißt **normalisiert** (*normalized*), wenn gilt:

$$\text{hgt}(A) = 1$$

**Definition 18.10** Eine unscharfe Menge  $A = \{(x, \mu_A(x)) | x \in X\}$  heißt **konvex**, wenn gilt:

$$\mu_A(x) \geq \min(\mu_A(x_1), \mu_A(x_2)) \quad \text{für alle } x_1, x_2 \in X, x \in [x_1, x_2]$$

Dies bedeutet, daß  $\mu_A$  außer am Rand keine lokalen Minima hat.

**Definition 18.11** Ein **Singleton** ist eine unscharfe Menge  $A$ , deren stützende Menge nur eine einzelne reelle Zahl  $x_0$  enthält:

$$\mu_A(x) = \begin{cases} a & \text{für } x = x_0 \\ 0 & \text{für } x \neq x_0 \end{cases}$$

Man schreibt dafür abkürzend  $A = a/x_0$ .

Abb. 18.5 zeigt ein Beispiel für ein Singleton.

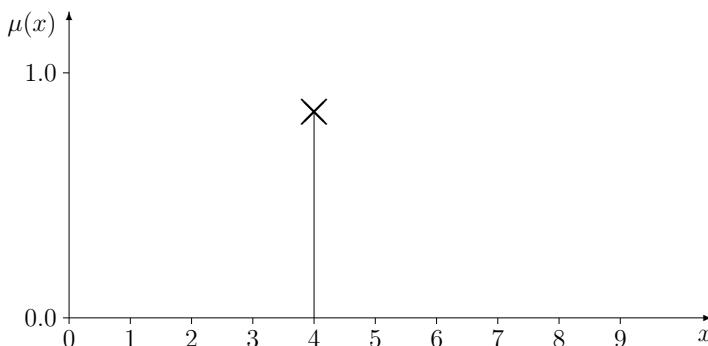
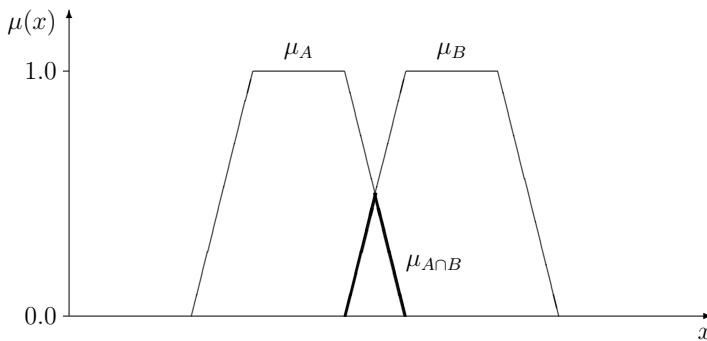


Abb. 18.5: Singleton 0.84/4

### 18.1.3 Grundlegende Mengenoperationen für unscharfe Mengen

Die grundlegenden Mengenoperatoren in der klassischen Mengenlehre sind Durchschnitt, Vereinigung und Komplement. Die Schnittmenge zweier scharfer Mengen  $A$  und  $B$  enthält alle Elemente  $x$ , für die gilt „ $x \in A$  UND  $x \in B$ “; die Vereinigungsmenge enthält die Elemente, für die gilt „ $x \in A$  ODER  $x \in B$ “; die Komplementmenge einer scharfen Menge  $A$  enthält die Elemente, für die gilt „NICHT ( $x \in A$ )“. Wie man sieht, besteht hier eine enge Beziehung zwischen Mengenlehre und Logik; darauf wird in Abschnitt 18.2 noch näher eingegangen.

Die Übertragung der Mengenoperationen auf unscharfe Mengen ist auf verschiedene Weise möglich. Zwei elementare Mengenoperatoren für unscharfe Mengen wurden bereits von Zadeh in [246] vorgeschlagen: der Minimumoperator und der Maximumoperator.



**Abb. 18.6:** Schnittmenge zweier unscharfer Mengen

**Definition 18.12** Die **Schnittmenge** (*intersection*) zweier unscharfer Mengen  $A \in \mathcal{P}(X)$  und  $B \in \mathcal{P}(X)$  ist definiert als

$$A \cap B := \{(x, \mu_{A \cap B}(x)) | x \in X\}$$

mit

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

(siehe Abb. 18.6).

**Definition 18.13** Die **Vereinigungsmenge** (*union*) zweier unscharfer Mengen  $A \in \mathcal{P}(X)$  und  $B \in \mathcal{P}(X)$  ist definiert als

$$A \cup B := \{(x, \mu_{A \cup B}(x)) | x \in X\}$$

mit

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

(siehe Abb. 18.7).

Der Zugehörigkeitsgrad zur Schnittmenge ist also gleich dem *kleineren* der beiden Zugehörigkeitsgrade zu den einzelnen Mengen, der Zugehörigkeitsgrad zur Vereinigungsmenge gleich dem *größeren* der beiden Zugehörigkeitsgrade. Auf die Beziehungen dieser Mengenoperatoren zur logischen UND- bzw. ODER-Verknüpfung wird in Abschnitt 18.2.4 noch näher eingegangen.

**Definition 18.14** Das **Komplement** (*complement*) einer unscharfen Menge  $A \in \mathcal{P}(X)$  ist definiert als

$${}_k A := \{(x, \mu_{_k A}(x)) | x \in X\}$$

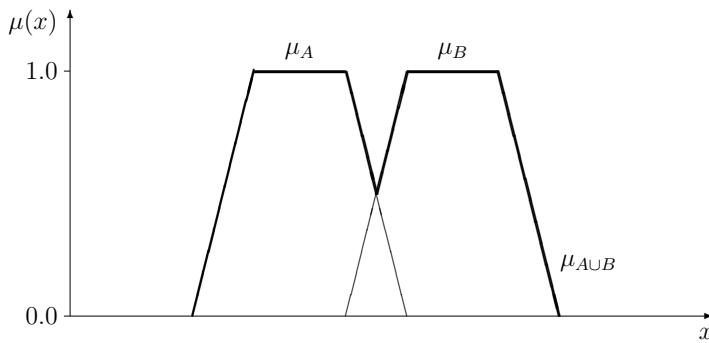


Abb. 18.7: Vereinigungsmenge zweier unscharfer Mengen

mit

$$\mu_{kA}(x) = 1 - \mu_A(x)$$

(siehe Abb. 18.8)

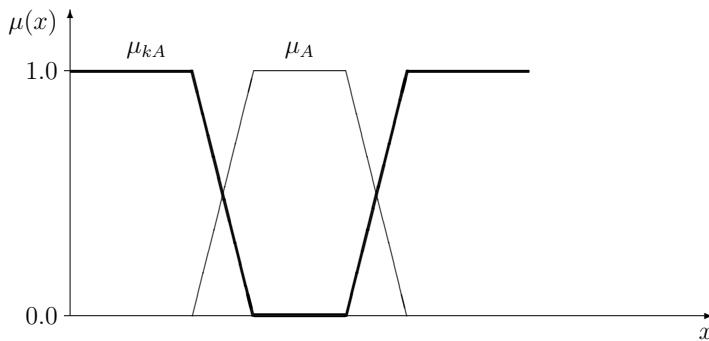


Abb. 18.8: Komplement einer unscharfen Menge

Für diese Mengenoperatoren gelten folgende Gesetze: ( $A$ ,  $B$  und  $C$  seien unscharfe Mengen  $\in \mathcal{P}(X)$ )

### Satz 18.1 Kommutativität

$$A \cap B = B \cap A$$

$$A \cup B = B \cup A$$

**Satz 18.2 Assoziativität**

$$\begin{aligned}(A \cap B) \cap C &= A \cap (B \cap C) \\ (A \cup B) \cup C &= A \cup (B \cup C)\end{aligned}$$

**Satz 18.3 Distributivität**

$$\begin{aligned}A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) &= (A \cup B) \cap (A \cup C)\end{aligned}$$

**Satz 18.4 Adjunktivität**

$$\begin{aligned}A \cap (A \cup B) &= A \\ A \cup (A \cap B) &= A\end{aligned}$$

Dagegen ist das Gesetz der Komplementarität für unscharfe Mengen nicht gültig:

$$\begin{aligned}A \cap {}_k A &\neq L \\ A \cup {}_k A &\neq X\end{aligned}$$

( $L$  = leere Menge (siehe Definition 18.2),  $X$  = Grundmenge von  $A$ ).

Somit bildet  $\mathcal{P}(X)$  bezüglich der Operatoren  $\cap$  und  $\cup$  einen distributiven Verband, der aber nicht komplementär ist.

Weiterhin gilt

**Satz 18.5 Gesetz von De Morgan**

$$\begin{aligned}{}_k(A \cap B) &= {}_k A \cup {}_k B \\ {}_k(A \cup B) &= {}_k A \cap {}_k B\end{aligned}$$

sowie

**Satz 18.6 Involution**

$${}_{kk} A = A$$

### 18.1.4 Modifikatoren

Eine weitere Klasse von Mengenoperatoren für unscharfe Mengen sind die so genannten *Modifikatoren* (*modifiers*). Dies sind Operatoren, die die Form der Zugehörigkeitsfunktion einer unscharfen Menge verändern. Hierzu gehören der *Konzentrationsoperator* (*concentration*), der *Dilationsoperator* (*dilation*) und der *Kontrastverstärkungsoperator* (*contrast intensification*). Diese Operatoren sind folgendermaßen definiert:

**Definition 18.15** Der **Konzentrationsoperator** ordnet einer unscharfen Menge  $A$  die unscharfe Menge  $\text{CON}(A)$  mit

$$\mu_{\text{CON}(A)}(x) = (\mu_A(x))^2$$

zu (siehe Abb. 18.9).

**Definition 18.16** Der **Dilationsoperator** ordnet einer unscharfen Menge  $A$  die unscharfe Menge  $\text{DIL}(A)$  mit

$$\mu_{\text{DIL}(A)}(x) = \sqrt{\mu_A(x)}$$

zu (siehe Abb. 18.10).

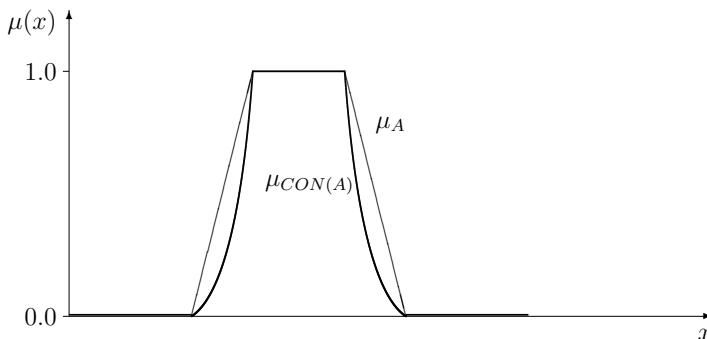


Abb. 18.9: Konzentrationsoperator

**Definition 18.17** Der **Kontrastverstärkungsoperator** ordnet einer unscharfen Menge  $A$  die unscharfe Menge  $\text{INT}(A)$  mit

$$\mu_{\text{INT}(A)}(x) = \begin{cases} 2(\mu_A(x))^2 & \text{für } \mu_A(x) < 0,5 \\ 1 - 2(1 - \mu_A(x))^2 & \text{für } \mu_A(x) \geq 0,5 \end{cases}$$

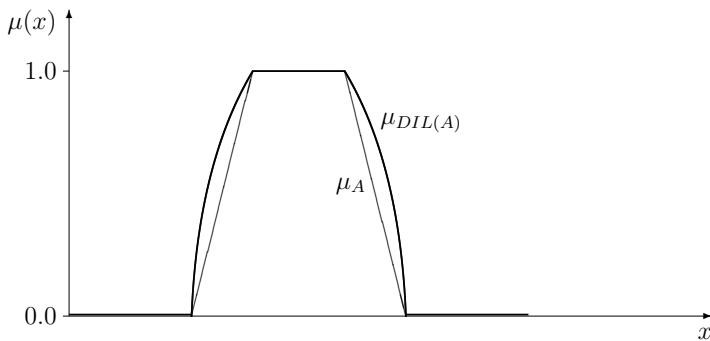


Abb. 18.10: Dilationsoperator

zu (siehe Abb. 18.11).

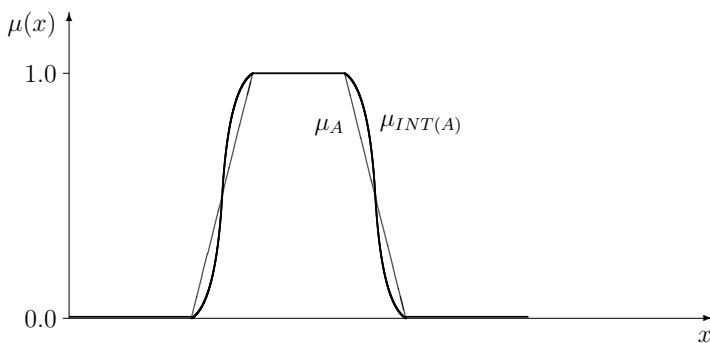


Abb. 18.11: Kontrastverstärkungsoperator

Diese Operatoren wurden ursprünglich in der Absicht definiert, linguistische Modifikatoren (*hedges*) wie „sehr“, „ziemlich“ etc. zu modellieren [247]. Beispielsweise sollte aus der Menge  $A$  der *großen* Menschen die Menge der *sehr großen* Menschen mit Hilfe des Konzentrationsoperators  $\text{CON}(A)$  gebildet werden. Es hat sich allerdings gezeigt, daß bereits die Semantik von einfachen Modifikatoren wie „sehr“ zu komplex ist, um sie durch einfache mathematische Modelle nachzubilden zu können.

## 18.2 Grundlagen der unscharfen Logik

### 18.2.1 Einführung

Das wichtigste Anwendungsgebiet der Theorie der unscharfen Mengen ist die *unscharfe Logik (fuzzy logic)*. Während in der klassischen „scharfen“ Logik nur Aussagen betrachtet werden, die entweder *wahr* oder *falsch* sind, sind in der unscharfen Logik auch ungenaue, qualitative Aussagen wie z. B. „die Temperatur ist ziemlich niedrig“ zulässig. Der Übergang von der unscharfen Mengenlehre zur unscharfen Logik ist naheliegend: beispielsweise lässt sich der Zugehörigkeitsgrad von Kolmogoroff zur Menge der berühmten Mathematiker ohne weiteres als eine Art „unscharfer Wahrheitswert“ der Aussage „Kolmogoroff war ein berühmter Mathematiker“ interpretieren.

Es muß darauf hingewiesen werden, daß es zwei unterschiedliche Versionen der unscharfen Logik gibt: eine mehr anwendungsorientierte und eine mehr theoretische. In der Praxis wird meistens die anwendungsorientiertere Darstellungsweise verwendet; in wissenschaftlichen Aufsätzen findet man dagegen häufig die theoretischere Darstellungsweise, da sie allgemeiner ist und die anwendungsorientiertere Darstellungsweise als Spezialfall enthält.

In dieser Einführung wird die mehr anwendungsorientierte Darstellungsweise verwendet, da sie einfacher verständlich ist; in Kapitel 18.4 wird auf die theoretischere Darstellungsweise und die Unterschiede zwischen den beiden Darstellungsweisen eingegangen.

Das folgende einfache Beispiel aus der Regelungstechnik soll dazu dienen, zunächst einige Grundideen der unscharfen Logik zu erläutern:

**Beispiel 18.3** Eine Heizung wird durch einen Fuzzy–Regler geregelt. Die Eingangsgröße dieses Reglers ist die Raumtemperatur  $\vartheta$ , die Ausgangsgröße die Stellung des Heizungsventils  $y$  ( $y = 100\%$  bedeutet voll geöffnet,  $y = 0\%$  voll geschlossen).

Das Verhalten eines Fuzzy–Reglers wird durch verbale **Regeln** beschrieben; daher wird die Fuzzy–Regelung auch als „regelbasiertes Regelungskonzept“ bezeichnet. Der Regler in unserem Beispiel besitzt nur zwei Regeln:

Regel 1: *Wenn die Temperatur niedrig ist, öffne das Ventil.*

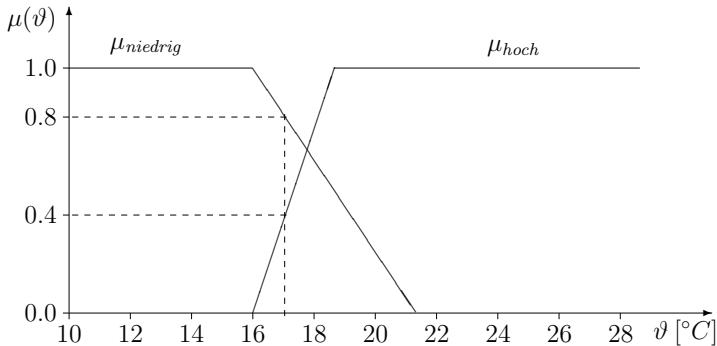
Regel 2: *Wenn die Temperatur hoch ist, schließe das Ventil.*

Der Algorithmus eines Fuzzy–Reglers besteht prinzipiell aus drei Schritten:

1. Umsetzen der gemessenen „scharfen“ Werte der Eingangsgrößen in unscharfe Aussagen (Fuzzifizierung);
2. Anwenden der Regeln auf diese unscharfen Aussagen (Inferenz);
3. Umsetzen der unscharfen Ergebnisse der Regeln in scharfe Werte der Ausgangsgröße (Defuzzifizierung).

Diese drei Schritte sollen nun für einen Eingangswert von  $\vartheta = 17^\circ\text{C}$  ausführlich dargestellt werden:

Die Fuzzyfizierung besteht darin, daß der gemessenen Temperatur ihre Zugehörigkeitsgrade zur Menge der hohen Temperaturen und zur Menge der niedrigen Temperaturen zugeordnet werden. Diese Mengen wurden beim Reglerentwurf definiert, z. B. wie in Abb. 18.12. (Üblicherweise werden mehr als zwei solche Mengen für jede Eingangsgröße verwendet. Auf die geeignete Definition dieser Mengen und ihren Einfluß auf das Reglerverhalten wird in Kapitel 18.3.1 noch näher eingegangen.)



**Abb. 18.12:** Fuzzyfizierung der Eingangsgröße  $\vartheta$

Für  $\vartheta = 17^\circ\text{C}$  lautet das Ergebnis der Fuzzyfizierung also  $\mu_{\text{niedrig}}(\vartheta) = 0.8$ ,  $\mu_{\text{hoch}}(\vartheta) = 0.4$ ; dies könnte man verbal etwa durch die Aussage „Die Temperatur ist relativ niedrig“ beschreiben.

Auf diese unscharfe Beschreibung der Eingangsgröße werden im zweiten Schritt die Regeln angewendet. Die Rechenvorschrift dafür lautet:

$$\begin{aligned} G_{\text{offen}} &= \mu_{\text{niedrig}}(\vartheta) \\ G_{\text{geschlossen}} &= \mu_{\text{hoch}}(\vartheta) \end{aligned}$$

dabei ist  $G_{\text{offen}}$  der Gültigkeitsgrad der Aussage „Ventil offen“ und  $G_{\text{geschlossen}}$  der Gültigkeitsgrad der Aussage „Ventil geschlossen“. (Zum Begriff des Gültigkeitsgrades s. Abschnitt 18.2.2.) Für  $\vartheta = 17^\circ\text{C}$  ergibt sich also  $G_{\text{offen}} = 0.8$ ,  $G_{\text{geschlossen}} = 0.4$ . Verbal könnte man dieses Ergebnis etwa mit „Ventil ziemlich weit offen“ übersetzen.

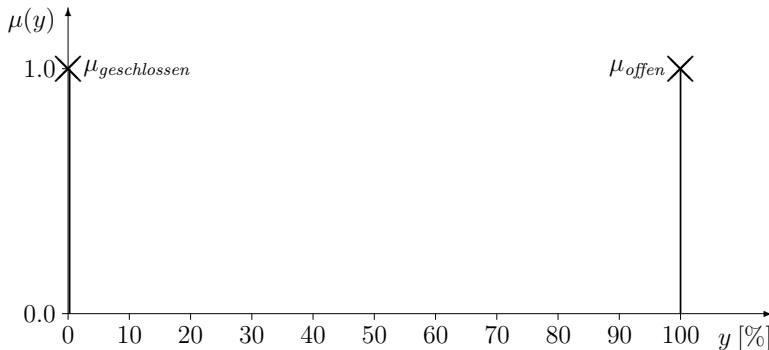
Der dritte Schritt ist die Defuzzyfizierung. Dafür gibt es verschiedene Verfahren. Ein sehr einfaches besteht darin, die Ventilstellung nach der Vorschrift

$$y = \frac{G_{\text{offen}} \cdot y_{\text{offen}} + G_{\text{geschlossen}} \cdot y_{\text{geschlossen}}}{G_{\text{offen}} + G_{\text{geschlossen}}}$$

zu berechnen; dabei sind  $y_{\text{offen}} = 100\%$  und  $y_{\text{geschlossen}} = 0\%$  scharfe Werte von  $y$ , die den Zuständen „Ventil offen“ bzw. „Ventil geschlossen“ entsprechen. In der Sprache der unscharfen Mengenlehre bedeutet das, daß als „Menge der offenen Ventilstellungen“ ein Singleton mit der Zugehörigkeitsfunktion

$$\mu_{\text{offen}}(y) = \begin{cases} 1 & \text{für } y = 100\% \\ 0 & \text{für } y \neq 100\% \end{cases}$$

definiert wird; die „Menge der geschlossenen Ventilstellungen“ wird entsprechend definiert (siehe Abb. 18.13).



**Abb. 18.13:** Defuzzifizierung der Ausgangsgröße  $y$

Für die Eingangsgröße  $\vartheta = 17^\circ\text{C}$  erhält man auf diese Weise als Ausgangsgröße  $y = 66,6\%$

### 18.2.2 Grundbegriffe der unscharfen Logik

Nach dem einführenden Beispiel aus dem vorigen Abschnitt sollen nun einige Begriffe und Vorgehensweisen der unscharfen Logik eingehender dargestellt werden. Am Anfang sollen die drei Grundbegriffe *linguistische Variable*, *unscharfe Aussage* und *Gültigkeitsgrad* stehen.

Ein grundlegendes Konzept der unscharfen Logik ist die *linguistische Variable* (*linguistic variable*). Alle in der unscharfen Logik vorkommenden Aussagen sind Aussagen über linguistische Variable. Linguistische Variable sind Variable, „deren Werte keine Zahlen sind, sondern Worte oder Sätze einer natürlichen oder künstlichen Sprache“ (L. Zadeh). Diese Worte bzw. Sätze werden als *Terme* (*terms*) bezeichnet. In dem Beispiel aus dem vorigen Abschnitt war also „Temperatur“ eine linguistische Variable mit den Termini „hoch“ und „niedrig“.

Die Aussagen, die in der unscharfen Logik behandelt werden, haben die Form „ $X$  ist  $A_i$ “, wobei  $X$  eine linguistische Variable und  $A_i$  ein Term von  $X$  ist.

Solche Aussagen werden als *unscharfe Aussagen* (*fuzzy proposition*, *fuzzy clause*, *fuzzy assertion*) bezeichnet; ein Beispiel dafür ist die Aussage „Die Temperatur ist niedrig.“

Während in der klassischen Logik jede Aussage entweder den Wahrheitswert WAHR oder den Wahrheitswert FALSCH hat, besitzt in der unscharfen Logik jede unscharfe Aussage einen *Gültigkeitsgrad* (*degree of validity*). Der Gültigkeitsgrad ist eine reelle Zahl zwischen Null und Eins. Ein Gültigkeitsgrad von Eins bedeutet, daß die Aussage mit Sicherheit wahr ist; ein Gültigkeitsgrad von Null bedeutet, daß der Wahrheitswert der Aussage völlig unsicher (aber nicht unbedingt falsch!) ist.

Der „Wert“ einer linguistischen Variablen  $X$  besteht aus den Gültigkeitsgraden der Aussagen „ $X$  ist  $A_1$ “ . . . „ $X$  ist  $A_n$ “, wobei  $A_1 \dots A_n$  die Terme von  $X$  sind<sup>7)</sup>. Die linguistische Variable „Temperatur“ hatte also in dem Beispiel den Wert „niedrig mit einem Gültigkeitsgrad von 0.8“, hoch mit einem Gültigkeitsgrad von 0.4“; die linguistische Variable „Ventilstellung“ hatte den Wert „geschlossen mit einem Gültigkeitsgrad von 0.4“, offen mit einem Gültigkeitsgrad von 0.8“.

Anmerkung: Die Begriffe linguistische Variable, unscharfe Aussage und Gültigkeitsgrad bilden die Grundlage der unscharfen Logik. Es sei darauf hingewiesen, daß diese Grundbegriffe zunächst völlig unabhängig von der Theorie der unscharfen Mengen definiert wurden.

### 18.2.3 Fuzzyfizierung und logisches Schließen

Im vorigen Abschnitt war der Begriff des Gültigkeitsgrades definiert worden. Es wurde aber noch nichts darüber gesagt, wie man den Gültigkeitsgrad einer unscharfen Aussage ermitteln kann. Darauf soll nun eingegangen werden.

Die erste Möglichkeit ist die Ermittlung von Gültigkeitsgraden aus gegebenen Fakten. Es wird davon ausgegangen, daß die Fakten scharfe Werte von numerischen Variablen sind (z. B. „ $\vartheta = 17^\circ C$ “) und daß diese numerischen Variablen durch linguistische Variable beschrieben werden. Die numerische Variable  $\vartheta$  wird z. B. durch die linguistische Variable „Temperatur“ beschrieben: ihren Termen „hoch“ und „niedrig“ sind unscharfe Mengen von Werten von  $\vartheta$  zugeordnet (Abb. 18.12). Die numerische Variable wird als *Basisvariable* (*base variable*) der linguistischen Variablen bezeichnet; die unscharfen Mengen, die den Termen zugeordnet sind, heißen *Bedeutungsmengen* (*meanings*).  $\vartheta$  ist also die Basisvariable von „Temperatur“, und die Menge der hohen Temperaturen  $\{(\vartheta, \mu_{\text{hoch}}(\vartheta))\}$  ist die Bedeutungsmenge des Terms „hoch“.

Die Regel zur Umsetzung der Fakten in Gültigkeitsgrade liegt auf der Hand:

**Satz 18.7 Fuzzyfizierungsregel** *Es sei  $X$  eine linguistische Variable,  $A_i$  ein Term dieser linguistischen Variablen und  $x$  die Basisvariable von  $X$ . Der Wert*

---

<sup>7)</sup> Dies gilt für die „anwendungsorientierte“ unscharfe Logik. In der „theoretischen“ unscharfen Logik ist der Wert einer linguistischen Variablen anders definiert.

von  $x$  sei bekannt (z. B. durch Messung). Dann ist der Gültigkeitsgrad der Aussage „ $X$  ist  $A_i$ “ gleich dem Zugehörigkeitsgrad von  $x$  zur Bedeutungsmenge des Terms  $A_i$ .

Anmerkung: Es ist wichtig, den Unterschied zwischen *Gültigkeitsgrad* und *Zugehörigkeitsgrad* zu beachten. Der Begriff Gültigkeitsgrad gehört in den Bereich der unscharfen Logik, der Begriff Zugehörigkeitsgrad dagegen in den Bereich der unscharfen Mengenlehre. Außerdem muß der Gültigkeitsgrad einer unscharfen Aussage nicht unbedingt etwas mit einem Zugehörigkeitsgrad zu tun haben: in unserem Beispiel war der Gültigkeitsgrad der Aussage „Ventil ist offen“ 0,8; es existiert aber kein Wert der Basisvariablen  $y$ , für den  $\mu_{\text{offen}}(y) = 0,8$  wäre (siehe Abb. 18.13).

Die zweite Möglichkeit zur Ermittlung von Gültigkeitsgraden ist das logische Schließen. Durch Regeln der Form

WENN *Aussage 1*  
DANN *Aussage 2*

wird ein Zusammenhang zwischen dem Gültigkeitsgrad von *Aussage 2* und dem Gültigkeitsgrad von *Aussage 1* hergestellt. *Aussage 1* wird als *Vorbedingung (antecedent)* bezeichnet, *Aussage 2* als *Schlußfolgerung (consequent)*.

In der unscharfen Logik gilt, daß bei einer solchen Regel die Schlußfolgerung den gleichen Gültigkeitsgrad wie die Vorbedingung hat.<sup>8)</sup>

$$\text{Gültigkeitsgrad}(\text{Aussage 2}) = \text{Gültigkeitsgrad}(\text{Aussage 1})$$

**Beispiel 18.4** Eine Regel bei der Qualitätskontrolle von Elektromotoren laute:

Regel 1): WENN *das Motorgeräusch sehr laut ist,*  
DANN *ist der Motor defekt*

Die Lautstärke  $x$  des Motorgeräusches wird gemessen. Aus der Fuzzyfizierung des gemessenen Wertes erhält man bei einem bestimmten Motor  $\mu_{\text{sehrlaut}}(x) = 0,1$ . Aus Regel 1) ergibt sich dann der Gültigkeitsgrad der Aussage „Der Motor ist defekt“ ebenfalls zu 0,1.

An diesem Beispiel wird deutlich, daß die Bezeichnung des Gültigkeitsgrades als „unscharfer Wahrheitswert“ nur bedingt richtig ist: bei einem völlig defekten Motor, der gar kein Geräusch mehr von sich gibt, wäre aufgrund der Regel 1) der Gültigkeitsgrad der Aussage „Der Motor ist defekt“ gleich Null. Ein niedriger Gültigkeitsgrad bedeutet also nicht, daß eine Aussage falsch ist, sondern nur, daß man aus den vorhandenen Regeln keine genaue Information über ihre Wahrheit gewinnen kann.

---

<sup>8)</sup> Der Gültigkeitsgrad der Vorbedingung einer Regel wird manchmal auch als *Feuerstärke (firing strength)* der Regel bezeichnet.

### 18.2.4 Logische Operatoren

Mit den logischen Operatoren UND und ODER kann man aus mehreren Aussagen Verbundaussagen bilden:

$$X \text{ ist } A \quad \text{UND} \quad Y \text{ ist } B$$

bzw.

$$X \text{ ist } A \quad \text{ODER} \quad Y \text{ ist } B$$

Eine einfache Möglichkeit zur Auswertung solcher Verbundaussagen besteht darin, das logische UND durch den Minimumoperator und das logische ODER durch den Maximumoperator darzustellen:

**Beispiel 18.5** Eine Regel für den Betrieb einer Anlage laute: „Wenn die Temperatur normal und der Kühlwasserdurchfluß hoch ist, dann ist der Prozeßzustand sicher“. „Temperatur“, „Kühlwasserdurchfluß“ und „Prozeßzustand“ seien linguistische Variable, und es sei  $\mu_{\text{normal}}(\vartheta) = 0,9$  und  $\mu_{\text{hoch}}(Q_K) = 0,3$ . Dann ergibt sich der Gültigkeitsgrad der Aussage „Die Temperatur ist normal UND der Kühlwasserdurchfluß ist hoch“ als das Minimum der Gültigkeitsgrade der beiden Teilaussagen zu 0,3. Damit erhält auch die Aussage „Der Prozeßzustand ist sicher“ den Gültigkeitsgrad 0,3.

Anmerkungen:

- Wie man an diesem Beispiel sieht, zeigt der Minimumoperator ein „pessimistisches“ Verhalten: Die am schlechtesten erfüllte Voraussetzung ist maßgeblich für die Folgerung.
- Wenn man nur 0 und 1 als Gültigkeitsgrade zuläßt und diese als FALSCH und WAHR interpretiert, erhält man mit dem Minimumoperator die gleichen Ergebnisse wie mit der UND–Verknüpfung der klassischen Logik.
- Die UND–Verknüpfung ist eng mit der Schnittmengenbildung verwandt. Der Unterschied besteht darin, daß man die Schnittmenge nur von zwei Mengen auf derselben Grundmenge bilden kann, aber nicht z. B. von der Menge der normalen Temperaturen und der hohen Kühlwasserdurchflüsse.

#### *t*–Normen und *s*–Normen

Die Darstellung des logischen UND durch den Minimumoperator ist durchaus nicht die einzige Möglichkeit. Wenn man überlegt, welche Bedingungen ein Operator erfüllen muß, um sinnvollerweise als logisches UND bzw. Durchschnittsoperator verwendet zu werden, gelangt man zu einer Klasse von Operatoren, die als *t*–Normen (von *triangular norm*) bezeichnet werden.

**Definition 18.18** *t*–Normen sind zweiwertige Funktionen  $t(x, y)$  mit  $x, y \in [0, 1]$ , die folgende Bedingungen erfüllen:

- $t(0, 0) = 0; \quad t(x, 1) = t(1, x) = x \quad \text{für alle } x \in [0, 1]$
- $t(x, y) \leq t(u, v)$  falls  $x \leq u$  und  $y \leq v$  (*Eigenschaft der Monotonie*)
- $t(x, y) = t(y, x)$  (*Kommutativgesetz*)
- $t(x, t(y, z)) = t(t(x, y), z)$  (*Assoziativgesetz*; diese Eigenschaft ist notwendig, um die UND-Verknüpfung von drei oder mehr Teilaussagen auswerten zu können.)

Beispiele für  $t$ -Normen:

- Minimumoperator (*minimum operator, Zadeh AND*)  $\min(x, y)$
- Hamacher-Produkt  $t_H(x, y) := \frac{xy}{x + y - xy}$
- Algebraisches Produkt (*algebraic product, probabilistic AND*)  $x \cdot y$
- Einstein-Produkt  $t_E(x, y) := \frac{xy}{2 - (x + y - xy)}$
- Beschränktes Produkt (*bounded product, Lukasiewicz AND; auch: bounded difference*)  $x \odot y := \max(0, x + y - 1)$
- Drastisches Produkt (*drastic product*)  $x \cap y := \begin{cases} x & \text{für } y = 1 \\ y & \text{für } x = 1 \\ 0 & \text{sonst.} \end{cases}$

In den Abbildungen 18.14 bis 18.19 sind diese  $t$ -Normen grafisch dargestellt. Funktionen, die zur Darstellung des logischen ODER geeignet sind, heißen  $s$ -Normen oder *Co-t-Normen*:

**Definition 18.19** *s-Normen* sind zweiwertige Funktionen  $s(x, y)$  mit  $x, y \in [0, 1]$ , die folgende Bedingungen erfüllen:

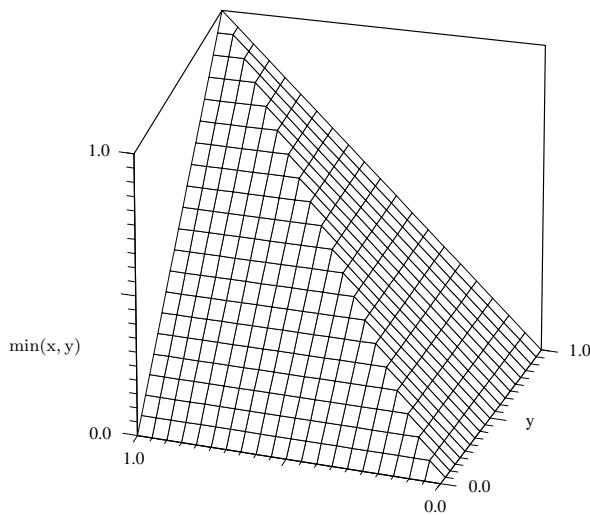
- $s(1, 1) = 1; \quad s(x, 0) = s(0, x) = x \quad \text{für alle } x \in [0, 1]$
- $s(x, y) \leq s(u, v)$  falls  $x \leq u$  und  $y \leq v$  (*Eigenschaft der Monotonie*)
- $s(x, y) = s(y, x)$  (*Kommutativgesetz*)
- $s(x, s(y, z)) = s(s(x, y), z)$  (*Assoziativgesetz*).

Zu jeder  $t$ -Norm lässt sich nach der Beziehung

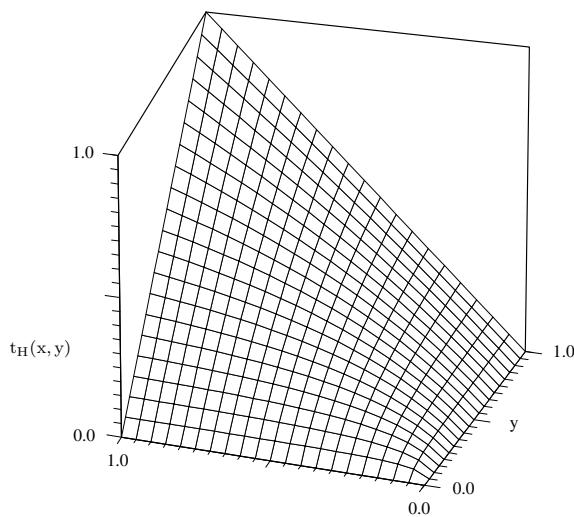
$$s(x, y) = 1 - t((1 - x), (1 - y))$$

eine gewissermaßen „spiegelbildliche“  $s$ -Norm konstruieren.

Beispiele für  $s$ -Normen:



**Abb. 18.14:** Minimumoperator



**Abb. 18.15:** Hamacher–Produkt

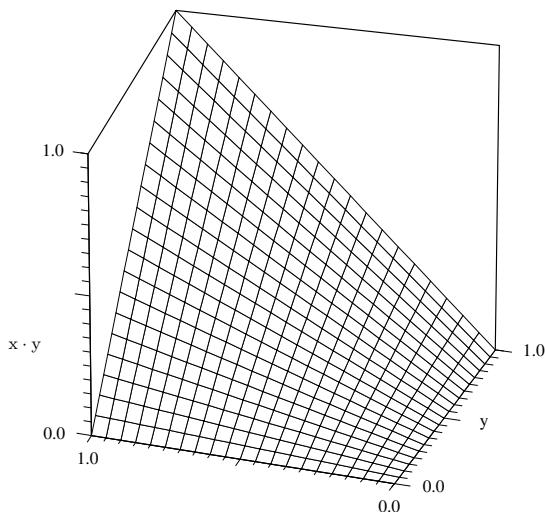


Abb. 18.16: Algebraisches Produkt

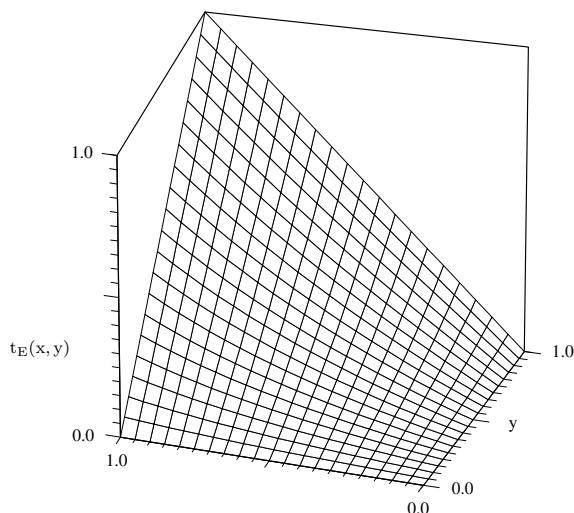
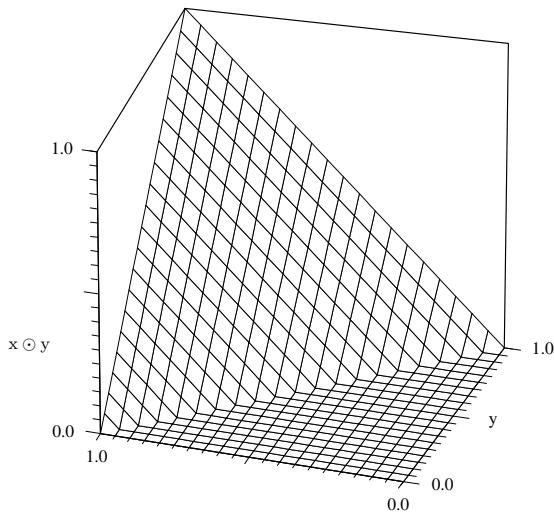
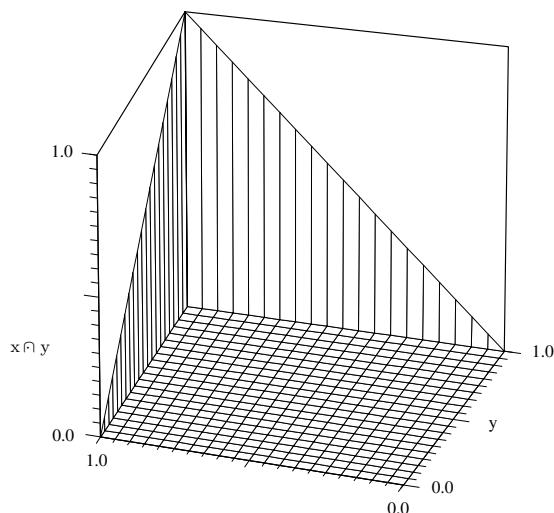


Abb. 18.17: Einstein-Produkt



**Abb. 18.18:** Beschränktes Produkt



**Abb. 18.19:** Drastisches Produkt

- Maximumoperator (*maximum operator, Zadeh OR*)  $\max(x, y)$
- Hamacher–Summe  $s_H(x, y) := \frac{x + y - 2xy}{1 - xy}$
- Algebraische Summe (*algebraic sum, probabilistic OR*)  $x\hat{+}y := x + y - xy$
- Einstein–Summe  $s_E(x, y) := \frac{x + y}{1 + xy}$
- Beschränkte Summe (*bounded sum, Lukasiewicz OR*)  $x\oplus y := \min(1, x + y)$
- Drastische Summe (*drastic sum*)  $x\cup y := \begin{cases} x & \text{für } y = 0 \\ y & \text{für } x = 0 \\ 1 & \text{sonst.} \end{cases}$

Die Abbildungen 18.20 bis 18.25 zeigen diese  $s$ –Normen in grafischer Darstellung.

### Weitere logische Operatoren

Eine Verallgemeinerung der oben beschriebenen  $t$ – bzw.  $s$ –Normen stellen Operatoren dar, deren Verhalten sich durch Variation eines Parameters stufenlos verändern lässt (*parametrierbare Operatoren*). Als Beispiele sollen hier der Yager–Durchschnittsoperator und der Hamacher–Durchschnittsoperator vorgestellt werden. Der Yager–Durchschnittsoperator ist definiert als

$$t_Y(x, y) := 1 - \min\left(1, \left((1 - x)^p + (1 - y)^p\right)^{1/p}\right)$$

Durch Variation des Parameters  $p$  im Bereich von 1 bis  $\infty$  wird das Verhalten des Operators eingestellt. Für  $p = 1$  ist der Yager–Operator gleich dem beschränkten Produkt, für  $p \rightarrow \infty$  gleich dem Minimumoperator<sup>9)</sup> (siehe Abb. 18.26 und 18.27).

Die Definition des Hamacher–Durchschnittsoperators lautet

$$t_{H\gamma}(x, y) := \frac{xy}{\gamma + (1 - \gamma)(x + y - xy)}$$

Für  $\gamma = 0$  ist der Hamacher–Durchschnittsoperator gleich dem Hamacher–Produkt, für  $\gamma = 1$  gleich dem algebraischen Produkt und für  $\gamma \rightarrow \infty$  gleich dem drastischen Produkt (siehe Abb. 18.28 und 18.29).

Alle bislang vorgestellten UND–Operatoren sind nicht kompensatorisch. Das bedeutet, daß der Gültigkeitsgrad der Verbundaussage immer kleiner oder gleich dem kleineren der beiden Gültigkeitsgrade der Teilaussagen ist. Dieses „pessimistische“ Verhalten ist z. B. bei sicherheitsrelevanten Anwendungen (Beispiel 18.5) durchaus berechtigt. In vielen Fällen wird aber das Wort „und“ umgangssprachlich so verwendet, daß eine Kompensation zwischen den Gültigkeitsgraden der beiden Teilaussagen sinnvoll ist.

---

<sup>9)</sup> Dies erkennt man, wenn man beachtet, daß für  $p \rightarrow \infty$  der Term  $(1 - y)^p$  gegenüber dem Term  $(1 - x)^p$  vernachlässigt werden kann, falls  $x < y$  ist und umgekehrt.

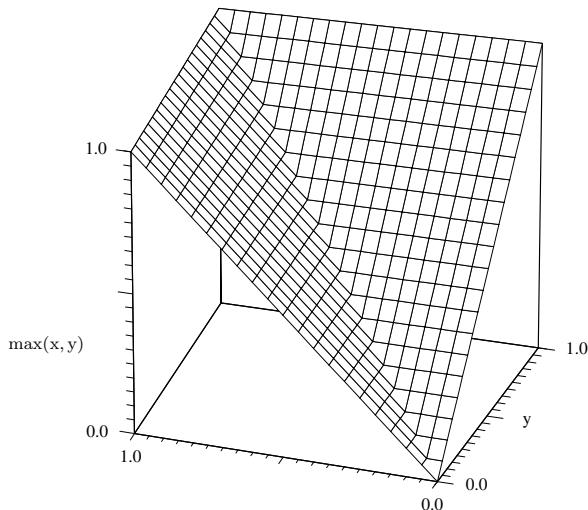


Abb. 18.20: Maximumoperator

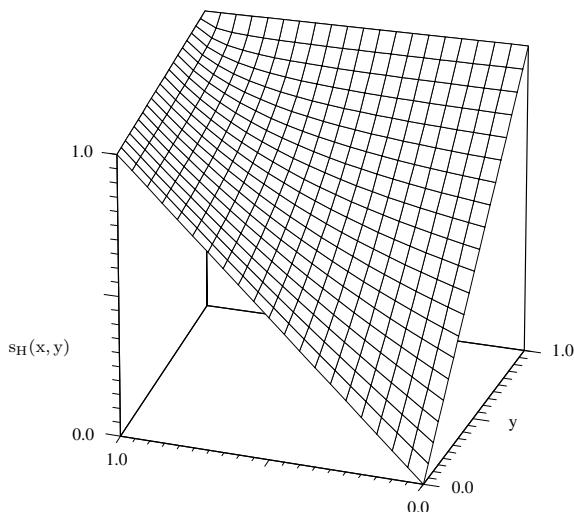
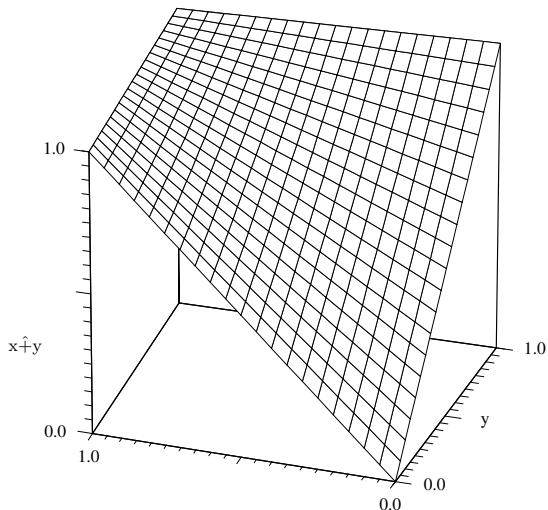
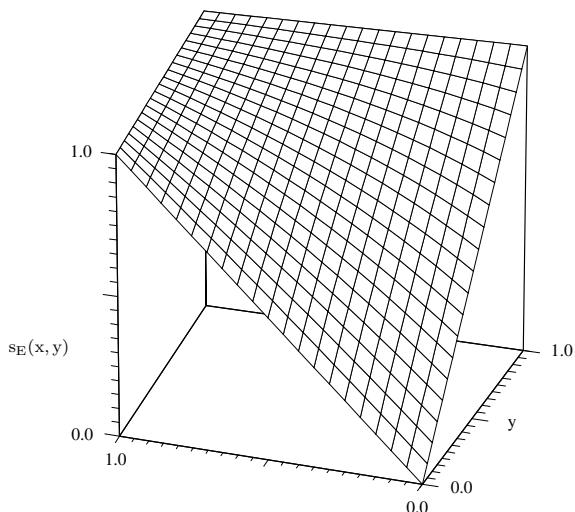


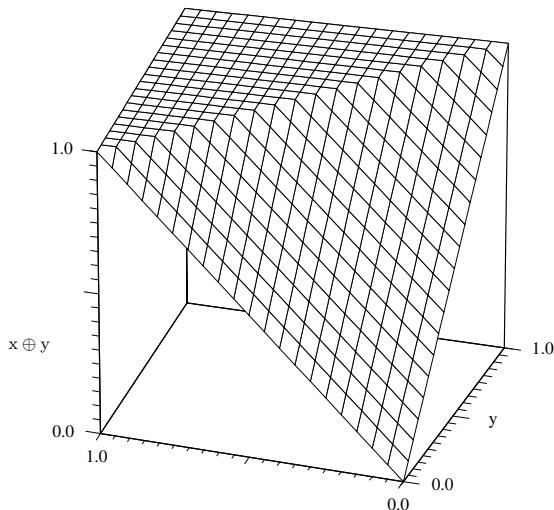
Abb. 18.21: Hamacher–Summe



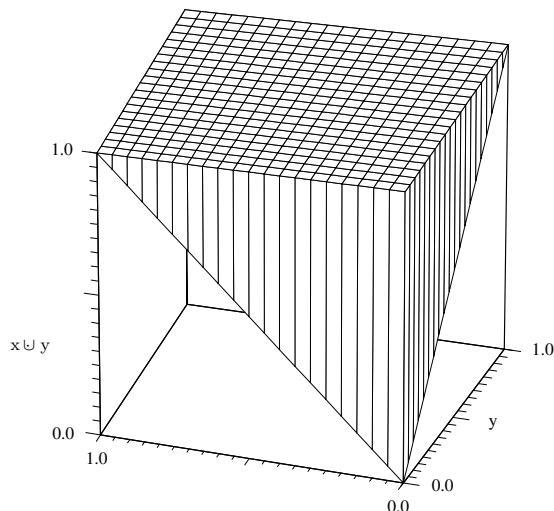
**Abb. 18.22:** Algebraische Summe



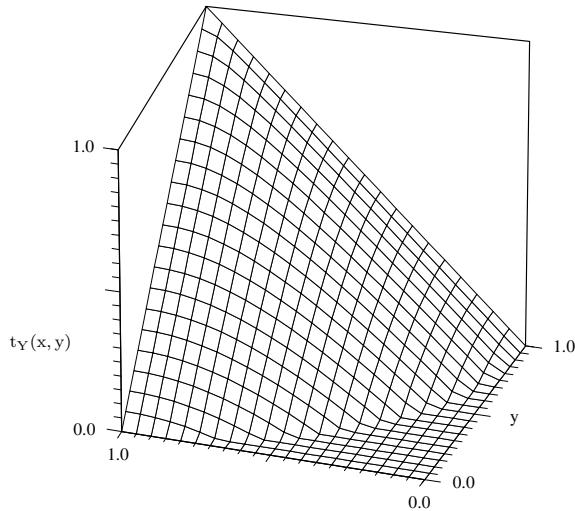
**Abb. 18.23:** Einstein-Summe



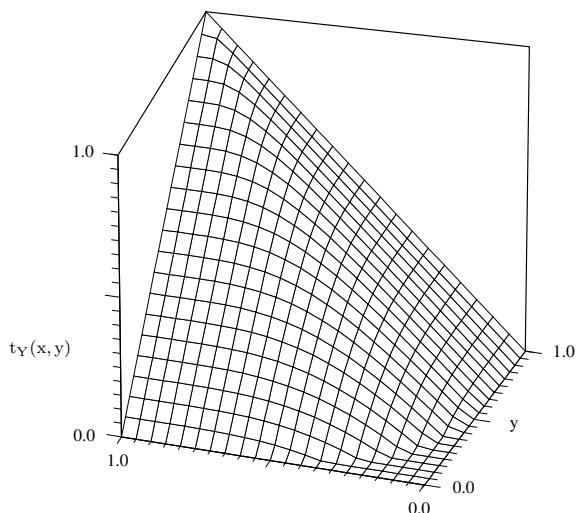
**Abb. 18.24:** Beschränkte Summe



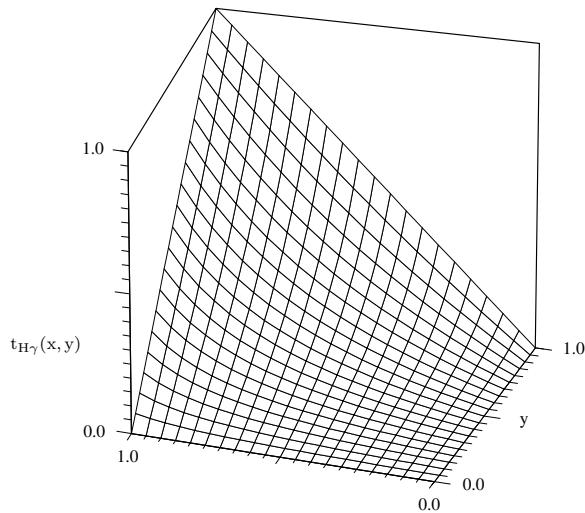
**Abb. 18.25:** Drastische Summe



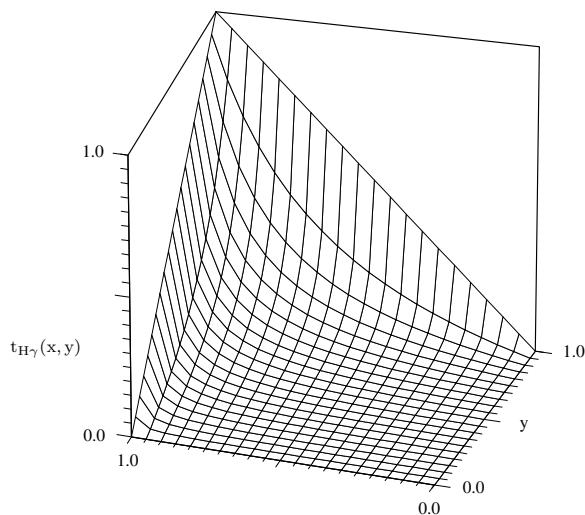
**Abb. 18.26:** *Yager-Durchschnittsoperator,  $p = 2,0$*



**Abb. 18.27:** *Yager-Durchschnittsoperator,  $p = 4,0$*



**Abb. 18.28:** Hamacher–Durchschnittsoperator,  $\gamma = 5,0$



**Abb. 18.29:** Hamacher–Durchschnittsoperator,  $\gamma = 50$

**Beispiel 18.6** Ein Kunde stellt beim Autokauf die Anforderung: „Ein gutes Auto muß zuverlässig UND preiswert sein“. Der Kunde hat zwei Autos zur Auswahl. Das erste kann mit einem Gültigkeitsgrad von 0,6 als zuverlässig und mit einem Gültigkeitsgrad von 0,55 als preiswert bezeichnet werden; das zweite mit einem Gültigkeitsgrad von 0,9 als zuverlässig und mit einem Gültigkeitsgrad von 0,5 als preiswert. Vermutlich wird der Kunde das zweite Auto als „besser“ einstufen, da er der Meinung ist, daß der geringe Nachteil beim Preis durch den großen Vorteil bei der Zuverlässigkeit kompensiert wird.

Drei wichtige kompensatorische Operatoren sind das *Fuzzy-UND (Minimum-Durchschnitts-Operator)*, der *Min-Max-Operator* und der  $\gamma$ -*Operator*. Alle drei sind parametrierbare Operatoren.

Das Fuzzy-UND ist eine Kombination aus Minimumoperator und arithmetischem Mittel:

$$c_{FU}(x, y) := \gamma \min(x, y) + (1 - \gamma) \frac{x + y}{2}$$

Der Parameter  $\gamma$  gibt den Grad der Kompensation an: für  $\gamma = 1$  (keine Kompensation) ist das Fuzzy-UND gleich dem Minimumoperator, für  $\gamma = 0$  (volle Kompensation) gleich dem arithmetischen Mittel. In Abb. 18.30 bis 18.32 ist das Verhalten des Fuzzy-UND für drei verschiedene Werte von  $\gamma$  grafisch dargestellt.

Durch Kombination des arithmetischen Mittels mit dem Maximumoperator in analoger Weise erhält man das *Fuzzy-ODER*. Fuzzy-UND und Fuzzy-ODER können zum Min-Max-Operator zusammengefaßt werden:

$$c_{MM}(x, y) := \gamma \min(x, y) + (1 - \gamma) \max(x, y)$$

Bei diesem Operator ist also die Trennung zwischen UND und ODER aufgehoben; durch Variation des Parameters  $\gamma$  von Null bis Eins läßt er sich stufenlos von ODER bis UND verstetigen. Für  $\gamma = 0,5$  ist der Min-Max-Operator gleich dem arithmetischen Mittel.

Auch der  $\gamma$ -Operator ermöglicht beliebige Zwischenstufen zwischen UND und ODER. Die Rechenvorschrift für den  $\gamma$ -Operator lautet

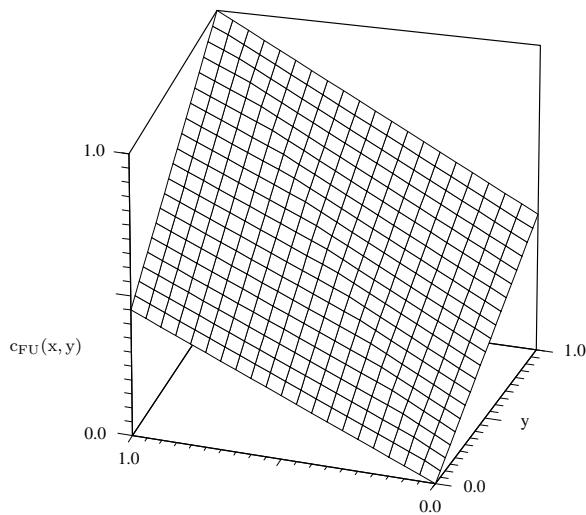
$$c_\gamma(x, y) := (xy)^{(1-\gamma)}(1 - (1-x)(1-y))^\gamma$$

Für  $\gamma = 0$  ist der  $\gamma$ -Operator gleich dem Produktoperator, für  $\gamma = 1$  gleich der algebraischen Summe (Abbildungen 18.33 bis 18.35).

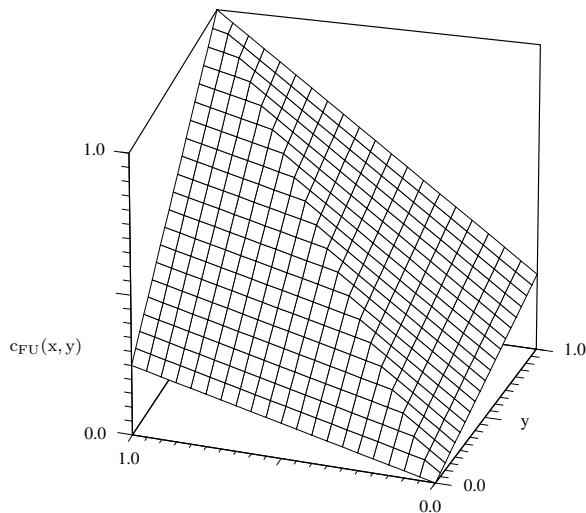
Bei der Verknüpfung von mehr als zwei Teilaussagen mit dem  $\gamma$ -Operator ist zu beachten, daß dieser Operator nicht assoziativ ist. Der  $\gamma$ -Operator kann aber auf mehrere Operanden erweitert werden, z.B.

$$c_{\gamma 3}(x, y, z) := (x \cdot y \cdot z)^{(1-\gamma)}(1 - (1-x)(1-y)(1-z))^\gamma$$

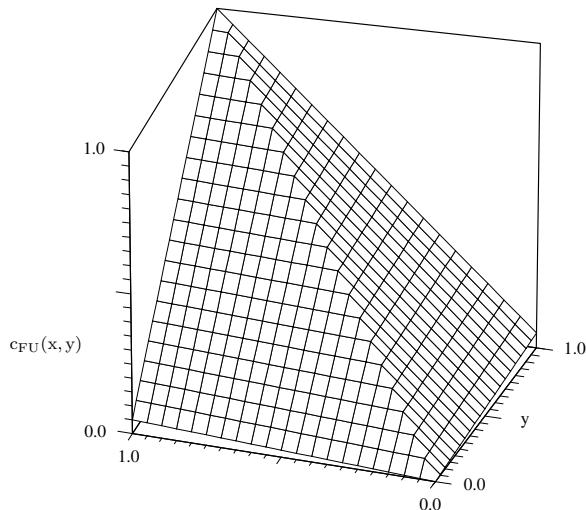
usw.



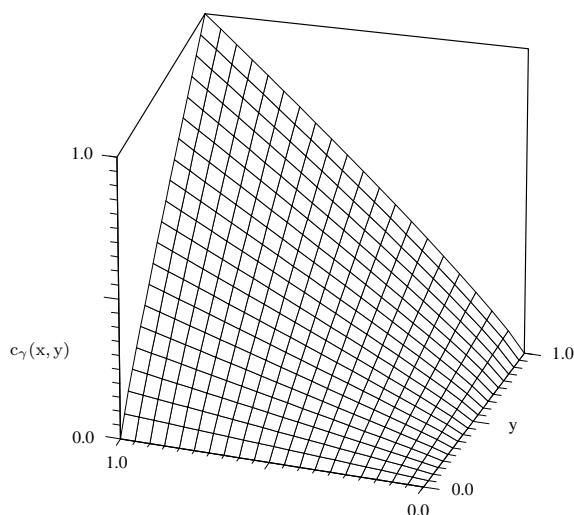
**Abb. 18.30:** *Fuzzy-UND,  $\gamma = 0, 1$*



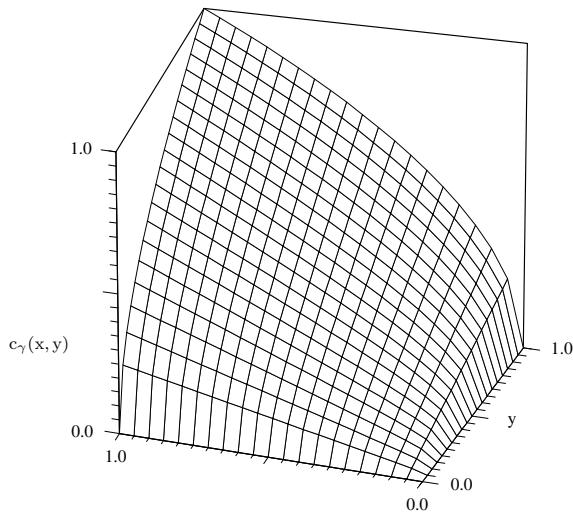
**Abb. 18.31:** *Fuzzy-UND,  $\gamma = 0, 5$*



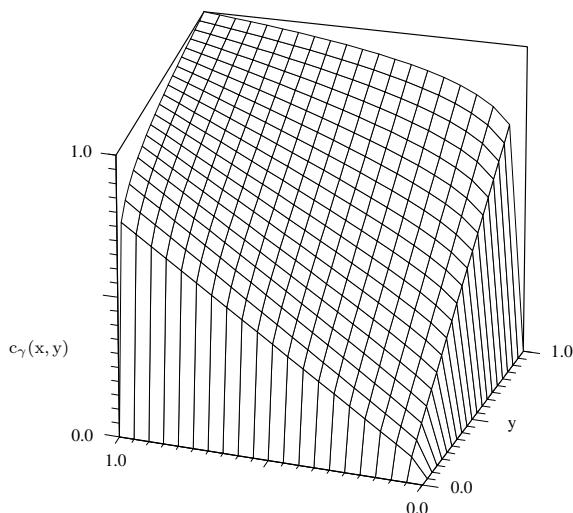
**Abb. 18.32:** *Fuzzy-UND*,  $\gamma = 0, 9$



**Abb. 18.33:**  *$\gamma$ -Operator*,  $\gamma = 0, 1$



**Abb. 18.34:**  $\gamma$ -Operator,  $\gamma = 0,5$



**Abb. 18.35:**  $\gamma$ -Operator,  $\gamma = 0,9$

### Auswahlkriterien für logische Operatoren

Wie wählt man nun unter den vielen logischen Operatoren denjenigen aus, der für eine bestimmte Aufgabenstellung am geeignetsten ist? Die meisten in der Praxis eingesetzten Fuzzy–Regler verwenden für das logische UND den Minimumoperator und für das logische ODER den Maximumoperator. Diese Operatorenkombination geht bereits auf den ersten Artikel über Fuzzy–Logik von Zadeh [246] zurück. Sie benötigt den geringsten Rechenaufwand, was in der Regelungstechnik einen entscheidenden Vorteil darstellt. Ein anderes wichtiges Kriterium bei Fuzzy–Reglern ist der Einfluß der verwendeten Operatoren auf das Übertragungsverhalten. Unter diesem Gesichtspunkt ist die Kombination Produkt/beschränkte Summe vorteilhaft; sie benötigt ebenfalls nur geringen Rechenaufwand. Daneben gibt es eine Reihe weiterer Auswahlkriterien für logische Operatoren; in der Literatur ([250], [224]) werden u.a. noch folgende genannt:

- **Adaptierbarkeit:** Bei Systemen, die universell einsetzbar sein sollen, ist die Verwendung parametrierbarer Operatoren vorteilhaft. Man muß nur wenige Operatoren implementieren, kann aber durch Variation der Parameter das Verhalten der jeweiligen Problemstellung anpassen.
- **Kompensation:** Ob ein kompensatorisches Verhalten der Operatoren sinnvoll ist oder nicht, hängt von der jeweiligen Aufgabenstellung ab.
- **Nachbildung der Realität bzw. menschlicher Denkweisen:** In manchen Anwendungen soll die Fuzzy–Logik zur Modellierung menschlichen Verhaltens benutzt werden. Z. B. werden Fuzzy–Regler für komplexe Anlagen entworfen, indem man einen erfahrenen Anlagenfahrer befragt, nach welchen Regeln er die Anlage bedient [135]. Für solche Anwendungen ist die Übereinstimmung der verwendeten Operatoren mit den umgangssprachlichen Vorstellungen von „und“ und „oder“ ein wichtiges Kriterium. Als besonders geeignet haben sich dafür in empirischen Untersuchungen der  $\gamma$ –Operator, das Fuzzy–UND und das Fuzzy–ODER erwiesen.

## 18.3 Grundlagen der Fuzzy–Regelung

Das Grundprinzip, nach dem die meisten in der Praxis eingesetzten Fuzzy–Regler arbeiten, wurde bereits anhand des Heizungsreglers aus Beispiel 18.3 dargestellt. Dieses Prinzip mit seinen drei Arbeitsschritten Fuzzyfizierung, Inferenz und Defuzzifizierung soll nun eingehender betrachtet werden. Vorher soll aber noch kurz auf einige Modifikationen bzw. Erweiterungen dieses Konzeptes eingegangen werden. Hier sind besonders Hybrid–Regler zu erwähnen, die Fuzzy–Regelung und konventionelle Regelungsverfahren verbinden. Beispiel für ein solches Hybridkonzept ist der *Sugeno–Fuzzy–Regler*.<sup>10)</sup> Während der reine Fuzzy–Regler (zur

---

<sup>10)</sup> Benannt nach seinem Erfinder M. Sugeno.

Unterscheidung auch als *Mamdani–Regler* bezeichnet) mit Regeln der Form

Regel  $k$ ): WENN Eingangsgröße 1 = Term  $A_{k_1}$   
 UND Eingangsgröße 2 = Term  $A_{k_2}$   
 $\vdots$   
 UND Eingangsgröße  $n$  = Term  $A_{k_n}$   
 DANN Stellgröße = Term  $B_l$

arbeitet, verwendet der Sugeno–Regler Regeln der Form

Regel  $k$ ): WENN Eingangsgröße 1 = Term  $A_{k_1}$   
 UND Eingangsgröße 2 = Term  $A_{k_2}$   
 $\vdots$   
 UND Eingangsgröße  $n$  = Term  $A_{k_n}$   
 DANN  $y = f_l(x_1, x_2, \dots, x_n)$  .

Der Sugeno–Regler überlagert also mit Hilfe von Fuzzy–Regeln die Ergebnisse mehrerer konventioneller Regelgesetze  $y = f_l(x_1, x_2, \dots, x_n)$ .

Ein weiteres Hybridkonzept ist die Einstellung bzw. Adaption von konventionellen Reglern mit Hilfe der unscharfen Logik. Dazu werden beispielsweise Regeln der Form

Regel  $k$ ): WENN Eingangsgröße 1 = Term  $A_{k_1}$   
 UND Eingangsgröße 2 = Term  $A_{k_2}$   
 $\vdots$   
 UND Eingangsgröße  $n$  = Term  $A_{k_n}$   
 DANN Proportional–Verstärkung = Term  $B_l$

verwendet; näheres zu diesem Verfahren s. z.B. [178], [249].

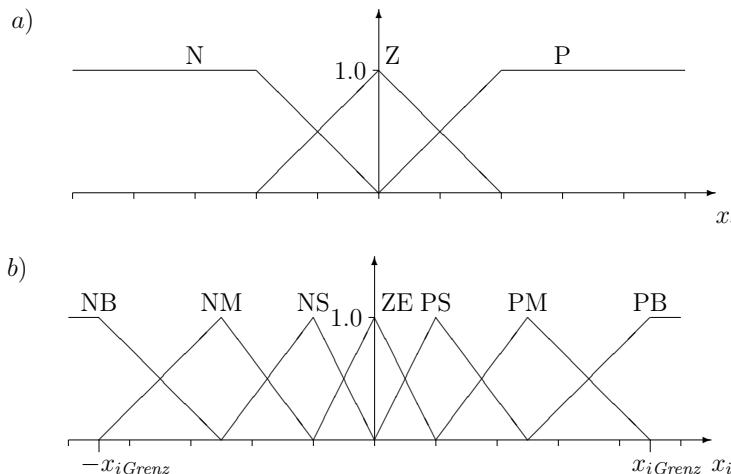
Eine weitergehende Entwicklung stellen *lernfähige* bzw. *adaptive* Fuzzy–Regler dar, bei denen Regeln und/oder Bedeutungsmengen automatisch generiert werden. Dabei kann eine Vielzahl von Adoptionsverfahren eingesetzt werden, z.B. Kombination von Fuzzy–Reglern mit Neuronalen Netzen („Neuro–Fuzzy“) [14] [118], Fuzzy–Adoptionsregeln [146], stabile Adoptionsgesetze nach Ljapunow [236] oder genetische Algorithmen [33]. Für Beschreibungen der einzelnen Konzepte sei auf die Literatur verwiesen.

In den folgenden Abschnitten sollen nun die Arbeitsschritte eines Fuzzy–Reglers — Fuzzifizierung, Inferenz und Defuzzifizierung — näher betrachtet werden. Anschließend wird noch auf das Übertragungsverhalten sowie auf Entwurf und Realisierung von Fuzzy–Reglern eingegangen.

### 18.3.1 Fuzzyfizierung

Die Fuzzyfizierung ist die Umsetzung eines scharfen Eingangswertes in eine unscharfe Beschreibung dieses Wertes. Die Vorgehensweise ist, wie im Beispiel 18.3 gezeigt wurde, sehr einfach: dem Eingangswert  $x$  werden die Zugehörigkeitsgrade  $\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)$  zugeordnet. Nach Satz 18.7 sind diese Zugehörigkeitsgrade die Gültigkeitsgrade der Aussagen „ $X$  ist  $A_1$ “ … „ $X$  ist  $A_n$ “.

Die Festlegung der Terme  $A_1 \dots A_n$  und ihrer Bedeutungsmengen ist eine wichtige Teilaufgabe beim Entwurf eines Fuzzy-Reglers und hat einen großen Einfluss auf das Reglerverhalten. Als erstes ist die Anzahl der Terme festzulegen, je nachdem wie grob oder fein der Wertebereich eingeteilt werden soll. Abb. 18.36 zeigt ein Beispiel für eine grobe Einteilung mit drei Termen (a) und eine feine Einteilung mit sieben Termen (b). Die Terme werden häufig mit den Abkürzungen NB, NM, NS, ZE, … für „negative big“, „negative medium“, „negative small“, „zero“ usw. bezeichnet. In der Praxis hat sich eine Einteilung in drei bis elf Terme als sinnvoll erwiesen; in Bereichen, in denen das Verhalten des Systems feinfühliger beeinflusst werden soll (z.B. um den Nullpunkt herum), können die Bedeutungsmengen dichter angeordnet werden (s. Abb. 18.36 (b)).



**Abb. 18.36:** Fuzzyfizierung mit größerer a) und feinerer b) Termeinteilung

Die zweite Entscheidung betrifft den Typ der verwendeten Zugehörigkeitsfunktionen. Meistens werden trapezförmige oder dreieckige Zugehörigkeitsfunktionen verwendet; es sind aber auch viele andere Typen möglich (siehe Abb. 18.37). In der Literatur werden dreieckige Zugehörigkeitsfunktionen wegen ihrer optischen Ähnlichkeit mit dem griechischen Großbuchstaben  $\Lambda$  manchmal als  $\Lambda$ -(*Lambda*-) Typ bezeichnet. Trapezförmige Zugehörigkeitsfunktionen heißen auch

$\Pi$ -*Typ*; auf der linken Seite abgeschnittene  $\Pi$ -Funktionen werden als  $Z$ -*Typ* bezeichnet (die dick gezeichnete Kurve in Abb. 18.37 hat gewisse Ähnlichkeit mit einem Z) und auf der rechten Seite abgeschnittene  $\Pi$ -Funktionen als  $S$ -*Typ*.

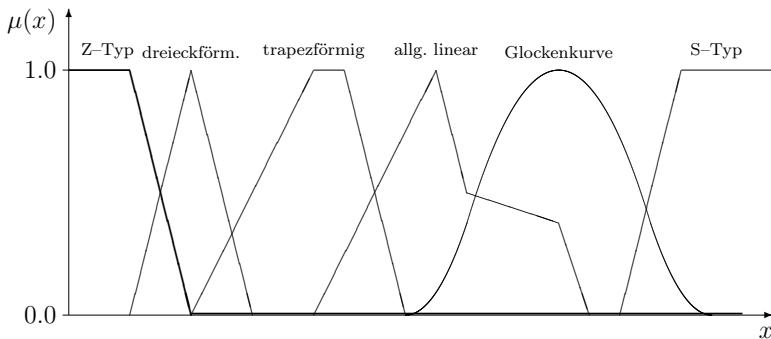


Abb. 18.37: Typen von Zugehörigkeitsfunktionen

Als dritte Entscheidung ist die Anordnung der Bedeutungsmengen festzulegen. Die Bedeutungsmengen müssen den gesamten zulässigen Wertebereich der Basisvariablen überdecken. Existieren Lücken, in denen alle Zugehörigkeitsfunktionen Null sind, enthält die linguistische Variable für den entsprechenden Bereich keine sinnvolle Information und kann daher auch nicht sinnvoll weiterverarbeitet werden. Es ist weiterhin zweckmäßig, die Zugehörigkeitsfunktionen so festzulegen, daß sich benachbarte Bedeutungsmengen überlappen; d. h. es sollte zwischen zwei Termen einen größeren Bereich geben, in dem die Zugehörigkeitsfunktionen für beide Terme  $> 0$  sind. Dadurch entstehen gleitende Übergänge zwischen den Termen. Wenn sich die Bedeutungsmengen nicht überlappen, reduziert sich die Fuzzyfizierung im wesentlichen auf ein „Umschalten“ zwischen den einzelnen Termen bei bestimmten Grenzwerten.

Eine Besonderheit bei Fuzzy-Reglern besteht darin, daß die Eingangsgrößen gewissermaßen „begrenzt“ werden: Da man in der Praxis nur endlich viele Terme verwenden kann, müssen die Bedeutungsmengen so festgelegt werden, daß oberhalb eines gewissen Grenzwertes  $x_{iGrenz}$  der Zugehörigkeitsgrad zur „positivsten“ Bedeutungsmenge (in Abb. 18.36 (b) „positive big“) gleich Eins und zu allen anderen Bedeutungsmengen gleich Null ist. (Für negative Werte der Eingangsgröße gilt entsprechendes.) Ist nun die Eingangsgröße  $x_i$  größer als  $x_{iGrenz}$ , haben Änderungen von  $x_i$  keinen Einfluß auf die Reglerausgangsgröße mehr. Dies muß beim Reglerentwurf berücksichtigt werden, z.B. indem  $x_{iGrenz}$  auf den maximal möglichen Wert von  $x_i$  gesetzt wird.

### 18.3.2 Inferenz

Der zweite Arbeitsschritt des Fuzzy–Reglers ist die Inferenz, d. h. das Anwenden der Regeln auf die unscharfen Eingangsgrößen. Meistens sind die Regeln von der Form

Regel  $k$ ): WENN Eingangsgröße 1 = Term  $A_{k_1}$   
 UND Eingangsgröße 2 = Term  $A_{k_2}$   
 :  
 UND Eingangsgröße  $n$  = Term  $A_{k_n}$   
 DANN Stellgröße = Term  $B_l$  .

Die Vorbedingung einer Regel stellt also die verbale Beschreibung eines bestimmten Streckenzustandes dar, die Schlussfolgerung die gewünschte Reaktion auf diesen Zustand.

Existieren mehrere Stellgrößen, so lauten die Schlussfolgerungen

:  
 DANN Stellgröße 1 = Term  $B_{l_1}$ , Stellgröße 2 = Term  $B_{l_2}$ ,  
 ..., Stellgröße  $m$  = Term  $B_{l_m}$  .

Die Inferenz erfolgt in diesem Falle so, als ob für jede der Stellgrößen ein eigener Fuzzy–Regler vorhanden wäre; ein Fuzzy–Regler mit  $m$  Ausgängen kann also in  $m$  Fuzzy–Regler mit je einem Ausgang zerlegt werden.

Prinzipiell können in den Regelvorbedingungen beliebige logische Operatoren (UND, ODER, NICHT) verwendet werden; in den meisten Fällen beschränkt man sich aber auf einfache UND–Verknüpfungen. Bei der Inferenz wird dann zunächst für jede Regel mit Hilfe eines UND–Operators der Gültigkeitsgrad der Vorbedingung berechnet (*aggregation*). Anschließend werden diese Gültigkeitsgrade für alle Regeln, die die gleiche Schlussfolgerung haben, mit einem ODER–Operator verknüpft:

**Beispiel 18.7** Die Regelbasis<sup>11)</sup> eines Fuzzy–Reglers zur Lageregelung eines elektrischen Antriebes enthalte die folgenden Regeln:

---

<sup>11)</sup> Die Gesamtheit aller Regeln wird als *Regelbasis* (*rule base*) oder *Regelsatz* (*rule set*) bezeichnet.

:

Regel 4): WENN der Abstand klein ist  
 UND die Geschwindigkeit mittelgroß ist  
 DANN muß der Drehzahlsollwert klein sein .

:

Regel 7): WENN der Abstand mittelgroß ist  
 UND die Geschwindigkeit klein ist  
 DANN muß der Drehzahlsollwert klein sein .

:

Wie man sieht, haben Regel 4) und Regel 7) dieselbe Folgerung: „Der Drehzahlsollwert muß klein sein“. Alle Regeln mit derselben Folgerung können unter Verwendung des logischen ODER zu einer Regel zusammengefaßt werden:

WENN (der Abstand klein ist UND die Geschwindigkeit mittelgroß ist)  
 ODER (der Abstand mittelgroß ist UND die Geschwindigkeit klein ist)  
 DANN muß der Drehzahlsollwert klein sein .

Diese Regel kann nun nach den Gesetzen der unscharfen Logik ausgewertet werden.

In manchen Anwendungen existieren Regeln, die nicht hundertprozentig gültig sind, oder einige Regeln haben ein stärkeres Gewicht als andere. In solchen Fällen wird ein erweitertes Inferenzverfahren angewandt: Jede Regel bekommt einen *Plausibilitätsgrad* (*degree of support*) von Null bis Eins zugewiesen; dieser Plausibilitätsgrad läßt sich als Zugehörigkeitsgrad der Regel zur Menge der absolut gültigen Regeln interpretieren. Der Plausibilitätsgrad wird bei der Inferenz berücksichtigt, indem er mit der Vorbedingung der Regel UND-verknüpft wird (*composition*). Hierfür kann ein anderer UND-Operator verwendet werden als für die Verknüpfung der Teilaussagen.

Es ist problemlos möglich, mehrstufige Reglerstrukturen aufzubauen, bei denen die Schlußfolgerung einer Regel in die Vorbedingung anderer Regeln eingeht. Im Beispiel 18.7 könnte z. B. der Drehzahlsollwert als Eingangsgröße für weitere Regeln dienen:

⋮

Regel 4): WENN der Abstand klein ist  
                   UND die Geschwindigkeit mittelgroß ist  
                   DANN muß der Drehzahlsollwert klein sein.

⋮

Regel 14) WENN der Drehzahlsollwert klein ist  
                   UND die Ist-Drehzahl klein ist  
                   DANN muß der Stromsollwert klein sein.

⋮

Bei Reglern mit mehr als zwei Eingangsgrößen ist ein solcher Aufbau vorteilhaft, da er die Regeln übersichtlicher macht.

### 18.3.3 Defuzzyfizierung

Als Ergebnis der Inferenz liegen die Werte der Ausgangsgrößen in unscharfer Form vor, d. h. als Gültigkeitsgrade von unscharfen Aussagen. Zur Ansteuerung eines Stellgliedes müssen diese Gültigkeitsgrade wieder in scharfe Werte überetzt werden; dieser Schritt heißt Defuzzyfizierung. Hierfür existieren im wesentlichen drei Verfahren: das *Flächenschwerpunktverfahren* (*center-of-area*, CoA), das *Maximumsmittelwertverfahren* (*mean-of-maxima*, MoM) und die *Maximumschwerpunktsmethode* (*center-of-maxima*, CoM).

Beim CoA-Verfahren und beim MoM-Verfahren erfolgt die Defuzzyfizierung in zwei Schritten: im ersten Schritt werden die vorliegenden Gültigkeitsgrade in eine unscharfe Menge umgesetzt, und im zweiten Schritt wird aus dieser unscharfen Menge ein scharfer Wert berechnet. Beim CoM-Verfahren werden dagegen die Gültigkeitsgrade direkt in einem Schritt in einen scharfen Ausgangswert umgerechnet.

Für die Umsetzung der Gültigkeitsgrade in eine unscharfe Menge bei CoA und MoM gibt es zwei Verfahren: *Clipping* und *Scaling*. Bei beiden Verfahren müssen für die Terme der Ausgangsgröße Bedeutungsmengen definiert sein.

Das Clipping-Verfahren funktioniert folgendermaßen: Es sei  $Y$  die linguistische Ausgangsvariable mit den Termen  $B_1 \dots B_n$  und der Basisvariablen  $y$ . Die Gültigkeitsgrade der Aussagen „ $Y$  ist  $B_i$ “ die bei der Inferenz ermittelt wurden, sollen mit  $G_{B_i}$  bezeichnet werden. Durch das Clipping werden diese Gültigkeitsgrade in eine unscharfe Menge  $B'$  umgewandelt. Die Zugehörigkeitsfunktion dieser unscharfen Menge wird punktweise für jedes  $y$  nach der Formel

$$\mu_{B'}(y) = \max(\min(\mu_{B_1}(y), G_{B_1}), \dots, \min(\mu_{B_n}(y), G_{B_n}))$$

berechnet. Grafisch bedeutet dies, daß die Zugehörigkeitsfunktionen der Bedeutungsmengen beim jeweiligen Gültigkeitsgrad „abgeschnitten“ werden und dann aus diesen abgeschnittenen Mengen die Vereinigungsmenge gebildet wird (siehe Abb. 18.39).

Beim Scaling ist

$$\mu_{B'}(y) = \max((\mu_{B_1}(y) \cdot G_{B_1}), \dots, (\mu_{B_n}(y) \cdot G_{B_n}))$$

das bedeutet grafisch, daß die Bedeutungsmengen mit dem jeweiligen Gültigkeitsgrad „gestaucht“ werden (siehe Abb. 18.40). Clipping bzw. Scaling werden auch als *Max-Min-Inferenz* bzw. *Max-Prod-Inferenz* bezeichnet.<sup>12)</sup>.

Als nächster Schritt erfolgt die eigentliche Defuzzyfizierung nach dem CoA- oder MoM-Verfahren. Beide Verfahren können sowohl mit Clipping als auch mit Scaling kombiniert werden. Beim CoA-Verfahren (Flächenschwerpunktverfahren) ist der scharfe Ausgangswert gleich der Position des „Flächenschwerpunktes“ der Menge  $B'$  auf der  $y$ -Achse (s. Abb. 18.41):

$$y_0 = \frac{\int y \cdot \mu_{B'}(y) dy}{\int \mu_{B'}(y) dy}$$

Beim Maximumsmittelwertverfahren ist der Ausgangswert gleich dem Mittelwert aller globalen Maxima von  $\mu_{B'}$ . Folgende Fälle sind zu unterscheiden:

1.  $\mu_{B'}(y)$  besitzt genau ein globales Maximum bei  $y = \hat{y}$ : Dann ist  $y_0 = \hat{y}$ .
2.  $\mu_{B'}(y)$  besitzt endlich viele globale Maxima bei  $y = \hat{y}_1 \dots y = \hat{y}_n$ : Dann ist  $y_0$  der Mittelwert aus  $\hat{y}_1 \dots \hat{y}_n$ .
3.  $\mu_{B'}(y)$  besitzt globale Maxima in einem zusammenhängenden Bereich  $\hat{y}_{min} \leq y \leq \hat{y}_{max}$  (wie z. B. in Abb. 18.39 und 18.40): Dann ist  $y_0$  der Mittelwert aus den Bereichsgrenzen  $\hat{y}_{min}$  und  $\hat{y}_{max}$  (siehe Abb. 18.42).
4.  $\mu_{B'}(y)$  besitzt globale Maxima in mehreren zusammenhängenden Bereichen: Dann ist  $y_0$  der Mittelwert aus allen Bereichsgrenzen.

Als drittes Defuzzyfizierungsverfahren soll noch die Maximumschwerpunktmethode (CoM) erläutert werden. Sie zeichnet sich durch einen geringen Rechenaufwand aus und ist deshalb in der Praxis das am meisten verwendete Verfahren. Die CoM-Methode wurde bereits in Beispiel 18.3 vorgestellt: Man definiert für jeden Term  $B_i$  einen scharfen Wert  $y_{B_i}$ <sup>13)</sup>. Die Umsetzung der Gültigkeitsgrade  $G_{B_i}$  in einen scharfen Ausgangswert  $y_0$  erfolgt dann in einem Schritt nach der Formel

$$y_0 = \frac{\sum G_{B_i} y_{B_i}}{\sum G_{B_i}} .$$

Ganz ähnlich funktioniert die Defuzzyfizierung beim Sugeno-Regler (s. S. 793). Sind  $G_{f_i}$  die in der Inferenz ermittelten Gültigkeitsgrade der Aussagen

<sup>12)</sup> Genaugenommen ist die Umwandlung der Gültigkeitsgrade in eine unscharfe Menge noch Teil der Inferenz. Clipping und Scaling ergeben sich als Spezialfälle des plausiblen Schließens bei Verwendung der Mamdani- bzw. der Larsen-Implikation (siehe Anhang A 18.4).

<sup>13)</sup> Die Bedeutungsmenge von  $B_i$  ist also ein Singleton

„ $y = f_i(x_1, x_2, \dots, x_n)$ “, so ergibt sich der scharfe Ausgangswert  $y_0$  nach der Gleichung

$$y_0 = \frac{\sum G_{f_i} f_i(x_{10}, x_{20}, \dots, x_{n0})}{\sum G_{f_i}} ;$$

dabei sind  $x_{10} \dots x_{n0}$  die scharfen Werte der Reglereingangsgrößen. Es wird also aus den Ergebnissen der einzelnen Regelgesetze  $f_i$  ein gewichteter Mittelwert mit den Gewichtungsfaktoren  $G_{f_i}$  gebildet<sup>14)</sup>.

Die verschiedenen Defuzzyfizierungsverfahren sollen nun an einem Beispiel erläutert werden.

**Beispiel 18.8** Die linguistische Variable „Ventilstellung“ mit der Basisvariablen  $y$  und den Termen „geschlossen“, „halb offen“ und „offen“ ist die Ausgangsgröße eines Fuzzy-Reglers. Die Inferenz ergibt folgende Gültigkeitsgrade:  $G_{geschlossen} = 0$ ,  $G_{halboffen} = 0,5$ ,  $G_{offen} = 0,6$ . Die Bedeutungsmengen der drei Terme wurden beim Reglerentwurf gemäß Abb. 18.38 definiert.

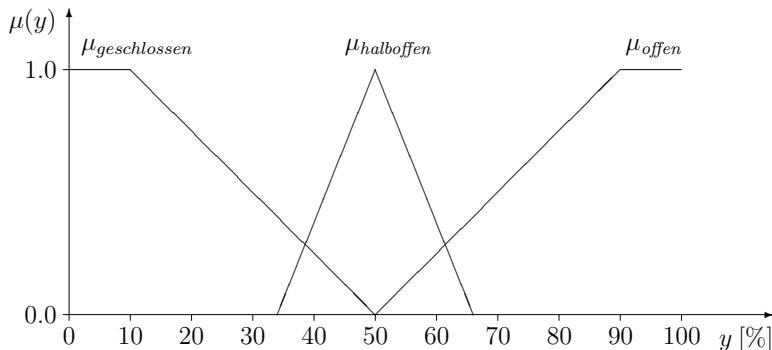


Abb. 18.38: Defuzzyfizierung der Ausgangsgröße  $y$

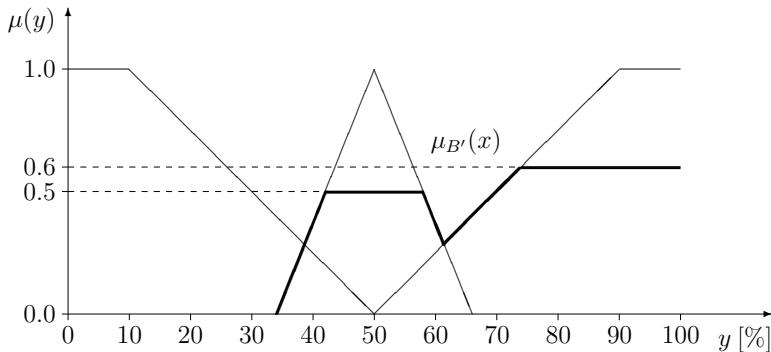
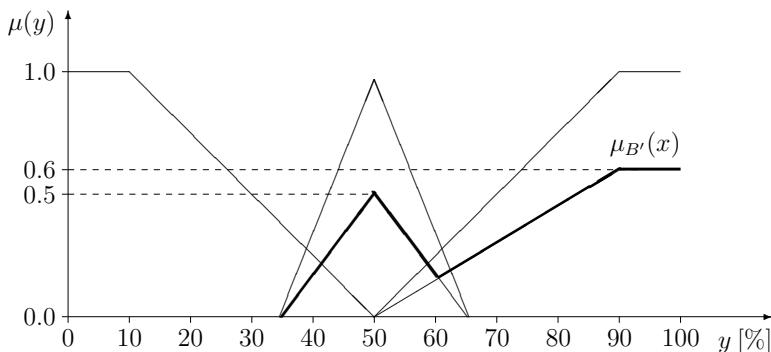
Durch Clipping bzw. Scaling erhält man aus diesen Gültigkeitsgraden die unscharfe Menge  $B'$  nach Abb. 18.39 bzw. 18.40.

Wendet man auf die durch Clipping ermittelte Menge  $B'$  das CoA-Verfahren an, erhält man den Ausgangswert  $y_0 = 70,6\%$  (Schwerpunkt der in Abb. 18.41 punktierten Fläche). Das MoM-Verfahren liefert  $y_0 = 87,0\%$ , da  $\mu_{B'}$  globale Maxima im Bereich  $74\% \leq y \leq 100\%$  besitzt (siehe Abb. 18.42).

Beim CoM-Verfahren werden die scharfen Werte  $y_{geschlossen} = 0\%$ ,  $y_{halboffen} = 50\%$  und  $y_{offen} = 100\%$  definiert. Damit ergibt sich

$$y_0 = \frac{G_{offen} \cdot y_{offen} + G_{halboffen} \cdot y_{halboffen} + G_{geschlossen} \cdot y_{geschlossen}}{G_{offen} + G_{halboffen} + G_{geschlossen}} = 79,5\%$$

<sup>14)</sup> Wie man sieht, kann der Mamdani-Regler mit CoM-Defuzzyfizierung auch als Spezialfall des Sugeno-Reglers aufgefaßt werden, bei dem alle  $f_i$  konstant sind.

Abb. 18.39: *Clipping*Abb. 18.40: *Scaling*

Anmerkungen:

- Beim CoA–Verfahren sind der minimale und maximale Wert der Ausgangsgröße durch die Flächenschwerpunkte der beiden äußeren Bedeutungsmengen festgelegt. In Beispiel 18.8 kann  $y$  nicht kleiner als 17,2% und nicht größer als 82,8% werden (s. Abb. 18.38).
- Beim MoM–Verfahren kann der Ausgangswert nicht größer als der Mittelwert der Maxima der am weitesten rechts liegenden Bedeutungsmenge und nicht kleiner als der Mittelwert der Maxima der am weitesten links liegenden Bedeutungsmenge werden. In Beispiel 18.8 sind die Grenzen für  $y$  bei diesem Verfahren 5% und 95%.
- Beim MoM–Verfahren führen stetige Änderungen der Gültigkeitsgrade zu sprungförmigen Änderungen der Ausgangsgröße. In Beispiel 18.8 würde der

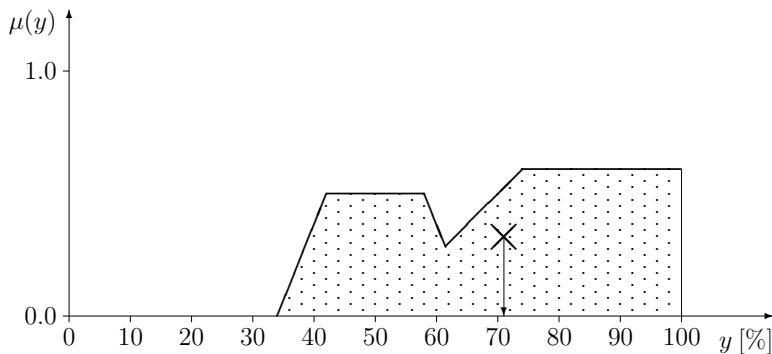


Abb. 18.41: Flächenschwerpunktverfahren

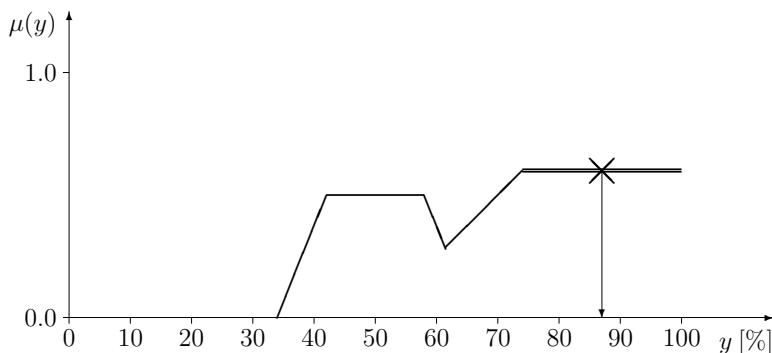


Abb. 18.42: Maximumsmittelwertverfahren

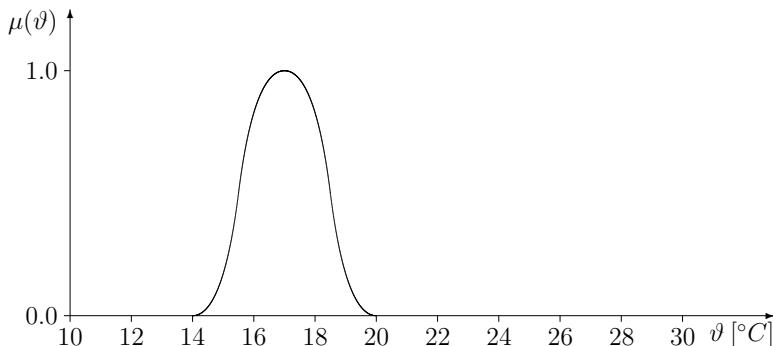
Ausgangswert z. B. auf  $y = 50\%$  springen, sobald  $G_{halboffen}$  größer als  $G_{offen}$  wird. Das MoM–Verfahren ist daher für Regelungstechnische Anwendungen weniger geeignet.

- Das CoM–Verfahren wurde als Näherung für das CoA–Verfahren entwickelt. Verwendet man beim CoA–Verfahren Bedeutungsmengen, die symmetrisch sind, sich nicht überschneiden und ihre Maxima bei den Werten  $y_{B_i}$  haben, erhält man dasselbe Ergebnis wie mit dem CoM–Verfahren. Das CoM–Verfahren hat den Vorteil, daß der Rechenaufwand erheblich geringer ist.

## 18.4 Anhang: Die „theoretische“ Darstellungsweise der unscharfen Logik

Wie bereits zu Beginn des Abschnittes 18.2 erwähnt wurde, existieren in der Literatur zwei unterschiedliche Darstellungsweisen der unscharfen Logik, eine mehr anwendungsorientierte und eine mehr theoretische. Nachdem in den vorangegangenen Abschnitten die „anwendungsorientierte“ Darstellungsweise verwendet wurde, soll nun auf die „theoretische“ Darstellungsweise eingegangen werden.

Die wichtigste Aufgabe der unscharfen Logik ist das *logische Schließen*, d. h. die Auswertung von *Fakten* mit Hilfe von *Regeln*. Der wesentliche Unterschied zwischen den beiden Darstellungsweisen der unscharfen Logik besteht darin, welcher Art die auszuwertenden Fakten und die Ergebnisse der Auswertung sind. In der bislang verwendeten „anwendungsorientierten“ Darstellungsweise waren die Fakten *scharfe Größen* und die Ergebnisse *Gültigkeitsgrade* von unscharfen Aussagen; diese konnten durch Defuzzifizierung wiederum in scharfe Ausgangsgrößen umgewandelt werden (vgl. Beispiel 18.3). Bei der „theoretischen“ Darstellungsweise sind dagegen Fakten und Ergebnisse *unscharfe Mengen*. Während z. B. in Beispiel 18.3 von dem scharfen Faktum „ $\vartheta = 17^\circ\text{C}$ “ ausgegangen wurde, könnte in der „theoretischen“ Darstellungsweise ein Faktum sein: „Die Temperatur ist ziemlich niedrig“ (siehe Abb. 18.43).



**Abb. 18.43:** Menge der ziemlich niedrigen Temperaturen

Eine solche unscharfe Menge wie die Menge der ziemlich niedrigen Temperaturen wird in der „theoretischen“ unscharfen Logik als *linguistischer Wert* (*linguistic value*) bezeichnet:

**Definition 18.20** Es sei  $X$  eine linguistische Variable mit der Basisvariablen  $x$ .  $U$  sei die Menge aller möglichen Werte von  $x$  (Grundmenge). Dann wird eine unscharfe Menge  $A$  auf  $U$  (also  $A \in \mathcal{P}(U)$ ) als **linguistischer Wert** von  $X$  bezeichnet.

Die Menge der ziemlich niedrigen Temperaturen nach Abb. 18.43 ist also z. B. ein linguistischer Wert der Variablen „Temperatur“. Die Bedeutungsmengen der Terme „hoch“ und „niedrig“ aus dem Heizungsregler–Beispiel (Abb. 18.12) sind ebenfalls linguistische Werte von „Temperatur“.

Diese Definition des linguistischen Wertes stellt einen weiteren Unterschied zwischen der „anwendungsorientierten“ und der „theoretischen“ Darstellungsweise der unscharfen Logik dar: in der „anwendungsorientierten“ Darstellungsweise bestand z. B. der Wert der linguistischen Variablen „Temperatur“ aus den Gültigkeitsgraden der Aussagen „Die Temperatur ist hoch“ und „Die Temperatur ist niedrig“.

Wie wird nun bei der Auswertung von solchen unscharfen Fakten vorgegangen? In der klassischen Logik sind zwei Vorgehensweisen bekannt, um eine Regel auf ein gegebenes Faktum anzuwenden: der *Modus Ponens* und der *Modus Tollens*.

**Satz 18.8 Modus Ponens** <sup>15)</sup> *Aus dem Faktum „Aussage 1 ist wahr“ und der Regel*

WENN Aussage 1 DANN Aussage 2

*kann geschlossen werden: Aussage 2 ist wahr.*

**Satz 18.9 Modus Tollens** <sup>16)</sup> *Aus dem Faktum „Aussage 2 ist falsch“ und der Regel*

WENN Aussage 1 DANN Aussage 2

*kann geschlossen werden: Aussage 1 ist falsch.*

Beim Modus Ponens wird die Regel also gewissermaßen „vorwärts“, beim Modus Tollens dagegen „rückwärts“ angewendet.

Modus Ponens und Modus Tollens können auf unscharfe Aussagen der Form „ $X$  ist  $A$ “ erweitert werden, wobei  $X$  eine linguistische Variable und  $A$  ein linguistischer Wert von  $X$  ist. Hier soll nur die Erweiterung des Modus Ponens behandelt werden, da der Modus Tollens für die Regelungstechnik nicht relevant ist.

Bei der Anwendung einer Regel auf ein unscharfes Faktum können prinzipiell zwei Fälle eintreten:

1. Das Faktum stimmt exakt mit der Vorbedingung der Regel überein. In diesem Fall liegt die Erweiterung von Satz 18.8 auf der Hand. Nehmen wir z. B. an, daß für einen verfahrenstechnischen Prozeß die Regel

---

<sup>15)</sup> von lat. *ponere* setzen

<sup>16)</sup> von lat. *tollere* aufheben

WENN *die Temperatur normal ist,*  
 DANN *ist der Wassergehalt mittelgroß*

gilt. Wenn der linguistische Wert von „Temperatur“ genau gleich der Menge der normalen Temperaturen ist, kann man analog zu Satz 18.8 schließen, daß „Wassergehalt“ den Wert „mittelgroß“ hat.

2. Das Faktum stimmt mit der Vorbedingung der Regel nur „mehr oder weniger“ überein. Dieser Fall ist der Normalfall. Man hat dann folgende Alternativen:
  - (a) Man stellt sich auf den Standpunkt, daß eine Regel (wie in der klassischen Logik) nur angewandt werden darf, wenn ihre Vorbedingung exakt erfüllt ist. Dann kann man im Fall 2) keine Schlußfolgerung ziehen.
  - (b) Man betrachtet eine Regel als „mehr oder weniger“ anwendbar, wenn das vorliegende Faktum „mehr oder weniger“ Ähnlichkeit mit ihrer Vorbedingung besitzt. Als Ergebnis der Regelauswertung erhält man dann „mehr oder weniger“ die Schlußfolgerung der Regel. Wie sich das „mehr oder weniger“ in der Zugehörigkeitsfunktion des resultierenden linguistischen Wertes ausdrücken läßt, wird im folgenden noch näher betrachtet. Diese Art des unscharfen Schließens heißt *plausibles Schließen (approximate reasoning)*.

#### 18.4.1 Grundlagen des plausiblen Schließens

Die Vorgehensweise beim plausiblen Schließen soll zunächst für den Fall einer einzelnen Regel erläutert werden, die nur eine einzelne Aussage als Vorbedingung hat. Für diesen Fall gilt der folgende Satz:

**Satz 18.10 (compositional rule of inference)** *Gegeben sei die Regel*

WENN  $X = A$  DANN  $Y = B$

*und das Faktum*

$$X = A' \quad ,$$

*wobei  $A'$  ein linguistischer Wert von  $X$  ist. Dann gilt für die Ausgangsvariable  $Y$*

$$Y = B' \quad ;$$

*der linguistische Wert  $B'$  ergibt sich nach der Formel*

$$\mu_{B'}(y) = \sup_x \left[ t\left(\mu_{A'}(x), I(\mu_A(x), \mu_B(y))\right) \right] \quad (18.1)$$

Dabei ist  $t(u, v)$  eine  $t$ -Norm (s. 18.2.4) und  $I(u, v)$  ein *Implikationsoperator* (s. u.). Die Anwendung von Satz 18.10 soll an einem Beispiel dargestellt werden:

**Beispiel 18.9** Zur Überwachung einer verfahrenstechnischen Anlage wird ein Expertensystem eingesetzt, das mit unscharfer Logik arbeitet. Eine Regel dieses Expertensystems lautet:

Regel 1): WENN die Temperatur normal ist,

DANN ist der Wassergehalt des Produktes mittelgroß

„Temperatur“ und „Wassergehalt“ sind linguistische Variable mit den Basisvariablen  $\vartheta$  und  $W$ ; ihre Terme sind nach Abb. 18.44 und 18.45 definiert.

Die Temperatur kann aus technischen Gründen nicht gemessen werden; durch Beobachtung des Prozesses läßt sich aber schließen, daß sie ungefähr zwischen 220 und 230°C liegt, mit einer maximalen Abweichung von 10°C nach oben oder unten. Diese Schätzung könnte dem Expertensystem in Form der unscharfen Menge  $A'$  (Abb. 18.46) eingegeben werden.

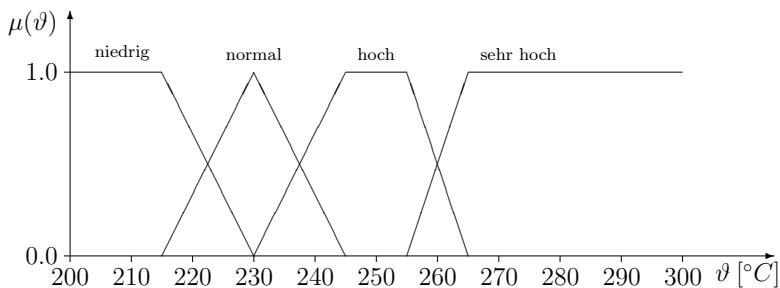


Abb. 18.44: Linguistische Variable „Temperatur“

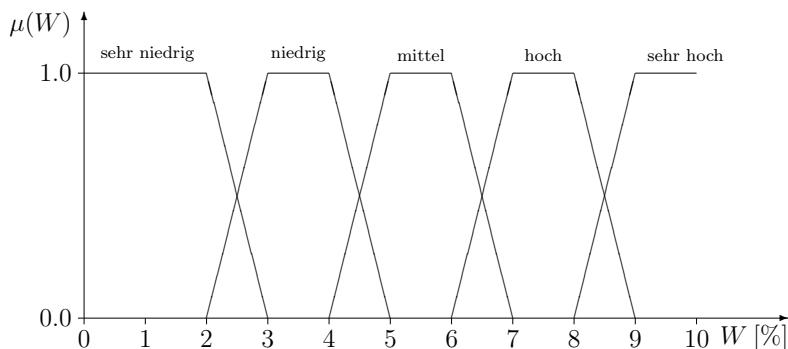
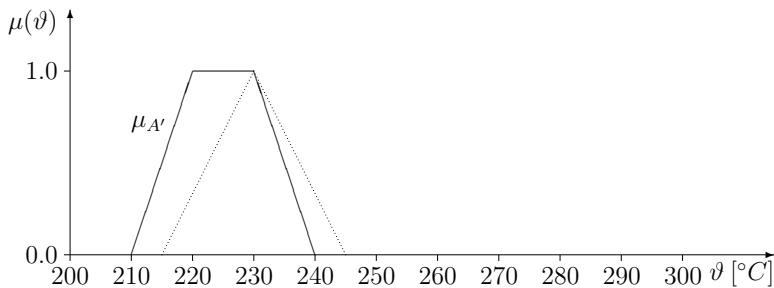


Abb. 18.45: Linguistische Variable „Wassergehalt“



**Abb. 18.46:** Geschätzte Temperatur; punktiert: Menge der normalen Temperaturen nach Abb. 18.44

Die Anwendung der Regel 1) auf das Faktum  $A'$  („Temperatur ungefähr 220 bis 230°C“) erfolgt nach Satz 18.10 in drei Schritten:

1. Für alle Wertepaare  $(\vartheta, W)$  aus dem Definitionsbereich:

Berechnung des Implikationsoperators  $I(\mu_{normal}(\vartheta), \mu_{mittel}(W))$

2. Für alle Wertepaare  $(\vartheta, W)$  aus dem Definitionsbereich:

$t$ -Norm-Verknüpfung der berechneten Werte mit der Zugehörigkeitsfunktion  $\mu_{A'}(\vartheta)$

3. Für alle Werte von  $W$ : Berechnung des Ergebnisses  $B'$  (berechneter Wassergehalt) nach der Formel

$$\mu_{B'}(W) = \sup_{\vartheta} (\text{Ergebnis von Schritt 2})$$

Anschaulich kann man Satz 18.10 so deuten, daß  $\mu_{B'}(W)$  punktweise berechnet wird, indem für jeden Wert von  $W$  (z. B.  $W = 7\%$ ) folgende Regeln (vgl. Satz 18.8, Modus Ponens) mit Hilfe der „anwendungsorientierten“ unscharfen Logik ausgewertet werden:

⋮

WENN  $\vartheta = 200^\circ C$  ist UND aus  $(\vartheta = 200^\circ C)$  ( $W = 7\%$ ) folgt

DANN ist  $W = 7\%$

WENN  $\vartheta = 201^\circ C$  ist UND aus  $(\vartheta = 201^\circ C)$  ( $W = 7\%$ ) folgt

DANN ist  $W = 7\%$

WENN  $\vartheta = 202^\circ C$  ist UND aus  $(\vartheta = 202^\circ C)$  ( $W = 7\%$ ) folgt

DANN ist  $W = 7\%$

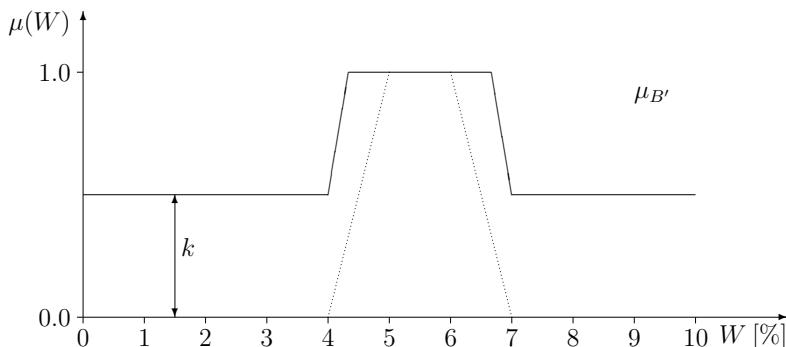
⋮

Die  $t$ -Norm entspricht dem UND in diesen Regeln, der Wert des Implikationsoperators  $I(\mu_{normal}(x), \mu_{mittel}(y))$  kann als Gültigkeitsgrad der Aussage „aus  $(\vartheta = x)$  folgt  $(W = y)$ “ aufgefaßt werden, und die Bildung des Supremums entspricht einer ODER-Verknüpfung der Ergebnisse der einzelnen Regeln.

Die verschiedenen möglichen Implikationsoperatoren werden im nächsten Abschnitt ausführlicher behandelt. Hier soll als Implikationsoperator die Goguen-Implikation

$$I_g(u, v) := \begin{cases} 1 & \text{für } v \geq u \\ v & \text{für } v < u \end{cases} \quad \text{für } u, v \in [0, 1]$$

eingesetzt werden (siehe Abb. 18.51); als  $t$ -Norm wird der Minimumoperator verwendet. Mit diesen Operatoren erhält man als Ergebnis von Regel 1) die unscharfe Menge  $B'$  nach Abb. 18.47; sie kann als „Wassergehalt mehr oder weniger mittelgroß“ interpretiert werden.



**Abb. 18.47:** Ergebnis von Regel 1): „Wassergehalt mehr oder weniger mittelgroß“ (Punktiert: Menge der mittelgroßen Wassergehalte nach Abb. 18.45 )

Man erkennt, daß die Menge der „mehr oder weniger mittelgroßen“ Wassergehalte von der Menge der mittelgroßen Wassergehalte (in Abb. 18.47 punktiert eingezeichnet) in zwei Punkten abweicht: sie ist „breiter“, und ihre Zugehörigkeitsfunktion weist einen konstanten Anteil  $k$  auf. Beide Abweichungen sind ein Maß für die Unsicherheit („mehr oder weniger“) des Ergebnisses; sie sind umso stärker, je geringer die Ähnlichkeit zwischen dem Faktum  $A'$  und der Menge der normalen Temperaturen ist. Wenn das Faktum genau gleich der Menge der normalen Temperaturen ist, erhält man als Ergebnis  $B'$  genau die Menge der mittelgroßen Wassergehalte. Ist dagegen die Schnittmenge zwischen  $A'$  und der Menge der normalen Temperaturen leer, so ist  $\mu_{B'}$  konstant gleich Eins; dies bedeutet, daß die Auswertung der Regel 1) keine Aussage ergibt.

### 18.4.2 Implikationsoperatoren

Eine zentrale Rolle in der „theoretischen“ unscharfen Logik spielen die Implikationsoperatoren. Anschaulich kann man sagen, daß ein Implikationsoperator für eine bestimmte Regel „WENN  $X = A$  DANN  $Y = B$ “ aus den beiden Zugehörigkeitsgraden  $\mu_A(x_0)$  und  $\mu_B(y_0)$  den Gültigkeitsgrad der Aussage „ $(x = x_0) \Rightarrow (y = y_0)$ “ berechnet (s. obiges Beispiel).

In der Literatur ([224], [136]) werden etwa 40 verschiedene Implikationsoperatoren beschrieben. Im folgenden sind die Definitionen für einige dieser Operatoren aufgeführt; für eine genauere Begründung der einzelnen Definitionen sei auf die Literatur verwiesen.

Einige wichtige Implikationsoperatoren:

- Die *arithmetische Implikation* nach Zadeh<sup>17)</sup>

$$I_z(u, v) := (1 - u) \oplus v = \min(1, 1 - u + v) \quad \text{für } u, v \in [0, 1]$$

(siehe Abb. 18.48);

- die *Boolesche Implikation*<sup>18)</sup>

$$I_b(u, v) := \max(1 - u, v) \quad \text{für } u, v \in [0, 1]$$

(siehe Abb. 18.49);

- die *Standard–Strict–Implikation* (*standard sequence fuzzy implication*)

$$I_s(u, v) := \begin{cases} 1 & \text{für } u \leq v \\ 0 & \text{für } u > v \end{cases} \quad \text{für } u, v \in [0, 1]$$

(siehe Abb. 18.50);

- die *Goguen–Implikation*<sup>19)</sup>

$$I_g(u, v) := \begin{cases} 1 & \text{für } u \leq v \\ v & \text{für } u > v \end{cases} \quad \text{für } u, v \in [0, 1]$$

(siehe Abb. 18.51);

- die *Gaines–Implikation*

$$I_G(u, v) := \begin{cases} 1 & \text{für } u \leq v \\ \frac{v}{u} & \text{für } u > v \end{cases} \quad \text{für } u, v \in [0, 1]$$

(siehe Abb. 18.52);

---

<sup>17)</sup> Auch: *Lukasiewicz–Implikation*

<sup>18)</sup> Auch: *Kleene–Dienes–Implikation*

<sup>19)</sup> Auch: *Gödel–Implikation*

Neben diesen Implikationsoperatoren im engeren Sinne können auch  $t$ –Normen als Implikationsoperatoren verwendet werden. Besonders der Minimumoperator (*Mamdani–Implikation*) und der Produktoperator (*Larsen–Implikation*) kommen hierfür infrage. Ihre Verwendung führt zur „anwendungsorientierten“ unscharfen Logik; näheres dazu im Unterabschnitt „Die anwendungsorientierte unscharfe Logik als Spezialfall des plausiblen Schließens“.

### 18.4.3 Berücksichtigung von Verbundaussagen und mehreren Regeln

Das plausible Schließen nach Satz 18.10 kann auf Regeln erweitert werden, deren Vorbedingung eine Verbundaussage ist:

WENN  $X_1 = A_1$  UND  $X_2 = A_2 \dots$  UND  $X_n = A_n$  DANN  $Y = B$

Wenn die Fakten  $X_1 = A'_1$ ,  $X_2 = A'_2 \dots X_n = A'_n$  sind, kann der linguistische Wert von  $Y$  analog zu Satz 18.10 berechnet werden; es ist lediglich  $\mu_A(x)$  durch

$$t(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n))$$

und  $\mu_{A'}(x)$  durch

$$t(\mu_{A'_1}(x_1), \mu_{A'_2}(x_2), \dots, \mu_{A'_n}(x_n))$$

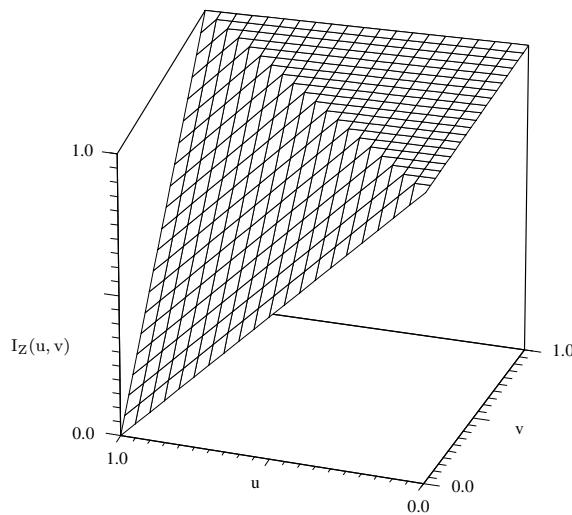
zu ersetzen. Die einzelnen Zugehörigkeitsgrade werden also vor Anwendung des Satzes 18.10 mit Hilfe einer  $t$ –Norm UND–verknüpft.

Enthält die Regelbasis mehrere Regeln, erhält man das Gesamtergebnis der Regelauswertung durch ODER–Verknüpfung der Ergebnisse der einzelnen Regeln.

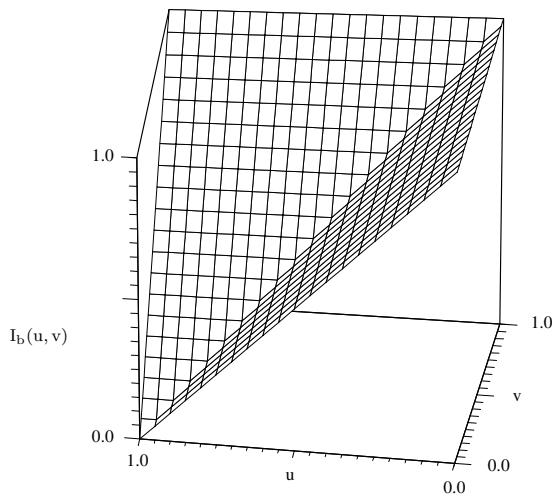
### 18.4.4 Berücksichtigung zusätzlicher Unsicherheiten

Eine andere Erweiterung des plausiblen Schließens besteht in der Berücksichtigung zusätzlicher Unsicherheiten. Bislang war davon ausgegangen worden, daß die Regeln zu 100% „richtig“ sind und die Fakten zwar unscharf, aber mit Sicherheit bekannt. In manchen Anwendungsgebieten existieren aber Regeln, die nicht immer voll gültig sind, sondern nur „in den meisten Fällen“; oder es gibt Regeln, die ein höheres Gewicht haben, und Regeln mit einem geringeren Gewicht. Außerdem können die Fakten neben der linguistischen Unschärfe noch mit zusätzlicher Unsicherheit behaftet sein („Es ist anzunehmen, daß die Temperatur ziemlich niedrig ist“).

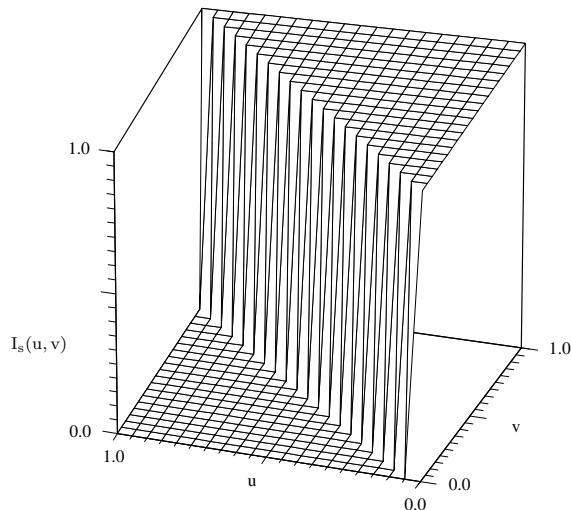
Die Berücksichtigung dieser Unsicherheiten beim unscharfen Schließen kann erfolgen, indem man den Regeln bzw. den Fakten *Plausibilitätsgrade* (*degrees of support*) zwischen Null und Eins zuordnet. Diese Plausibilitätsgrade können als Zugehörigkeitsgrade zur Menge der absolut gültigen Regeln bzw. zur Menge der absolut sicheren Fakten interpretiert werden. Diese Plausibilitätsgrade werden dann durch UND–Verknüpfung in die Inferenz einbezogen.



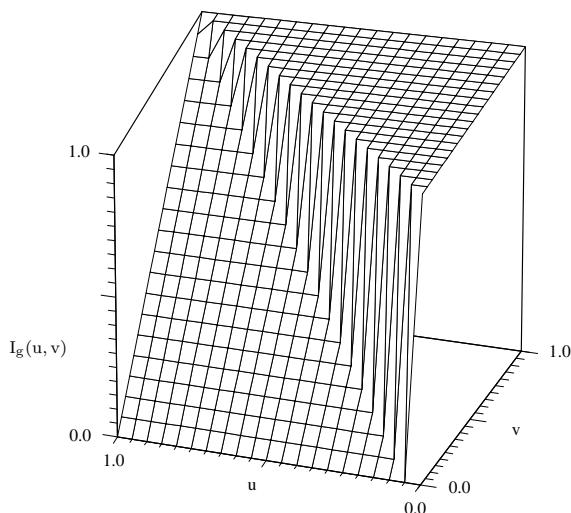
**Abb. 18.48:** Arithmetische Implikation



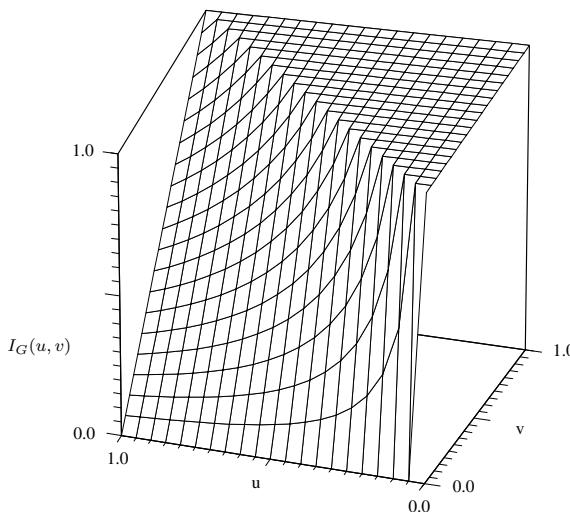
**Abb. 18.49:** Boolesche Implikation



**Abb. 18.50:** Standard–Strict–Implikation



**Abb. 18.51:** Goguen–Implikation



**Abb. 18.52:** Gaines–Implikation

#### 18.4.5 Die „anwendungsorientierte“ unscharfe Logik als Spezialfall des plausiblen Schließens

Nachdem in den vorangegangenen Abschnitten die „anwendungsorientierte“ und die „theoretische“ Darstellungsweise der unscharfen Logik vorgestellt wurden, soll nun auf die Beziehungen zwischen diesen beiden Darstellungsweisen eingegangen werden. Eine mathematische Untersuchung dieser Beziehungen führt zu folgenden Ergebnissen:

- Die „anwendungsorientierte“ unscharfe Logik stellt im wesentlichen einen Spezialfall des plausiblen Schließens dar. Bei der „anwendungsorientierten“ unscharfen Logik gelten folgende Voraussetzungen:

- Die Fakten sind scharfe Größen  $x = x_0$ . Eine scharfe Größe kann als Sonderfall einer unscharfen Menge angesehen werden; sie lässt sich als normalisiertes Singleton darstellen:

$$\mu_{A'}(x) = \begin{cases} 1 & \text{für } x = x_0 \\ 0 & \text{für } x \neq x_0 \end{cases}$$

- Implikationsoperator ist der Minimumoperator (Mamdani–Implikation) oder der Produktoperator (Larsen–Implikation); als  $t$ –Norm in Gleichung 18.1 (S. 806) wird der gleiche Operator verwendet.

Bei Verwendung der Mamdani-Implikation erhält man für den linguistischen Wert  $B'$  der Ausgangsgröße dasselbe Ergebnis, wie wenn man die Inferenz mit Hilfe der „anwendungsorientierten“ unscharfen Logik durchführt und die berechneten Gültigkeitsgrade anschließend durch „Clipping“ (s. Abschnitt 18.3.3) in eine unscharfe Menge umwandelt. Bei Verwendung der Larsen-Implikation entspricht das Ergebnis der „anwendungsorientierten“ Inferenz mit anschließendem „Scaling“.

- „Anwendungsorientierte“ und „theoretische“ unscharfe Logik unterscheiden sich in der Darstellungsweise: in der „anwendungsorientierten“ unscharfen Logik kann der Begriff des Gültigkeitsgrades eingeführt werden, der in der allgemeinen Theorie des plausiblen Schließens explizit nicht vorkommt. Dadurch wird die Darstellung wesentlich einfacher und anschaulicher.
- „Anwendungsorientierte“ unscharfe Logik und plausibles Schließen unterscheiden sich *inhaltlich* für den Fall, daß mehrere Regeln verkettet werden, z.B.

Regel 1): WENN  $X = A$  DANN  $Y = B$

Regel 1\*): WENN  $X = A^*$  DANN  $Y = B^*$

Regel 2): WENN  $Y = B$  DANN  $Z = C$

In der „anwendungsorientierten“ unscharfen Logik ist in einem solchen Fall der Gültigkeitsgrad von „ $Z = C$ “ gleich dem Gültigkeitsgrad von „ $Y = B^*$ “, der aus Regel 1) berechnet wird. Das Ergebnis der Regel 1\* hat auf den Gültigkeitsgrad von „ $Z = C$ “ keinen Einfluß. Beim plausiblen Schließen hängt dagegen im allgemeinen das Ergebnis von Regel 2) sowohl vom Ergebnis der Regel 1) als auch vom Ergebnis der Regel 1\*) ab.

# Literaturverzeichnis

- [1] ABERG, E.R. und A.R.T. GUSTAVSSON: *Design and evaluation of modified simplex methods*. *Analytica chimica acta*, 144:39–53, 1982.
- [2] ACKERMANN, J.: *Robuste Regelung*. Springer-Verlag Berlin Heidelberg New York, 1993.
- [3] ALT, W.: *Nichtlineare Optimierung*. Vieweg, 2002.
- [4] ANDERSON, B. D. O., BITMEAD R. R. JOHNSON C. R. JR. KOKOTOVIC P. V. KOSUT R. L. MAREELS I. M. Y. PRALY L. RIEDLE B. D.: *Stability of Adaptive Systems: Passivity and Averaging Analysis*. The MIT Press, 1986.
- [5] ANGERER, B.: *Online Identifikation mechatronischer Systeme mit Hilfe rekurrenter Netze*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2001.
- [6] ANGERER, B.: *Fortschritte in der Erforschung der repetitiven peripheren Magnetstimulation*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2007.
- [7] ANGERER, B.T., C. HINTZ und D. SCHRÖDER: *Online identification of a nonlinear mechatronic system*. *Control Engineering Practice*, 12(11):1465–1478, November 2004.
- [8] 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.
- [9] ARMSTRONG-HÉLOUVRY, B., DUPONT P. CANUDAS DE WIT C.: *A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction*. *Automatica*, 30(7):1083–1138, 1994.
- [10] ASTROM, K. J., WITTENMARK B.: *Adaptive Control*. Addison-Wesley Publishing Company, Inc., 1995.
- [11] AYOUBI, M.: *Nonlinear System Identification Based on Neuronal Networks with Locally Distributed Dynamics and Application to Technical Processes*. Dissertation, VDI Verlag, Düsseldorf, 1996.
- [12] BAUMGARTNER, U., T. EBNER und C. MAGELE: *Optimization in Electrical Engineering*. Vorlesungsskript, Institute for Fundamentals and Theory in Electrical Engineering, TU Graz, 2005.
- [13] BECKER, T. T.: *Methoden kleinster Fehlerquadrate zur parametrischen Identifikation dynamischer Systeme*. VDI Verlag, 1989.
- [14] BERENJI, H.: *A Reinforcement Learning-Based Architecture for Fuzzy Logic Control*. *International Journal of Approximate Reasoning*, vol. 6, pp. 267 – 292, 1992.

- [15] BERGER, J. und W. VOLTERRA: *Technische Mechanik für Ingenieure, Band 3: Dynamik*. Vieweg, 1998.
- [16] BETTERIDGE, D., A.P. WADE und A.G. HOWARD: *Reflections on the modified simplex*. *Talanta*, 32:723–734, 1985.
- [17] BEUSCHEL, M.: *Neuronale Netze zur Diagnose und Tilgung von Drehmomentschwingungen am Verbrennungsmotor*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2000.
- [18] BISHOP, C. M.: *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [19] BONABEAU, E., M. DORIGO und G. THERAULAZ: *Swarm Intelligence - From Natural to Artificial Systems*. Oxford University Press, 1999.
- [20] BRAUSE, R.: *Neuronale Netze – Eine Einführung in die Neuroinformatik*. B.G. Teubner Verlag, 1995.
- [21] BROMMUNDT, E und G. SACHS: *Technische Mechanik*. Springer–Verlag, 1988.
- [22] BRONSTEIN, I. N., K. A. SEMENDJAJEW, G. MUSIOL und H. MÜHLIG: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 3. Auflage, 1997.
- [23] BROYDEN, C. G.: *A Class of Methods for Solving Nonlinear Simultaneous Equations*. Mathematics of Computation, 19:577–593, 1965.
- [24] BROYDEN, C. G.: *The Convergence of a Class of Double-rank Minimization Algorithms*. Journal of the Institute of Mathematics and Its Applications, 6:76–90, 1970.
- [25] BRYCHCY, T.: *Modellierung dynamischer Systeme mit vorstrukturierten neuronalen Netzen*. Akad. Verl.-Ges., 2000.
- [26] BULLINGER, E.: *Adaptive  $\lambda$ -tracking for Systems with Higher Relative Degree*. Dissertation, Institut für Automatik, ETH Zürich No. 13858, 2001.
- [27] BUSS, M. und G. SCHMIDT: *Computational Intelligence*. Vorlesungsskript, Lehrstuhl für Steuerungs- und Regelungstechnik, TU München, 2004.
- [28] BUTTELmann, M. und B. LOHMANN: *Optimierung mit Genetischen Algorithmen und eine Anwendung zur Modellreduktion*. Automatisierungstechnik (at), 52(4):151–163, 2004.
- [29] CABRERA, J.B.D. und K. FURUTA: *Improving the robustness of Nussbaum-type regulators by the use of  $\sigma$ -modification - Local results*. *Systems & Control Letters*, 12(5):412–429, June 1989.
- [30] CABRERA, J.B.D und K.S. NARENDRA: *Issues in the application of neural networks for tracking based on inverse control*. IEEE Transactions on Automatic Control, vol. 44, no. 11, pp. 2007–2027, 1999.
- [31] CAORSI, S., A. MASSA und M. PASTORINO: *A computational technique based on a real-coded genetic algorithm for microwave imaging purposes*. IEEE Transactions on Geoscience and Remote Sensing, 38(4):1228–1233, Jule 2000.
- [32] CERNY, V.: *Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm*. Journal of Optimization Theory and Applications, 45(1):41–51, Januar 1985.
- [33] COOPER, M. G. und J. J. VIDAL: *Genetic Design of Fuzzy Controllers: The Cart and Jointed-Pole Problem*. Proc. Third IEEE International Conference on Fuzzy Systems (Fuzz-IEEE'94), Orlando, 1994.
- [34] DAVIDON, W.C.: *Variable metric method for minimization*. Technical Report ANL-5990, Argonne National Laboratory, Argonne, II, 1959.

- [35] DORIGO, M.: *Ottimizzazione, Apprendimento Automatico, ed Algoritmi Basati su Metafora Naturale*. Doktorarbeit, Politecnico di Milano, 1992.
- [36] DORSCH, M.: *Globale Optimierungsstrategien für mehrschichtige Perzeptronen-Netze*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, Technische Universität München, Juni 2006.
- [37] DOYLE, F. J., R. K. PEARSON und B. A. OGUNNAIKE: *Identification and Control Using Volterra Models*. Springer-Verlag, 2002.
- [38] DUECK, G. und T. SCHEUER: *Threshold Accepting: A general purpose optimization algorithm appearing superior to simulated annealing*. Journal of Computational Physics, Seiten 161–175, 1990.
- [39] DUECK, G., T. SCHEUER und H.-M. WALLMEIER: *Toleranzschwelle und Sintflut: Neue Ideen zur Optimierung*. Spektrum der Wissenschaft, Seiten 42–51, März 1993.
- [40] EBERHART, R. und J. KENNEDY: *A New Optimizer Using Particle Swarm Theory*. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Seiten 39–43, 1995.
- [41] EBERHART, R. und Y. SHI: *Special Issue on Particle Swarm Optimization*. IEEE Transactions on Evolutionary Computation, 8(3):201–203, Juni 2004.
- [42] EIBEN, A. E. und J. E. SMITH: *Introduction to Evolutionary Computing*. Springer, 2003.
- [43] ELMAN, J. L: *Finding Structure in Time*. Cognitive Science, Vol 14, pp.179–211, 1990.
- [44] ENDISCH, C.: *Optimierungsstrategien für die Identifikation mechatronischer Systeme*. Shaker Verlag, Aachen, 2009.
- [45] ENDISCH, C. und D. SCHRÖDER: *Fast Nonlinear Dynamic System Identification using Time Delay Neural Networks and its Application in Mechatronic Systems*. Proceedings of the International Conference on Instrumentation, Communication and Information Technology (ICICI), Bandung, Indonesien, Seiten 122–128, 2005.
- [46] ENTENMANN, W.: *Optimierungsverfahren*. Hüthig Verlag, Heidelberg, 1976.
- [47] FAHLMAN, S. E.: *Faster-Learning Variations on Back-Propagation: An Empirical Study*. Proceedings of the 1988 Connectionist Models Summer School, Seiten 38–51, 1988.
- [48] FEILER, M. J.: *Adaptive Control in the Presence of Disturbances*. Doctoral Thesis, 2004.
- [49] FEILER, M., WESTERMAIER C. SCHROEDER D.: *Adaptive Speed Control of a Two-Mass System*. Transactions of the IEEE, Conference on Control Applications, 2003.
- [50] FISCHER, A., W. SCHIROTZEK und K. VETTERS: *Linear Algebra – Eine Einführung für Ingenieure und Naturwissenschaftler*. Mathematik für Ingenieure und Naturwissenschaftler. Vieweg+Teubner, 2003.
- [51] FISCHLE, K.: *Ein Beitrag zur stabilen adaptiven Regelung nichtlinearer Systeme*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU-München, 1998.
- [52] FLETCHER, R.: *A New Approach to Variable Metric Algorithms*. Computer Journal, 13:317–322, 1970.
- [53] FLETCHER, R.: *Practical Methods of Optimization*. John Wiley, New York, 2 Auflage, 1987.

- [54] FLETCHER, R. und M. J. D. POWELL: *A rapidly convergent descent method for minimization.* The Computer Journal, 6:163–168, 1963.
- [55] FLETCHER, R. und C. M. REEVES: *Function minimisation by conjugate gradients.* Computer Journal, 7:149–154, 1964.
- [56] FÖLLINGER, O.: *Regelungstechnik (7. Auflage).* Hüthig Buch Verlag Heidelberg, 1992.
- [57] FÖLLINGER, O.: *Regelungstechnik, Einführung in die Methoden und ihre Anwendung.* Hüthigbuch Verlag, 1992.
- [58] FRANCIS, B.A; WONHAM, W.M.: *The internal model principle of control theory.* Automatica Vol. 12, pp. 457–465, 1976.
- [59] FRANCIS, B., WONHAM W.: *The Internal Model Principle of Control Theory.* Automatica, 12(5):457–465, 1976.
- [60] FUCHS, M.: *Lernverfahren mit konjugierten Richtungen zur Identifikation dynamischer Nichtlinearitäten.* Diplomarbeit, Lehrstuhl für Elektrische Antriebsysteme, Technische Universität München, 2005.
- [61] GEIGER, CARL: *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben.* Springer, 1999.
- [62] GERDES, I., F. KLAWONN und R. KRUSE: *Evolutionäre Algorithmen.* Vieweg, Wiesbaden, 2004.
- [63] GERTHSEN, C. und H. VOGEL: *Physik.* Springer, Berlin, 18. Auflage, 1995.
- [64] GILL, P.E, W. MURRAY und M.H. WRIGHT: *Practical Optimization.* Academic Press, London, 1981.
- [65] GOLDBERG, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass., 1989.
- [66] GOLDFARB, D.: *A Family of Variable Metric Updates Derived by Variational Means.* Mathematics of Computation, 24:23–26, 1970.
- [67] GOLDFARB, M. und N. CELANOVIC: *Modeling Piezoelectric Stack Actuators for Control of Micromanipulation.* IEEE Transactions of Control Systems, vol. 17, no. 3, 1997.
- [68] GOODWIN, G. C. und K. S. SIN: *Adaptive Filtering, Prediction and Control.* Prentice-Hall, 1984.
- [69] HACKL, C. M., C. ENDISCH und D. SCHRÖDER: *Error Reference Control of Nonlinear Two-Mass Flexible Servo Systems.* Proceedings of the 16th Mediterranean Conference on Control and Automation, MED 2008 - Ajaccio, France, June 25-27, Seiten 1047–1053, 2008. ISBN: 978-1-4244-2505-1 IEEE Catalog Number: CFP08MED-CDR.
- [70] HACKL, C. M., C. ENDISCH und D. SCHRÖDER: *Specially Designed Funnel-Control in Mechatronics.* to be published in Proceedings of the 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2008 - Linz, Austria, June 19-21, 2008.
- [71] HACKL, C. M., Y. JI und D. SCHRÖDER: *Enhanced Funnel-Control with Improved Performance.* Proceedings of the 15th Mediterranean Conference on Control and Automation (Paper T01-016), MED 2007 - Athens, Greek, Jun 27-29, 2007. ISBN: 978-960-254-664-2.
- [72] HACKL, C. M., Y. JI und D. SCHRÖDER: *Funnel-Control with Constrained Control Input Compensation.* Proceedings of the 9th IASTED International Conference CONTROL AND APPLICATIONS, CA 2007, May 30 - June 1,

- Montreal, Quebec, Canada, ISBN Hardcopy: 978-0-88986-665-2 / CD: 978-0-88986-666-9:568-087, 2007.
- [73] HACKL, C. M., Y. JI und D. SCHRÖDER: *Nonidentifier-based Adaptive Control with Saturated Control Input Compensation*. Proceedings of the 15th Mediterranean Conference on Control and Automation (Paper T01-015), MED 2007 - Athens, Greek, Jun 27-29, 2007. ISBN: 978-960-254-664-2.
- [74] HACKL, C. M. und D. SCHRÖDER: *Extension of High-Gain Controllable Systems For Improved Accuracy*. Joint CCA, ISIC and CACSD Proceedings 2006, October 4-6, 2006, Munich, Germany (CDROM), Seiten 2231–2236, 2006.
- [75] HACKL, C. M. und D. SCHRÖDER: *Funnel-Control For Nonlinear Multi-Mass Flexible Systems*. IECON'06 Proceedings (CDROM), November 7-10, Paris, 2006.
- [76] HACKL, C. M. und D. SCHRÖDER: *Funnel-Control with Online Foresight*. Proceedings of the 26th IASTED International Conference MODELLING, IDENTIFICATION AND CONTROL, MIC 2007, February 12 -14, Innsbruck, Seiten 171 – 176, 2007.
- [77] HACKL, C. M., H. SCHUSTER, C. WESTERMAIER und D. SCHRÖDER: *Funnel-Control with Integrating Prefilter for Nonlinear, Time-varying Two-Mass Flexible Servo Systems*. The 9th International Workshop on Advanced Motion Control, AMC 2006 - Istanbul, Seiten 456 – 461, March 27-29 2006.
- [78] HACKL, CHRISTOPH MICHAEL, CHRISTIAN ENDISCH und DIERK SCHRÖDER: *Contributions to non-identifier based adaptive control in mechatronics*. Robotics and Autonomous Systems, Elsevier, 57(10):996–1005, October 2009.
- [79] HAGAN, M. T., H. B. DEMUTH und M. H. BEALE: *Neural Network Design*. PWS Publishing Company, Boston, 1996.
- [80] HAGAN, M. T. und M. MENHAJ: *Training Feedforward Networks with the Marquardt Algorithm*. IEEE Transactions on Neural Networks, 5(6):989–993, November 1994.
- [81] HAGAN, M., DEMUTH, H., BEALE, M.: *Neural Network Design*. PWS Publishing Company, 1996.
- [82] HAMM, L., B. BRORSEN und M. HAGAN: *Global Optimization of Neural Network Weights*. Proceedings of the International Joint Conference of Neural Networks, 2:1228–1233, Mai 2002.
- [83] HAYKIN, S.: *Neural Networks a Comprehensive Foundation*. Prentice Hall, New Jersey, 2 Auflage, 1999.
- [84] HECHT-NIELSEN, R.: *Neurocomputing*. Addison-Wesley, 1990.
- [85] HESTENES, M. R. und E. STIEFEL: *Methods of conjugate gradients for solving linear systems*. Journal of Research of the National Bureau of Standards, 49(6):409–436, 1952.
- [86] 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.
- [87] HEUSER, HARRO: *Gewöhnliche Differentialgleichungen*, Band (6., aktualisierte Auflage). Vieweg+Teubner GWV Fachverlag GmbH, Wiesbaden, 2009.
- [88] HINTZ, C.: *Identifikation nichtlinearer mechatronischer Systeme mit strukturierten rekurrenten Netzen*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2003.

- [89] HINTZ, C., B. ANGERER und D. SCHRÖDER: *Online identification of a mechatronic system with structured recurrent neural networks*. Proceedings of the 2002 IEEE International Symposium on Industrial Electronics (ISIE), 1:288–293, 2002.
- [90] HINTZ, C., B. ANGERER und D. SCHRÖDER: *Online Identification of Nonlinear Mechatronic System*. Proceedings of the IFAC Conference on Mechatronic Systems December 9-11, Berkeley, California, USA, 2002.
- [91] HOFFMANN, N.: *Neuronale Netze – Kleines Handbuch*. Vieweg, 1993.
- [92] HOFMANN, S.: *Identifikation von nichtlinearen mechatronischen Systemen auf der Basis von Volterra-Reihen*. Dissertation, Lehrstuhl für Elektrische Antriebsysteme, TU München, 2003.
- [93] HOFMANN, S., T. TREICHL und D. SCHRÖDER: *Identification and Observation of Mechatronic Systems including Multidimensional Nonlinear Dynamic Functions*. Proceedings of the 7th International on Advanced Motion Control AMC 2002, pp. 285-290, Maribor Slovenia, 2002.
- [94] HOLLAND, J.: *Adaptation in natural and artificial systems*. MIT press, Cambridge, Mass., 1992.
- [95] HOOKE, R. und T.A. JEEVES: *Direct search solution of numerical and statistical problems*. Journal of the ACM, 8:212–229, 1961.
- [96] HU, X., Y. SHI und R. EBERHART: *Recent Advances in Particle Swarm*. Congress on Evolutionary Computation CEC2004, 1:90–97, Juni 2004.
- [97] HUIFU, X.: *MATH3016: Optimization*. School of Mathematics, University of Southampton, Highfield SO17 1BJ, Southampton, 2005.
- [98] HUSH, D.R. und J.M. SALAS: *Improving the learning rate of backpropagation with the gradient re-use algorithm*. IEEE International Conference on Neural Networks, 1:441–447, 1988.
- [99] ILCHMANN, A.: *Non-Identifier-Based Adaptive Control of Dynamical Systems: A Survey*. IMA Journal of Mathematical Control & Information, 8:321–366, 1991.
- [100] ILCHMANN, A.: *Non-Identifier-Based Adaptive Control of Dynamical Systems: A Survey*. IMA Journal of Mathematical Control & Information, 8:321 – 366, 1991.
- [101] ILCHMANN, A.: *Non-Identifier-Based High-Gain Adaptive Control*. Lecture Notes in Control and Information Sciences 189, Springer-Verlag, 1993.
- [102] ILCHMANN, A. und E. P. RYAN: *Asymptotic Tracking With Prescribed Transient Behaviour For Linear Systems*. International Journal of Control, 79(8):910–917, May 2006.
- [103] ILCHMANN, A., E. P. RYAN und C. J. SANGWIN: *Tracking with Prescribed Transient Behaviour*. ESAIM: Control, Optimisation and Calculus of Variations, EDP Sciences, SMAI, 2002.
- [104] ILCHMANN, A., E. P. RYAN und S. TRENN: *Tracking Control: Performance Funnels and Prescribed Transient Behaviour*. Systems & Control Letters, 54:655 – 670, 2005.
- [105] ILCHMANN, A., E.P. RYAN und P. TOWNSEND: *Tracking With Prescribed Transient Behaviour For Nonlinear Systems Of Known Relative Degree*. SIAM Journal on Control and Optimization, 46(1):210 – 230, 2007.

- [106] ILCHMANN, A. und H. SCHUSTER: *PI-funnel control for two mass systems*. IEEE Transactions on Automatic Control, 54(4):918–923, April 2009.
- [107] ILCHMANN, A., RYAN E. SANGWIN C.: *Tracking with prescribed transient behaviour*. ESAIM: Control, Optimisation and Calculus of Variations, 7:471–493, 2002.
- [108] ILCHMANN, A., SCHUSTER H.: *PI-Funnel Control for Two Mass Systems*. IEEE Trans. on Automatic Control, 54(4):918–923, 2009.
- [109] IOANNOU, P.A.; SUN, J.: *Robust Adaptive Control*. Prentice-Hall, 1996.
- [110] ISAACSON, D. L., MADSEN R.W.: *Markov Chains, theory and applications*. John Wiley & Sons, 1976.
- [111] ISERMANN, R.: *Identifikation dynamischer Systeme – Band 1*. Springer–Verlag, 1992.
- [112] ISERMANN, R.: *Identifikation dynamischer Systeme – Band 2*. Springer–Verlag, 1992.
- [113] ISIDORI, A.: *Nonlinear Control Systems*. Springer–Verlag, 1989.
- [114] ISIDORI, A.: *Nonlinear Control Systems*. Springer-Verlag, London, 3rd Auflage, 1995.
- [115] ISIDORI, A.: *Nonlinear Control Systems*. Springer–Verlag, 2001.
- [116] JACOBS, R. A.: *Increased rates of convergence through learning rate adaption*. Neural Networks, 1(4):205–308, 1988.
- [117] JUNGE, T. F.: „*On-line“-Identifikation und lernende Regelung nichtlinearer Regelstrecken mittels neuronaler Netze*. VDI Verlag, Düsseldorf, 1999.
- [118] JYH-SHING, R. J.: *ANFIS: Adaptive-Network-Based Fuzzy Inference System*. IEEE Transactions on Systems, Man and Cybernetics, vol. 23, pp. 665 – 684, 1993.
- [119] KALMAN, R. E. und R. S. BUCY: *New results in linear filtering and prediction theory*. Proc. ASME Journal of Basic Engineering, March, Seiten 95–108, 1961.
- [120] KENNEDY, J. und R. EBERHART: *Particle Swarm Optimization*. Proceedings of the IEEE International Conference on Neural Networks, Seiten 1942–1948, 1995.
- [121] KENNEDY, J. und R. EBERHART: *Swarm Intelligence*. Academic Press, 2001.
- [122] KHALIL, H.: *Nonlinear Systems*. Prentice – Hall, 1996.
- [123] KILLICH, A.: *Prozeßidentifikation durch Gewichtsfolgenschätzung*. Dissertation, VDI Verlag, Düsseldorf, 1991.
- [124] KIRKPATRICK, S., C. GELATT und M. VECCHI.: *Optimization by Simulated Annealing*. Science, 220(4598):671–680, Mai 1983.
- [125] KNOHL, T.: *Anwendung künstlicher neuronaler Netze zur nichtlinearen adaptiven Regelung*. VDI Verlag, Düsseldorf, 2001.
- [126] KOHONEN, T.: *The Self-Organizing Map*. Proc. of the IEEE, Vol 78, No.9 pp. 1464-1480 Sept., 1989.
- [127] KORTMANN, M.: *Die Identifikation nichtlinearer Ein- und Mehrgrößensysteme auf der Basis nichtlinearer Modellansätze*. VDI Verlag, 1989.
- [128] KOST, B.: *Optimierung mit Evolutionsstrategien*. Harri Deutsch, 2003.
- [129] KREISSELMEIER, G. und K.S. NARENDRA: *Stable model reference adaptive control in the presence of bounded disturbances*. IEEE Transactions on Automatic Control, vol. 27, pp. 1169–1175, 1982.

- [130] KÚRKOVÁ, V. und R. NERUDA: *Uniqueness of Functional Representation by Gaussian Basis Function Networks*. ICANN, Sorrento, 1994.
- [131] KURTH, J.: *Identifikation nichtlinearer Systeme mit komprimierten Volterra-Reihen*. Dissertation, VDI Verlag, Düsseldorf, 1995.
- [132] KURZE, M.: *Modellbasierte Regelung von Robotern mit elastischen Gelenken ohne abtriebsseitige Sensorik*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2008.
- [133] LANDAU, I.D., R. LOZANO und M. M'SAAD: *Adaptive Control*. Springer Verlag, 1998.
- [134] LANG, M.: *Signaldarstellung*. Vorlesungsskriptum Lehrstuhl für Mensch-Maschine-Kommunikation der TU-München, 1999.
- [135] LARSEN, P. M.: *Industrial Applications of Fuzzy Logic*. International Journal of Man-Machine Studies, vol. 12, no. 1, pp. 3–10, 1980.
- [136] LEE, C. C.: *Fuzzy Logic in Control Systems: Fuzzy Logic Controller*. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-20, No. 2, 1990.
- [137] 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.
- [138] LEVENBERG, K.: *A Method for the Solution of Certain Problems in Least Squares*. Quarterly of Applied Mathematics, 2:164–168, 1944.
- [139] LEVINE, W.S. (Herausgeber): *The Control Handbook*. CRC Press LLC (with IEEE Press), Boca Raton, Florida, USA, 1. Auflage, April 1996.
- [140] LJUNG, L.: *System Identification*. Prentice Hall, 1999.
- [141] LJUNG, L. und T. SÖDERSTRÖM: *Theory and Practice of Recursive Identification*. MIT Press, 1987.
- [142] LUDYK, G.: *Theoretische Regelungstechnik 2*. Springer-Verlag, 1995.
- [143] LUENBERGER, D.G.: *Observing the State of Linear Systems*. IEEE Transactions on Military Electronics, pp 74–80, 1964.
- [144] LUNZE, J.: *Regelungstechnik 1 - Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen* (4. erweiterte und überarbeitete Auflage). Springer-Verlag Berlin Heidelberg, 2004.
- [145] LUNZE, J.: *Regelungstechnik 2 - Mehrgrößensysteme, Digitale Regelung* (3. neu bearbeitete Auflage). Springer-Verlag Berlin Heidelberg, 2005.
- [146] MAEDA, M. und S. MURAKAMI: *A Self-tuning Fuzzy Controller*. Fuzzy Sets and Systems, vol. 51, 1992.
- [147] MAMDANI, E.: *Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis*. IEEE Transactions on Computers, Vol. C-26, No. 12, 1977.
- [148] MAREELS, I., POLDERMAN J. W.: *Adaptive Systems, An Introduction*. Birkhäuser, 1996.
- [149] MARQUARDT, D.: *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. SIAM Journal, 11:431–441, 1963.
- [150] MCCULLOCH, W. S. und W. PITTS: *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics 5, pp 115–133, auch in J. Anderson, E. Rosenfeld (Eds): Neurocomputing: Foundations of Research, Kap. 2, pp. 18–28, MIT Press, 1988, 1943.

- [151] METROPOLIS, N., M. ROSENBLUTH, A. ROSENBLUTH, A. TELLER und E. TELLER: *Equation of state calculations by fast computing machines*. Journal of Chemical Physics, 21:1087–1092, 1953.
- [152] MIDDLETON, R. H., GOODWIN G. C.: *Digital Control and Estimation, A Unified Approach*. Prentice-Hall, Inc., 1990.
- [153] MINSKY, M. und S. PAPERT: *Perceptrons - Expanded Edition : An Introduction to Computational Geometry*. MIT-Press pp. 1-20 und 73, auch in J. Anderson, E. Rosenfeld (Eds): Neurocomputing: Foundations of Research, Kap. 13, pp. 161-170, MIT Press, 1988, 1969.
- [154] MØLLER, M.: *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Neural Networks, 6(4):525–533, 1993.
- [155] MØLLER, M.: *Efficient training of feed-forward neural networks*. Ph.D. thesis, Computer Science Department, Aarhus University, Arhus, Denmark, 1997.
- [156] MURRAY-SMITH, R.: *A Local Linear Model Network Approach to Nonlinear Modelling*. Ph.D. Thesis, University of Strathclyde, UK, 1994.
- [157] MURRAY-SMITH, R. und T. A. JOHANSEN: *Multiple Model Approaches to Modelling and Control*. Taylor & Francis, 1997.
- [158] NARENDRA, K. S. und A. M. ANNASWAMY: *Stable Adaptive Systems*. Prentice Hall, 1989.
- [159] NARENDRA, K.S. und J. BALAKRISHNAN: *Improving transient response of adaptive control systems using multiple models and switching*. Proc. of the 7th Yale Workshop on Adaptive and Learning Systems, Yale University, 1992.
- [160] NARENDRA, K.S. und J. BALAKRISHNAN: *Improving transient response of adaptive control systems using multiple models and switching*. IEEE Transactions on Automatic Control, 39(9), pp. 1861-1866, 1994.
- [161] NARENDRA, K.S. und J. BALAKRISHNAN: *Adaptive Control Using Multiple Models*. IEEE Transactions on Automatic Control, Vol. 42, No. 2, 1997.
- [162] NARENDRA, K.S., O.A. DRIOLET, M.J. FEILER und G. KOSHY: *Adaptive Control Using Multiple Models, Switching and Tuning*. International Journal of Adaptive Control and Signal Processing, vol. 17, pp.87-102, 2003.
- [163] NARENDRA, K.S. und A.M. LEE: *Stable direct adaptive control of time-varying discrete-time systems*. Technical Report No. 8720, Center for Systems Science, Yale University, New Haven, 1987.
- [164] NARENDRA, K. S., THATHACHAR M. A. L: *Learning automata, an introduction*. Prentice Hall, 1989.
- [165] NELDER, J.A. und R. MEAD: *A simplex method for function minimization*. Computer Journal, 7:308–313, 1965.
- [166] NELLES, O.: *LOLIMOT-Lokale, lineare Modelle zur Identifikation nichtlinearer, dynamischer Systeme*. at - Automatisierungstechnik, Nr. 45, 4/1997, Oldenburg, Wien, 1997.
- [167] NELLES, O.: *Nonlinear System Identification with Local Linear Neuro-Fuzzy Models*. Ph.D. Thesis, TU Darmstadt, 1999.
- [168] NELLES, O.: *Nonlinear System Identification*. Springer-Verlag, 2001.
- [169] NELLES, O.: *Nonlinear System Identification*. Springer Verlag, Berlin, Heidelberg, New York, 2001.

- [170] NELLES, O., S. ERNST und R. ISERMANN: *Neuronale Netze zur Identifikation nichtlinearer Systeme: Ein Überblick.* at - Automatisierungstechnik, Nr. 45, 6/1997, Oldenburg, Wien, 1997.
- [171] NELLES, O., O. HECKER und R. ISERMANN: *Automatische Strukturselektion für Fuzzy-Modelle zur Identifikation nichtlinearer, dynamischer Prozesse.* at - Automatisierungstechnik, Nr. 46, 6/1998, Oldenburg, Wien, 1998.
- [172] NISSEN, V.: *Einführung in evolutionäre Algorithmen.* Vieweg, Braunschweig, 1997.
- [173] NOCEDAL, J. und S. J. WRIGHT: *Numerical Optimization.* Springer-Verlag, 1999.
- [174] NOCEDAL, J. und S.J. WRIGHT: *Numerical Optimization.* Springer-Verlag, 1999.
- [175] NOSSEK, J. A.: *Vorlesungsmanuskript: Netzwerktheorie 1.* Lehrstuhl für Netzwerktheorie und Schaltungstechnik der TU München, 1997.
- [176] PAPAGEORGIOU, M.: *Optimierung. Statische, dynamische, stochastische Verfahren für die Anwendung.* Oldenburg Verlag, 1991.
- [177] PAPAGEORGIOU, M.: *Optimierung. Statische, dynamische, stochastiche Verfahren für die Anwendung.* Oldenburg Verlag, München, Wien, 1991.
- [178] PFEIFFER, B. M.: *Selbsteinstellende klassische Regler mit Fuzzy-Logik.* Automatisierungstechnik, Jg. 42, Heft 2, 1994.
- [179] PLAUT, D. C., NOWLAN S. J. und G. E. HINTON: *Experiments on learning by back propagation.* Technical Report CMU-CS-86-126, Seiten 1–40, 1986.
- [180] POHLHEIM, H.: *Evolutionäre Algorithmen.* Springer, Berlin, 2000.
- [181] POLAK, E. und G. RIBIÈRE: *Note sur la convergence de methodes de directions conjugueés.* Revue Francaise d’Informatique et de Recherche Operationnelle, 16:35–43, 1969.
- [182] RECHENBERG, I.: *Evolutionsstrategie '94.* Frommann Holzboog, Stuttgart-Bad Cannstatt, 1994.
- [183] REED, R. D. und R. J. MARKS: *Neural Smithing: Supervised Learning in Feed-forward Artificial Networks.* MIT Press, 1998.
- [184] REINSCHKE, K.: *Lineare Regelungs- und Steuerungstheorie.* Springer Berlin Heidelberg New York, 2006.
- [185] REKLAITIS, G. V., A. RAVINDRAN und K. M. RAGSDELL: *Engineering Optimization – Methods and Applications.* John Wiley & Sons, 1983.
- [186] RENDERS, J. und S. FLASSE: *Hybrid Methods using Genetic Algorithms for Global Optimization.* IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, 26(2):243–258, April 1996.
- [187] RITTER, K.: *Nichtlineare Optimierung.* Vorlesungsskript, Institut für Angewandte Mathematik und Statistik, Technische Universität München, München, 1992.
- [188] ROOIJ, A. VAN, L. JAIN und R. JOHNSON: *Neural Network Training using Genetic Algorithms.* World Scientific, 1996.
- [189] ROSENBALTT, F: *The perceptron: a probabilistic model for information storage and organization in the brain.* Psychological review 65, pp 386-408, auch in J. Anderson, E. Rosenfeld (Eds): Neurocomputing: Foundations of Research, Kap. 8, pp. 92-114, MIT Press, 1988, 1958.

- [190] RUMELHART, D. E., G. E. HINTON und R. J. WILLIAMS: *Learning internal representations by error propagation.* D.E. Rumelhart, J.L. McClelland (Eds) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, auch in J. Anderson, E. Rosenfeld (Eds): Neurocomputing: Foundations of Research, Kap. 41, pp. 696-700, MIT Press, 1988, 1986.
- [191] RUMELHART, D. E. und J. L. MCCLELLAND: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 1: Foundations.* The MIT-Press, 1986.
- [192] SCALES, L.E.: *Introduction to non-linear optimization.* Macmillan, London, 1985.
- [193] SCHÄFFNER, C.: *Analyse und Synthese neuronaler Regelungsverfahren.* Dissertation, Lehrstuhl für Elektrische Antriebssysteme der TU München, 1996.
- [194] SCHETZEN, M.: *The Volterra and Wiener Theories of Nonlinear Systems.* John Wiley & Sons, 1980.
- [195] SCHLURMANN, J.: *Elektrisches System und Regelung des Optimierten CVT-Hybrid-Antriebsstranges.* Dissertation, Lehrstuhl für Elektrische Antriebssysteme, TU München, 2009.
- [196] SCHMIDT, G.: *Regelungs- und Steuerungstechnik 2.* Vorlesungsskriptum Lehrstuhl für Steuerungs- und Regelungstechnik der TU-München, 1996.
- [197] SCHMIDT, G.: *Lernverfahren in der Automatisierungstechnik.* Vorlesungsskriptum Lehrstuhl für Steuerungs- und Regelungstechnik der TU München, 1997.
- [198] SCHÖNEBURG, E., F. HEINZMANN und S. FEDDERSEN: *Genetische Algorithmen und Evolutionsstrategien.* Addison-Wesley, 1994.
- [199] SCHRÖDER, D.: *Verfahren zur Beobachtung nicht meßbarer Größen nichtlinearer dynamischer Systeme.* Patent 19531692, 1995.
- [200] SCHRÖDER, D.: *Intelligent Observer and Control Design for Nonlinear Systems.* Springer Verlag, Berlin, 2000.
- [201] SCHRÖDER, D.: *Elektrische Antriebe – Regelung von Antriebssystemen.* Springer–Verlag, 2001.
- [202] SCHRÖDER, D.: *Elektrische Antriebe - Grundlagen (3., erw. Auflage).* Springer Verlag, Berlin, 2007.
- [203] SCHRÖDER, D.: *Elektrische Antriebe - Regelung von Antriebssystemen (3., bearb. Auflage).* Springer Verlag, Berlin, 2009.
- [204] SCHRÖDER, D.: *Elektrische Antriebe 2, Regelung von Antrieben.* Springer–Verlag, 1995.
- [205] SCHRÖDER, D.: *Intelligent Observer and Control Design for Nonlinear Systems.* Springer–Verlag, 2000.
- [206] SCHRÖDER, D.: *Elektrische Antriebe – Regelung von Antriebssystemen.* Springer–Verlag, 2001.
- [207] SCHRÖDER, D.: *Elektrische Antriebe – Regelung von Antriebssystemen.* Springer–Verlag, 2008.
- [208] SCHUSTER, H.: *Hochverstärkungsbasierte Regelung nichtlinearer Antriebssysteme.* Dissertation, Lehrstuhl für Elektrische Antriebssysteme, Technische Universität München, 2009.
- [209] SCHUSTER, H., C.M. HACKL, C. WESTERMAIER und D. SCHRÖDER: *Funnel Control for Electrical Drives with Uncertain Parameters.* 7th International

- Power Engineering Conference - IPEC Proceedings, 2005 Singapore, Proceedings CDROM, Nov. 29 - Dec. 2 2005.
- [210] SCHUSTER, H., C. WESTERMAIER und D. SCHRÖDER: *High-Gain Control of Systems with Arbitrary Relative Degree: Speed Control for a Two Mass Flexible Servo System*. Proceedings of the 8th IEEE Int. Conf. on Intelligent Engineering Systems INES, Cluj-Napoca, Romania, Seiten 486 – 491, September 2004.
- [211] SCHUSTER, H., WESTERMAIER C. SCHRÖDER D.: *Non-Identifier-Based Adaptive Control for a Mechatronic System Achieving Stability and Steady State Accuracy*. Proceedings of the 2006 IEEE International Conference on Control Applications, CCA, München, Deutschland, Seiten 1819–1824, 2006.
- [212] SHANNO, D. F.: *Conditioning of Quasi-Newton Methods for Function Minimization*. Mathematics of Computation, 24:647–656, 1970.
- [213] SHEPHERD, A. J.: *Second-order methods for neural networks: fast and reliable training methods for multi-layer perceptrons*. Springer, London, 1997.
- [214] SHEWCHUK, JONATHAN: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. School of Computer Science, 1994.
- [215] SIMOVICI, D. A., TENNEY R. L.: *Theory of Formal Languages with Applications*. World Scientific, 1999.
- [216] SÖDERSTRÖM, T. und P. STOICA: *System Identification*. University Press Cambridge, 1989.
- [217] SONTAG, E.: *Adaptation and regulation with signal detection implies internal model*. Systems & Control Letters, 50(2):119–126, 2003.
- [218] SPECHT, D.: *A General Regression Neural Network*. IEEE Transactions on Neural Networks. Vol 2, pp. 568-576, November, 1991.
- [219] SPELLUCCI, P.: *Numerische Verfahren der nichtlinearen Optimierung*. Birkhäuser, 1993.
- [220] SPENDLEY, W., G.R. HEXT und F.R. HIMSWORTH: *Sequential Application of Simplex Designs in Optimization and Evolutionary Operation*. Technometrics, 4:441–461, 1962.
- [221] STROBL, D.: *Identifikation nichtlinearer mechatronischer Systeme mittels neuronaler Beobachter*. Hertbert Utz Verlag Wissenschaft, 1999.
- [222] SZU, H. und R. HARTLEY: *Fast Simulated Annealing*. Physics Letters A, 122(3,4):157–162, Juni 1987.
- [223] TAO, G.: *Adaptive Control Design and Analysis*. John Wiley and Sons, Inc., 2003.
- [224] TILLI, T.: *Fuzzy-Logik: Grundlagen, Anwendungen, Hard- und Software*. Franzis Verlag, 1991.
- [225] TOLLENAERE, T.: *SuperSAB: Fast adaptive back propagation with good scaling properties*. Neural Networks, 3(4):561–573, 1990.
- [226] TSAKALIS, K. und P. IOANNOU: *Adaptive Control of linear time-varying plants: A new model reference control structure*. Technical Report No. 86-10-1, University of Southern California, 1987.
- [227] TSETLIN, M. L.: *Automaton Theory and Modeling of Biological Systems*. Academic Press, 1973.
- [228] TZIRKEL-HANCOCK, E., FALLSIDE F.: *A Direct Control Method For a Class of Nonlinear Systems Using Neural Networks*. Technical Report CUED/F-

- INFENG/TR.65, Cambridge University Engineering Department. Cambridge, England, 1991.
- [229] UNBEHAUEN, H.: *Regelungstechnik I*. Vieweg Verlag, 1992.
- [230] UNBEHAUEN, H. und G. P. RAO: *Identification of Continuous Systems*. North-Holland Systems and Control Series, Vol. 10, 1987.
- [231] VACHENAUER, PETER: *Springers Mathematische Formeln, Taschenbuch für Ingenieure, Naturwissenschaftler, Wirtschaftswissenschaftler*. Springer, 1997.
- [232] VOLTERRA, W.: *Theory of Functionals*. Blackie and Sons, 1930.
- [233] WAHLBERG, B.: *System Identification using Laguerre Models*. IEEE Transactions on Automatic Control, Vol. 36, pp. 551–562, 1991.
- [234] WAHLBERG, B.: *System Identification using Kautz Models*. IEEE Transactions on Automatic Control, Vol. 39, pp. 1276–1282, 1994.
- [235] WALTER, W.: *Gewöhnliche Differentialgleichungen*, Band (7. Auflage). Springer Berlin Heidelberg New York, 2000.
- [236] WANG, L.: *Stable Adaptive Fuzzy Control of Nonlinear Systems*. IEEE Transactions on Fuzzy Systems, vol. 1,, 1993.
- [237] WELLERS, M. und N. KOSITZA: *Identifikation nichtlinearer Systeme mit Wiener- und Hammerstein- Modellansätzen auf Basis der Volterra-Reihe*. at - Automatisierungstechnik, Nr. 47, 1999, Oldenburg, Wien, 1999.
- [238] WERBOS, P. J.: *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
- [239] WESTERMAIER, C.: *Adaptive Drehzahlregelung für ein Zwei-Massen-Systems*. Diplomarbeit, Lehrstuhl für Elektrische Antriebssysteme, Technische Universität München, 2003.
- [240] WESTERMAIER, C.: *Hochdynamisches Regeln unbekannter nichtlinearer Antriebssysteme mittels Model-Reference-Adaptive-Control unter Beachtung der Stabilität*. Dissertation, Lehrstuhl für Elektrische Antriebssysteme und Leistungselektronik, Technische Universität München, 2010.
- [241] WESTERMAIER, C., H. SCHUSTER UND D. SCHRÖDER: *Controlling the Loop-Gain for Robust Adaptive Control of a Mechatronic System*. Proceedings of the 5th IEEE International Conference on Technology and Automation, Thessaloniki, Griechenland, Seiten 160–165, 2005.
- [242] WESTERMAIER, C., H. SCHUSTER UND D. SCHRÖDER: *Industrietauglichkeit adaptiver Konzepte: Robuster MRAC-Ansatz durch Gain-Regelung*. Tagungsband des 2005 Internationalem Forum Mechatronik, Augsburg, Deutschland, Seiten 511–534, 2005.
- [243] WILLIAMS, R. J. und D. ZIPSER: *A learning algorithm for continually running fully recurrent neural networks*. Neural Computation, Vol. 1, No.2, pp:270–280, 1989.
- [244] WONHAM, W.: *Towards an abstract internal model principle*. IEEE Trans. on Systems, Man & Cybernetics, 6(11):735–740, 1976.
- [245] WONHAM, W.: *Linear Multivariable Control: a Geometric Approach*. 1979.
- [246] ZADEH, L. A.: *Fuzzy Sets*. Information and Control, Vol. 8, 1965.
- [247] ZADEH, L. A.: *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, No. 1, 1973.
- [248] ZELL, A.: *Simulation Neuronaler Netze*. Addison-Wesley, 1994.

- [249] ZHAO, Z. Y., M. TOMIZUKA und S. ISAKA: *Fuzzy Gain Scheduling of PID Controllers*. IEEE Transactions of Systems, Man, and Cybernetics, vol. 2, 1993.
- [250] ZIMMERMANN, H.-J.: *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, 1991.

# Stichwortverzeichnis

- $\varepsilon$ -Optimalität, 604
- Übergangswahrscheinlichkeit, 593
- Linear Reward-Inaction* Lerngesetz, 610
- Linear Reward-Penalty* Lerngesetz, 606
- 1-Schritt-Prädiktionsfehler, 225
  
- Ableitung entlang von Lösungstrajektorien, *siehe* Orbital derivative
- Abstiegsrichtung, 346
- Adaption, 40
- Adaptionsfehler, 45
- Adaptionsgesetz, 631
- Adaptive Multi-Modellregelung, *siehe* Multiple-Model Adaptive Control
- Adaptive Regelung, 487
- adaptiver Regler, 632
- adaptiver Stabilisator, 631
- Adaptives Umschalten, *siehe* Switching and Tuning
- Adjunktivität, 770
- Affensattel, 55
- aktive Bedämpfung, 657
- Aktivierung, 40
- Aktivierungsdynamik, 252
- Aktivierungsfunktion, 44, 78
- Aktivierungsvektor, 44
- algebraische Darstellung, 40
- ALIS, 337, 341, 343, 357, 376
- amplitudenmoduliertes Pseudo-Rausch-Binär-Signal, 278
- Anfangswertproblem, *siehe* Initial Value Problem (IVP)
- Anregesignal, 219
- Anregung, ausreichende, 99, 100
- Anregungsphase, 652
- Anregungssignale, 276
- Antriebsmaschine, 10
  
- Antriebsmaschinendrehzahl, 13
- Antwortlänge, 228
- Approximation
  - inhärenter Fehler, 44, 58
- Arbeitsmaschine, 10
- Arbeitsmaschinendrehzahl, 12
- arithmetische Implikation, 810
- ARMA-Modell, 490, 496
- ARX-Modell, 222, 234
- Assoziativität, 770
- asymptotische Modellfolgeregelung, 459
- asymptotische Stabilität, 98
- Attraktionsgebiet, *siehe* Basin (region) of attraction
- Aufdatierungsformel, 368, 371–373, 376
- Ausgangsfehler, 94
- Ausgangsfehleranordnung, 91
- Ausgangsfehlermodell, 222, 225
- Auto Regressive, 221
- Auto Regressive Moving Average, 221, 490, 496
- Auto Regressive Moving Average with eXogenous input, 221
- Auto Regressive with eXogenous input, 221
- Automat, 599
- Autoregressive with Exogenous Input, 222
- Axon, 77
  
- Backpropagation, 38, 111
- Backpropagation-Regel, 102
- Basin (region) of attraction, 581
- Basisfunktion
  - Gauß'sche Glockenkurve, 43
  - lokale, 43
  - Manhattan-Distanz, 43
  - Zentrum, 44

- Batch Lernen, 106  
 Beharrungswert, 260  
 Beobachtbarkeit, 138  
 Beobachter  
   – neuronal, 467  
 Beobachterentwurf, 287, 291  
 Beobachterentwurf bei messbarem Eingangsraum, 129  
 Beschleunigungsmoment der Masse, 10  
 BFGS, 372, 373, 376  
 BINF, 426  
 Black-Box-Modelle, 245  
 blockorientierte nichtlineare Modelle, 247  
 Bode-Diagramm, 12, 16  
 Boltzmann-Verteilung, 388  
 Boolesche Implikation, 810  
 Broyden, 371  
 Byrnes-Isidori Form, 419  
 Byrnes-Isidori Norm, 426  
 Byrnes-Isidori Normalform, 426  
 Certainty-Equivalence Prinzip, 518  
 Chapman-Kolmogorov-Identität, 594  
 Cholesky-Zerlegung, 232  
 Clipping, 799  
 compositional rule of inference, 806  
 Continuity (of a function), 572  
 Correlated noise, 540  
 Coulombsches Reibungsmodell, 37  
 Cybenko, 82  
 Dämpfung, Übertragungsmoment, 10  
 Dämpfung  
   – mechanische, 10  
 De Morgan, 770  
 Defuzzyfizierung, 773  
 Dehnungsdiagramm, 32  
 delta-bar-delta-Algorithmus, 352  
 Dendriten, 77  
 deterministisch, 220  
 deterministische Optimierungsverfahren, 319  
 DFP, 371  
 Die Quasi-Newton-Bedingung, 369  
 Differentialgleichung, 26  
 Dilationsoperator, 771  
 Distributivität, 770  
 Disturbance Rejection, 533  
 Drehfedersteifigkeit, 10  
 Drehwinkel, 10  
 Dreimassensystem, 25  
 dynamische Neuronale Netze, 249  
 dynamische Nichtlinearität, 281, 283, 285, 287, 291  
 dynamischer Aktivierungsvektor, 262  
 dynamischer Operator, 677  
 Eigenwert, 378  
 Ein- Ausgangslinearisierung, 439  
 Ein-Ausgangsstabilität, *siehe* Input-Output Stability  
 Eindeutigkeit der Adaption, 45  
 eindimensionale statische Nichtlinearität, 51, 68, 82  
 Eingangsraum, 45  
   – mehrdimensionaler, 49  
 Einheitsmatrix, 193  
 Elman-Netzwerk, 253  
 Entscheidungsfunktion, 78  
 Ergodizität, 594  
 Error Reference Control, 741  
   – erweiterte Referenz, 742  
   – virtueller Schlauch, 742  
 Erweiterte Regelung mit multiplen Modellen, 625  
 Erweiterung der Systemordnung, *siehe* Order augmentation  
 Evolutionäre Algorithmen, 393  
 Evolutionsstrategien, 392, 398, 399  
 Extended least-squares algorithm (ELS), 546  
 Extrapolation, 48  
 Extrapolationsverhalten, 52, 55  
 Führungsintegrator, 671  
 Faltungssumme, 228  
 Farbiges Rauschen, *siehe* Correlated noise  
 fast sichere Konvergenz, *siehe* Konvergenz mit Wahrscheinlichkeit 1  
 Feder, Übertragungsmoment, 10  
 Fehler  
   – quadratischer, 96, 97  
 Fehlerübertragungsfunktion, 287  
 Fehlerübertragungsfunktion, 129, 130  
 Fehlermodell 1, 150

- Fehlermodell 2, 152
- Fehlermodell 3, 134, 152, 289
- Fehlermodell 4, 136, 154, 290
- Fehlermodelle, 99, 150
- Fehlerschranke, 641
- Finite Impulse Response, 221, 222
- Finite Impulse Response Model, 228
- FIR-Modell, 222, 238
- Flächenschwerpunktverfahren, 799
- Flache Plateaus, 117
- Fletcher-Reeves, 355
- Fourierkoeffizienten, 102
- Funktional, 644
- Funktionsapproximation
  - algebraisch, 40
  - konnektionistisch, 42
  - Methoden, 40
  - tabellarisch, 41
  - universelle, 44
- Funktionsapproximatoren, 37
  - Bewertung, 87
- Funnel Control, 647
- Funnel-Control, 640, 644, 648, 649, 655, 656, 659, 662, 693, 699
  - Anforderungen an LTI SISO System, 713
  - Dämpfungsskalierung, 728
  - Erlaubte Referenzsignale, 708
  - Error Reference Control, 741
    - erweiterte Referenz, 742
    - virtueller Schlauch, 742
  - Kundenanforderungen, 723, 724
    - Überschwingweite, 723
    - Anregelzeit, 723
    - Ausregelzeit, 723
  - Mindestverstärkung, 727
  - Operatorklasse, 711
  - Regelfehler, 700
  - Reglerverstärkung, 700
  - Skalierungsfunktion, 701, 727
  - Stellgröße, 699
  - Systemanforderungen, 710
  - Systemklasse  $\mathcal{S}$ , 712
  - Trichterfunktion, 702
  - Trichterrand, 701, 702
    - angepasster Standard-, 705
    - exponentieller, 705
- Gaußglocken-, 706
- simpler, 707
- unendlicher Standard-, 703
  - Vertikale Distanz, 700
  - Zukünftige Distanz, 700
  - zukünftige Distanz, 731
  - Zwei-Massen-System, 748
    - aktive Dämpfung, 748
    - Dämpfungskoeffizient, 750
    - Eigenfrequenz, 750
    - erweitertes Hilfssystem, 751
    - Hilfsauskopplung, 751
    - Hilfsfehler, 756
    - Hilfssollwertverlauf, 756
    - instantane Verstärkung, 754
    - Nulldynamik, 754
    - Vorverstärkung, 756
- Funnel-Operator, 644
- Fuzzy-Logik
  - Implikationsoperatoren, 810
  - logische Operatoren, 778
  - logisches Schließen, 777
  - theoretische Darstellungsweise, 804
- Fuzzy-ODER, 789
- Fuzzy-Regelung, 761
- Fuzzy-Regler, 793
  - Beispiel, 773
- Fuzzy-UND, 789
- Fuzzyfizierung, 773, 776, 795
- Gaines-Implikation, 810
- Gauß'sche Glockenkurve, 43, 46
- Gauss-Newton-Verfahren, 367
- General-Regression-Neural-Network, 38, 48
- Generation, 394, 395
- Genetische Algorithmen, 394
- Genotyp, 395
- Gewicht, 40
- Gewichtsvektor, 44
- Glättungsfaktor, 46, 50, 53
  - normiert, 48
- Gleichungsfehler, 222, 234
- Gleichungsfehlermodelle, 222, 234
- Gleitreibung, 22, 25, 37
- global integrierende Systeme, 294
- Goguen-Implikation, 809, 810

- Goldener Schnitt, 330, 332, 333, 335
- Gradient, 346
- Gradientenabstieg, 114, 345–352
- Gradientenabstieg mit Liniensuche, 352
- Gradientenabstieg mit Momentumterm, 349
- Gradientenabstieg mit variabler Schrittweite, 351
- Gradientenabstiegsverfahren, 51, 94, 108
- Gradientenberechnung, 111
- Grey–Box–Modelle, 246
- GRNN, 38, 48, 62, 96
  - Extrapolationverhalten, 55
- Grundintervall, 327, 337
- Gültigkeitsgrad, 776
- Höhenlinie, 348
- Haftriebung, 22, 25, 37
- Hammerstein-Modell, 247
- HANN, 38, 58
  - erweiterte Struktur, 61
  - Grundstruktur, 60
  - Lerngesetz, 99
  - Parameterkonvergenz, 100
  - Schätzwert, 61
  - Stabilität, 100
- harmonisch aktiviertes Netz, 38, 58, 311
- Hessematrix, 358, 367, 378
- hidden layers, 79
- High-Frequency-Gain, 649
- high-gain-fähige Strecke, 644
- high-gain-Fähigkeit, 639, 647
- High-Gain-Feedback, 629, 630
- Hochpassfilter, 671, 672, 677, 687, 690
- hochverstärkungsbasierte Regelung, 629
- homogener Generator, 653
- Hooke-Jeeves-Tastverfahren, 324
- Hornik, 82
- hunting-Effekt, 24
- Hurwitz, 674, 678
- Hysterese
  - Identifikation, 171, 177
  - Modellierung, 171
- Identifikation, 40, 160, 218, 276
  - mechatronisches Antriebssystem, 307
  - Plattenaufbau dynamisch, 407
  - Reibkennlinie, 383
- stochastische Optimierungsverfahren, 406
- Identifikation nichtlinearer dynamischer Systeme, 245
- Identifikations-Modelle, Grundlagen, 217
- Identifikationsansatz, allgemeiner, 270
- Identifikationsansatz, Erweiterung, 271
- Identifikationsbeispiele, 234
- Identifikationsfehler, 499
- Identifikationsverfahren, 203
- Implementierung, 382
- Impulsantwort, 228
- Individuum, 394, 395
- Inferenz, 773, 797
- ingenieurtechnischer Ansatz, 663
- inhärenter Approximationsfehler, 44
- Initial Value Problem (IVP), 573
- Input-Output Stability, 586
- INT, 328, 333, 335
- Integrale Zustandsregelung, 20
- intelligenter Ansatz, 671
- Internal Model Principle, 650
- Internes Modell, 650, 659, 660
- internes Modell, 654, 655
- Interpolationsverfahren, 327
- Interpolationsverhalten, 53
- Intervallsuche, 327, 329
- Intervallvergrößerung, 337, 339
- Intervallverkleinerung, 327, 330, 337, 340
- Invarianzprinzip, *siehe* La Salle's Invariance Principle
- Involution, 770
- isiolierte Nichtlinearität, 127
- Jacobimatrix, 105, 367
- Jordan-Netzwerk, 252
- Jordanform, 647
- Kühlschema, 389
- Kausalität, 645
- Kautz–Filter, 231
- KG, 354
- kleinste Quadrate, 119
- Kohonen-Netze, 92
- Kommutativität, 769
- Kompensation, 165
- kompenatorische Operatoren, 783
- Konjugationskoeffizient, 355
- Konjugierter Gradientenabstieg, 354

- konnektionistische Darstellung, 40  
kontinuierliche Produktionsanlage, 32  
Kontrastverstärkungsoperator, 771  
Konvergenz, 349, 352, 366, 381  
Konvergenz in Verteilung, 598  
Konvergenz mit Wahrscheinlichkeit 1, 598  
Konvergenzgeschwindigkeit, 186  
Konvergenzrate, 381  
Konzentrationsoperator, 771  
Kovarianzmatrix, 120  
kumulierte Fehlermaß, 106
- La Salle's Invariance Principle, 580  
Lagrange-Interpolation, 337  
Laguerre-Filter, 231  
lambda-Tracker, 631  
Larsen-Implikation, 811  
Least Squares, 119  
leere Menge, 763  
Lernen, überwachtes, 91  
Lernen, bestärkendes, 91  
Lernen, unüberwachtes, 91  
Lernfähiger Beobachter, 127  
Lernfähigkeit, 42  
Lernfaktor, 95, 96, 100  
Lernfehler  
– HANN, 59  
– RBF-Netz, 45  
Lerngesetz, 96  
– HANN, 99  
Lernschrittweite, 95  
Levenberg-Marquardt-Algorithmus, *siehe LM*  
Lie-Ableitungen, 427  
Linear disturbance model, 534  
Linear vector space, 567  
lineare dynamische Modellstrukturen, 220  
lineare Gewichte, 225  
lineare Konvergenz, 347, 381  
lineare Modellstruktur, 217  
Linearer Vektorraum, *siehe* Linear vector space  
Lineares Störmodell, *siehe* Linear disturbance model  
Linearkombination, 665  
linguistische Variable, 775  
linguistischer Wert, 804
- Liniensuche, 326, 328, 337, 352  
Lipschitz continuity, 575  
Lipschitz Stetigkeit, *siehe* Lipschitz continuity  
LM, 377, 379  
Local-Linear-Model-Tree, 38, 62, 256  
logisches Schließen, 776  
Logistikfunktion, 79  
lokale Basisfunktion, 43  
Lokale Minima, 116  
lokale Optimierungsverfahren, 319  
lokale Zuordnung, 45  
LOLIMOT, 38, 62, 256  
Look-Up Table, 41  
losebehaftetes Zweimassensystem, 209  
Losemodellierung, 205  
Luenberger-Beobachter, 192  
Lyapunov, 97, 100  
Lyapunov Function, 578  
Lyapunov Funktion, *siehe* Lyapunov Function
- Mamdani-Implikation, 811  
Mamdani-Regler, 794  
Manhattan-Distanz, 43  
Markov-Kette, 591  
Matrix norm, 569  
Matrixinversionslemma, 121  
Matrizendarstellung, 11  
Maximumschwerpunktsmethode, 799  
Maximumsmittelwertverfahren, 799  
McCulloch-Pitts-Neuronen, 38  
mehrdimensionale Nichtlinearität, 32  
mehrdimensionaler Eingangsraum, 49  
mehrere Nichtlinearitäten, 147  
mehrschichtige Netze, 108  
– Lerngesetz, 102  
Mehrschichtiges MLP-Netz, 79  
Methode der kleinsten Quadrate, 119  
Min-Max-Operator, 789  
Minima  
– lokale, 116  
Minimum-Varianz-Regler, 519  
Minimumnorm, 123  
MLP, 38, 42, 77  
MLP-Fehlerfläche, 110  
Modalform, 647

- Model Reference Adaptive Control, 519
- Modellbasierte Adaptive Regelung, 487
- Modelle ohne Ausgangsrückkopplung, 227
- Modifikatoren, 770
- Modus Ponens, 805
- Momentumterm, 97, 109, 349, 351, 352
- monotone Verstärkung, 632
- monotones Adaptionsgesetz, 640
- Moving Average, 221
- Multi-Layer-Perzeptronen Netzwerk, 38, 42, 77
- Multiple-Model Adaptive Control, 555
- Mutation, 394, 397
- NARX, 250
- NARX-Modell, 255
- Netze mit externer Dynamik, 249
- Netze mit interner Dynamik, 249
- Neuron, 40
- Neuronale Netze, 38
- Neuronales Netz
  - Aktivierung, 40
  - Gewicht, 40
  - GRNN, 48
  - Neuron, 40
  - RBF-Netz, 45
  - Stützwert, 45
- Neuronenmodell-Darstellung, 252, 253
- Neuroregelung
  - referenzmodellbasierte, 479
- Newton-Suchrichtung, 366
- Newton-Verfahren, 365, 367, 368, 381
- NFIR, 250
- nicht messbarem Eingangsraum, 291
- nichtlinear, 420
- Nichtlineare Differentialgleichung, *siehe* Nonlinear differential equation
- Nichtlineare Ein-Ausgansdarstellung, *siehe* Nonlinear input-output representation
- nichtlineare Optimierung, 319
- nichtlineare Parameter, 225
- nichtlineare Regelungsnormalform, 426
- nichtlineare Umrichterdynamik, 308
- nichtlineare Zustandsdarstellung, 247
- Nichtlinearer Konjugierter Gradientenabstieg, *siehe* NKG
- nichtlinearer Signalflußplan, 32
- nichtlineares Ausgangsfehlermodell, 253
- nichtlineares dynamisches Teilsystem, 282
- nichtlineares Gleichungsfehlermodell, 253
- nichtlineares Zweimassensystem, 22, 677
- nichtparametrische Identifikationsverfahren, 217
- nichtparametrische Modelle, 218
- Nichtrekursiver Least-Squares-Algorithmus, 119
- Niveau-Menge, 765
- NKG, 354, 357
- NN
  - Bewertung, 87
- NOBF, 250
- NOE, 250
- NOE-Modell, 255
- Nonlinear Auto Regressive with eXogenous input, 250
- Nonlinear differential equation, 566
- Nonlinear Finite Impuls Response, 250
- Nonlinear input-output representation, 552
- Nonlinear Orthonormal Basis Function, 266
- Nonlinear Output Error, 250
- Nonlinear with Orthonormal Basis Functions, 250
- Norm einer Matrix, *siehe* Matrix norm
- Normalform, 426
- Normalized Radial Basis Function Network, 49
- Normpolynom 2. Ordnung, 15
- NRBF, 49
- NRNF, 426
- Nulldynamik, 411, 420, 646, 647, 652
- OBF-Modell, 240
- OE-Modell, 225, 236
- Operator, 645
- Operatorklasse, 645
- Optimierungsverfahren 0. Ordnung, 321
- Optimierungsverfahren 1. Ordnung, 345
- Optimierungsverfahren 2. Ordnung, 353
- Orbital derivative, 578
- Order augmentation, 536
- Ordnungsreduktion, 22

- orthogonalisierter Projektionsalgorithmus, 512
- Orthogonalität, 101
- Orthonormal Basis Function, 222
- Orthonormal Basis Function Model, 230
- Oszillationen, 118
- Output Error, 221
- Output Error Model, 225, 236
- Overfitting, 43
- paralleles Modell, 225
- Parameterfehler, 45, 59, 98
- Parameterkonvergenz
  - HANN, 100
  - RBF-Netz, 98
- Parameteroptimierung, 63, 319
- Parameteroptimierungsverfahren
  - Überblick, 319, 320, 353
  - 0. Ordnung, 321
  - 1. Ordnung, 345
  - 2. Ordnung, 353
  - Adaptive Lagrange Interpolation Search, *siehe* ALIS
  - BFGS, 372, 373, 376
  - Broyden, 371
  - delta-bar-delta-Algorithmus, 352
  - DFP, 371
  - eindimensionale Optimierung, 326
  - Gauss-Newton-Verfahren, 367
  - global, stochastisch, 319, 387
  - Goldene-Schnitt-Verfahren, 326, 330, 332, 333, 335
  - Gradientenabstieg, 345–347
  - Hooke-Jeeves-Tastverfahren, 324
  - Konjugierter Gradientenabstieg, *siehe* KG
  - Levenberg-Marquardt-Algorithmus, *siehe* LM
  - Liniensuche, 326, 333, 341
  - lokal, deterministisch, 319
  - Newton-Verfahren, 365, 367, 368, 381
  - Nichtlinearer Konjugierter Gradientenabstieg, *siehe* NKG
  - Quasi-Newton, 368
  - Quickprop, 352
  - Simplex-Methode, 321
  - Skalierter Konjugierter Gradientenabstieg, *siehe* SKG
  - SuperSAB, 352
- Parametervektor, 496
- parametrierbare Operatoren, 783
- Parametrierung, 37
- parametrische Identifikationsverfahren, 217
- parametrische Modelle, 217
- Particle Swarm Optimization (PSO), 400, 402, 404
- Partielle Ableitungen, 193
- Perceptron, 38
- Persistent Excitation, 99, 100, 137
- Perzeptron, 77
- Phänotyp, 395
- physikalische Interpretierbarkeit, 31, 45
- physikalische Modellbildung, 157
- physikalisches Modell, 25
- PI-Regler, 659, 660, 690
- Plausibilitätsgrad, 798, 811
- plausible Schließen, 806
- Pol-Nullstellen-Kürzung, 654
- Pol-Nullstellen-Kompensation, 633
- Polak-Ribi  re, 355, 357
- Polynominterpolation, 328
- Population, 395
- Prädiktionsmodell, 490
- Prktitor-ARMA-Modell, 497
- praxistauglicher Lsungsansatz, 647
- Problemstellungen, konomische, 221
- Produktionsanlagen
  - kontinuierlich, 32
- Prognose der hufigsten Transition, 625
- Projektionsalgorithmus, 501, 508
- Proportionale Zustandsregelung, 18
- proportionales Regelgesetz, 644
- Pseudo-Links-Inverse, 120
- Pseudo-Rausch-Bin  rsignal, 279
- PT<sub>1</sub>-Strecke, 630
- quadratische Konvergenz, 381
- quadratischer Fehler, 96, 97
- quadratisches Fehlerma  , 94
- Quasi-Newton-Bedingung, 370
- Quasi-Newton-Verfahren, 368, 371–373, 376
- Quickprop, 352
- Radial-Basis-Function-Netz, 38, 45
- RBF, 38, 96

- Extrapolationverhalten, 52
- RBF-Netz, 45
- Parameterkonvergenz, 98
- Schätzwert, 44
- Stabilität, 97
- Rechenaufwand, 382
- Referenzmodell-Regler, 519
- Referenzsignal, 448
- Regelung, 448
  - neuronal, 473
- Reglerentwurf, 439
- Regressionsmatrix, 119
- Regressionsvektor, 234, 238, 496
- Reibungskennlinie, 160, 208
- Rekombination, 394
- Rekonstruktion der blockorientierten Modellstruktur, 272
- Rekonstruktion der Hammerstein-Modellstruktur, 272
- Rekonstruktion der Wiener-Modellstruktur, 274
- Rekonstruktionsmatrix, 232
- rekursive Netze, 181, 249
  - Identifikation, 181
- rekurrenter Zustand, 596
- rekursiver Least-Squares-Algorithmus, 120, 512, 515
- Relativgrad, 409
- Relativgradreduzierung, 662
- Repräsentationsfähigkeit, 42
- RLS-Algorithmus, 512, 515
- s-Normen, 778, 779
- Sattelpunkt, 368
- Scaling, 800
- Schätzwert, 61
  - RBF-Netz, 44
- Schnittmenge, 767
- Schornsteinregler, 640
- Schrittweite, 323, 346, 358, 378
- selbstorganisierte Karten, 92
- Selektion, 394
- Selektionsdruck, 397
- seriell-paralleles Modell, 225
- Sichtbarkeit, 129
- sigmoide Funktionen, 78
- Signalflussplan des Zweimassensystems, 11
- Signumfunktion, 79
- Simplex-Methode, 321
- Simulated Annealing, 387, 390, 391
- Singleton, 767
- Skalierter Konjugierter Gradientenabstieg, *siehe* SKG
- Skalierung, 359, 378
- Skalierungsfaktor, 359, 378
- SKG, 358, 361, 365
- Sollwertskalierung, 666
- Speicheraufwand, 382
- SPR-Übertragungsfunktion, 134
- SPR-Bedingung, 134
- squashing Funktion, 78
- SRN
  - Anwendungsbeispiel, 205
  - Luenberger-Beobachter, 192
- SRNR, 479
- Störgrößenaufschaltung, 165
- Störgrößenunterdrückung, 651
- Störgrößenunterdrückung, *siehe* Disturbance Rejection
- stützende Menge, 764
- stützwertbasierte Approximation, 41
- Stützwerte, 45
- stabile referenzmodellbasierte Neuroregelung, 479
- Stabilisator, 631
- Stabilität
  - HANN, 100
  - RBF-Netz, 97
- Stabilität des OE-Modells, 227
- Standard-Strict-Implikation, 810
- stationäre Genauigkeit, 684
- stationärer Regelfehler, 644
- statische Funktionsapproximatoren, 37
- Stetigkeit, *siehe* Continuity (of a function)
- stick-slip-Effekt, 24
- Stoßkoeffizient, 206
- Stoßvorgang, 206
- Stochastic disturbance model, 538
- stochastisch, 220
- stochastische Optimierungsverfahren
  - Überblick, 387, 393, 400
  - Ant Colony Optimization (ACO), 400, 406

- Evolutionäre Algorithmen, 393
- Evolutionsstrategien, 392, 398, 399
- Genetische Algorithmen, 394
- Particle Swarm Optimization (PSO), 400, 402, 404
- Simulated Annealing, 387, 390, 391
- Systemidentifikation mit Neuronalen Netzen, 406
- stochastischer Prozess, 591
- Stochastisches Störmodell, *siehe* Stochastic disturbance model
- Stribeck-Effekt, 37
- strukturierte rekurrente Netze, 182
  - Parameteradaption, 184
- Strukturoptimierung, 63, 66
- Stützwert, 45
- Stufenfunktion, 79
- Suchrichtung, 326, 346
- Sugeno–Fuzzy–Regler, 793
- superlineare Konvergenz, 381
- SuperSAB, 352
- Swarm Intelligence, 400
- Switching and Tuning, 558
- Synapse, 77
- Synapsendynamik, 252
- t–Normen, 778
- tabellarische Darstellung, 40
- Taylorapproximation, 345, 365, 379
- technologische Aufgabenstellung, 32
- Teilsystem, 32
- Time Delay Neural Network, 255
- Time–varying parameters, 555
- Torsionsschwingung, 657
- Tracking, 448
- Trainingsdaten, 37
- Transferfunktion, 78
- Transformation, 183
- Transformationsmatrix, 424
- transienter Zustand, 596
- Transitionsmatrix, 601
- Transmissionswelle, 657
- Trichterrand, 641, 644
- Trichterrandfunktion, 649
- Tsetlin–Automat, 617
- unscharfe Aussage, 776
- unscharfe Logik, 762, 773
- Implikationsoperatoren, 810
- logische Operatoren, 778
- logisches Schließen, 777
- theoretische Darstellungsweise, 804
- unscharfe Menge, 762
- Mengenoperation, 767
- Modifikatoren, 771
- Rechengesetze, 769
- Varianz, 46
- Verallgemeinertes Hysteresemodell, 174
- Verallgemeinerungsfähigkeit, 42
- Verbundaussagen, 811
- Vereinigungsmenge, 768
- Vergessensfaktor, 122
- Verlassen guter Minima, 118
- versteckte Schichten, 79
- verzögerte Aktivierung, 135, 290
- virtuelle Stützwerte, 140
- virtuellen Parameter, 292
- viskose Reibung, 677
- Volterra–Funktionalpotenzreihe, 250, 260
- Volterra–Funktionalpotenzreihe mit Basisfunktionen, 266
- Volterra–Kerne, 260
- Vorschubantrieb, 156
- Wahrscheinlichkeit, 388
- Weierstrass, 82
- Weighted–Least–Squares–Algorithmus, 123
- White–Box–Modelle, 245
- Wiener–Modell, 247
- Winkelbeschleunigung, 10
- Winkelgeschwindigkeit, 10
- Wurzelortskurve, 636, 638
- Zeitkonstante, 634
- Zeitreihenmodelle, 221
- Zeitvariable Parameter, *siehe* Time–varying parameters
  - zeitvariantes Regelgesetz, 643
  - Zentrum einer Basisfunktion, 44
  - Zielkonflikt, 670
  - Zufallsvariable, 591
  - Zugehörigkeitsfunktion, 763, 795
  - Zugehörigkeitsgrad, 763
  - Zustandsgleichung, 10
  - Zustandsrückführung, 662

- zweidimensionale statische Nichtlinearität, 55, 73, 84
- Zweimassensystem, 9, 657–659, 662, 663, 690
  - Matrizendarstellung, 11
  - nichtlineares, 22
- Zweimassensystem mit Reibung, 141
- Zweimassensystem-Zustandsregelung, 18
- Zweimassensystemkaskadenregelung, 12, 13