



Fraunhofer
ISE

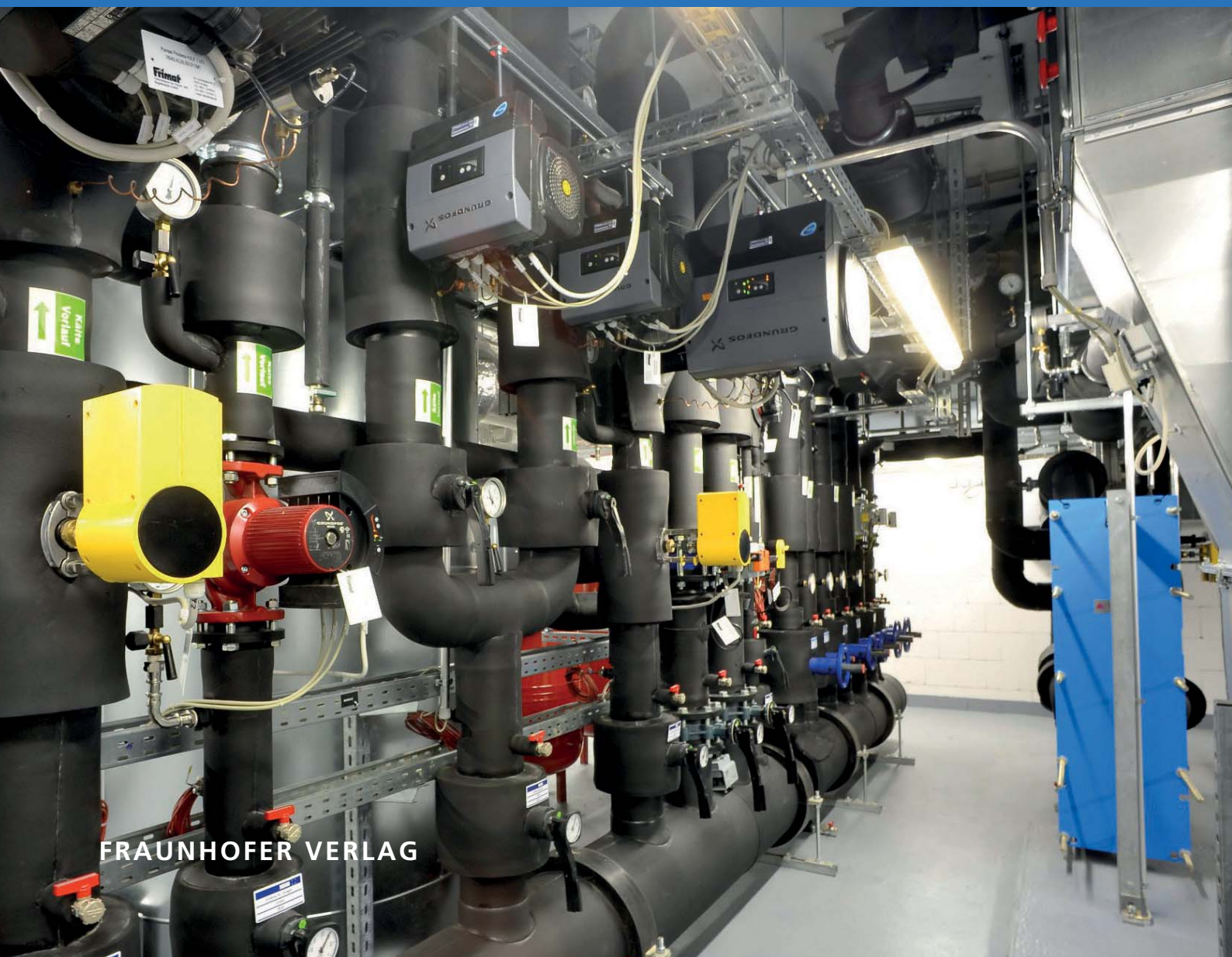
FRAUNHOFER-INSTITUT FÜR SOLARE ENERGIESYSTEME ISE

SOLARE ENERGIE- UND SYSTEMFORSCHUNG

Thorsten Müller-Eping

TENSORDEKOMPOSITION QUALITATIVER MODELLE ZUR FEHLERERKENNUNG

Anwendung in der Gebäudeautomation



FRAUNHOFER VERLAG

Fraunhofer-Institut für Solare Energiesysteme ISE

SOLARE ENERGIE- UND SYSTEMFORSCHUNG /
SOLAR ENERGY AND SYSTEMS RESEARCH

Tensordekomposition qualitativer Modelle zur Fehlererkennung

Anwendung in der Gebäudeautomation

Thorsten Müller-Eping

FRAUNHOFER VERLAG

Kontakt:

Fraunhofer-Institut für Solare Energiesysteme ISE
Heidenhofstraße 2
79110 Freiburg
Telefon +49 761/4588-5150
Fax +49 761/4588-9342
E-Mail info@ise.fraunhofer.de
URL www.ise.fraunhofer.de

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über www.dnb.de abrufbar.

ISSN: 2512-3629

ISBN (Print): 978-3-8396-1628-4

Reihe: »Solare Energie- und Systemforschung / Solar Energy and Systems Research«

D 90

Zugl.: Karlsruhe, KIT, Diss., 2019

Druck: Fraunhofer Verlag Mediendienstleistung

Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2020

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 80 04 69, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 07 11 9 70-25 00
Telefax 07 11 9 70-25 08
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

Tensordekomposition qualitativer Modelle zur Fehlererkennung

– Anwendung in der Gebäudeautomation –

Von der Fakultät für Architektur
des Karlsruher Instituts für Technologie (KIT)
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von

Thorsten Müller-Eping

2019

Referent: Prof. Dipl.-Ing. Andreas Wagner

Korreferent: Prof. Dr.-Ing. habil. Gerwald Lichtenberg (HAW Hamburg)

Korreferent: Prof. Dr.-Ing Sören Hohmann (KIT IRS)

Weiteres Mitglied des Prüfungsausschusses: PD Dr. Marcel Schweiker

Vorsitzende des Promotionsausschusses: Prof. Dr.-Ing. Petra von Both

Tag der mündlichen Prüfung: 05.08.2019

Kurzfassung

Diese Arbeit liefert einen Beitrag zur technischen Fehlererkennung auf Grundlage der Beobachtung gebäudetechnischer Systeme durch qualitative Modelle. Als qualitatives Modell wird ein stochastischer Automat verwendet, der das qualitative Verhalten eines Prozesses anhand diskreter Wahrscheinlichkeitsverteilungen beschreibt. Das Systemverhalten des zu beschreibenden Prozesses wird dabei nur näherungsweise abgebildet, wobei jedoch die grundlegende Systemdynamik erhalten bleibt. Stochastische Automaten und qualitative Modelle leiden dabei unter dem Problem, dass ihr Speicherbedarf mit einer steigenden Anzahl zu beschreibender Systemgrößen exponentiell ansteigt, womit es selbst für Systeme mit nur wenigen Systemgrößen kaum möglich ist, deren gesamtes qualitatives Verhalten abzubilden und die entsprechenden Wahrscheinlichkeiten effizient zu speichern. Zur Lösung dieses Komplexitätsproblems wird in dieser Arbeit ein neuer Ansatz verfolgt, bei dem die diskreten Wahrscheinlichkeitsverteilungen des stochastischen Automaten in Tensoren gespeichert werden. Dabei wird eine moderne Methode zur Modellreduktion vorgestellt, die auf der Dekomposition der multilinearen Tensorstruktur basiert. Die in dieser Arbeit vorgestellten Algorithmen, welche auf der speicher-reduzierten dekomponierten Tensorrepräsentation definiert sind, ermöglichen die effiziente Realisierung der qualitativen Fehlererkennung.

*We cannot solve our problems with
the same thinking we used when we
created them*

(Albert Einstein)

Vorwort

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der vergangenen Jahre bei der Ausarbeitung dieser Dissertation unterstützt haben.

Als Erstes danke ich Herrn Prof. Dipl.-Ing. Andreas Wagner vom Karlsruher Institut für Technologie dafür, dass er mir diese Arbeit ermöglichte und sie über die Jahre begleitete und betreute.

Mein ganz besonderer Dank gilt Prof. Dr.-Ing. habil. Gerwald Lichtenberg von der Hochschule für angewandte Wissenschaften Hamburg. Meine Aufenthalte in Hamburg gehörten zu den schönsten und produktivsten Zeiten während dieser Dissertation. Die vielen freundschaftlichen und wissenschaftlichen Gespräche waren stets inspirierend und – was vielleicht am wichtigsten ist – sie waren motivierend. Ich kann mit Sicherheit sagen, dass ohne diese Unterstützung, diese Arbeit so nicht möglich gewesen wäre. Vielen Dank für Alles und das Aufzeigen dessen, was alles möglich ist!

Besonders herzlich bedanken möchte ich mich auch bei Dr.-Ing. Kai Kruppa und Dipl.-Phys. Björn Lautenschlager von der Hochschule für angewandte Wissenschaften Hamburg. Die immer spannenden Diskussionen und die gute Zusammenarbeit bei gemeinsamen wissenschaftlichen Veröffentlichungen haben mir stets große Freude bereitet.

Ebenfalls danke ich meinen Kollegen und Freunden am Fraunhofer-Institut für Solare Energiesysteme in Freiburg. Dipl.-Ing. Nicolas Réhault hat mich während der gesamten Zeit immer unterstützt und mir die nötigen Freiräume geschaffen, die zum Gelingen einer solchen Arbeit notwendig sind. Dipl.-Phys. Tim Rist danke ich für seine stete Unterstützung in der Beantwortung wissenschaftlicher Fragestellungen, aber vor allem für seine Hilfe bei der Umsetzung und Implementierung mathematischer Algorithmen in der Programmiersprache PYTHONTM. Felix Ohr, M. Eng. gebührt ebenfalls besonderer Dank für die oftmals sehr willkommenen Ablenkungen und die stete Versorgung mit gekühlten Getränken für den gemeinsamen Konsum auf dem Dach des Instituts.

Natürlich tragen zu einer Dissertation, die über mehrere Jahre entsteht, auch Bachelorrenden, Masterrenden und studentische Hilfskräfte durch ihrer Arbeiten bei. Hier möchte ich mich insbesondere bei Vivien Vogelmann, B. Sc. für ihre Unterstützung bei der mathematischen Beweisführung bedanken. Außerdem haben Markus Saas, M. Eng. und Jakob Merck, M. Sc. durch ihre Arbeiten im Bereich der Simulation gebäudetechnischer Anlagen wichtige Erkenntnisse generiert.

Danken möchte ich zudem meinen Eltern Jürgen und Elke Müller, die mich seit dem Beginn meines Studiums über all die Jahre in jeder Hinsicht unterstützt haben.

Für die immer liebevolle und bedingungslose Unterstützung, insbesondere während der zeitintensiven Endphase der Dissertation, gilt mein größter Dank meiner Frau Yoshiko.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Technische Fehlererkennung und Diagnose	2
1.3	Methodik	3
1.4	Forschungsfragen	12
1.5	Struktur der Arbeit	14
2	Fehlererkennung und Diagnose in der Gebäudeautomation	15
2.1	Übersicht der Methoden	15
2.2	Stand der Technik	16
2.2.1	Zeitreihenplots	16
2.2.2	Carpetplots	18
2.2.3	Scatterplots	19
2.2.4	Grenzwertüberwachung und Expertensysteme	21
2.3	Forschungsrelevante Methoden	23
2.3.1	Regressionsanalyse	26
2.3.2	Künstliche neuronale Netze	27
2.3.3	Entscheidungsbäume	29
2.3.4	Clusteranalyse	30
2.4	Abgrenzung der Methoden zu qualitativen Modellen	31
3	Stochastische Automaten und Prozesse	33
3.1	Diskrete stochastische Prozesse	33
3.1.1	Definition	33
3.1.2	Homogene Markow-Ketten	34
3.2	Autonomer stochastischer Automat	36
3.2.1	Definition	36
3.2.2	Matrixrepräsentation	37
3.3	Diskrete Stochastische Prozesse mit Ein- und Ausgabe	41
3.3.1	Definition	41
3.3.2	Homogene Markow-Ketten mit Ein- und Ausgabe	41
3.4	Allgemeiner stochastischer Automat	42
3.4.1	Definiton	42
3.4.2	Stochastische Automaten mit besonderen Eigenschaften	44
3.4.3	Matrixrepräsentation	44
3.4.4	Modellierungsziel	45
3.5	Beobachtung und Fehlererkennung	48
3.5.1	Projektion und Prädiktion	48
3.5.2	Beobachtergleichung	50
3.5.3	Fehlererkennung	51
4	Qualitatives Modell	55
4.1	Einleitung	55
4.2	Quantisierung der Messgrößen	55
4.2.1	Quantisiertes System	55
4.2.2	Allgemeine Quantisierung	57

4.3	Qualitatives Modell des quantisierten Systems	60
4.3.1	Modellbeschreibung	60
4.3.2	Modellbildung	61
4.4	Qualitative Beobachtung und Fehlererkennung	64
4.4.1	Modellierungsziel	64
4.4.2	Qualitative Beobachtergleichung	65
4.4.3	Qualitative Fehlererkennung	67
4.4.4	Algorithmus zur qualitativen Fehlererkennung	68
5	Anwendung qualitativer Modelle in der Gebäudeautomation	71
5.1	Einleitung	71
5.2	Der Weg zum qualitativen Modell	72
5.2.1	Systembeschreibung und Trainingsdaten	72
5.2.2	Partitionierung der Signalräume und Quantisierung	75
5.2.3	Qualitative Modellbildung	79
5.2.4	Qualitative Beobachtung	84
5.2.5	Qualitative Fehlererkennung	89
5.2.6	Variation der Partitionierungen und der Beobachterzeitschrittdauer	92
5.3	Fehlererkennung in einem Erhitzer	96
5.3.1	Systembeschreibung	96
5.3.2	Qualitative Fehlererkennung	98
5.4	Fehlererkennung in einem Heizkreis	99
5.4.1	Systembeschreibung	100
5.4.2	Qualitative Fehlererkennung	100
5.5	Fehlererkennung in einem thermischen Speicher	103
5.5.1	Systembeschreibung	103
5.5.2	Qualitative Fehlererkennung	104
5.6	Zwischenfazit	105
6	Lineare und multilineare Algebra	107
6.1	Grundlagen	107
6.1.1	Notationen	107
6.1.2	Vektoren und Matrizen	107
6.2	Tensoren	108
6.2.1	Definition	109
6.2.2	Tensorscheiben	109
6.2.3	Tensorfibern	110
6.3	Tensortransformationen	111
6.3.1	Multiindex	111
6.3.2	Matrizisierung	112
6.3.3	Vektorisierung	115
6.4	Tensor- und Matrixprodukte	115
6.4.1	Hadamard-Produkt	115
6.4.2	Dyadisches Vektorprodukt	116
6.4.3	Kronecker-Produkt	117
6.4.4	Modus- n Tensor-Vektor Produkt	118
6.4.5	Kontrahiertes Modus- $(P 1)$ Tensorprodukt	119
6.4.6	Kontrahiertes Tensorprodukt	121
6.5	Spezielle Tensoren und Tensorrang	124
6.5.1	Rang-1 Tensor	124

6.5.2	Tensorrang	124
6.5.3	Kruskal-Tensor	125
7	Qualitative Modelle in Tensorrepräsentation	127
7.1	Tensorrepräsentation stochastischer Automaten	127
7.1.1	Verhaltenstensor	127
7.1.2	Beobachtergleichung	128
7.1.3	Erweiterte Beobachtergleichung	129
7.2	Tensorrepräsentation qualitativer Modelle	134
7.2.1	Verhaltenstensor	134
7.2.2	Qualitative Beobachtergleichung	135
7.2.3	Algorithmus zur qualitativen Fehlererkennung	136
7.2.4	Erweiterte qualitative Beobachtergleichung	136
7.2.5	Erweiterter Algorithmus zur qualitativen Fehlererkennung	139
8	Tensordekomposition	141
8.1	Einleitung	141
8.2	CP-Dekomposition	142
8.2.1	Grundlagen der CP-Dekomposition	142
8.2.2	Berechnung der CP-Dekomposition	144
8.2.3	Speicherbedarf der CP-Dekomposition	146
8.2.4	CP-Repräsentation des Verhaltenstensors	148
8.3	TT-Dekomposition*	149
8.3.1	Grundlagen der TT-Dekomposition*	150
8.3.2	Berechnung der TT-Dekomposition*	151
8.3.3	Speicherbedarf der TT-Dekomposition*	152
8.3.4	TT-Repräsentation des Verhaltenstensors*	153
9	Beobachteralgorithmen für die CP- und TT-Repräsentation	155
9.1	Beobachtung mit der CP-Repräsentation des Verhaltenstensors	155
9.1.1	Beobachtung durch Komposition von Submatrizen	155
9.1.2	CP-Beobachtergleichung	158
9.1.3	CP-Algorithmus zur qualitativen Fehlererkennung	159
9.1.4	Vergleich der Rechenzeiten	165
9.2	Beobachtung mit der TT-Repräsentation des Verhaltenstensors*	165
9.2.1	TT-Beobachtergleichung*	165
9.2.2	TT-Algorithmus zur qualitativen Fehlererkennung*	167
10	Beobachteralgorithmen für die erweiterte CP-Repräsentation	171
10.1	Stochastischer Automat mit vektoriellen Ein- und Ausgaben	171
10.1.1	Stochastischer Prozess	171
10.1.2	Stochastischer Automat	172
10.1.3	Tensorrepräsentation	173
10.1.4	Allgemeine Beobachtung	173
10.1.5	Beobachtung mit der CP-Repräsentation	175
10.2	Qualitatives Modell für komponentenweise quantisierte Ein- und Ausgänge	176
10.2.1	Komponentenweise Quantisierung	176
10.2.2	Quantisiertes System	181
10.2.3	Qualitative Beobachtung	182
10.2.4	CP-Algorithmus zur qualitativen Fehlererkennung	184

10.2.5	Äquivalenz qualitativer Modelle	184
10.3	Autonomer stochastischer Automat mit vektoriellen Zustandsfolgen*	188
10.3.1	Betrachtung stochastischer Teilprozesse*	188
10.3.2	Betrachtung des stochastischen Gesamtprozesses*	190
10.4	Stochastischer Automat mit vektoriellen Ein-, Ausgabe-, und Zustandsfolgen*	193
10.4.1	Stochastischer Prozess*	193
10.4.2	Stochastischer Automat*	193
10.4.3	Tensorrepräsentation*	194
10.4.4	Betrachtung stochastischer Teilprozesse*	194
10.4.5	Beobachtung mit der CP-Repräsentation*	196
10.5	Qualitatives Modell für komponentenweise quantisierte Ein-, Ausgänge und Zustände*	198
10.5.1	Quantisiertes System*	198
10.5.2	CP-Algorithmus zur qualitativen Fehlererkennung*	199
11	Zusammenfassung und Ausblick	203
	Literatur	207
A	Symbole und Symbolkonventionen	217
B	Beweise	223
B.1	Beweis der Proposition 7.1	223
B.2	Beweis der Proposition 9.1	224
B.3	Beweis der Proposition 9.2	226
B.4	Beweis der Proposition 9.3	227
B.5	Beweis der Proposition 9.4	229
B.6	Beweis der Proposition 10.1	231
B.7	Beweis der Proposition 10.2	232
B.8	Beweis der Proposition 10.3	235
B.9	Beweis der Proposition 10.4	238

1

Einleitung

1.1 Motivation

Im Jahr 2016 betrug der Endenergieverbrauch in Deutschland 2516.7 TWh [2], wovon ca. 35.4 % und damit 890.9 TWh für die Wärme- und Kältebereitstellung sowie die Beleuchtung¹ in Gebäuden aufgewendet wurden [19, S. 64]. Das von der Bundesregierung im Rahmen des *nationalen Aktionsplan Energieeffizienz* (NAPE) [16] und der *Energieeffizienzstrategie Gebäude* (ESG) [17] ausgegebene Ziel, den gebäudespezifischen Endenergiebedarf von 958.6 TWh im Referenzjahr 2008, um 20 % Prozent auf 766.9 TWh im Jahr 2020 zu reduzieren, ist damit kaum noch zu erreichen [19, S. 64].² Der für das Jahr 2016 ermittelte Endenergiebedarf entspricht gegenüber dem Referenzjahr lediglich einer Einsparung von ca. 7.1 %³, womit zur Erfüllung der Zielvorgabe bis 2020 weitere 124 TWh eingespart werden müssten. Das eine solche Reduktion innerhalb von nur vier Jahren wenig wahrscheinlich ist, zeigt der vorläufige Gesamtendenergiebedarf für das Jahr 2017 (2591.4 TWh [2]), von dem ca. 917.4 TWh⁴ auf den Gebäudesektor entfallen, was einem Anstieg von rund 3 % gegenüber dem Vorjahr entspricht. Diese Zahlen verdeutlichen, dass die bisher umgesetzten Maßnahmen zur Energieeinsparung im Gebäudesektor nicht ausreichen, um die energiepolitisch gesetzten Ziele zu erreichen und dass weitere Initiativen nötig sind [19, S. 64]. Ein großes Potenzial bieten dabei die Nichtwohngebäude, die den geringsten Anteil aller Gebäude ausmachen aber im Jahr 2016 für 37 % des gebäudespezifischen Endenergiebedarfs verantwortlich waren [29, S. 18–19].

Die innerhalb des NAPE und der ESG adressierten Maßnahmen und Förderprogramme zur Reduktion des gebäudespezifischen Endenergiebedarfs setzten insbesondere auf den vermehrten Einsatz erneuerbarer Energien, energetische Sanierungen von Bestandsgebäuden und niedrigenergetische Neubauten sowie die Verjüngung von Gebäudeanlagen- und Regelungstechnik – insbesondere im Bereich der Raumluftechnik [16, S. 22–23], [17, S. 27]. Neben diesen zumeist kostenintensiven Optimierungen werden aber auch geringinvestive Maßnahmen zur Energieeffizienzsteigerung in Bestandsgebäuden gefördert. Die im Besonderen für die Forschung relevante förderpolitische Grundlage bildet hierfür das 6. Energieforschungsprogramm der Bundesregierung aus dem Jahr 2011 (erweitert 2014) [14], [15] mit der Forschungsinitiative *Energieoptimiertes Bauen* (EnOB) [18]. Die Forschungsinitiative EnOB fördert die Weiterentwicklung von Methoden zur energetischen Betriebsoptimierung und zum Monitoring bestehender Gebäude- und Anlagentechnik sowie die Entwicklung moderner Automatisierungswerkzeuge und entsprechender Soft- und Hardwarelösungen [18, S. 4]. Forschungsprojekte im Rahmen von EnOB, wie beispielsweise das Projekt ModBen [96, S. 1], zeigen, dass sich durch eine kontinuierliche Betriebsüberwachung von Nichtwohngebäuden zwischen 5 – 20 % des Energiebedarfs einsparen lassen. Andere nationale Forschungsprojekte wie ModQS [107, S. 1], aber auch internationale Vorhaben [33, S. iii], gehen sogar von Einsparpotenzialen bis zu 30 % aus, was der kontinuierlichen

¹Energiemengen für Beleuchtung werden nur für Nichtwohngebäude bilanziert [19, S. 64].

²Energiemengen aus [19, S. 65].

³Die für das Jahr 2016 berechnete Einsparung von 7.1 % gegenüber dem Referenzjahr 2008 bezieht sich auf den in [2] gegebenen Datenstand zum Zeitpunkt 07/2018. In [19, S. 64] wird eine Reduktion von 6.3 % genannt, die jedoch auf älteren Erhebungsdaten zum Zeitpunkt 11/2017 beruht.

⁴Berechnung unter der Annahme, dass auf den für das Jahr 2017 vorläufig bestimmten deutschen Gesamtenergiebedarf von 2591.4 TWh, wiederum 35.4 % auf den Gebäudesektor entfallen.

Gebäudebetrieboptimierung eine besondere Relevanz im Hinblick auf geringinvestiv erschließbare Einsparpotenziale zukommen lässt.

Ein grundlegender Aspekt der kontinuierlichen Gebäudebetriebsüberwachung- und Optimierung ist das Sicherstellen eines fehlerfreien und damit energieeffizienten Gebäude- und Anlagenbetriebs. In der Praxis ist ein solcher Idealzustand nur selten gegeben, denn wie alle technischen Systeme unterliegen gebäudetechnische Anlagen wie Heiz- und Kühlkreise, Wärme- und Kälteerzeuger, Kühltürme, raumlufthtechnische Anlagen (RLT-Anlagen) und Wärmepumpen, Ausfällen und Defekten, die dazu führen, dass die betroffenen Systeme nicht mehr ordnungsgemäß und jenseits des energetischen Optimums operieren. Defekte Komponenten wie Pumpen und Ventile, sich durch Alterungsprozesse verschlechternde Wärmeübertragungsraten und falsche oder nicht eingehaltene Betriebszeiten sind nur einige Beispiele für meist unentdeckte Fehler, die sowohl zu thermischen Komforteinbußen als auch zur höheren Energievierbräuchen und -kosten führen. Die Gründe dafür, dass solche Fehler unerkannt bleiben sind vielfältig. Zum einen mangelt es meist an einer übergeordneten Instanz, welche die Messdaten der heterogenen Gebäudesysteme in guter Qualität sammelt, zentral bereitstellt und damit eine generelle Überprüfung des Gebäudebetriebs möglich macht. Zum anderen fehlen geeignete Methoden zur Fehlererkennung, um die Anlagenmessdaten interpretieren und Fehler frühzeitig erkennen zu können. Moderne Nichtwohngebäude verfügen teils über tausende Datenpunkte, deren kontinuierliche Auswertung einen kaum zu realisierenden Zeitaufwand bedeutet und darüber hinaus ein enormes Fachwissen bezüglich der entsprechenden Systeme voraussetzt. Dieser Aspekt gewinnt insbesondere vor dem Hintergrund, dass sich Gebäude durch aktuelle Themen wie das Internet der Dinge (*engl.*: Internet of Things) und die Gebäudeautomation 4.0 immer mehr zu technisch hoch entwickelten und -komplexen *Smart Buildings* entwickeln, zunehmend an Brisanz. Dieser Trend bietet jedoch auch große Potenziale, denn die zunehmende Digitalisierung und Vernetzung der Gebäudesysteme lässt eine Quantitäts- und Qualitätssteigerung der Gebäudemessdaten erwarten und eröffnet damit neue Möglichkeiten für den Einsatz von Analysemethoden aus dem Bereich der künstlichen Intelligenz und des maschinellen Lernens. Die grundlegende Voraussetzung zum Ausschöpfen dieses Potenzials ist, dass den Gebäudebetreibern ebenso moderne und zukunftsorientierte Werkzeuge zur Fehlererkennung und Diagnose bereitgestellt werden. Diese Arbeit leistet einen Beitrag hierzu.

1.2 Technische Fehlererkennung und Diagnose

Unter einem *Fehler* wird ein Anlagenbetriebszustand verstanden, der nicht dem nominalen Verhalten entspricht und der dazu führt, dass die Anlage ihre Funktion nicht oder nur eingeschränkt erfüllen kann.⁵ Ziel der technischen *Fehlererkennung* (*engl.*: Fault Detection) ist es, Anlagen kontinuierlich zu überwachen und zu prüfen, ob diese innerhalb ihrer nominalen Betriebsparameter operieren oder ob ein fehlerhafter Betriebszustand vorliegt.⁶ Während die technische Fehlererkennung ausschließlich die Frage beantwortet, ob und wann ein Fehler aufgetreten ist, adressieren die *Fehlerisolation* (*engl.*: Fault Isolation) und die *Fehleridentifikation* (*engl.*: Fault Identification) auch die Bestimmung der verursachenden Anlagenkomponente und des Fehlertyps [12, S. 14], [87, S. 287–288]:

- Fehlererkennung: Bestimmung ob und wann ein Fehler in einer Anlage auftrat.
- Fehlerisolation: Bestimmung wo der Fehler auftrat (Feststellung der Anlagenkomponente oder des technischen Teilsystems).
- Fehleridentifikation: Bestimmung welcher Fehler auftrat (Feststellung des Fehlertyps und Abschätzung der Schwere des Fehlers).

⁵In Fachliteratur wie [11], [45] wird zwischen Anlagen-, Sensor-, und Aktorfehlern unterschieden. Eine solche Differenzierung findet hier nicht statt. Ein Anlagenbetriebszustand wird als fehlerhaft bewertet, wenn er nicht dem zu erwartenden nominalen Verhalten entspricht – unabhängig davon – ob der Fehler in der Anlage selbst oder durch fehlerhafte Sensoren oder Aktoren verursacht wurde.

⁶Die Erkennung von Planungs-, Ausführungs- und Inbetriebnahmefehlern ist nicht unmittelbar Gegenstand der technischen Fehlererkennung.

In der Literatur existieren alternative und von der oben angegebenen Quelle abweichende Definitionen dieser Fachbegriffe. In [45, S. 413–414] wird beispielsweise die Bestimmung der fehlerhaften Anlagekomponente und des Fehlertyps als Fehlerisolation verstanden, während unter dem Begriff der Fehleridentifikation nur die Abschätzung der Fehlerschwere sowie eine zusätzliche Bestimmung des zeitlichen Verhaltens des Fehlers (kontinuierlich oder transient) geführt wird. Unabhängig davon, wie weit die einzelnen Definitionen der Fehlererkennung, -isolation und -identifikation gefasst sind, werden diese oftmals unter einem Oberbegriff vereint und man spricht entweder von *Fehlerdiagnose* (engl.: Fault Diagnosis) [45, S. 413–414], [12, S. 14], oder wie auch im Folgenden, von *Fehlererkennung und Diagnose* (engl.: Fault Detection and Diagnosis) [47, S. 5], [34, S. 3]. Die einzelnen Fachbegriffe lassen sich anhand eines kleinen Beispiels leicht nachvollziehen: Betrachtet man eine raumlufttechnische Anlage (RLT-Anlage), die aus den Komponenten Ventilator, Erhitzer, Kühler und Befeuchter besteht, dann kann die Feststellung eines Fehlers wie »Anlage heizt nicht«, als Fehlererkennung betrachtet werden. Die Diagnose folgt mit der Erkenntnis, dass der Erhitzer der RLT-Anlage ohne Funktion (Fehlerisolation) und die Ursache ein Defekt der Erhitzerpumpe ist (Fehleridentifikation). Die Schwere des Fehlers kann dabei monetär (Reparaturkosten) oder qualitativ in Form von Komforteinbußen (hoch, mittel, gering, keine) beziffert werden. In der praktischen Anwendung der Methoden zur Fehlererkennung und Diagnose wird meist nur die Fehlererkennung und bei Bedarf die Fehlerisolation realisiert, da die vollständige Fehlerdiagnose einen in der Regel erheblichen Mehraufwand bedeutet [34, S. 3], [87, S. 294]. Diese Dissertation adressiert in erster Linie die Fehlererkennung, wobei sich die hier verwendeten qualitativen Modelle grundsätzlich auch zur Isolation und Identifikation von Fehlern eignen. Auch wenn die Fokussierung auf die Fehlerdetektion zunächst wie eine Einschränkung wirkt, bedeutet dies im Vergleich zum Status-Quo in der Gebäudeautomation bereits einen deutlichen Fortschritt, denn die derzeit verfügbaren Automationssysteme bieten keine oder nur unzureichende proprietäre Funktionalitäten, um Fehler zuverlässig zu detektieren.

1.3 Methodik

Dieser Abschnitt beschreibt ausführlich die *Methodik*, mit der die Fehlererkennung in gebäudetechnischen Systemen in dieser Arbeit realisiert wird. Zuerst erfolgt eine Einführung in eine der wichtigsten Grundlagen technischer Systeme – die *Signale*. Technische Prozesse treten über Signale in Wechselwirkung mit ihrer Umgebung und deren Messungen bilden die Grundlage aller weiteren Untersuchungen. Anschließend erfolgt die *Systemvorstellung*, welche die Art und Weise erläutert, in der die gebäudetechnischen Systeme betrachtet werden. Auf die Systemvorstellung folgt die *Modellvorstellung* und damit die Einführung der qualitativen Modelle zur Fehlerdetektion.

Signale

Signale beschreiben den zeitlichen Verlauf messbarer Größen. Dabei existieren unterschiedliche Signaltypen, die man anhand der Werte, die das Signal annehmen kann als auch an der Zeit, für welche die Signalwerte definiert sind, unterscheidet. Die Werte eines Signals und die zugrundeliegende Zeit können jeweils durch *kontinuierliche* oder *diskrete* Größen definiert sein:

- Unter einer kontinuierlichen Größe versteht man eine *reellwertige* Größe wie etwa $a \in \mathbb{R}$, $a \in \mathbb{R}_{\geq 0}$ oder $a \in [0, 1]$.⁷
- Unter einer diskreten Größe versteht man eine *ganzzahlige* Größe wie beispielsweise $a \in \mathbb{N}$ oder $a \in \{0, 1\}$.⁸

⁷Das Symbol \mathbb{R} bezeichnet die reellen Zahlen, $\mathbb{R}_{\geq 0}$ die nichtnegativen reellen Zahlen und $[0, 1]$ das Einheitsintervall, welches alle reellen Zahlen von Null bis Eins enthält.

⁸Das Symbol \mathbb{N} bezeichnet die natürlichen ganzen Zahlen inkl. der Null: $\mathbb{N} := \{0, 1, 2, 3, \dots\}$. Die Menge $\{0, 1\}$ enthält nur zwei ganzzahlige Werte, die Null und die Eins.

Anhand der Begriffe der kontinuierlichen und diskreten Größen lassen sich die in der Praxis auftretenden Signaltypen leicht beschreiben. Dabei unterscheidet man häufig zwischen *kontinuierlichen*, *zeitdiskreten* und *quantisierten Signalen*, welche in Abbildung 1.1 dargestellt sind und deren Eigenschaften im Folgenden erläutert werden.

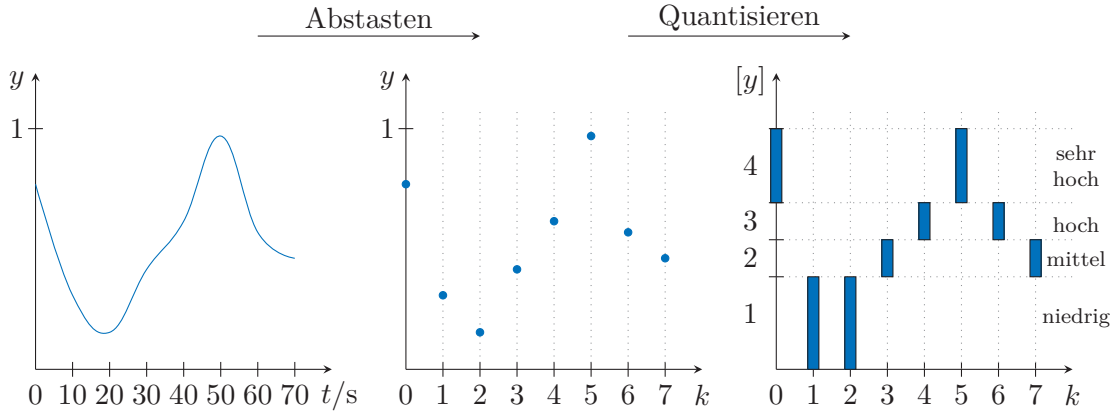


Abbildung 1.1: Abtasten und Quantisieren. Kontinuierliches Signal (links), zeitdiskretes Signal (mitte) und quantisiertes Signal (rechts)

Kontinuierliche Signale: In der Abbildung 1.1 ist links ein *kontinuierliches Signal* als Funktion $y(t)$ dargestellt. Dieses Signal ist *wert- und zeitkontinuierlich*, denn es kann zu beliebigen Zeiten $t \in \mathbb{R}_{\geq 0}$ die Signalwerte $y(t) \in [0, 1]$ annehmen. Einfach formuliert: Zu jeder *kontinuierlichen Zeit* t existiert ein eindeutig definierter Signalwert $y(t)$, der einen reellen Wert zwischen Null und Eins besitzt. Für kontinuierliche Signale sind somit genaue Aussagen bezüglich des Signalwerts zu jeder Zeit möglich, wie beispielsweise: Zum Zeitpunkt $t = 2.3456 \dots$ Sekunden besitzt das Signal den Wert $0.7752 \dots$

Zeitdiskrete Signale: Wird das kontinuierliche Signal $y(t)$ im linken Teil von Abbildung 1.1 mit einer festen *Abtastzeit* T_a zeitlich diskretisiert, ergibt sich das in der mittleren Darstellung abgebildete *zeitdiskrete und wertkontinuierliche* Signal $y(k)$. Die Signalwerte $y(k)$ liegen damit nur noch zu *diskreten Zeiten* $k \in \mathbb{N}$ vor und entsprechen den Signalwerten $y(t)$ zur kontinuierlichen Zeit $t = k \cdot T_a$. Am besten lässt sich dies anhand des linken Teils der Abbildung 1.1 veranschaulichen, in dem ein kontinuierliches Signal über einen Zeitraum von $t = 0$ s bis $t = 70$ s dargestellt ist. Tastet man das Signal $y(t)$ mit einer Abtastzeit von $T_a = 10$ s ab, erhält man die zeitdiskrete Darstellung $y(k)$ in der mittleren Abbildung. Der Wert des zeitdiskreten Signals $y(k)$ zur Zeit $k = 7$ entspricht dann dem Wert des kontinuierlichen Signals $y(t)$ zum Zeitpunkt $t = k \cdot T_a = 7 \cdot 10$ s = 70 s. Im Gegensatz zu kontinuierlichen Signalwerten, die durch eine Funktion definiert sind, können die zeitdiskreten Signalwerte $y(k)$ für fortlaufende Zeiten $k = 0, 1, 2, \dots, T$ als Folgen

$$Y(0..T) = (y(0), y(1), y(2), \dots, y(T)) \quad (1.1)$$

notiert werden, die man als *Zeitreihen* bezeichnet. In der Praxis sind solche Folgen oft endlich und nur bis zu einer bestimmten Zeit verfügbar oder relevant, die man als *Zeithorizont* T bezeichnet. Die Zeitreihe in der Mitte von Abbildung 1.1 ist beispielsweise bis zu einem Zeithorizont von $T = 7$ gegeben und besitzt die Form

$$\begin{aligned} Y(0..7) &= (y(0), y(1), y(2), y(3), y(4), y(5), y(6), y(7)) \\ &= (0.77, 0.30, 0.15, 0.41, 0.62, 0.97, 0.57, 0.46) . \end{aligned} \quad (1.2)$$

Zeitdiskrete Signale oder Zeitreihen bilden in der Gebäudetechnik die signaltheoretische Grundlage für nahezu alle Auswertungen, Bilanzierungen und Bewertungen (s. Kapitel 2). Die Gebäudeautomationssysteme erfassen die Messwerte häufig mit Abtastzeiten von $T_a = 60$ s, weshalb man auch

von *Minutendaten* spricht. Weitere übliche Abtastzeiten sind 5-Minutendaten ($T_a = 300$ s) und 15-Minutendaten ($T_a = 900$ s), wobei beliebige weitere Varianten existieren.

Quantisierte Signale: Wird ausgehend von einem zeitdiskreten Signal auch dessen Wertebereich diskretisiert, erhält man ein *zeit- und wertdiskretes* Signal $[y(k)]$. Diesen Vorgang bezeichnet man als *Quantisierung* (s. Abbildung 1.1). Ein solches Signal, das sowohl zeit- als auch wertdiskret ist, wird als *Digitalsignal* oder *quantisiertes Signal* bezeichnet.⁹ Der Wertebereich des Signals ist nun nicht mehr durch reelle Zahlen gegeben, sondern durch ganzzahlige Werte, womit für die Signalwerte im Allgemeinen $[y(k)] \in \mathbb{N}$ gilt. Die Werte die ein quantisiertes Signal $[y(k)]$ annehmen kann, bezeichnet man als *qualitative Werte*.

Rechts in Abbildung 1.1 wurde der Wertebereich des Signals in vier Bereiche diskretisiert, sodass für die Signalwerte $[y(k)] \in \{1, 2, 3, 4\}$ gilt. Der quantisierte Signalwert $[y(k)]$ kann somit nur die qualitativen Werte »Eins, Zwei, Drei oder Vier« annehmen. Allgemein können den quantisierten Signalwerten $[y(k)]$ anstelle von ganzen Zahlen aber auch symbolische oder linguistische Werte wie {niedrig, mittel, hoch, sehr hoch} zugeordnet werden. Im Gegensatz zu einem zeitdiskreten aber wertkontinuierlichen Signalwert $y(k)$ erlaubt der quantisierte Signalwert $[y(k)]$ nur noch eine qualitative Einschätzung der zugrunde liegenden Größe. In Abbildung 1.1 ist zum Beispiel zur Zeit $k = 5$, der zeitdiskrete Signalwert durch $y(5) = 0.97$ gegeben, während für den quantisierten Signalwert $[y(5)] = 4$ oder $[y(5)] = \{\text{sehr hoch}\}$ gilt. Das quantisierte Signal besitzt somit den qualitativen Wert »Vier« oder linguistisch ausgedrückt, den symbolischen Wert »sehr hoch«. Wie bei den zeitdiskreten Signalwerten in Gleichung (1.2) können die qualitativen Werte des quantisierten Signals $[y(k)]$ im rechten Teil von Abbildung 1.1 als eine ganzzahlige Folge

$$\begin{aligned} Y(0\dots7) &= ([y(0)], [y(1)], [y(2)], [y(3)], [y(4)], [y(5)], [y(6)], [y(7)]) \\ &= (4, 1, 1, 2, 3, 4, 3, 2) \end{aligned} \quad (1.3)$$

notiert werden. In der Gebäudetechnik existieren auch natürliche zeit- und wertdiskrete Signale. So sind beispielsweise diverse Pumpen und Ventilatoren nicht drehzahl geregelt, sondern können nur ein- oder ausgeschaltet sein. Ein solches Signal kann ausschließlich Werte aus der Menge $\{0, 1\}$ annehmen und hat entweder den Wert Eins (An) oder den Wert Null (Aus).

Das Abtasten und Quantisieren eines kontinuierlichen Signals hin zu einem quantisierten Signal stellt letztlich eine Abstraktion dar [86, S. 3], bei der sowohl Informationen bezüglich des zeitlichen Verhaltens als auch die Möglichkeit, reellwertige und damit präzise Aussagen über den Signalwert zu treffen, verloren gehen. Quantisierte Signale bilden in dieser Arbeit die Grundlage der systemtheoretischen Betrachtungen und damit die Basis für die Erstellung und Anwendung *qualitativer Modelle* zur Fehlerdetektion in gebäudetechnischen Systemen.

Systemvorstellung

Um technische Gebäudesysteme anhand von Modellen beschreiben zu können, bedarf es zunächst einer *Systemvorstellung*. Wie erwähnt, liegen die aus den gebäudetechnischen Anlagen wie Heiz- und Kühlkreisen, thermischen Speichern, Wärmepumpen oder RLT-Anlagen erfassten Messwerte im Allgemeinen in Form von Zeitreihen, also in zeitdiskreten Folgen von Messwerten vor (s. Gleichung 1.2). Solche Anlagen gehören damit zur Systemklasse der sogenannten *zeitdiskreten dynamischen Systeme* (s. Abbildung 1.2).

Zeitdiskretes dynamisches System: Unter einem dynamischen System wird Allgemein eine Funktionseinheit verstanden, deren *Ausgangsgrößen* sich als eine Reaktion auf die Einwirkung der *Eingangsgrößen* zeitlich ändern [85, S. 2], [120, S. 1]. Dynamische Systeme treten dabei über ihre Ein-

⁹In dieser Arbeit werden ausschließlich quantisierte Signale betrachtet, denen eine konstante Abtastzeit zugrunde liegt und deren Signalwerte somit zu äquidistanten Zeitpunkten vorliegen. Ereignisdiskrete Signale werden nicht betrachtet.

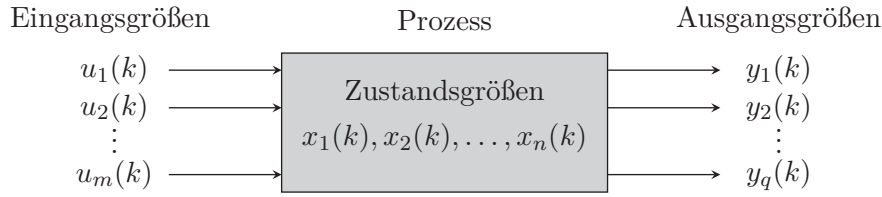


Abbildung 1.2: Zeitdiskretes dynamisches Mehrgrößensystem

und Ausgangsgrößen in Wechselwirkung mit ihrer Umgebung und zeichnen sich durch ein »Gedächtnis« aus, denn die aktuellen Ausgangsgrößen hängen nicht nur von den aktuellen Eingangsgrößen ab, sondern auch von den in der Vergangenheit liegenden [131, S. 17]. Die Art und Weise, wie sich die Eingangsgrößen auf die Ausgangsgrößen auswirken, hängt dabei von der inneren Systemstruktur und damit von den *Zustandsgrößen* ab, welche als funktionales Bindeglied zwischen den Ein- und Ausgangsgrößen interpretiert werden können [66, S. 4]. Der in Abbildung 1.2 dargestellte *Prozess* mit den zeitabhängigen Eingangs-, Zustands-, und Ausgangsgrößen stellt somit ein zeitdiskretes dynamisches System dar und repräsentiert die allgemeine Vorstellung technischer Systeme.¹⁰

Besitzt ein dynamisches System nur eine einzige Ein- und Ausgangsgröße, so spricht man von einem *Eingrößensystem* (engl.: single input, single output system (SISO)), andernfalls von einem *Mehrgrößensystem* (engl.: multiple input, multiple output system (MIMO)). Bei der Beschreibung und Darstellung dynamischer Mehrgrößensysteme, die wie in Abbildung 1.2 über eine Anzahl von m Eingangsgrößen, n Zustandsgrößen und q Ausgangsgrößen verfügen, werden diese Systemgrößen üblicherweise in einem *Eingangsvektor* $\mathbf{u}(k)$, einem *Zustandsvektor* $\mathbf{x}(k)$ und einem *Ausgangsvektor* $\mathbf{y}(k)$ zusammengefasst:

$$\mathbf{u}(k) = \begin{pmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_m(k) \end{pmatrix} \in \mathbb{R}^m, \quad \mathbf{x}(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{pmatrix} \in \mathbb{R}^n, \quad \mathbf{y}(k) = \begin{pmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_q(k) \end{pmatrix} \in \mathbb{R}^q. \quad (1.4)$$

Typografisch klein und fett gedruckte Buchstaben symbolisieren Vektoren. Die Notation eines Eingangsvektors $\mathbf{u}(k) \in \mathbb{R}^m$ bedeutet, dass dieser Vektor eine Anzahl von m reellen Eingangsgrößen repräsentiert und damit die Länge m besitzt.¹¹ Gleiches gilt für den Zustands- und Ausgangsvektor. Mit dem Wissen, dass die Eingangs-, Zustands- und Ausgangsvektoren in Gleichung (1.4) stets mehrere Größen repräsentieren, spricht man trotz der vektoriellen Repräsentation vereinfachend auch nur von Eingängen, Zuständen und Ausgängen – oder wie zuvor – von Eingangs-, Zustands-, und Ausgangsgrößen. Die vektorielle Notation der Systemgrößen ist in den Ingenieurwissenschaften üblich und ermöglicht nicht nur eine einfache und kompakte Notation, sondern bietet auch Vorteile in den späteren modelltheoretischen Berechnungen. Darstellungen wie die des Mehrgrößensystems in Abbildung 1.2 erfolgen typischerweise ebenso in vektorieller Form, was zu einer übersichtlichen Repräsentation führt (s. Abbildung 1.3).

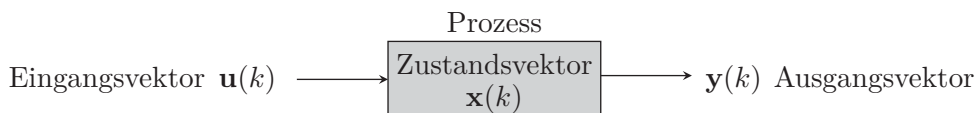


Abbildung 1.3: Zeitdiskretes dynamisches Mehrgrößensystem (vektorielle Darstellung)

¹⁰Die Bezeichnungen »System« und »Prozess« werden in der Literatur oft synonym verwendet [85, S. 2].

¹¹Ein formale Definition des Vektorbegriffs wird in Kapitel 6 gegeben.

Eingangs-, Ausgangs- und Zustandsgrößen realer Systeme: Die Zustandsgrößen eines Prozess beschreiben dessen energetische Speicherfähigkeit. Typische Zustandsgrößen sind potenzielle oder thermische Energien, Enthalpien oder Kapazitäten. Aus theoretischer Sicht stellen die Zustandsgrößen im Allgemeinen keine messbaren Größen dar und sind für einen außenstehenden Betrachter des Systems nicht ersichtlich. Sie spielen jedoch eine grundlegende Rolle bei der klassischen Modellierung dynamischer Systeme und werden dabei durch Differential- oder Differenzgleichungen beschrieben, mit denen die innere Systemdynamik abgebildet wird. In der Praxis werden dabei häufig solche Zustandsgrößen gewählt, die sich als eine Funktion der dem System zugrunde liegenden Energiegröße beschreiben lassen und die entgegen der theoretischen Vorstellung auch messbar sein können. Ein typisches Beispiel, an dem sich dies erläutern lässt, ist ein Behälter der mit Wasser gefüllt ist und der damit einen Speicher potentieller Energie E_{pot} darstellt (s. Abbildung 1.4).

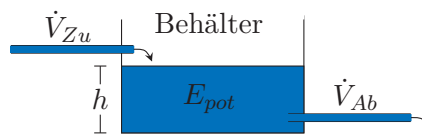


Abbildung 1.4: Wasserbehälter als dynamisches Eingrößensystem

Die potentielle Energie des Wassers E_{pot} ist nicht unmittelbar messbar, kann jedoch wenn man die Abmaße des Behälters kennt, als eine Funktion der Füllhöhe h beschrieben werden. Die Füllhöhe h repräsentiert damit die Zustandsgröße und ist proportional zu der im Behälter gespeicherten potenziellen Energie. Für die qualitativen Modelle ist eine solche Betrachtung der Zustandsgrößen wichtiger Bestandteil der Modellbildung.

Im Gegensatz zu den Zustandsgrößen stellen die Eingangs- und Ausgangsgrößen dynamischer Systeme dagegen grundsätzlich messbare Größen dar, die beispielsweise durch Temperaturen, Massen-, Volumen- oder Wärmeströme gegeben sind. In der Vorstellung sind die Ein- und Ausgangsgrößen diejenigen Variablen, welche die Energie in das System einbringen oder aus diesem abführen. Im Beispiel des Wasserbehälters in Abbildung 1.4 stellt der obere Zufluss \dot{V}_{Zu} somit die Eingangsgröße dar und der untere freie Abfluss \dot{V}_{Ab} die Ausgangsgröße. Ein hoher Zufluss in den Behälter führt zu einer steigenden Füllhöhe und damit zu einer Zunahme der potenziellen Energie des Systems über die Zeit. Wie sich der Zufluss auf den Abfluss auswirkt, hängt von der Zustandsgröße, also von der Füllhöhe ab. Der Volumenstrom durch den freien Abfluss, der die Energie aus dem System abführt, ist dabei umso größer, desto höher der Füllstand des Behälters ist.

Gebäudetechnische Anlagen zur Wärme- und Kältebereitstellung sind thermische dynamische Systeme, welche sich wie in Abbildung 1.5 dargestellt, interpretieren lassen.

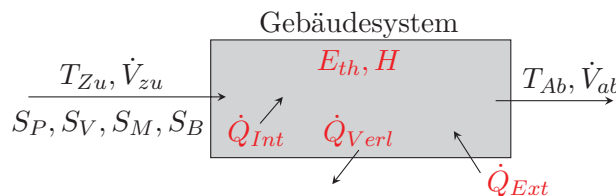


Abbildung 1.5: Generisches Gebäudesystem als dynamisches Mehrgrößensystem. Die üblicherweise messbaren Eingangs- und Ausgangsgrößen sind in schwarz und die nicht messbaren in rot dargestellt.

Die generalisierte Darstellung eines dynamischen Gebäudesystems in Abbildung 1.5 kann sowohl als thermischer Speicher, als Heizkessel, Heizkörper oder auch als ein Raum, der beispielsweise über eine RLT-Anlage beheizt oder gekühlt wird, verstanden werden. In allen Fällen existieren Wärmeströme in

Form mechanisch erzeugter Luft- oder Wasservolumenströme \dot{V}_{Zu} und gegebenen Temperaturen T_{Zu} , die als Eingangsgrößen in das System hinein wirken und zu einer zeitlichen Änderung seiner Zustandsgrößen, wie beispielsweise der thermischen Energie E_{th} oder der Enthalpie H führen. Auf der anderen Seite verlassen ebensolche Wärmeströme, beispielsweise gegeben durch Abluftvolumenströme \dot{V}_{Ab} mit Temperatur T_{Ab} das System als Ausgangsgrößen und führen Energie ab. Hinzu kommen Störgrößen wie Wärmeverluste \dot{Q}_{Verl} , die aus dem System hinaus wirken, äußere Wärmeeinträge \dot{Q}_{Ext} und gegebenenfalls innere Lasten \dot{Q}_{Int} , die in der Regel ebenso wenig messtechnisch erfassbar sind wie die Zustandsgrößen, diese jedoch entsprechend beeinflussen. Im Hinblick auf die Modellierung solcher Systeme muss auch hier zwischen der grundsätzlichen Annahme, dass die Zustandsgrößen nicht als Messgrößen zur Verfügung stehen und der Notwendigkeit, diese mathematisch beschreiben zu können, unterschieden werden. Ähnlich wie beim Wasserbehälter in Abbildung 1.4, in dem die Füllhöhe als Zustandsgröße das Niveau der potentiellen Energie beschreibt, können bei Gebäudesystemen beispielsweise die Raumtemperatur oder die Temperatur in einem Speicher geeignete Zustandsgrößen sein, um das Niveau der im System vorhandenen Energie abzubilden. Neben den oben erwähnten typischen Eingangsgrößen können auch Mischventilstellungen S_M , Pumpen- und Ventilatorendrehzahlen S_P, S_V sowie Bedarfsanforderungen S_B als solche betrachtet werden. Diese Eingangsgrößen stellen oftmals wichtige Indikatoren für die Bewertung des Betriebszustandes der Anlage dar.

Abbildung 1.5 kann als typisches Beispiel dafür angesehen werden, wie sich ein dynamisches System aus Sicht eines Gebäudebetreibers darstellt: Die Ein- und Ausgangsgrößen stehen als beobachtbare und messbare Größen in Form von Zeitreihen (1.1) zur Verfügung und können entsprechend ausgewertet werden (s. a. Kapitel 2). Die inneren Zustandsgrößen, welche die energetische Speicherfähigkeit des Systems beschreiben, sind nicht ersichtlich oder messtechnisch erfassbar. Für den außenstehenden Beobachter oder Betreiber des dynamischen Systems, tritt dieses nur anhand seines *Eingangs-Ausgangs-Verhaltens* in Erscheinung. Kürzer spricht man auch einfach vom *Verhalten* des dynamischen Systems.

Verhalten dynamischer Systeme: Zum sogenannten *Verhalten* eines dynamischen Systems gehören alle Folgen von Signalwerten der Eingangs- und Ausgangsgrößen, die dem System möglich sind [11, S. 4–5], [86, S. 26]. Die Bewertung des Eingangs-Ausgangs-Verhaltens dynamischer Systeme spielt eine grundlegende Rolle im Hinblick auf die Fehlererkennung. Fehler, die auf einen Prozess einwirken, führen dazu, dass sich dessen Eingangs-Ausgangs-Verhalten ändert und damit vom Nominalverhalten abweicht. Beispiel 1.1 erläutert dies.

Beispiel 1.1 (Unplausibles Verhalten): *Angenommen der Wasserbehälter in Abbildung 1.4 besitzt eine Leckage, durch die Wasser austritt. In diesem Fall wird sich der Wert der Ausgangsgröße \dot{V}_{Ab} für eine gegebene Eingangsgröße \dot{V}_{Zu} gegenüber dem Nominalfall ändern. Aufgrund der Leckage wird weniger Wasser aus dem Abfluss fließen als im Normalfall und das Eingangs-Ausgangs-Verhalten ändert sich.*

Ein anderes Beispiel, welches man jedoch nicht als Fehler im Sinne der Definition bezeichnen sollte, sondern vielmehr als eine Anomalie, ist Folgendes: Angenommen Abbildung 1.5 stellt einen Raum eines Gebäudes dar, der über eine RLT-Anlage beheizt wird. Diese bringt einen Zuluftvolumenstrom \dot{V}_{Zu} mit Temperatur T_{Zu} in den Raum ein und führt den Abluftvolumenstrom $\dot{V}_{Ab} = \dot{V}_{Zu}$ mit der Temperatur T_{Ab} aus dem Raum ab. Ein nun beispielsweise im Heizbetrieb vollständig geöffnetes Fenster im Raum führt entsprechend dazu, dass dieser auskühlt und sich damit die thermische Energie im Raum reduziert. Unabhängig davon, ob die Lüftungsanlage auf diese Situation mit höheren Zulufttemperaturen und/oder größerem Zuluftvolumenstrom reagiert, ändert sich hierdurch das Eingangs-Ausgangs-Verhalten des dynamischen Systems »Raum«. Die Werte der Eingangsgrößen \dot{V}_{Zu}, T_{Zu} werden gegenüber dem Nominalfall gleich oder höher ausfallen, während jedoch der Wert der Ausgangsgröße T_{Ab} niedriger sein wird.

Das Verhalten technischer dynamischer Systeme zu beschreiben und zu analysieren hat somit eine hohe Priorität im Hinblick auf die Sicherstellung einer ordnungsgemäßen und energieeffizienten Anlagenfunktion. Dies automatisiert zu realisieren, ist die Aufgabe von Modellen. Es ist somit notwendig, Funktionen aufzustellen oder abzuleiten, welche die Dynamik des Systems beschreiben und damit eine kausale Beziehung zwischen den Eingängen und Ausgängen herstellen. Dabei existieren viele unterschiedliche Möglichkeiten, dies zu realisieren. So kann der funktionale Zusammenhang zwischen Ein- und Ausgangsgrößen durch lineare, nichtlineare, zeitvariante oder zeitinvariante Modelle beschrieben werden. Die Modellbildung kann auf White-Box Verfahren, welche auf der expliziten Modellierung der Zustandsgrößen in Form von Differential- oder Differenzgleichungen basieren, aber auch auf Black-Box Ansätzen, welche die innere Systemdynamik anhand von historischen Messungen der Eingangs- und Ausgangsgrößen abstrahieren. Außerdem kann das Modellierungsprinzip *quantitativ* sein und das Systemverhalten anhand der zeitdiskreten und reellwertigen Systemgrößen beschreiben, oder es kann *qualitativer* Natur sein und das Verhalten nur näherungsweise abbilden, basierend auf der Betrachtung der quantisierten Eingangs-, Zustands-, und Ausgangsgrößen. Ein solches System, bei dem die Systemgrößen ausschließlich quantisiert und damit nur durch deren qualitative Werte gegeben sind, nennt man ein *quantisiertes System* [84]. Das Verhalten eines quantisierten Systems zu beschreiben, ist das Ziel der qualitativen Modellierung.

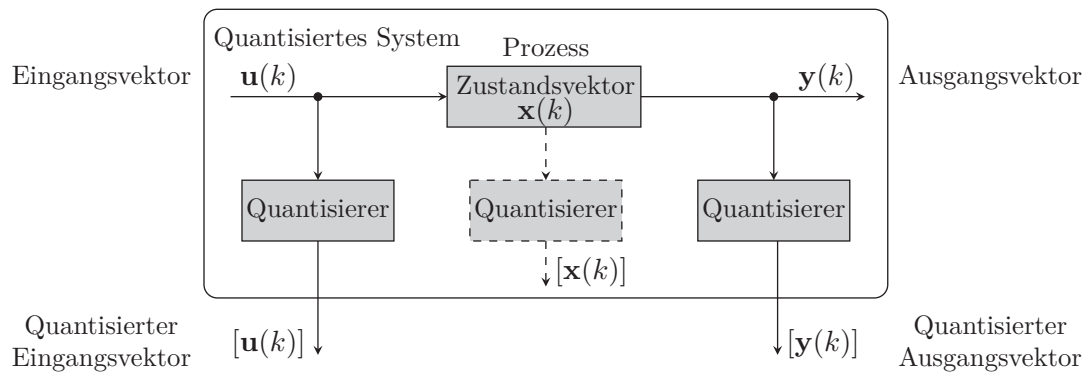


Abbildung 1.6: Quantisiertes System aus Sicht eines Beobachters, nach [113, S. 170]

Quantisiertes System: Wie Abbildung 1.6 zeigt, besteht das quantisierte System aus dem zeitdiskreten Prozess aus Abbildung 1.3 und den *Quantisierern*. Die Quantisierer konvertieren den reellwertigen Eingangsvektor $\mathbf{u}(k)$, Zustandsvektor $\mathbf{x}(k)$ und Ausgangsvektor $\mathbf{y}(k)$ in die quantisierten Eingangs- $[\mathbf{u}(k)]$, Zustands- $[\mathbf{x}(k)]$ und Ausgangsvektor $[\mathbf{y}(k)]$. Im Gegensatz zum zeitdiskreten System mit den reellwertigen Eingangs- und Ausgangsvektoren, tritt das quantisierte System für einen externen Beobachter nur durch die qualitativen Werte der quantisierten Ein- und Ausgangsvektoren in Erscheinung. Das quantisierte System spiegelt somit das *qualitative Eingangs-Ausgangs-Verhalten* des zeitdiskreten Systems wieder [69, S. 4]. Die Quantisierung des Zustandsvektors ist in der Abbildung gestrichelt eingezeichnet, da dieser auf der einen Seite für einen Beobachter des Systems weder quantitativ noch qualitativ ersichtlich ist, auf der anderen Seite jedoch bei der qualitativen Modellbildung von Bedeutung ist, weil er die Beziehung zwischen Ein- und Ausgängen beschreibt.

Überführt man ein wertkontinuierliches und zeitdiskretes dynamisches System wie in Abbildung 1.4 oder 1.5 in ein quantisiertes System, dann wird man dessen Verhalten nur noch näherungsweise, anhand der qualitativen Werte der Eingangs- und Ausgangsgrößen, bewerten können (s. Abbildung 1.1). Man kann dann beispielsweise über einen Volumenstrom oder eine Temperatur nur noch sagen, ob deren Signalwerte sehr hoch, hoch, mittel oder niedrig sind. Um zu bewerten, ob sich das System nominal oder fehlerhaft verhält, können solche qualitativen Einschätzungen ausreichend sein. Das für den Raum und die RLT-Anlage im Beispiel 1.1 beschriebene unplausible Systemverhalten, ist dann beispielsweise dadurch beschrieben, dass die quantisierten Eingangsgrößen der Zulufttemperatur $[T_{Zu}(k)]$ und des Zuluftvolumenstroms $[\dot{V}_{Zu}(k)]$, den qualitativen Wert »sehr hoch« besitzen, aber

die Ablufttemperatur $[T_{Ab}(k)]$ nur »niedrig« ist. Um technische Systeme in dieser Art und Weise, auf Basis der qualitativen Werte ihrer Ein- und Ausgangsgrößen, *kontinuierlich und automatisiert* zu überwachen, kann ein qualitatives Modell verwendet werden.

Modellvorstellung

Qualitative Modelle sind *nichtdeterministische Modelle*, deren Grundlage ein sogenannter *stochastischer Automat* bildet und der das qualitative Verhalten des zugrundeliegenden Systems in einem wahrscheinlichkeitstheoretischen Kontext beschreibt [69], [83], [113]. Die Modelle repräsentieren dabei das Verhalten eines Systems nur näherungsweise, bilden jedoch die grundlegende Systemdynamik ab. Ein qualitatives Modell soll dabei das zugrundeliegende System stets nur »so genau wie nötig und so grob wie möglich beschreiben« [113, S. 6], um die Modellkomplexität niedrig zu halten. Die *Modellbildung* kann durch unterschiedliche Methoden erfolgen und wird in dieser Arbeit durch ein Black-Box Verfahren realisiert, bei dem das Modell anhand historischer Messdaten der Gebäudesysteme identifiziert wird. Dies bietet den Vorteil einer sehr schnellen Modellbildung, bei der prinzipiell keine physikalischen Kenntnisse des zugrundeliegenden Systems benötigt werden, da das explizite Aufstellen von Modellgleichungen entfällt.

Im Hinblick auf die Zielsetzung der Modellierung unterscheidet man sowohl bei quantitativen als auch bei qualitativen Modellen zwischen zwei unterschiedlichen Ansätzen, die man als *Simulation* und *Beobachtung* bezeichnet. Bei der Simulation wird ein Modell eines dynamischen Systems dazu verwendet, um für gegebene oder gemessene Eingangsgrößen, die Ausgangsgrößen zu bestimmen (s. Kapitel 2). Bei der Beobachtung dient das Modell dem Ziel, anhand der messbaren Ein- und Ausgangsgrößen des Systems, die in der Praxis nicht messbaren Zustandsgrößen zu berechnen. Das Konzept der Beobachtung bietet dabei die Möglichkeit, das Eingangs-Ausgangs-Verhalten des Systems hinsichtlich seines fehlerfreien Betriebs zu bewerten. Die Fehlererkennung mit qualitativen Modellen basiert auf diesem Beobachterprinzip. Modellbedingt spricht man dabei von der sogenannten *qualitativen Beobachtung* [71].

Qualitatives Modell zur Beobachtung: Im Sinne der qualitativen Beobachtung erfolgt die Modellierung technischer Systeme unter der folgenden Zielsetzung:

Das qualitative Modell soll anhand der zur Zeit k gemessenen qualitativen Werte des quantisierten Eingangs- $[\mathbf{u}(k)]$ und Ausgangsvektors $[\mathbf{y}(k)]$ des quantisierten Systems, die Wahrscheinlichkeiten dafür bestimmen, welche qualitativen Werte der quantisierte Zustandsvektor $[\mathbf{x}(k+1)]$ des Prozesses zur Zeit $k+1$ annehmen kann.

Das Konzept der qualitativen Beobachtung ist in Abbildung 1.7 dargestellt und lässt sich anhand des folgenden Beispiels einfach erläutern.

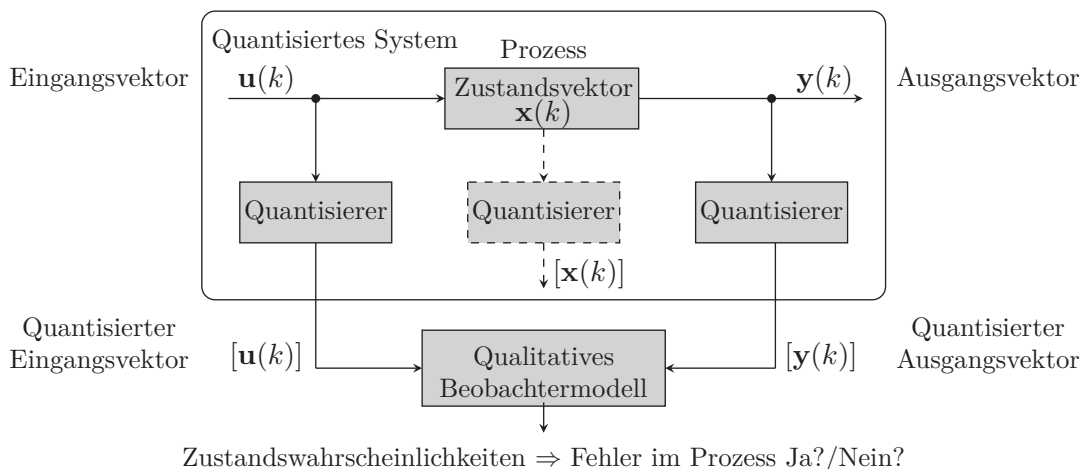


Abbildung 1.7: Qualitative Beobachtung und Fehlererkennung, nach [113, S. 170]

Beispiel 1.2 (Qualitative Beobachtung): Der Wasserbehälter aus Abbildung 1.4 soll anhand seiner quantisierten Ein- und Ausgänge beobachtet werden. Dabei wird angenommen, dass die quantisierten Ein- und Ausgangsgrößen $[\dot{V}_{Zu}(k)]$, $[\dot{V}_{Ab}(k)]$ zur Zeit $k = 0$ gemessen wurden und die in Abbildung 1.8 gezeigten qualitativen Werte besitzen.



Abbildung 1.8: Wasserbehälter mit quantisierten Systemgrößen

Das qualitative Modell soll anhand der zur Zeit $k = 0$ gegebenen qualitativen Werte der Ein- und Ausgänge bestimmen, welche qualitativen Werte die quantisierte Zustandsgröße $[h(k)]$, sprich die Füllhöhe, zur Zeit $k = 1$ annehmen kann (s. Abbildung 1.8). Dabei sind die qualitativen Werte $\{\text{niedrig, mittel, hoch, sehr hoch}\}$ möglich. Da das qualitative Modell ein stochastisches Modell ist, wird es diese Frage nicht mit einem konkreten qualitativen Wert beantworten, sondern es wird alle qualitativen Werte der Zustandsgröße mit einer Wahrscheinlichkeit berechnen, wie zum Beispiel:¹²

$$\begin{aligned} \Pr([h(1)] = \{\text{niedrig}\}) &= 0, \\ \Pr([h(1)] = \{\text{mittel}\}) &= 0.1, \\ \Pr([h(1)] = \{\text{hoch}\}) &= 0.2, \\ \Pr([h(1)] = \{\text{sehr hoch}\}) &= 0.7. \end{aligned} \tag{1.5}$$

Die Abkürzung \Pr steht für die »Wahrscheinlichkeit«. Auf Basis der zur Zeit $k = 0$ gemessenen quantisierten Ein- und Ausgangsgrößen $[\dot{V}_{Zu}(0)]$, $[\dot{V}_{Ab}(0)]$ trifft das qualitative Modell für die Zeit $k = 1$ die folgende Prognose: Die Füllhöhe des Wasserbehälters $[h(1)]$ wird mit 70-prozentiger Wahrscheinlichkeit den qualitativen Wert »sehr hoch« besitzen und mit 20-prozentiger Wahrscheinlichkeit den Wert »hoch« usw. Die Zustandswahrscheinlichkeiten in Gleichung (1.5) stellen somit das Ergebnis der qualitativen Beobachtung für einen Zeitschritt k dar (s. a. Abbildung 1.7).

Qualitatives Modell zur Fehlererkennung: Um ein qualitatives Modell zur Beobachtung und Fehlererkennung in realen Gebäudesystemen verwenden zu können, muss dieses eine grundsätzliche Forderung erfüllen:

Das qualitative Modell muss ein Nominalmodell des zugrundeliegenden Systems darstellen und damit dessen fehlerfreies qualitatives Verhalten abbilden.

Die Fehlererkennung erfolgt nach einem *Konsistenzprinzip*. Dieses basiert auf der Bewertung, ob die am realen System gemessenen quantisierten Eingänge $[\mathbf{u}(k)]$ und Ausgänge $[\mathbf{y}(k)]$ mit dem qualitativen Nominalmodell konsistent sind [11], [84]. Diese Bedingung ist immer dann erfüllt, wenn für die gemessenen quantisierten Ein- und Ausgänge positive Zustandswahrscheinlichkeiten existieren, wie etwa in Beispiel 1.2 in Gleichung (1.5). Tritt am realen System nun ein Fehler auf, wie zum Beispiel eine Leckage im Wasserbehälter, dann zeigen die Eingangs- und Ausgangsgrößen ein vom fehlerfreien Betrieb abweichendes Verhalten und das qualitative Nominalmodell wird keine positiven Zustandswahrscheinlichkeiten bestimmen können. In einem solchen Fall sind alle Zustandswahrscheinlichkeiten null, was bedeutet, dass die gemessenen quantisierten Ein- und Ausgänge inkonsistent mit dem qualitativen Nominalmodell sind und ein Fehler im beobachteten System vorliegt.

¹²Bei den in Gleichung (1.5) gegebenen Wahrscheinlichkeiten handelt es sich genau genommen um bedingte Wahrscheinlichkeiten, da diese die qualitativen Zustandswahrscheinlichkeiten unter der Bedingung gegebener Eingangs- und Ausgangsgrößen repräsentieren. Zudem wird das qualitative Modell die qualitativen Zustandswahrscheinlichkeiten in der Regel nur näherungsweise bestimmen können und wird darüber hinaus sogar unzutreffende Lösungen generieren. Diese Modelleigenschaften werden in Kapitel 4 erläutert.

Eigenschaften qualitativer Modelle: Die wesentlichen Eigenschaften qualitativer Modelle, die im Verlauf dieser Arbeit detailliert erläutert werden, sind nachstehend zusammengefasst:

- Qualitative Modelle können, wie in dieser Arbeit realisiert, durch *Black-Box Verfahren*, basierend auf historischen Anlagenmessdaten oder durch Abstraktion anhand von White-Box Modellen, identifiziert werden [69, S. 50–60].
- Qualitative Modelle sind mathematisch einfach verständlich und vollständig beschreibbar, denn die stochastischen Automaten, die das Modell repräsentieren, besitzen die *Markow-Eigenschaft* [113, S. 18]. Zur Beschreibung und Anwendung der Modelle werden in dieser Arbeit weder Differentialgleichungen noch Integralrechnungen benötigt. Alle Berechnungen basieren auf einfachen Additionen und Multiplikationen, die sich meist durch Produkte der linearen und multilinearen Algebra generalisieren lassen.
- Qualitative Modelle können *Unsicherheiten* in den gemessenen Ein- und Ausgangsgrößen berücksichtigen [113, S. 163–165].
- Qualitative Modelle können sowohl zur Simulation, zur Beobachtung, zur Steuerung und Regelung als auch zur Fehlererkennung und Diagnose verwendet werden [69, S. 60–83].
- Qualitative Modelle eignen sich ebenso zur Beschreibung *linearer* als auch *nichtlinearer* Systeme [72], [73].
- Qualitative Modelle sind dynamische Modelle, welche die zugrundeliegende Systemdynamik näherungsweise abbilden. Im Vergleich zu vielen anderen Methoden eignen sie sich damit auch zur Erkennung von Fehlern, die sich in der Systemdynamik manifestieren (vgl. hierzu Kapitel 2).
- Qualitative Modelle besitzen in Bezug auf ihren Speicherbedarf ein *Komplexitätsproblem*. In Abhängigkeit davon, durch wie viele verschiedene qualitative Werte die quantisierten Eingangs-, Zustands- und Ausgangsgrößen des dynamischen Systems beschrieben werden, kann bereits die Berücksichtigung nur weniger Systemgrößen zu qualitativen Modellen führen, die mit handelsüblichen PC's nicht mehr prozessierbar sind.
- Qualitative Modelle sind durch *Tensoren* darstellbar und somit durch moderne Verfahren wie die *Tensordekomposition* in ihrem Speicherbedarf erheblich reduzierbar [91], [92].

1.4 Forschungsfragen

Qualitative Modelle zur Fehlererkennung sind bereits seit den 1990er Jahren bekannt und in der Literatur umfangreich beschrieben [69], [76], [79], [80], [113]. Der Ursprung der Modelle liegt in der Prozess- und Regelungstechnik, in deren unterschiedlichen Bereichen auch ihr Anwendungsschwerpunkt lag. Während sich in der Industrie- und Prozesstechnik die Systeme oft durch sehr gute messtechnische Ausstattungen mit geringen Abtastzeiten auszeichnen, besteht in Gebäudesystemen oft ein Mangel an Datenpunkten und die Abtastzeiten liegen im Minuten- oder gar im Stundenbereich. Technische Gebäudeanlagen stellen mit diesen Eigenschaften insbesondere für modellbasierte Methoden zur Fehlererkennung eine Herausforderung dar. Ob sich die qualitativen Modelle zur Fehlererkennung in technischen Gebäudesystemen eignen, ist in der Literatur bisher nur in eigenen Arbeiten untersucht worden [91], [92], [93], [94]. Im Rahmen dieser Dissertation werden die wesentlichen Voraussetzungen dafür geschaffen, qualitative Modelle in der Gebäudeautomation nutzbar zu machen. Aus anwendungsorientierter Sicht bedeutet dies, die folgenden Fragestellungen zu beantworten:

- Was sind die Anforderungen an die Gebäudemessdaten? Welche Daten können zur Modellbildung verwendet werden und über welchen Zeitraum müssen diese vorliegen?
- Welche Informationen müssen bei der Modellbildung bekannt und berücksichtigt werden? Durch wie viele qualitative Werte sollen die quantisierten Eingangs-, Zustands- und Ausgangsgrößen repräsentiert werden?

- Welche Informationen müssen einem Gebäudebetreiber vorliegen, um die Ergebnisse eines qualitativen Modells bewerten zu können?
- Welche Vorteile ergeben sich aus der Verwendung qualitativer Modelle zur Fehlererkennung, im Vergleich zum Stand der Technik in der Gebäudeautomation?
- Was sind die Grenzen qualitativer Modelle? Was können diese leisten und was nicht? Bei welchen Systemen ist der Einsatz qualitativer Modelle sinnvoll?

Über den oben aufgezählten anwendungsorientierten Fragestellungen steht eine diesen übergeordnete wissenschaftliche Fragestellung, die das *Komplexitätsproblem qualitativer Modelle* betrifft. Eine der größten Hürden bei der Anwendung qualitativer Modelle ist deren sehr großer Speicherbedarf.¹³ Dieser wird bestimmt von der Anzahl der möglichen qualitativen Werte, welche die quantisierten Eingangs-, Zustands- und Ausgangsgrößen annehmen können. Tabelle 1.1 verdeutlicht dies anhand einiger Beispiele.

Tabelle 1.1: Speicherbedarf qualitativer Modelle

Anzahl Zustandsgrößen	Qual. Werte pro Zustandsgröße	Anzahl Eingangsgrößen	Qual. Werte pro Eingangsgröße	Anzahl Ausgangsgrößen	Qual. Werte pro Ausgangsgröße	Speicherbedarf des qual. Modells
1	4	1	4	1	4	$4^{2 \cdot 1} \cdot 4^1 \cdot 4^1 = 256$
2	4	2	4	2	4	$4^{2 \cdot 2} \cdot 4^2 \cdot 4^2 = 65\,536$
2	6	2	6	2	6	$6^{2 \cdot 2} \cdot 6^2 \cdot 6^2 = 1\,679\,616$
4	5	5	6	3	4	$5^{2 \cdot 4} \cdot 6^5 \cdot 4^3 = 1.944 \cdot 10^{11}$

Die erste Zeile der Tabelle 1.1 zeigt die Berechnung des Speicherbedarfs für ein kleines qualitatives Modell, wie beispielsweise das des Wasserbehälters in Beispiel 1.2. Da der Wasserbehälter als System nur über je eine Eingangs-, Zustands- und Ausgangsgröße verfügt, die durch jeweils vier qualitative Werte repräsentiert werden, ergibt sich ein Speicherbedarf von 256 Werten. Die zweite Zeile der Tabelle zeigt den Speicherbedarf für ein System mit jeweils zwei Eingangs-, Zustands- und Ausgangsgrößen mit je vier qualitativen Werten. Der Speicherbedarf beträgt dann bereits 65 536 Werte, wobei angemerkt sei, dass es sich bei einem technischen System mit nur je 2 Systemgrößen immer noch um ein eher weniger komplexes System handelt. Wie die Tabelle zeigt, besitzen insbesondere die Zustandsgrößen einen großen Einfluss auf den Speicherbedarf. Modellbedingt geht deren Anzahl stets zweifach in die jeweilige Potenz ein, was bei der Berechnung in der letzten Tabellenspalte durch die in Schwarz dargestellte Zwei ausgedrückt ist, die eine Konstante darstellt. Die Zeilen drei und vier zeigen weitere Beispiele für den Speicherbedarf qualitativer Modelle. Während ein Modell mit einem Speicherbedarf von 1 679 616 Werten noch mit einem handelsüblichen PC berechenbar ist, übersteigt ein Modell mit einem Speicherbedarf von $= 1.944 \cdot 10^{11}$ dessen Kapazitäten bereits erheblich.

Bisher wurde der Speicherbedarf qualitativer Modelle durch die Verwendung sogenannter *spärlich besetzter Matrizen*¹⁴ reduziert. Ein wesentlicher Nachteil bei der Verwendung solcher Datenformate besteht jedoch darin, dass die Effizienz dieser Methode stets von der Anzahl der Nicht-Nullelemente

¹³Als Speicherbedarf wird in dieser Arbeit die Anzahl der Werte verstanden, die man speichern muss, um ein technisches System durch ein qualitatives Modell beschreiben zu können.

¹⁴In der Mathematik werden Matrizen, die nur wenige von Null verschiedene Elemente besitzen, als spärlich besetzte Matrizen bezeichnet. In der Informatik und Datenverarbeitung ist damit ein spezieller Datentyp gemeint, der die Spärlichkeit einer solchen Matrix ausnutzt und bei der Speicherung nur die Werte der Matrix berücksichtigt, die ungleich Null sind.

der zugrundeliegenden Daten abhängt. Zwar wird es sich insbesondere bei der Modellerstellung nicht vermeiden lassen, auf spärlich besetzte Datenformate zurückzugreifen, jedoch können für die Anwendung qualitativer Modelle auch andere Ansätze verfolgt werden. In dieser Arbeit wird eine solche Methode vorgestellt, die im Gegensatz zu den spärlichen Datenformaten eine Modellreduktion im mathematischen Sinne darstellt und die auf der Speicherung der Modelldaten in Form von *Tensoren* basiert. Dabei kommt eine moderne und aktuelle Methode zur Modellreduktion zum Einsatz, die man als *Tensor-Dekomposition* bezeichnet und die es ermöglicht, den Speicherbedarf qualitativer Modelle soweit zu reduzieren, dass diese auch für größere praktische Anwendungen nutzbar sind.

1.5 Struktur der Arbeit

Die Kapitel dieser Arbeit sind wie folgt strukturiert: Kapitel 2 gibt zunächst einen umfangreichen Überblick bezüglich des Stands der Technik der Fehlererkennung und Diagnose in der Gebäudeautomation. In Kapitel 3 werden die stochastischen Automaten eingeführt, welche die theoretische Grundlage der qualitativen Modelle für die Fehlererkennung in den Gebäudesystemen bilden. Die Grundlagen der qualitativen Modelle und wie diese zur Beobachtung und Fehlerdetektion verwendet werden können, sind in Kapitel 4 beschrieben. Kapitel 5 erläutert die theoretischen Grundlagen aus den Kapiteln 3 und 4. Dabei wird die gesamte Prozedur der qualitativen Modellbildung an einfachen Beispielen illustriert und die grundlegenden Modelleigenschaften werden anhand einfacher Rechenbeispiele erläutert. Außerdem fasst dieses Kapitel Anwendungsbeispiele der qualitativen Fehlererkennung an RLT-Anlagen, Heizkreisen und thermischen Speichern zusammen. Kapitel 6 führt die algebraischen Grundlagen der Tensormathematik ein, welche das Fundament für die notwendige Modellreduktion bilden. In Kapitel 7 wird die Tensorrepräsentation qualitativer Modelle eingeführt, die wiederum die Grundlage zur Lösung des Komplexitätsproblems stochastischer Automaten und qualitativer Modelle auf Basis der Tensordekomposition bildet. In Kapitel 9 sind die Algorithmen zur Fehlerdetektion für die dekomponierten Tensorrepräsentationen stochastischer Automaten und qualitativer Modelle beschrieben und werden anhand von Anwendungsbeispielen erläutert. Spezielle methodische Herangehensweisen zur weiteren Speicherreduktion qualitativer Modelle und stochastischer Automaten sind in Kapitel 10 erläutert. Kapitel oder Abschnitte, die mit einem Stern »*« gekennzeichnet sind, enthalten komplexe und wissenschaftlich interessante Verfahren zur Modellreduktion, die jedoch nicht notwendigerweise in der Praxis angewandt werden müssen.

2

Fehlererkennung und Diagnose in der Gebäudeautomation

Dieses Kapitel bietet einen Überblick der Methoden zur Fehlererkennung und Diagnose in der Gebäudeautomation. Abschnitt 2.1 gibt eine Übersicht bezüglich der für die Fehlererkennung und Diagnose in der Gebäudetechnik relevanten Methoden. In Abschnitt 2.2 wird der Stand der Technik erläutert und die Funktionsweise der üblicherweise verwendeten Ansätze anhand einfacher Beispiele erklärt. Die forschungsrelevanten Methoden sowie ihre Vor- und Nachteile werden in Abschnitt 2.3 beschrieben.

2.1 Übersicht der Methoden

In der Literatur existiert eine sehr große Anzahl von Methoden zur Fehlererkennung und Diagnose, wovon die meisten ihren Ursprung in sicherheitsrelevanten Bereichen der Industrie- und Prozesstechnik haben. Die methodischen Herangehensweisen unterscheiden sich mitunter erheblich und sind letztlich unterschiedlichen Problem- und Fragestellungen geschuldet. Zusätzlich unterscheiden sich die Methoden darin, was sie imstande sind zu leisten, denn nicht jedes Verfahren eignet sich über die Fehlererkennung hinaus auch zur Diagnose. Die Übersicht in Abbildung 2.1 zeigt eine mögliche Klassifizierung der Verfahren zur Fehlererkennung und Diagnose.

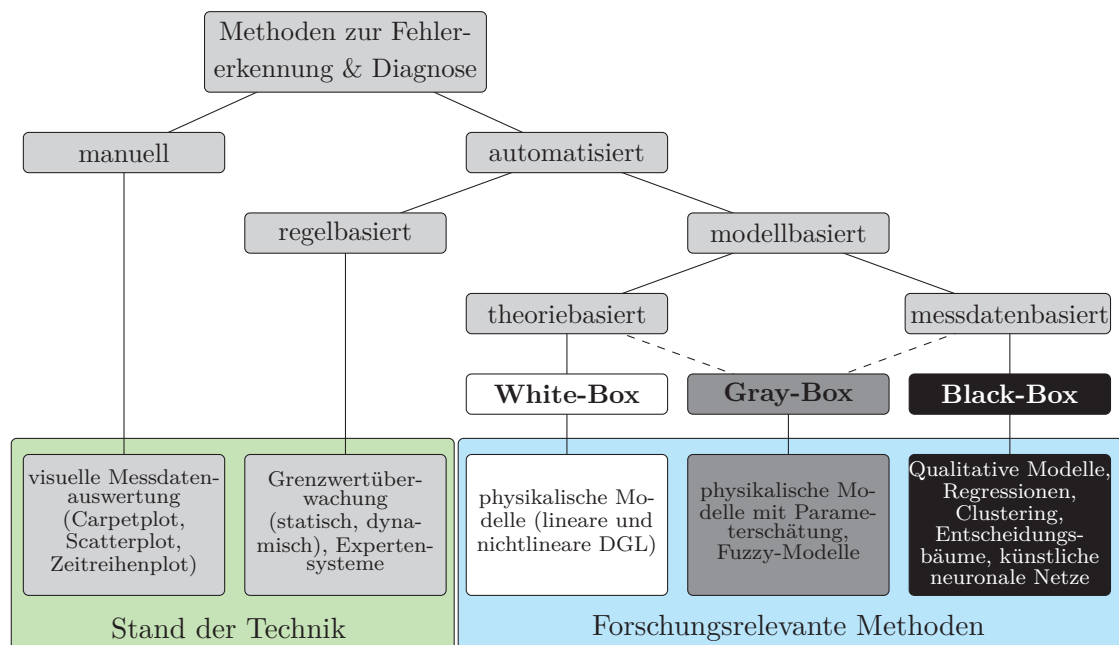


Abbildung 2.1: Methoden zur Fehlererkennung und Diagnose, nach [45, S. 73], [106, S. 235]

Alternative Möglichkeiten, die Methoden zur Fehlererkennung und Diagnose einzuordnen, finden sich unter anderem in der Literatur [45], [47], [124]. Literatur mit einem gebäudetechnischen Schwerpunkt

findet sich insbesondere in [8], [47], [54]. Wie Abbildung 2.1 zeigt, wurden die Verfahren hier in zwei Gruppen eingeteilt: Die erste Gruppe fasst die manuellen und regelbasierten Ansätze zusammen und spiegelt damit den *Stand der Technik* in der Gebäudeautomation wieder. Die modellbasierten Methoden, die entweder durch theoretische oder messdatenbasierte Modellierungsprinzipien generiert werden, bilden die zweite Gruppe und repräsentieren die *forschungsrelevanten Methoden* zur Fehlererkennung und Diagnose.¹ Die einzelnen Verfahren beider Gruppen sowie ihre Vor- und Nachteile werden im Folgenden erläutert.

2.2 Stand der Technik

Der Stand der Technik in der Gebäudeautomation ist dadurch gekennzeichnet, dass die in Form von Zeitreihen zur Verfügung stehenden Ein- und Ausgangsgrößen dynamischer Gebäudesysteme visuell ausgewertet oder mithilfe einfacher Genwerte überwacht werden. Die Verfahren, die dem Stand der Technik in Abbildung 2.1 zugeordnet sind, werden im Folgenden erläutert.

2.2.1 Zeitreihenplots

Zeitreihenplots stellen die in der Gebäudeautomation am häufigsten anzutreffende Variante zur Visualisierung von Messdaten und deren Überprüfung hinsichtlich fehlerhafter Betriebszustände dar. Sie eignen sich insbesondere zur Darstellung kürzerer Zeitbereiche, da die Übersichtlichkeit mit zunehmender Anzahl von Messwerten und Messsignalen stetig abnimmt. Eine kontinuierliche Überwachung der Gebäudesysteme durch die ausschließliche manuelle Analyse von Zeitreihenplots, ist insbesondere für größere Gebäude mit komplexer Anlagentechnik kaum möglich. Abbildung 2.2 zeigt beispielhaft ein fehlerhaftes Signal eines Außenlufttemperatursensors einer RLT-Anlage in Mittelitalien. Anhand der gemessenen Temperaturen, die von -25 °C bis $+55\text{ °C}$ reichen, lässt sich leicht nachvollziehen, dass der Außentemperatursensor offensichtlich defekt ist. Da bei RLT-Anlagen die Außenlufttemperatur eine grundlegende Größe zur korrekten Regelung der Anlagenkomponenten wie Erhitzer, Kühler, Befeuchter und Wärmerückgewinnung darstellt, hat ein solcher Fehler mitunter gravierenden Einfluss auf die allgemeine Anlagenfunktion, die energetische Effizienz und den thermischen Komfort der zur versorgenden Gebäudezone.

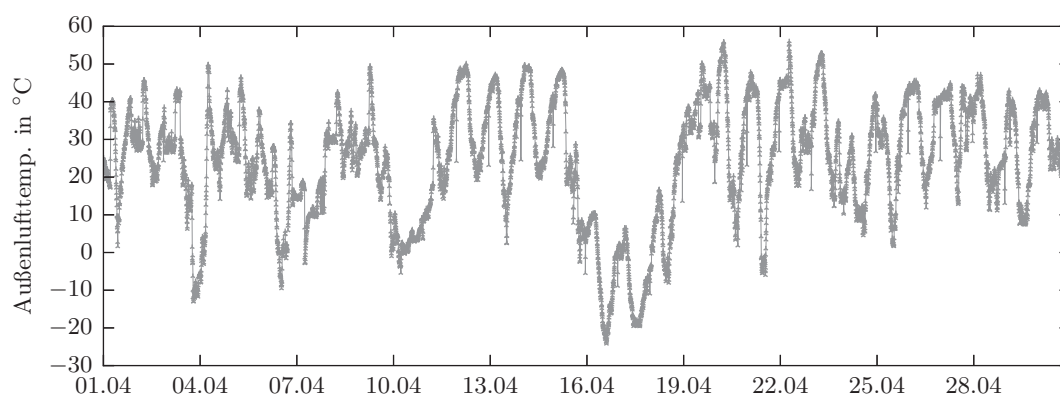


Abbildung 2.2: Zeitreihe eines defekten Außentemperatursensors einer RLT-Anlage, dargestellt über einen Monat (Jahr 2016). Abtastzeit $T_a = 300\text{ s}$

¹Abbildung 2.1 zeigt lediglich eine Auswahl der forschungsrelevanten Methoden. Signalbasierte Verfahren zur Fehlererkennung und Diagnose, die beispielsweise auf Fourier-Transformationen oder Korrelationsanalysen basieren, werden hier nicht berücksichtigt. Siehe hierzu beispielsweise [45].

Zeitreihenplots werden in der Praxis häufig durch interpolierte Signale dargestellt, das heißt, die eigentlich zeitdiskreten Messwerte (s. Gleichung (1.1)), die stets nur zu bestimmten Abtastzeiten vorliegen, werden meist linear miteinander verbunden und als kontinuierlicher Kurvenverlauf abgebildet. Durch diese Vorgehensweise erhöht sich zwar die Übersichtlichkeit des Signalverlaufs, allerdings steigt auch die Wahrscheinlichkeit für Fehlinterpretationen an. Dies ist insbesondere bei den in der Gebäudeautomation üblichen Abtastzeiten, von häufig mehr als 300 Sekunden, problematisch, weil dadurch der Anschein erweckt wird, dass die Bereiche zwischen den einzelnen Messwerten tatsächlich bekannt sind. Während der dadurch generierte Fehler bei trägen Signalen wie Außentemperaturen relativ gering ist, kann dies bei Ventilsignalen oder zeit- und wertdiskreten Signalen (An/Aus) zu falschen Schlüssen bezüglich der Anlagendynamik führen.

Um mithilfe von Zeitreihenplots Fehler im Anlagenbetrieb aufzudecken, ist es in der Regel notwendig, mehrere Signale einer Anlage oder einer Komponente miteinander zu vergleichen. Die Entscheidung darüber, welche Signale einer Anlage zur Überprüfung der ordnungsgemäßen Funktion betrachtet werden müssen, erfordert häufig ein hohes Maß an Expertenwissen und Verständnis bezüglich der zugrundeliegenden Anlagendynamik. Abbildung 2.3 zeigt beispielsweise drei charakteristische Signale eines Erhitzers einer RLT-Anlage in Mittelitalien sowie die Außentemperatur. Bei den Messwerten des Lufterhitzers handelt es sich um die Wasservorlauftemperatur, den Öffnungsgrad des 3-Wege-Ventils und den Massenstrom des Heißwassers.

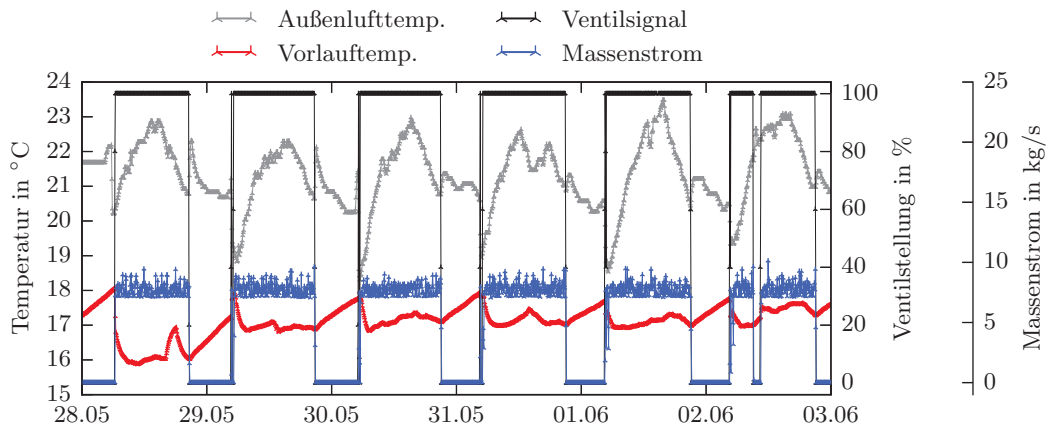


Abbildung 2.3: Zeitreihen von vier Sensoren eines Erhitzers einer RLT-Anlage, dargestellt über 6 Tage (Jahr 2014). Abtastzeit $T_a = 300$ s.

In der Abbildung ist deutlich zu erkennen, dass wenn das Ventil öffnet, der Massenstrom des Heißwassers ordnungsgemäß ansteigt, jedoch die Heißwasservorlauftemperatur nur ca. 16 – 18 °C beträgt, wodurch eine Erwärmung der Zuluft nicht gewährleistet werden kann. Für dieses Problem kommen unterschiedliche Ursachen in Betracht. Zunächst muss die Frage gestellt werden, ob bei den gegebenen Außentemperaturen ein Betrieb des Erhitzers überhaupt nötig ist. Falls ja, muss entsprechend sichergestellt werden, dass eine ausreichende Heißwasserversorgung gewährleistet ist und ein Defekt am Wärmeerzeuger ausgeschlossen werden. Anders stellt sich die Ursachenforschung dar, wenn der Erhitzer aufgrund der gegebenen Rahmenbedingungen grundsätzlich ausgeschaltet sein sollte. In diesem Fall kommen als Fehlerursache auch zu hoch eingestellte Sollwerte für die Zuluft- oder Zonentemperaturen in Frage.

Wie dieses einfache Beispiel zeigt, ist die Detektion des Fehlers relativ einfach. Anhand der in Abbildung 2.3 dargestellten Signale ist unmittelbar ersichtlich, dass ein grundlegendes Problem vorliegt und es sich nicht um einen nominalen Anlagenbetrieb handeln kann. Die Fehleridentifikation und Fehlerisolation sind hingegen schwieriger zu realisieren, was auch daran liegt, dass der Fehler in der Übergangjahreszeit auftrat. In dieser Zeit wechseln sich Erhitzerbetrieb, freie Kühlung mit der Außenluft und Kühlerbetrieb häufig ab, was die Ursachenforschung zusätzlich erschwert.

2.2.2 Carpetplots

Carpetplots, auch Teppichdiagramme genannt, dienen der grafischen Darstellung *einzelner* Zeitreihen und eignen sich besonders zur Repräsentation langer Zeitbereiche. Carpetplots bieten dabei eine tendenziell qualitative Repräsentation der zugrundeliegenden Messdaten, da exakte Werte nicht ablesbar sind. Diese Art der Darstellung besitzt ihre Stärke in der Hervorhebung von Mustern, Zeitplänen sowie zyklischen Betriebscharakteristiken und bietet eine übersichtliche Datenrepräsentation. Abbildung 2.4 zeigt beispielsweise einen Carpetplot, basierend auf stündlichen Temperaturmittelwerten. Dabei wurde oben die Außenlufttemperatur und unten die Raumtemperatur eines ausgewählten Raumes eines Bürogebäudes in Westdeutschland dargestellt. Die Carpetplots sind dabei so aufgebaut, dass jeweils 24 Stundenmittelwerte vertikal zu einem Tag übereinander angeordnet werden. Die Stundenmittelwerte werden durch unterschiedliche Farben repräsentiert, deren Zuordnung zu den jeweiligen Temperaturniveaus der Skala auf der rechten Seite der Abbildung entnommen werden kann. Auf der y -Achse der Carpetplots ist somit die Uhrzeit und auf der x -Achse das Datum aufgetragen.

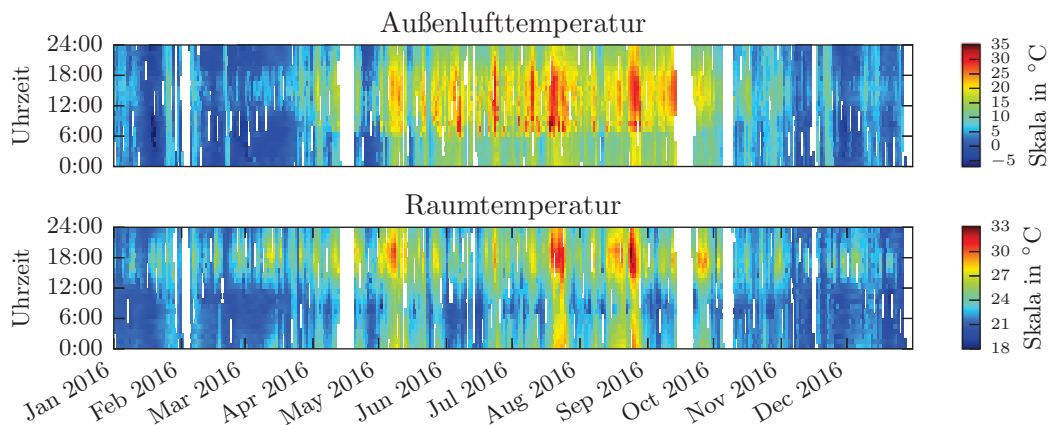


Abbildung 2.4: Carpetplot der Außenluft- und Raumtemperatur eines Bürogebäudes in Deutschland, basierend auf Stundenmittelwerten

Wie in der Abbildung 2.4 unmittelbar ersichtlich ist, kommt es im Sommer und in den Übergangsjahreszeiten, trotz maschineller Kühlung in Form thermoaktiver Bauteilsysteme, zu erheblichen Komforteinbußen mit Raumtemperaturen von über 30 °C. Deutlich zu erkennen ist auch die aufgrund der thermischen Gebäudemasse etwas versetzte, aber starke Korrelation der Raumtemperatur mit der Außenlufttemperatur. Die in den Abbildungen auftretenden vertikalen und in weiß dargestellten Bereiche kennzeichnen fehlende oder unplausible Messdaten, was ein typisches Problem der Messdatenerfassung in der Gebäudetechnik darstellt.

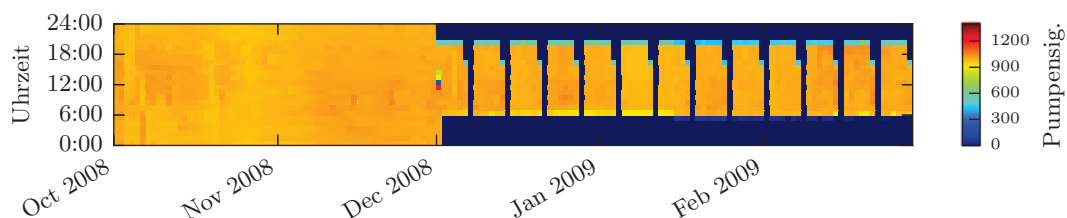


Abbildung 2.5: Carpetplot des Betriebssignals einer Doppelpumpe, basierend auf Stundenmittelwerten

Abbildung 2.5 zeigt das Betriebssignal einer Heizkreis-Doppelpumpe. Wie aus der Abbildung ersichtlich ist, befindet sich die Pumpe während der Monate Oktober und November 2008 im Dauerbetrieb,

während zu Beginn des Monats Dezember ein Zeitplan mit Nacht- und Wochenendabsenkung eingeführt wurde.

Weitere Informationen zu Carpetplots und wie die Muster, die sich aus der Visualisierung verschiedener Gebäudedaten ergeben, interpretiert werden können, finden sich in [96, S. 112–132].

2.2.3 Scatterplots

Scatterplots, auch Streudiagramme genannt, dienen häufig der Veranschaulichung der Abhängigkeit zweier Variablen oder Messgrößen voneinander. Dabei wird eine beliebige Anzahl von Datenpaaren (2-Tupeln) wie $(u^{(1)}, y^{(1)})$, $(u^{(2)}, y^{(2)})$, \dots , $(u^{(n)}, y^{(n)})$ in ein zweidimensionales kartesisches Koordinatensystem eingetragen.² Scatterplots berücksichtigen keine Zeit, was bedeutet, dass die Information bezüglich der zeitliche Abfolge in der die Messdaten zueinander stehen, verloren geht. In der Gebäudetechnik werden Scatterplots in der Regel dazu verwendet, Messgrößen wie Raumtemperaturen, Heiz- und Kühlleistungen, beispielsweise in Abhängigkeit von der Außenlufttemperatur zu visualisieren. Abbildung 2.6 zeigt einen Scatterplot, basierend auf Stundenmittelwerten der Außenlufttemperatur (x -Achse) und der Heizleistung (y -Achse) eines öffentlichen Gebäudes in Deutschland. Der Zusammenhang zwischen hoher Heizleistung und kalten Außenlufttemperaturen sowie die Abnahme der Heizleistung mit steigender Außenlufttemperatur, ist deutlich zu erkennen. Außerdem wurden in der Abbildung die Leistungswerte der Wochenenden in der Farbe rot eingezeichnet.

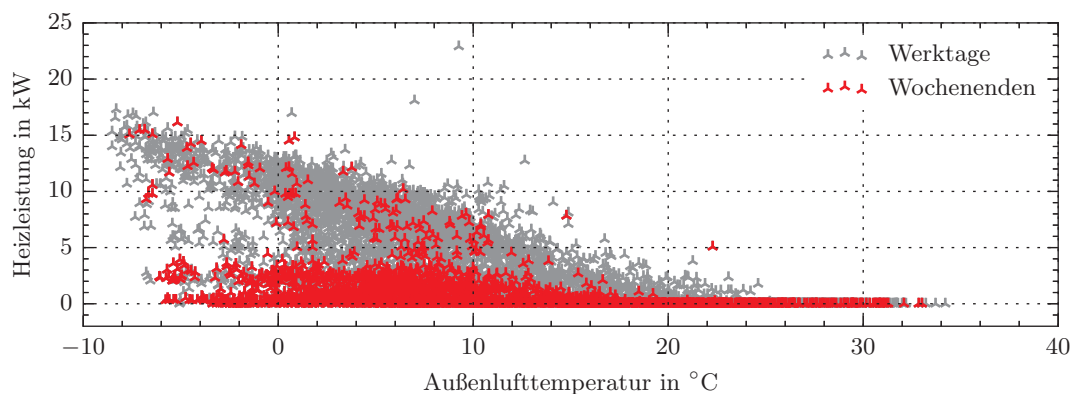


Abbildung 2.6: Scatterplot der Heizleistung und Außenlufttemperatur eines öffentlichen Gebäudes in Deutschland mit Absenkbetrieb, basierend auf Stundenmittelwerten des Jahres 2016

Wie man in Abbildung 2.6 erkennen kann, ist die Heizleistung an den Wochenenden erheblich geringer, was für eine funktionierende Wochenendabsenkung spricht. Bei Gebäuden ohne Wochenendabsenkung liegen die Punktwolken für Werktagen und Wochenenden entsprechend übereinander. Dies zeigt Abbildung 2.7 beispielhaft anhand eines Wohngebäudes, bei dem naturgemäß keine Absenkung an den Wochenenden erfolgt.

Scatterplots bieten somit eine sehr einfache Möglichkeit, Gebäude, bei denen ein Absenkbetrieb an Wochenenden prinzipiell möglich ist, dahingehend zu überprüfen und gegebenenfalls Energieeinsparpotenziale aufzudecken.

²Ein *Tupel* ist eine geordnete Liste von Objekten. Ein Tupel mit zwei Einträgen wie (a, b) oder $(5.8, 7.9)$ nennt man 2-Tupel. Ein Tupel mit drei Einträgen heißt 3-Tupel usw. Bei n Einträgen spricht man von einem n -Tupel. Die Nummerierung der Werte des Messwerttupels erfolgt mit einem hochgestellten Index und dient der Vermeidung von Verwechslungen mit der Nummerierung von Variablen, deren Indexierung tiefgestellt erfolgt.

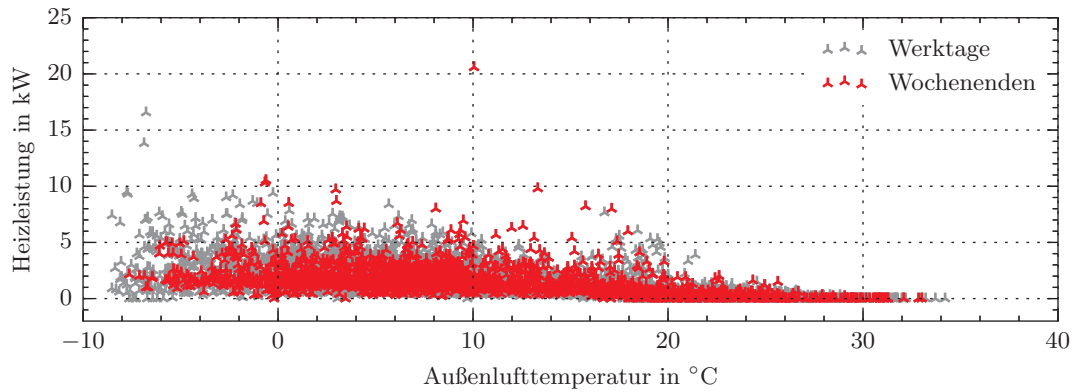


Abbildung 2.7: Scatterplot der Heizleistung und Außenlufttemperatur eines Wohngebäudes in Deutschland ohne Absenkbetrieb, basierend auf Stundenmittelwerten des Jahres 2016

Scatterplots eignen sich zudem zur Bewertung des thermischen Komforts in Gebäuden. Diese Bewertung erfolgt anhand des nationalen Anhangs der DIN EN 15251 [30] und ist beispielhaft in Abbildung 2.8 für einen Büroraum dargestellt. Dem abgebildeten Scatterplot, bei dem auf der y -Achse die Raumtemperatur und auf der x -Achse die Außenlufttemperatur dargestellt ist, liegen grundsätzlich die gleichen Messdaten zugrunde, wie dem unteren Carpetplot in Abbildung 2.4. Allerdings wurden für den Scatterplot in Abbildung 2.8 nur die Wertepaare abgebildet, die innerhalb der Gebäudenutzungszeit liegen. Da es sich hierbei um ein Bürogebäude handelt, wurde die Nutzungszeit auf die Wochentage Montag bis Freitag und die Zeiten 7:00 bis 19:00 Uhr festgelegt. Die gestrichelt in Rot eingezeichnete Linie stellt die quasi ideale Komfortraumtemperatur dar. Von dieser Temperatur gilt nach DIN EN 15251 eine Abweichung von ± 2 K als tolerierbar, was in der Abbildung mithilfe durchgehend roter Linien dargestellt ist.

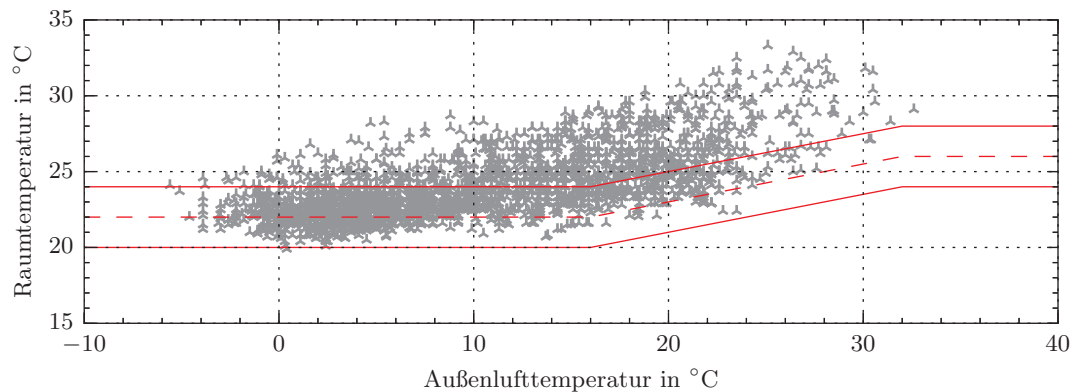


Abbildung 2.8: Scatterplot der Raumtemperatur und Außenlufttemperatur eines Bürogebäudes in Deutschland, basierend auf Stundenmittelwerten des Jahres 2016. Angenommene Nutzungszeit: Mo. - Fr., 7:00 - 19:00 Uhr

Wie Abbildung 2.8 zeigt, lässt die Darstellung der Messdaten mittels eines Scatterplots eine deutlich präzisere Bewertung des thermischen Komforts zu, als die Darstellung in einem Carpetplot. Es ist unmittelbar ersichtlich, dass es insbesondere in der Übergangsjahreszeit und im Sommer zu erheblichen Komfortverletzungen kommt. Die Summe aller Stundenmittelwerte in denen die Komfortgrenzen nicht eingehalten wurden, beläuft sich auf 920 Werte, beziehungsweise. Stunden, was

einem Anteil von 36 % entspricht. Weitere Informationen zu Scatterplots und deren Interpretation findet sich in [96, S. 106–111].

Die Vor- und Nachteile der visuellen Methoden zur Fehlererkennung und Diagnose wie Zeitreihen-, Carpet- und Scatterplots sind nachfolgenden zusammengefasst.

Vorteile:

- Die Darstellungen von Zeitreihen-, Carpet- und Scatterplots sind intuitiv bewertbar.
- Die Implementierung ist sehr leicht umsetzbar und auch mit Standardprogrammen ohne Programmierkenntnisse realisierbar.

Nachteile:

- Das visuelle Auswerten von Messdaten ist zeitaufwändig und erfordert eine sehr gute Anlagenkenntnis und ein hohes Maß an Expertenwissen.
- Für komplexe Gebäude mit vielen Datenpunkten ist es kaum möglich, alle Messdaten manuell zu bewerten.
- Fehler, die sich in der Systemdynamik zeigen, sind nicht unmittelbar detektierbar.

Der letzte unter den Nachteilen aufgezählte Punkt besitzt eine hohe Relevanz und stellt mitunter einen erheblichen Nachteil dar. Die Detektion von Fehlern, die sich in Form einer sich ändernden Systemdynamik zeigen, wird von vielen Methoden zur Fehlererkennung nicht ohne Weiteres ermöglicht. Hierunter versteht man Fehler, die nicht zwangsläufig zu Überschreitungen von Grenzwerten führen, sondern zur Vergrößerung oder Verringerung von Zeitkonstanten (s. Abbildung 2.9). Typische Beispiele hierfür sind Verkalkungen in Speichern oder Wärmeübertragern, die dazu führen, dass sich die Wärmeübertragungsraten oder Aufheizprozesse gegenüber dem Nominalfall ändern, ohne dass jedoch die grundsätzliche Funktion eingeschränkt wird. Solche Fehler zeigen sich wiederum im Eingangs-Ausgangs-Verhalten dynamischer Systeme. Die Ausgangsgröße verhält sich in solchen Fällen für gegebene Eingänge anders, als dies im Nominalbetrieb der Fall ist. Da solche Fehler einen hohen Einfluss auf die energetische Performance des entsprechenden Systems haben können, ist deren Detektion in jedem Falle wünschenswert.

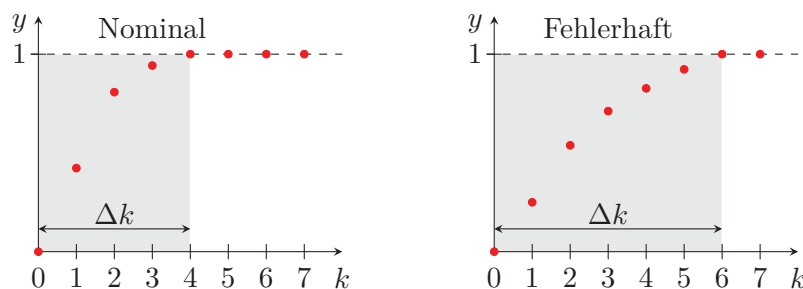


Abbildung 2.9: Fehler, die sich in der Systemdynamik manifestieren, führen dazu, dass sich Zeitkonstanten ändern

2.2.4 Grenzwertüberwachung und Expertensysteme

Die Grenzwertüberwachung sowie die sogenannten Expertensysteme stellen die im Rahmen der *automatisierten* Verfahren zur Fehlererkennung nahezu ausschließlich anzutreffenden Methoden in der Gebäudeautomation dar. Die Abgrenzung dieser Verfahren zueinander ist fließend, denn beide arbeiten mit sogenannten *wenn-dann-sonst-Regeln* (engl.: if-then-else-rules). Mit der Grenzwertüberwachung sind im Allgemeinen jedoch einfachste Regeln gemeint, die anhand fester Grenzwerte, die

Überwachung einzelner Signale hinsichtlich plausibler Werte sicherstellen sollen. Ein Beispiel für eine einfache regelbasierte Grenzwertüberwachung lässt sich anhand des Außentemperatursignals in Abbildung 2.2 zeigen. Die in Abbildung 2.2 dargestellte Zeitreihe der Außentemperatur T_A ist eine Folge

$$T_A(0...T) = (t_A(0), t_A(1), t_A(2), \dots, t_A(T))$$

mit Folgengliedern $t_A(k)$ für $k = 0, 1, 2, \dots, T$ (s. Gleichung (1.1)). Für den in der Abbildung 2.2 dargestellten Zeitbereich von 30 Tagen und eine gegebene Abtastzeit von $T_a = 300$ s, besteht diese Folge aus $T = 8640$ Werten. Eine einfache Regel zur Grenzwertüberwachung der Außentemperatur kann dann folgendermaßen definiert werden:³

$$\forall k = 0, 1, 2, \dots, T : \quad d(k) = \begin{cases} 1 & \text{wenn } -5 \leq t_A(k) \leq 35 \\ 0 & \text{sonst} \end{cases} \quad (2.1)$$

Mit der Regel (2.1) erhält man für jeden Wert der Außentemperatur $t_A(k)$ einen entsprechenden Wert $d(k) \in \{0, 1\}$ und damit eine Folge

$$D(0...T) = (d(0), d(1), d(2), \dots, d(T)) \quad (2.2)$$

Diese Folge beinhaltet für alle Zeiten $k = 0, 1, 2, \dots, T$ die Information, ob die Außentemperatur in dem als plausibel eingeschätzten Temperaturbereich von -5 bis 35°C liegt ($d(k) = 1$) oder ob diese Grenzen überschritten wurden ($d(k) = 0$). Folgen wie (2.2), die somit Aufschluss über das nominale oder fehlerhafte Verhalten von Sensoren, Anlagen oder Komponenten geben, werden in dieser Arbeit als *Verhaltenssignal* bezeichnet. Wendet man die Regel (2.1) auf das zeitdiskrete Signal der Außenlufttemperatur aus Abbildung 2.2 an, erhält man das Verhaltenssignal D aus Gleichung (2.2), was in Abbildung 2.10 dargestellt ist.

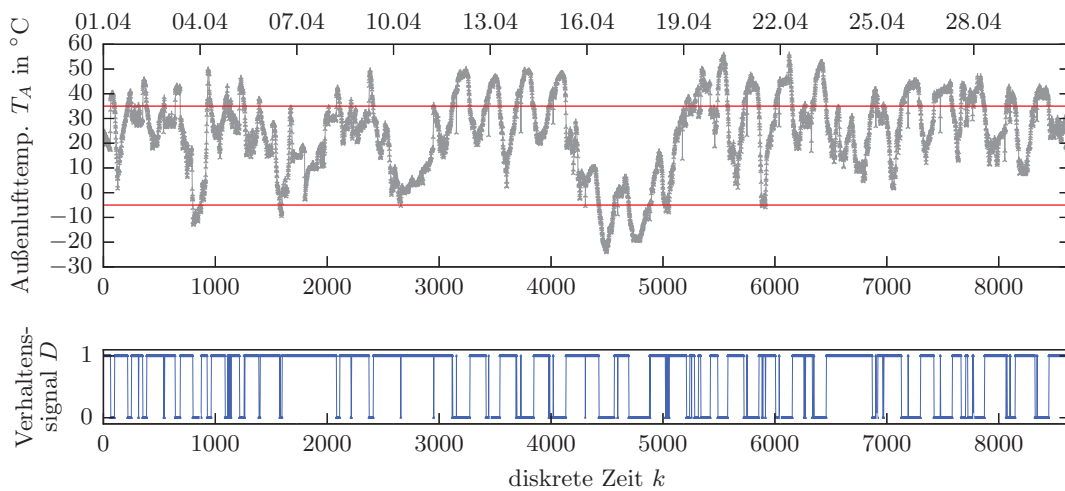


Abbildung 2.10: Zeitreihe der Außenlufttemperatur (oben) und durch einfache Grenzwertüberwachung generiertes Verhaltenssignal (unten)

Wie man am Verhaltenssignal D in Abbildung 2.10 erkennen kann, bietet dieses eine für Nutzer oder Gebäudebetreiber übersichtliche und unmittelbar interpretierbare Form, anhand derer sich die Validität des zugrundeliegenden Sensors ablesen lässt. Die Festlegung der Temperaturgrenzen, ab denen das Signal als unplausibel betrachtet wird, sollten dabei zumindest jahrzeitabhängig festgelegt

³Das Symbol \forall bedeutet »für alle« und wird als *Allquantor* bezeichnet.

werden, um fehlerhafte Signalwerte zuverlässiger detektieren zu können. Die im obigen Beispiel für das Frühjahr und den Monat April konservativ festgelegten Temperaturgrenzen von -5 und 35 °C eignen sich demnach nur unzureichend für die Sommer- und Wintermonate.

Die sogenannten Expertensysteme stellen eine Erweiterung der reinen Grenzwertüberwachung dar und basieren auf komplexeren Regeln. So würde beispielsweise eine ausschließlich auf heuristischen Grenzwerten basierte Signalüberwachung nicht ausreichen, um den Fehler an dem RLT-Erhitzer in Abbildung 2.3 zuverlässig zu detektieren. Zwar kann auch in diesem Fall die Vorlauftemperatur des Heißwassers mit einer Regel wie (2.1) und entsprechend angepassten Grenzwerten überwacht werden, jedoch müssen dazu weitere Kriterien berücksichtigt werden. So darf diese Regel zum Beispiel nur dann aktiv sein, wenn der Erhitzer in Betrieb ist. Das bedeutet, es müssen sowohl der Betriebsmodus der RLT-Anlage (freie Kühlung, Kühlen, Heizen usw.) als auch die Anlagenbetriebszeiten als Eingangskriterien für die entsprechende Regel berücksichtigt werden. Dies ist zwingend notwendig, da sonst eine Regel, welche die Vorlauftemperatur des Heißwassers etwa auf Temperaturen < 30 °C überwachen soll, in den Stillstandszeiten sowie in den Betriebsmodi in denen der Erhitzer ausgeschaltet ist, falsch-positive Fehlermeldungen generiert.

Im Allgemeinen zeigen die Expertensysteme eine mit der zu überwachenden Systemgröße stark ansteigende Komplexität. Für große Gebäude mit hunderten Messpunkten, vielen unterschiedlichen Zeitplänen der Komponenten sowie komplexen Regelungsstrategien, bedeutet es einen nahezu nicht zu bewältigenden Aufwand, Expertensysteme zu entwickeln, die eine Überwachung der gesamten Anlagentechnik sicherstellen. Dies liegt mitunter daran, dass für die oft sehr heterogenen Gebäudesysteme, stets individuelle Anpassungen vorgenommen oder neue Regeln entwickelt werden müssen. Vordefinierte Regelsätze, die einen Basisumfang abdecken, existieren beispielsweise für RLT-Anlagen und sind in [112] beschrieben.

Abschließend lassen sich die Vor- und Nachteile der regelbasierten Methoden zur Fehlererkennung und Diagnose zusammenfassen.

Vorteile:

- Einzelne Regeln sind direkt interpretierbar und einfach verständlich (s. Gleichung (2.1)).
- Kleinere Regelsätze sind leicht implementierbar.
- Komplexe Regeln ermöglichen auch die Fehleridentifikation und -isolation.

Nachteile:

- Das Aufstellen der Regeln für komplexe Systeme ist zeitaufwändig und erfordert eine sehr gute Anlagenkenntnis und ein hohes Maß an Expertenwissen.
- Größere Regelsätze sind mitunter unübersichtlich.
- Hoher Wartungsaufwand, wenn sich Umgebungsbedingungen ändern.
- Fehler, die sich in der Systemdynamik zeigen, sind nicht unmittelbar detektierbar.

2.3 Forschungsrelevante Methoden

Bei den *Forschungsrelevanten Methoden* zur Fehlererkennung und Diagnose in Abbildung 2.1 handelt es sich ausschließlich um *modellbasierte* Ansätze. Die Modellierungsansätze dieser Methoden unterscheiden sich dabei mitunter grundlegend und reichen von *White-Box* Ansätzen, die auf theoretischen physikalischen Gleichungen beruhen, hin zu *Black-Box* Modellen, die auf dem *maschinellen Lernen* basieren und mithilfe historischer Messdaten *trainiert* werden. Außerdem sind Kombinationen beider Modellierungsprinzipien möglich, die man als *Gray-Box* Modelle bezeichnet. Eine grundlegende Eigenschaft, die alle modellbasierten Methoden eint, ist, dass zur Fehlerdetektion, Modelle des Nominalverhaltens der zugrundeliegenden technischen Anlage benötigt werden. Darüber hinaus müssen

zur Fehlerisolation, Nominalmodelle der Teilsysteme, beziehungsweise der einzelnen Anlagenkomponenten, und zur Fehleridentifikation, zusätzliche Fehlermodelle generiert werden. Wie bereits in Abschnitt 1.2 erwähnt, bedeutet die Fehleridentifikation und -isolation damit einen erheblich Mehraufwand, der häufig dem daraus resultierenden Nutzen nicht gerecht wird.

White-Box Modelle

White-Box Modelle, wie beispielsweise die sogenannten Zustandsraummodelle, basieren auf linearen oder nichtlinearen Differential-, beziehungsweise Differenzengleichungen, mit denen die innere Systemdynamik abgebildet wird. In der allgemeinen Form eines nichtlinearen und zeitdiskreten Zustandsraummodells, ist dieses durch die Gleichungen

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) , \quad (2.3)$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) , \quad (2.4)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.5)$$

gegeben [113, S. 38]. Die Vektorfunktion $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ in der *Zustandsgleichung* (2.3) und die Vektorfunktion $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^q$ in der *Ausgangsgleichung* (2.4) beschreiben das Verhalten des zeitdiskreten Systems. Die Zustandsgleichung (2.3) definiert dabei, wie sich der aktuelle Eingangsvektor $\mathbf{u}(k) \in \mathbb{R}^m$ und Zustandsvektor $\mathbf{x}(k) \in \mathbb{R}^n$ auf den Zustandsvektor $\mathbf{x}(k+1) \in \mathbb{R}^n$ zur Zeit $k+1$ auswirkt. Die Zustandsgleichung beschreibt somit die Dynamik des Systems. Die Ausgangsgleichung (2.4) zeigt, wie sich der Ausgangsvektor $\mathbf{y}(k) \in \mathbb{R}^q$ unter Einwirkung des aktuellen Eingangsvektors $\mathbf{u}(k) \in \mathbb{R}^m$ und des Zustandsvektors $\mathbf{x}(k) \in \mathbb{R}^n$ verhält. Die Gleichung 2.5 ist die Startbedingung mit der das Modell initialisiert wird. Das Eingangs-Ausgangs-Verhalten eines dynamischen Systems wie in Abbildung 1.4 oder 1.5, kann durch das Zustandsraummodell (2.3)–(2.5) vollständig beschrieben werden. Solche Modelle erfordern ein umfassendes theoretisches Verständnis der zu modellierenden Systeme, da jede Zustandsgröße des Zustandsvektors durch eine entsprechende Differentialgleichung beschrieben werden muss, was deren Erstellung entsprechend zeitintensiv und komplex macht.

Zur Fehlererkennung und Diagnose können Zustandsraummodelle in Form klassischer Simulations- oder Beobachtermodelle verwendet werden. Bei der Simulation werden die Modellausgaben und Systemausgaben meist anhand von Residuen verglichen. Bei der Beobachtung werden die Residuen der Zustands- und oder Ausgangsgrößen miteinander verglichen [45, S. 231–252]. Außerdem können White-Box Modelle als Grundlage für andere modellbasierte Methoden der Fehlererkennung und Diagnose verwendet werden. Zur Fehlerdetektion an RLT-Anlagen werden in [117] beispielsweise White-Box MODELICA[®] Modelle verwendet, um sogenannte *qualitative deviation models* abzuleiten, welche die Abweichung von Modellgrößen und realen Anlagenmesswerten qualitativ bewerten. In den Publikationen [47], [54] findet sich eine Vielzahl von Literaturverweisen zur Anwendung von White-Box Modellen zur Fehlererkennung und Diagnose in der Gebäudetechnik.

Black-Box Modelle

Black-Box Modelle zeichnen sich, zumindest aus Anwendersicht, durch eine niedrige Komplexität aus. Der Nutzer des Modells benötigt prinzipiell keinerlei Informationen bezüglich der physikalischen Zusammenhänge des betrachteten Systems. Die innere Modellstruktur, die darüber entscheidet wie die Modelleingaben auf die Ausgaben abgebildet werden, wird anhand von verfügbaren Messdaten der Ein- und Ausgangsgrößen eines technischen Systems algorithmisch identifiziert und stellt für den Nutzer in diesem Sinne eine »Black-box« dar (s. Abbildung 2.11). Die damit komfortable Handhabung von Black-Box Modellen, als auch die Vielzahl von verfügbaren Werkzeugen des maschinellen Lernens, wie GOOGLE[™] TENSORFLOW[™] [1], die PYTHON[™] Bibliothek SCIKIT-LEARN [104] oder

die STATISTICS AND MACHINE LEARNING TOOLBOX™ [116] für MATLAB®, haben auch im Bereich der Fehlererkennung und Diagnose in der Gebäudeautomation zu einer großen Verbreitung dieses Modelltyps geführt. Eine Studie, in der 197 Veröffentlichungen mit dem Schwerpunkt der Fehlererkennung und Diagnose in der Gebäudetechnik untersucht wurden, zeigt, dass 123 (62 %) dieser Publikationen auf Black- und Gray-Box Modellierungsansätze entfallen [54]. Von diesen 123 Werken haben die Black-Box Ansätze mit 72 % den mit Abstand größten Anteil, gefolgt von den Gray-Box Modellen (12 %) und verschiedenen weiteren Modellkombinationen [54].



Abbildung 2.11: Black-Box Modell

Black-Box Modelle fallen häufig unter die Methoden des maschinellen Lernens und unterscheiden sich damit von den White-Box Modellen in erster Linie in der Modellbildung.⁴ Während White-Box Modelle durch Differentialgleichungssysteme beschrieben werden, erfolgt die Modellbildung bei Black-Box Modellen auf Basis des *Lernens* anhand historischer Messdaten des technischen Systems. Der Lernprozess des Modells wird dabei als *Trainieren* bezeichnet und die dazu erforderlichen Messdaten als *Trainingsdaten*. Die zum Trainieren eines Black-Box Modells verwendeten Variablen bezeichnet man als *Merkmale* (engl.: Features) und die Anzahl der Messungen, die von diesen Merkmalen zur Verfügung steht, nennt man Stichprobe (engl. Samples). Im Hinblick auf die Modellierungsansätze unterscheidet man zwischen *Regressionsmodellen* und *Klassifikationsmodellen*.

Bei den Regressionsmodellen steht meist die klassische Modellbildung im Sinne eines Simulationsmodells im Vordergrund. Im Hinblick auf die Systemvorstellung in Abbildung 1.2 bedeutet dies, dass anhand verfügbarer Messungen der Eingangs- und Ausgangsgrößen des technischen Systems, ein Modell trainiert wird, welches das Eingangs-Ausgangs-Verhalten des Systems näherungsweise beschreibt. Ist diese Modellbildung abgeschlossen, können *Anwendungsdaten* mit dem Modell prozessiert werden. Das bedeutet, das Modell kann dazu verwendet werden, um für neue Messwerte der Eingangsgrößen des technischen Systems, Schätzungen der Ausgangsgrößen zu berechnen (s. Abbildung 2.12, links). Zur Fehlererkennung kann ein solches Modell verwendet werden, indem die vom Modell berechneten Ausgangsgrößen, mit den tatsächlich gemessenen Ausgangswerten des Systems verglichen werden. Ein anderer Ansatz zur Fehlererkennung wird bei der Verwendung von Klassifikationsmodellen verfolgt. Solche Modelle können dazu genutzt werden, um die Werte der Ein- und Ausgänge des realen Systems anhand verschiedener Kriterien wie »fehlerhaft« oder »nominal« zu klassifizieren (s. Abbildung 2.12, rechts). Als Modelleingangsgrößen werden dann zum Beispiel die Ein- und Ausgänge des technischen Systems verwendet und die Modellausgangsgröße stellt eine diskrete Größe dar, welche den Wert null besitzt, wenn die Messwerte der Systemeingänge und -ausgänge als fehlerhaft klassifiziert wurden und die den Wert eins annimmt, wenn das System sich im Nominalbetrieb befindet.

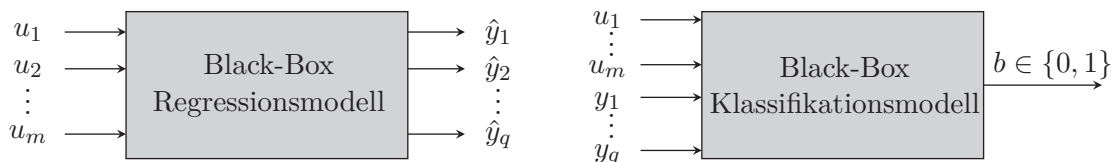


Abbildung 2.12: Regression und Klassifikation mit Black-Box Modellen

⁴Sogenannte *Subspace* Modellidentifikationsverfahren wie N4SID [122], mit denen Zustandsraummodelle anhand von Systemmessdaten identifiziert werden können, werden in der Regel nicht den Methoden des maschinellen Lernens zugeordnet und hier nicht näher erläutert.

Den bereits erwähnten Vorteilen, wie der einfachen Handhabung und Modellbildung der Black-Box Modelle, stehen in der praktischen Anwendung jedoch auch Nachteile gegenüber. So stellen manche Black-Box Modelle in Abbildung 2.1 in ihrer Grundform *statische* Modelle dar oder werden als solche verwendet. Das bedeutet, dass modellseitig für die Berechnung der aktuellen Ausgabe, ausschließlich die aktuellen Eingaben berücksichtigt werden, während die weiter in der Vergangenheit liegenden Eingaben keinen Einfluss mehr besitzen. Das Modell hat damit, im Gegensatz zum dynamischen System, kein Gedächtnis. Es ist für ein solches Modell also völlig unerheblich, zu welchem Zeitpunkt es eine Eingabe erhält, denn es wird darauf stets mit der gleichen Ausgabe antworten. Dieser Umstand ist häufig dem Anspruch an speicher- und recheneffiziente Modelle geschuldet, denn oft bedeutet die Berücksichtigung der Systemdynamik eine deutlich höhere mathematische Komplexität und damit größere Rechenzeiten und Speicherbedarfe. Ein weiterer Grund hierfür ist, dass die meisten Black-Box Ansätze ursprünglich nicht für die Modellierung technischer Systeme, sondern zum Beispiel für die Mustererkennung in Daten konzipiert wurden, bei der zeitliche Zusammenhänge eine untergeordnete Rolle spielen. Auch wenn der Verzicht auf die modellseitige Abbildung der Systemdynamik oft gerechtfertigt ist, stellt dies im Hinblick auf die Detektion von Fehlern, die sich in einer Veränderung der Systemdynamik manifestieren, einen Nachteil dar. Denn statische Modelle können keine dynamischen Fehler detektieren. Bei Black-Box Modellen gilt es damit stets zwischen Modellkomplexität und den erforderlichen Anforderungen an das Modell abzuwägen.

Nachstehend werden die bereits in Abbildung 2.1 aufgeführten und für die Gebäudeautomation besonders relevanten Black-Box Modelle erläutert und deren Vor- und Nachteile in Bezug auf die Fehlererkennung und Diagnose dargelegt.

2.3.1 Regressionsanalyse

Regressionsanalysen wie die einfache lineare, die lineare, die multiple lineare und polynomiale Regression stellen die wohl bekanntesten Verfahren des maschinellen Lernens dar. Abbildung 2.13 zeigt, wie ein einfaches lineares Regressionsmodell zur Fehlererkennung in einem technischen System mit nur einer Eingangsgröße u und einer Ausgangsgröße y (SISO-System) verwendet werden kann.

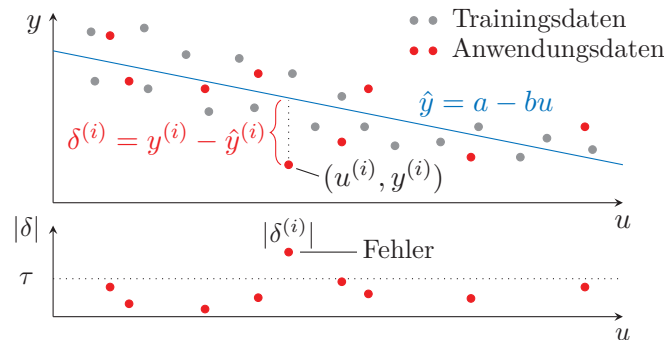


Abbildung 2.13: Prinzipdarstellung Fehlererkennung durch lineare Regression: Datenpaare eines Trainingsdatensatzes (grau) führen zur Regressionsgeraden (blau). Zur Fehlererkennung wird für jedes Datentupel eines Anwendungsdatensatzes (rot) das Residuum berechnet und anhand eines Grenzwerts τ bewertet (unten).

Das einfache lineare Regressionsmodell kann anhand von historischen Messdaten der Ein- und Ausgangsgrößen trainiert werden. Der Trainingsdatensatz ist dann durch eine Menge von Messwertpaaren

$$\{(u^{(1)}, y^{(1)}), (u^{(2)}, y^{(2)}), \dots, (u^{(n)}, y^{(n)})\} \quad (2.6)$$

gegeben, wobei n die Anzahl der Samples, beziehungsweise die Stichprobengröße ist. Ein Trainingsdatensatz wie (2.6) besitzt zwei Merkmale oder Features, da jedes Datentupel zwei für das zugrundelie-

gende System charakteristische Größen enthält. Der Trainingsdatensatz besitzt hier also die gleiche Form wie die Daten, die man zur Erstellung eines Scatterplots verwendet. Ein einfaches lineares Regressionsmodell ist durch eine lineare Funktion

$$\hat{y} = a + bu \quad (2.7)$$

gegeben, die man auch als *Regressionsgerade* bezeichnet. Die algorithmisch anhand der Trainingsdaten bestimmten Regressionsparameter a und b geben den y -Achsenabschnitt sowie die Steigung der Geraden an. Die Funktion (2.7) stellt somit ein lineares Black-Box Modell eines einfachen Eingrößensystems dar. Nach der Modellbildung können Anwendungsdaten prozessiert werden, welche in ihrer Form identisch mit Gleichung (2.6) sind. Anschließend können mit Gleichung (2.7) und den neuen Eingaben $u^{(i)}$, die Schätzwerte der Ausgaben $\hat{y}^{(i)}$ berechnet werden. Zur Fehlererkennung kann die Abweichung zwischen Modell- und Systemausgangsgröße durch Bildung des Residuums

$$\delta^{(i)} = y^{(i)} - \hat{y}^{(i)}$$

bestimmt und anhand eines heuristischen Grenzwerts τ bewertet werden. Dieser entscheidet somit darüber, ob das entsprechende Datenpaar den Nominalbetrieb der Anlage ($|\delta^{(i)}| \leq \tau$) repräsentiert oder ob ein davon abweichender Zustand vorliegt ($|\delta^{(i)}| > \tau$) (s. Abbildung 2.13).

Im Bereich der Fehlererkennung und auch Allgemein haben die einfachen linearen Regressionen wenig Relevanz, da diese nur eine abhängige und eine unabhängige Variable berücksichtigen und darüber hinaus ausschließlich lineare Zusammenhänge modellieren. Da die meisten realen Systeme Mehrgrößensysteme sind, die zudem häufig auch Nichtlinearitäten beinhalten, kommen in der Regel multiple lineare oder polynomiale Regressionen zum Einsatz. Komplexe Anwendungsbeispiele zur Gebäudebetriebsoptimierung und Fehlerdetektion mittels Regressionsverfahren finden sich in [46].

Die Vor- und Nachteile der Regressionsverfahren lassen sich wie folgt beschreiben:

Vorteile:

- Das Modell ist in Form einer konkreten Funktion gegeben und gut interpretierbar.
- Residuen geben im Fehlerfall Aufschluss über die Höhe der Abweichung vom Nominalbetrieb.

Nachteile:

- Empfindlich gegenüber Ausreißern beim Trainieren (Wenige Ausreißer können die zu berechnende Funktionsgleichung deutlich beeinflussen)
- Gefahr, Fehler zu trainieren.
- Gefahr der Überanpassung.
- Fehler, die sich in der Systemdynamik zeigen, sind nicht unmittelbar detektierbar.

2.3.2 Künstliche neuronale Netze

Die künstlichen neuronalen Netze (KNN) (*engl.*: artificial neural networks (ANN)) stellen eine weitere Methode des maschinellen Lernens dar und sind in ihrer Struktur biologischen neuronalen Verknüpfungen nachempfunden [89, S. 1]. Künstliche neuronale Netze bestehen aus einzelnen, miteinander verknüpften und interagierenden *Neuronen*. Ein solches Neuron ist links in Abbildung 2.14 dargestellt und besteht aus einer Anzahl von m Eingaben u_1, u_2, \dots, u_m , deren Gewichten w_1, w_2, \dots, w_m , einer Übertragungs- und Aktivierungsfunktion – die hier vereinfacht mit f bezeichnet ist – sowie einer Ausgabe \hat{y} .

Ein künstliches vorwärts-gerichtetes neuronales Netz, wie es rechts in Abbildung 2.14 dargestellt ist, besteht aus vielen einzelnen Neuronen, die sich in eine *Eingabeschicht*, sogenannte *verdeckte Schichten* und eine *Ausgabeschicht* einordnen lassen [89, S. 20]. Während jedes künstliche neuronale Netz

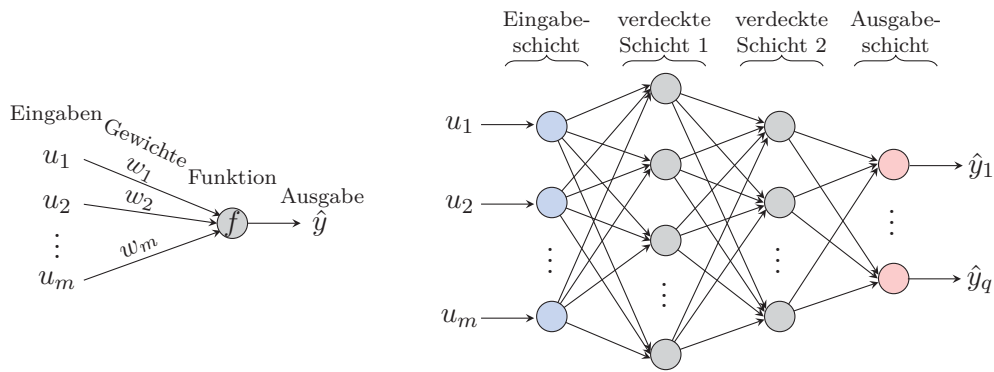


Abbildung 2.14: Künstliches Neuron (links) und zweischichtiges vorwärts-gerichtetes künstliches neuronales Netz (rechts), nach [89, S. 10, 20]

eine Eingabe- und eine Ausgabeschicht besitzt, kann die Anzahl der verdeckten Schichten variieren. Bei nur einer verdeckten Schicht spricht man von einem einlagigen Netz, bei mehreren verdeckten Schichten von einem mehrlagigen Netz. Der Name der verdeckten Schichten rührt von deren Black-Box Charakter, denn die Ausgaben der Neuronen innerhalb dieser Schichten sind von außen nicht sichtbar. Ausgehend von den Eingaben u_1, u_2, \dots, u_m verteilt jedes Neuron der Eingabeschicht, diese an alle Neuronen der ersten verdeckten Schicht. Die Ausgaben der ersten verdeckten Schicht werden dann entsprechend gewichtet und dienen als Eingaben für die zweite verdeckte Schicht. Zuletzt werden die für den Nutzer zugänglichen Ausgaben $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_q$ durch die Ausgabeschicht des Netzes bereitgestellt. Bei einfachen vorwärts-gerichteten Netzen werden die Ausgaben der einzelnen Neuronen stets nur an die nachfolgende Schicht weitergegeben [89, S. 20]. Diese Netze besitzen kein zeitliches Gedächtnis. Das künstliche neuronale Netz in Abbildung 2.14 kann als ein Black-Box Model eines Mehrgrößensystems wie in Abbildung 1.2 interpretiert werden. Trainiert wird ein solches künstliches neuronales Netz anhand eines Trainingsdatensatzes, der sich wiederum aus Messwerten der Ein- und Ausgangsgrößen des zugrundeliegenden technischen Systems zusammensetzt. Der Lernprozess des neuronalen Netzes basiert auf der Bestimmung der Gewichte, die anhand der Trainingsdaten parametrisiert werden, sodass für gegebene Eingaben u_1, u_2, \dots, u_m möglichst präzise Näherungen der Ausgaben $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_q$ generiert werden. Die Fehlererkennung mit neuronalen Netzen kann wiederum durch die Bewertung der Residuen erfolgen.

Neuronale Netze können sowohl zur Regression als auch zur Klassifikation verwendet werden, womit sich zwei verschiedene Möglichkeiten zur Fehlerdetektion ergeben. In beiden Fällen können neuronale Netze außerdem zur Fehlerdiagnose eingesetzt werden, vorausgesetzt es sind Daten verschiedener Fehlerfälle verfügbar. Beispiele zur Anwendung künstlicher neuronaler Netze zur Fehlerdetektion in der Gebäudetechnik finden sich in [32]; auf weitere Anwendungen wird in [47], [54] verwiesen.

Abschließend lassen sich die Vor- und Nachteile künstlicher neuronaler Netze wie folgt zusammenfassen:

Vorteile:

- Einfach zu trainieren.
- Auch nichtlineare Zusammenhänge werden abgebildet.
- Ermöglichen die Fehlerdiagnose wenn Fehlerdaten verfügbar sind.

Nachteile:

- Die Kostenfunktion des Optimierungsproblems ist in der Regel nicht-konvex und nicht-quadratisch, sodass ein Konvergieren in einem globalen Optimum häufig nicht sichergestellt werden kann [90, S. 13].

- Innere Struktur mathematisch schwer oder gar nicht interpretierbar.
- Gefahr, Fehler zu trainieren.
- Gefahr der Überanpassung.
- Fehler, die sich in der Systemdynamik zeigen, können mithilfe einfacher vorwärts-gerichteter Netze nicht unmittelbar detektiert werden. Dies ist jedoch zum Beispiel mit sogenannten *rekurrenten* Netzen möglich, bei denen die Neuronenausgaben einer Netzschicht, auch an die gleiche oder vorige Schichten kommuniziert werden [89, S. 136]. Rekurrente Netze können somit zeitliche Zusammenhänge berücksichtigen und die zugrundelegende Systemdynamik abbilden [45, S. 182–185].

2.3.3 Entscheidungsbäume

Entscheidungsbäume stellen eine Möglichkeit zur automatisierten *Klassifizierung* von Daten sowie zur *Regressionsanalyse* dar. Während bei *Regressionsbäumen* eine reellwertige Ziel- oder Ausgangsvariable für gegebene Eingangsgrößen berechnet wird, erfolgt bei *Klassifizierungsbäumen* die Ausgabe in Form diskreter, also ganzzahliger Werte – den sogenannten *Klassen*. Die Beziehung zwischen Eingangs- und Ausgangsgrößen wird, wie bei allen Black-Box Modellen, anhand von vorhandenen Messdaten *trainiert* und basiert auf dem Erlernen von hierarchischen, logischen Regeln, die durch die *Knoten* des Entscheidungsbaums repräsentiert werden. Das allgemeine Funktionsprinzip der Entscheidungsbäume sowie deren Verwendung zur Fehlerdetektion ist beispielhaft anhand eines Klassifizierungsbaums in Abbildung 2.15 dargestellt und wird im Folgenden erläutert. Allgemeine Grundlagen zu Entscheidungsbäumen finden sich zum Beispiel in [39, S. 305–317], [109].

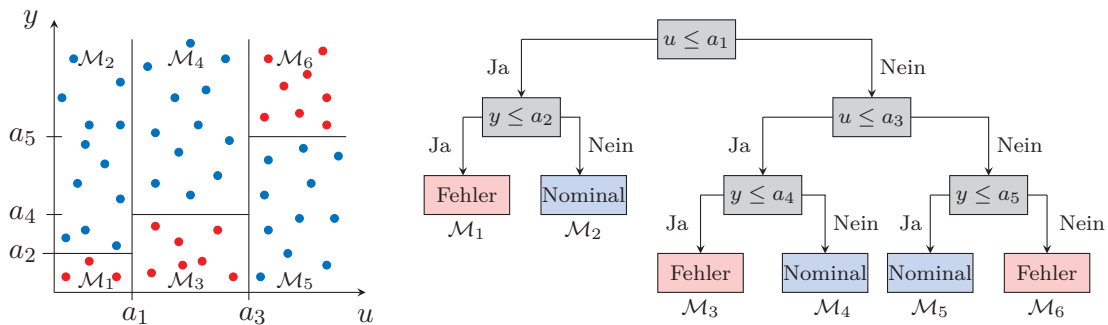


Abbildung 2.15: Prinzipdarstellung Fehlererkennung mit Entscheidungsbäumen: Partitionierung eines zweidimensionalen Merkmalsraums (links) und der daraus resultierende Entscheidungsbaum (rechts) mit zwei Klassen »Fehler« und »Nominal«. Als Fehler klassifizierte Daten sind in Rot, nominale in Blau dargestellt.

Um einen Entscheidungsbaum zur Fehlerdetektion nutzen zu können, müssen sowohl Nominaldaten als auch Fehlerdaten des zugrundeliegenden Systems verfügbar sein. Dies lässt sich wieder anhand eines Eingrößensystems mit einer Eingangsgröße u und einer Ausgangsgröße y erläutern. Der Trainingsdatensatz für eines solches System ist dann durch die Menge

$$\{(u^{(1)}, y^{(1)}, b^{(1)}), (u^{(2)}, y^{(2)}, b^{(2)}) \dots, (u^{(n)}, y^{(n)}, b^{(n)})\}$$

von Messwerttupeln gegeben. Jedes Datentupel $(u^{(i)}, y^{(i)}, b^{(i)})$ enthält somit je einen Messwert der Ein- und Ausgangsgröße sowie die zusätzliche Information, ob dieses Messwertpaar den nominalen oder fehlerhaften Betrieb des Systems beschreibt. Diese Information wird durch die diskrete Zielvariable $b^{(i)} \in \{0, 1\}$ beschrieben, für die $b^{(i)} = 0$ im Fehlerfall und $b^{(i)} = 1$ im Nominalfall gilt. Die

Variable b stellt somit die Ausgangsgröße des Modells dar (s. a. Abbildung 2.12, rechts). Da diese nur zwei Werte annehmen kann, nämlich die Klassen »Nominal« und »Fehler«, spricht man bei einem solchen Entscheidungsbaum auch von einem binären Klassifikator.

Die Grundlage eines Entscheidungsbaums bildet die Partitionierung des sogenannten *Merkmalsraumes* (engl.: feature space) [39, S. 305], in den die Trainingsdaten abgebildet werden. Der in der Abbildung 2.15 links dargestellte zweidimensionale Merkmalsraum der Systemgrößen u und y zeigt dabei sechs Partitionierungen $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_6$. Die Anzahl der Partitionierungen entspricht der Anzahl der *Endknoten* des Entscheidungsbaums. Die Grenzen a_1, a_2, \dots, a_5 dieser Partitionen werden algorithmisch anhand des Trainingsdatensatzes bestimmt, sodass im Idealfall eine scharfe Trennung zwischen den Klassen »Fehler« und »Nominal« entsteht. In der Abbildung sind die Trainingsdaten, die anhand der Variablen b als fehlerhaft bewertet waren, in Rot und die nominal bewerteten in Blau dargestellt. Auf Grundlage der mit dieser Information generierten Partitionen ergeben sich die logischen Regeln, die in den Knoten des Entscheidungsbaumes rechts in Abbildung 2.15 dargestellt sind. Ist der Entscheidungsbaum anhand eines Trainingsdatensatzes generiert worden, kann er anschließend zur Klassifizierung neuer Anwendungsdaten des zugrundeliegenden Systems verwendet werden. Ausgehend vom oberen *Ursprungsknoten* des Entscheidungsbaumes durchläuft dann jedes Datentupel $(u^{(i)}, y^{(i)})$ die nachfolgenden Entscheidungsregeln und terminiert in einem *Endknoten*, der die Klassenzugehörigkeit »Fehler« ($b^{(i)} = 0$) und »Nominal« ($b^{(i)} = 1$) wiedergibt.

Konkrete Beispiele und Erläuterungen zur Anwendung von Fehlerbäumen in der Gebäudetechnik finden sich in [129]. Die Vor- und Nachteile der Entscheidungsbäume können wie folgt zusammengefasst werden:

Vorteile:

- Sehr gut interpretierbar. Die Regeln sind für den Nutzer anhand der Baumstruktur einfach nachvollziehbar.
- Können als Klassifikations- oder Regressionsmethode verwendet werden.
- Ermöglichen die Fehlerdiagnose, wenn entsprechende Fehlermodelle oder Messdaten von Fehlern verfügbar sind.

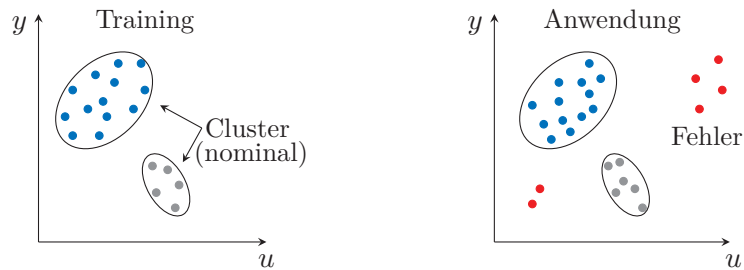
Nachteile:

- Fehlerdaten und Nominaldaten müssen zum Trainieren verfügbar sein.
- Gefahr der Überanpassung (Entscheidungsbaum ist komplexer als er sein müsste).
- Interpretierbarkeit nimmt mit steigender Baumkomplexität ab.
- Die Berechnung eines *idealen* Entscheidungsbaums gehört zu den NP-schweren Problemen und ist damit nicht möglich [44].
- Fehler, die sich in der Systemdynamik zeigen, sind nicht unmittelbar detektierbar.

2.3.4 Clusteranalyse

Die Clusteranalyse ist ein Verfahren zur Klassifizierung von Daten anhand von Ähnlichkeitskriterien. Dabei wird eine Gesamtmenge von Daten in Untermengen – sogenannte *Cluster* – separiert, sodass die Daten eines Clusters möglichst homogen sind und damit zueinander eine größere *Ähnlichkeit* besitzen, als zu Daten eines anderen Clusters [39, S. 501], [3, S. 16]. Zur Berechnung der Cluster existieren verschiedene Ansätze, wie partitionierende, hierarchische oder dichte-basierte Verfahren, für die wiederum unterschiedliche Lösungsalgorithmen verfügbar sind [132].

Abbildung 2.16 zeigt das Prinzip der Fehlerdetektion mittels Clusteranalyse anhand eines einfachen zweidimensionalen Merkmalsraums. Die im linken Teil der Abbildung dargestellten blauen und grauen Punkte, stellen die aus einem Trainingsdatensatz verfügbaren Datentupel $(u_1, y_1), \dots, (u_n, y_n)$ dar.



Abbildungung 2.16: Prinzipdarstellung Fehlererkennung durch Clusteranalyse: Zweidimensionaler Merkmalsraum mit zwei Clustern und Trainingsdatensatz (links) sowie Anwendungsdatensatz (rechts).

Diese können wieder als Messdaten der Ein- und Ausgangsgrößen eines Eingrößensystems interpretiert werden und besitzen damit die gleiche Form wie in Gleichung (2.6). Die sich daraus ergebende Verteilung im Merkmalsraum führt in der Folge auf die eingezeichneten Cluster, die den Nominalzustand beschreiben. Nachdem die Cluster identifiziert sind, können Anwendungsdaten prozessiert werden, welche die gleiche Struktur wie die Trainingsdaten besitzen. Das zugrundeliegende System befindet sich demnach im Nominalbetrieb, wenn die Anwendungsdaten einem der zuvor identifizierten Cluster zugeordnet werden können. Liegen die Daten außerhalb dieser Cluster (rote Punkte), befindet sich das System in einem Zustand der nicht nominal ist und als Fehler bewertet werden kann. Anwendungsbeispiele aus der Gebäudeautomation finden sich in [9].

Die Methode der Clusteranalyse bietet die folgenden Vor- und Nachteile:

Vorteile:

- Training basiert nur auf Nominaldaten.
- Alle Abweichungen vom Nominalbetrieb werden erkannt.
- Ermöglichen die Fehlerdiagnose wenn entsprechende Fehlermodelle erzeugt werden.

Nachteile:

- Neigen zu falsch positiv detektieren Fehlern.
- Gefahr, Fehler zu trainieren.
- Modell schwierig interpretierbar.
- Fehler, die sich in der Systemdynamik zeigen, sind nicht unmittelbar detektierbar.

2.4 Abgrenzung der Methoden zu qualitativen Modellen

Qualitative Modelle unterscheiden sich von den in diesem Abschnitt beschriebenen Methoden vor allem darin, dass diese dynamische Modelle darstellen. Sie ermöglichen es somit, Fehler die sich in der Systemdynamik zeigen, zu detektieren (vgl. Abbildung 2.9). Die qualitativen Modelle besitzen zudem einen direkten Bezug zur ingenieurstechnischen Vorstellung dynamischer Systeme und basieren auf einer klassischen Systeminterpretation. Die Auswahl der Modelleingaben -zustände und -ausgaben erfolgt anhand ihrer Entsprechungen des realen Systems. Qualitative Modelle sind zudem vielseitig anwendbar und können nicht nur zur Beobachtung und Fehlererkennung, sondern auch für andere Aufgaben der Prozess- und Regelungstechnik wie beispielsweise zur Simulation verwendet werden. Zudem bieten sie die Möglichkeit, das zugrundeliegende Verhalten technischer Systeme mit einer einfach einstellbaren Genauigkeit zu approximieren. Die Modellbildung ist für einen Black-Box Ansatz sehr einfach und kann mathematisch leicht nachvollzogen werden. Auch das Modell selbst und seine

innere Struktur sind leicht interpretierbar, was bei vielen anderen Black-Box Ansätzen nicht ohne weiteres möglich ist. In Bezug auf die Fehlererkennung eignen sich die qualitativen Modelle sowohl zur Detektion, aber auch zur Isolation und Identifikation von Fehlern. Die vom qualitativen Modell bei der Beobachtung generierten Zustandswahrscheinlichkeiten lassen sich außerdem sehr gut grafisch darstellen und bieten eine einfache Möglichkeit das Systemverhalten unmittelbar zu visualisieren. Im nächsten Kapitel werden die Grundlagen der qualitativen Modelle – die sogenannten stochastischen Automaten – eingeführt und deren Eigenschaften erläutert.

3

Stochastische Automaten und Prozesse

In diesem Kapitel werden die stochastischen Automaten eingeführt sowie deren Eigenschaften als Modell beschrieben. Dazu werden in Abschnitt 3.1 zunächst die einfachen stochastischen Prozesse, deren Verhalten ausschließlich durch ihre Zustände bestimmt ist, eingeführt. Anschließend wird in Abschnitt 3.2 die Beziehung zwischen diesen Prozessen und den sogenannten autonomen stochastischen Automaten, die den einfachsten Spezialfall eines stochastischen Automaten darstellen, erläutert. In Abschnitt 3.3 werden die einfachen stochastischen Prozesse um Ein- und Ausgaben erweitert, wodurch die Basis für die Einführung der allgemeinen stochastischen Automaten geschaffen wird. Die stochastischen Automaten, welche die modelltheoretische Grundlage für alle in dieser Arbeit betrachteten Prozesse bilden, und welche Fragestellungen diese als Modell beantworten sollen, wird in Abschnitt 3.4 erläutert. Abschnitt 3.5 führt die theoretischen Grundlagen der Beobachtung und Fehlererkennung mit stochastischen Automaten ein.

3.1 Diskrete stochastische Prozesse

3.1.1 Definition

Als diskreter stochastischer Prozess wird ein System bezeichnet, das für diskrete Zeiten $k = 0, 1, 2, \dots$ Zufallsvariablen $Z(k)$ erzeugt, die ausschließlich Werte aus einer endlichen Zustandsmenge

$$\mathcal{Z} := \{1, 2, \dots, N\} \quad (3.1)$$

annehmen können. Ein diskreter stochastischer Prozess ist damit durch eine unendliche Folge von Zufallsvariablen $(Z(k))_{k \in \mathbb{N}}$ definiert, bei der jedes Folgenglied $Z(k)$ einen Zustand $z(k) \in \mathcal{Z}$ annimmt [7, S. 53]. Das Erzeugen einer einzelnen Folge von Zufallsvariablen wird als *Experiment* bezeichnet und führt zu einer Folge von Zuständen $(z(0), z(1), z(2), \dots)$, die man auch als *Trajektorie* oder *Realisierung* des diskreten stochastischen Prozess bezeichnet [113, S. 17]. Auch wenn stochastische Prozesse definitionsgemäß unendliche Folgen darstellen, so ist man in der Praxis häufig an der Beschreibung einer solchen Realisierung bis zu einem beliebigen Zeithorizont $T \in \mathbb{N}$ interessiert. Betrachtet man einen stochastischen Prozess bis zu einer Zeit T , so erhält man eine zufällige *Zustandsfolge* oder *-trajektorie*¹

$$Z(0 \dots T) = (z(0), z(1), z(2), \dots, z(T)) \in \mathcal{Z}^{T+1}, \quad (3.2)$$

die durch die Zustände $z(k)$ für $k = 0, 1, 2, \dots, T$ gegeben ist.

Da der stochastische Prozess ein Zufallsprozess ist, führen Wiederholungen eines Experiments in der Regel auch zu anderen Folgen von Zuständen, also zu verschiedenen Trajektorien (s. Beispiel 3.1). Üblicherweise ist es somit nicht möglich, den Zustand zu einem Zeitpunkt eindeutig zu bestimmen, sondern lediglich die Wahrscheinlichkeit dafür, welchen Zustand $z(k) \in \mathcal{Z}$ der stochastische Prozess, beziehungsweise die Zufallsvariable $Z(k)$, zu einer bestimmten Zeit k annimmt. Die Berechnung dieser *Zustandswahrscheinlichkeiten* wird im nächsten Abschnitt für sogenannte *homogene Markow-Ketten*, die eine wichtige Klasse der stochastischen Prozesse repräsentieren, beschrieben.

¹Zur Notation der Zustandsfolge/Zustandstrajektorie siehe auch Anmerkung 3.1 und 3.2

Anmerkung 3.1 (Kartesisches Produkt von Mengen): *Der in der Notation der Zustandsfolge (3.2) an der Zustandsmenge \mathcal{Z} auftretende Exponent repräsentiert das $(T+1)$ -fache kartesische Produkt der Menge. Sei beispielsweise $\mathcal{Z} = \{1, 2, 3\}$ die Zustandsmenge des stochastischen Prozesses und $T = 1$ der Zeithorizont, dann repräsentiert die Menge*

$$\mathcal{Z}^2 = \mathcal{Z} \times \mathcal{Z} = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$$

alle Zustandsfolgen, die innerhalb von zwei Zeitschritten möglich sind. Eine Zustandsfolge wie zum Beispiel $Z(0..1) = (2, 3)$ ist damit ein Element der Menge \mathcal{Z}^2 .

Anmerkung 3.2 (Zustandstrajektorie): *Die Zustandsfolge $Z(0..T)$ in Gleichung (3.2) stellt eine verkürzte Schreibweise der expliziten Notation*

$$Z(0..T) = (Z(0) = z(0), Z(1) = z(1), Z(2) = z(2), \dots, Z(T) = z(T))$$

einer Zustandstrajektorie bis zu einem Zeithorizont T dar.

Beispiel 3.1 (Zustandstrajektorie): *Das mehrmalige werfen eines Würfels ist ein einfacher diskreter stochastischer Prozess, dessen Zustandsmenge durch die Anzahl der Augen des Würfels gegeben ist. Es ist also $\mathcal{Z} \in \{1, 2, 3, 4, 5, 6\}$. Führt man nun ein Experiment durch, indem man beispielsweise fünf mal hintereinander würfelt und die Zahlen 4, 1, 6, 3, 5 erhält, dann ist die Zustandstrajektorie des Experiments durch*

$$Z(0..4) = (Z(0) = 4, Z(1) = 1, Z(2) = 6, Z(3) = 3, Z(4) = 5) \quad (3.3)$$

gegeben. Wiederholt man das Experiment und würfelt wieder fünf mal, dann wird man sehr wahrscheinlich eine andere Zustandstrajektorie erhalten.

3.1.2 Homogene Markow-Ketten

Ein diskreter stochastischer Prozess besitzt die *Markow-Eigenschaft*, wenn für alle diskreten Zeiten $k \in \mathbb{N}$ sowie für alle Zustände $z(k+1), \dots, z(0) \in \mathcal{Z}$ folgendes gilt [128, S. 11]:

$$\begin{aligned} & \Pr(Z(k+1) = z(k+1) \mid Z(k) = z(k), Z(k-1) = z(k-1), \dots, Z(0) = z(0)) \\ &= \Pr(Z(k+1) = z(k+1) \mid Z(k) = z(k)) . \end{aligned} \quad (3.4)$$

Die bedingte Wahrscheinlichkeit $\Pr(Z(k+1) = z(k+1) \mid Z(k) = z(k))$ beschreibt die Wahrscheinlichkeit dafür, dass der stochastische Prozess in den Zustand $z(k+1)$ übergeht, unter der Bedingung, dass er sich zuvor im Zustand $z(k)$ befand. Da diese *Übergangswahrscheinlichkeit* ausschließlich vom aktuellen Zustand $z(k)$ abhängt und nicht von der weiteren Vergangenheit, spricht man in diesem Fall von einer Markow-Kette erster Ordnung oder auch von einer gedächtnislosen Markow-Kette [86, S. 345–346]. Sind außerdem die Übergangswahrscheinlichkeiten (3.4) zeitinvariant, dann ist die Markow-Kette *homogen* [128, S. 11].

Definition 3.1 (Homogene Markow-Kette [113, S. 18]): *Ein diskreter stochastischer Prozess, bei dem für die Übergangswahrscheinlichkeiten*

$$\forall \bar{z}, z \in \mathcal{Z}, k \in \mathbb{N} : \Pr(Z(k+1) = \bar{z} \mid Z(k) = z) = \Pr(Z(1) = \bar{z} \mid Z(0) = z) \quad (3.5)$$

gilt, wird als homogene Markow-Kette bezeichnet.

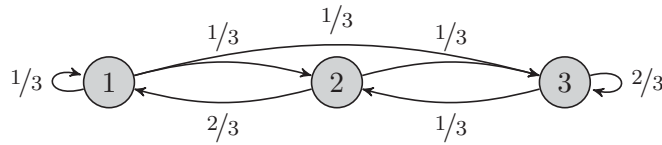
Die Übergangswahrscheinlichkeiten $\Pr(Z(k+1) = \bar{z} \mid Z(k) = z)$, $\bar{z}, z \in \mathcal{Z}$ der homogenen Markow-Kette beschreiben die bedingte Wahrscheinlichkeit dafür, dass der stochastische Prozess zur Zeit $k+1$ in den Folgezustand $\bar{z} \in \mathcal{Z}$ übergeht, wenn er sich zur Zeit k im Zustand $z \in \mathcal{Z}$ befand.

Für die Übergangswahrscheinlichkeiten (3.5) der homogenen Markow-Kette gelten naturgemäß die folgenden Bedingungen:

$$\forall \bar{z}, z \in \mathcal{Z} : 0 \leq \Pr(Z(k+1) = \bar{z} \mid Z(k) = z) \leq 1, \quad (3.6)$$

$$\forall z \in \mathcal{Z} : \sum_{\bar{z}=1}^N \Pr(Z(k+1) = \bar{z} \mid Z(k) = z) = 1. \quad (3.7)$$

Wie Abbildung 3.1 beispielhaft zeigt, lassen sich die Übergangswahrscheinlichkeiten (3.5) sehr anschaulich in Form eines *Zustands-* oder *Übergangsgraphen* darstellen.



Abbildungung 3.1: Zustandsgraph einer homogenen Markow-Kette mit drei Zuständen und eingezeichneten Übergangswahrscheinlichkeiten

Die in der Abbildung dargestellte homogene Markow-Kette besitzt drei Zustände, welche in Form der grauen *Knoten* dargestellt sind und zeigt die an den gerichteten *Kanten* eingezeichneten Zustandsübergangswahrscheinlichkeiten $\Pr(Z(k+1) = \bar{z} \mid Z(k) = z)$. Ein Übergang innerhalb eines Zeitschrittes vom Zustand Eins in Zustand Drei, hat in diesem Beispiel somit die Übergangswahrscheinlichkeit $\Pr(Z(k+1) = 3 \mid Z(k) = 1) = 1/3$.

Diskrete stochastische Prozesse, deren Zustandsübergangsverhalten durch eine homogene Markow-Kette (3.5) gegeben ist, werden auch als *homogene Markow-Prozesse* bezeichnet [11, S. 379]. Im Allgemeinen sind solchen stochastischen Prozessen zu einer Zeit k stets mehrere Zustände mit unterschiedlicher Wahrscheinlichkeit möglich, weshalb man auch von einem *nicht-deterministischen* Prozess spricht [11, S. 378]. Das Interesse bei der Beschreibung solcher Prozesse gilt demzufolge der Wahrscheinlichkeit *jedes* Zustandes $z \in \mathcal{Z}$, den die Zufallsvariable $Z(k)$ zur Zeit k annehmen kann [86, S. 331]:

$$\Pr(Z(k) = z), \quad z \in \mathcal{Z}. \quad (3.8)$$

Die Betrachtung der *Zustandswahrscheinlichkeiten* (3.8) für alle Zustände $z \in \mathcal{Z}$ bildet eine *Wahrscheinlichkeitsverteilung* [87, S. 370], deren Berechnung im folgenden Abschnitt erläutert wird. Für das einfache Beispiel 3.1 des Würfels ist diese Wahrscheinlichkeit direkt ersichtlich, was Beispiel 3.2 erläutert.

Beispiel 3.2 (Zustandswahrscheinlichkeit): *Das in Beispiel 3.1 beschriebene Werfen eines Würfels, stellt einen homogenen Markow-Prozess dar, da die Wahrscheinlichkeit eines Wurfes nicht vom vergangenen Wurf abhängt. Es ist offensichtlich, dass die Übergangswahrscheinlichkeiten (3.5) stets $1/6$ betragen, da es beispielsweise ebenso Wahrscheinlich ist, eine Fünf zu würfeln, wie eine Eins. Für die Wahrscheinlichkeit (3.8) gilt damit für beliebige Zustände $z \in \mathcal{Z} = \{1, 2, 3, 4, 5, 6\}$ und alle Zeiten $k = 0, 1, 2, \dots$*

$$\Pr(Z(k) = z) = 1/6.$$

Zustandswahrscheinlichkeiten

Die Berechnung der Zustandswahrscheinlichkeiten (3.8) erfolgt üblicherweise für zur Zeit $k = 0$ gegebene *initiale* Zustandswahrscheinlichkeiten

$$(p_0)_z := \Pr(Z(0) = z), \quad z \in \mathcal{Z}, \quad (3.9)$$

die für alle $z \in \mathcal{Z}$ definiert werden und damit die initiale Zustandswahrscheinlichkeitsverteilung bilden [87, S. 371–372]. Mit den Startwahrscheinlichkeiten (3.9) und den Übergangswahrscheinlichkeiten (3.5) kann die zeitliche Entwicklung der Zustandswahrscheinlichkeiten für alle Zeiten $k \in \mathbb{N}$ berechnet werden [87, S. 373]:

$$\begin{aligned} k \geq 0 : \quad & \Pr(Z(k+1) = \bar{z}) = \sum_{z=1}^N \Pr(Z(k+1) = \bar{z} \mid Z(k) = z) \cdot \Pr(Z(k) = z) \\ k = 0 : \quad & \Pr(Z(0) = z) = (p_0)_z \end{aligned} \quad (3.10)$$

Gleichung (3.10) muss in jedem Zeitschritt k für alle $\bar{z} \in \mathcal{Z}$ berechnet werden und führt mit den zur Zeit $k = 0$ gegebenen Startwahrscheinlichkeiten, zur Wahrscheinlichkeitsverteilung der Folgezustände $\Pr(Z(1) = \bar{z}), \bar{z} \in \mathcal{Z}$. Diese Wahrscheinlichkeiten repräsentieren die Zustandswahrscheinlichkeiten zur Zeit $k = 1$ und werden wieder in die Gleichung eingesetzt, um die Zustandswahrscheinlichkeitsverteilung zur Zeit $k = 2$ zu berechnen. Durch diese rekursive Anwendung von Gleichung (3.10) können damit die Zustandswahrscheinlichkeiten für beliebige Zeiten $k \in \mathbb{N}$ berechnet werden.

3.2 Autonomer stochastischer Automat

3.2.1 Definition

Die *autonomen stochastischen Automaten* stellen den einfachsten Sonderfall der erst in Abschnitt 3.4 vollständig eingeführten stochastischen Automaten dar und dienen hier als Einführung in die Automatentheorie. Autonome stochastische Automaten sind selbst stochastische Prozesse [113, S. 16] und besitzen in der hier verwendeten Definition die gleichen Eigenschaften wie homogene Markow-Ketten. Solche Automaten können damit als das ingenieurstechnische äquivalent der in der Mathematik als homogene Markow-Ketten bezeichneten Prozesse interpretiert werden [87, S. 372]. In dieser Arbeit werden die stochastischen Automaten dabei allgemein als ein stochastisches Modell verstanden, mit dem das Verhalten technischer dynamischer Prozesse, qualitativ abgebildet werden soll.

Definition 3.2 (Autonomer stochastischer Automat [86, S. 333]): *Das 3-Tupel*

$$\mathcal{A}_a = (\mathcal{Z}, \mathcal{F}, (p_0)_z), \quad (3.11)$$

in dem \mathcal{Z} die Zustandsmenge (3.1) ist und

$$\mathcal{F} : \mathcal{Z} \times \mathcal{Z} \longrightarrow [0, 1] \quad (3.12)$$

eine Wahrscheinlichkeitsverteilung, welche die bedingte Wahrscheinlichkeit

$$\mathcal{F}(\bar{z}, z) = \Pr(Z(k+1) = \bar{z} \mid Z(k) = z) \quad (3.13)$$

für alle möglichen Kombinationen von Folgezuständen $\bar{z} \in \mathcal{Z}$ und Zuständen $z \in \mathcal{Z}$ beschreibt, heißt autonomer stochastischer Automat.

Die Funktion \mathcal{F} in (3.12) wird allgemein auch als *Zustandsübergangsrelation*² bezeichnet [11, S. 381], [113, S. 16]. Da die Zustandsübergangsrelation (3.13) die bedingte Wahrscheinlichkeit dafür angibt, dass der Automat vom Zustand z in den Folgezustand \bar{z} wechselt, schreibt man in der Regel $\mathcal{F}(\bar{z} | z)$ anstatt $\mathcal{F}(\bar{z}, z)$, um diese Bedingung hervorzuheben [87, S. 372]. Da die Startwahrscheinlichkeiten (3.9) als fester Bestandteil der Definition des Automaten \mathcal{A}_a berücksichtigt sind, wird dieser auch als *initialer autonomer stochastischer Automat* bezeichnet (s. a. Anmerkung 3.3).

Es ist offensichtlich, dass der autonome stochastische Automat (3.11) einer homogenen Markow-Kette mit gegebener Startverteilung entspricht. Die Zeitinvarianz der Übergangswahrscheinlichkeiten (3.13) ist wie auch bei den Markow-Ketten nicht zwingend, wird in dieser Arbeit jedoch vorausgesetzt. Damit besteht eine Eins-zu-Eins Entsprechung zwischen den Übergangswahrscheinlichkeiten des autonomen stochastischen Automaten (3.13) und denen der homogenen Markow-Kette (3.5). Es gelten also ebenso die Bedingungen (3.6) und (3.7) und die Berechnung der Zustandswahrscheinlichkeitsverteilungen erfolgt analog zu Gleichung (3.10):

$$\begin{aligned} k \geq 0 : \quad & \Pr(Z(k+1) = \bar{z}) = \sum_{z=1}^N \mathcal{F}(\bar{z} | z) \cdot \Pr(Z(k) = z) \\ k = 0 : \quad & \Pr(Z(0) = z) = (p_0)_z \end{aligned} \quad (3.14)$$

Allgemein lassen sich die Berechnungen der Zustandswahrscheinlichkeiten des Automaten (3.14) übersichtlicher in Form des Matrix-Vektor³ Produkts notieren, welches wie folgt definiert ist.

Definition 3.3 (Matrix-Vektor Produkt): *Das Produkt einer Matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ mit Elementen x_{ij} und eines Spaltenvektors $\mathbf{y} \in \mathbb{R}^J$ mit Komponenten y_j ergibt einen Spaltenvektor $\mathbf{z} \in \mathbb{R}^I$, der die Länge I besitzt:*

$$\mathbf{z} = \mathbf{X} \cdot \mathbf{y} = \begin{pmatrix} x_{11}y_1 + x_{12}y_2 + \dots + x_{1J}y_J \\ x_{21}y_1 + x_{22}y_2 + \dots + x_{2J}y_J \\ \vdots \\ x_{I1}y_1 + x_{I2}y_2 + \dots + x_{IJ}y_J \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_I \end{pmatrix} \in \mathbb{R}^I. \quad (3.15)$$

Die elementweise Berechnung der Komponenten z_i des Ergebnisvektors $\mathbf{z} \in \mathbb{R}^I$ erhält man mit

$$z_i = \sum_{j=1}^J x_{ij} y_j. \quad (3.16)$$

Im nächsten Abschnitt wird gezeigt, wie sich die Zustandswahrscheinlichkeitsverteilungen (3.10) mithilfe des Matrix-Vektor-Produkts berechnen lassen.

3.2.2 Matrixrepräsentation

Um Gleichung (3.14) in Form des Matrix-Vektor Produkts (3.15) zu generalisieren, werden zunächst die Zustandswahrscheinlichkeiten in einem diskreten *Wahrscheinlichkeitsvektor* $\mathbf{p}_z(k) \in [0, 1]^N$ angeordnet, für dessen Komponenten $p_z(k, z)$ folgendes gilt:

$$\forall z \in \mathcal{Z} : \quad p_z(k, z) = \Pr(Z(k) = z). \quad (3.17)$$

²Die Bezeichnung der Funktion \mathcal{F} als *Relation* hat ihren Ursprung in einer mengentheoretischen Beschreibung des Zustandsübergangsverhaltens des Automaten. Allgemeine Informationen hierzu finden sich in [11], [86], [87], [113]

³Eine formale Definition des Matrizen- und Vektorbegriffs findet sich in Abschnitt 6.1.

Mit (3.17) und den Zustandswahrscheinlichkeiten (3.8) erhält man somit einen Wahrscheinlichkeitsvektor

$$\mathbf{p}_z(k) = \begin{pmatrix} p_z(k, 1) \\ p_z(k, 2) \\ \vdots \\ p_z(k, N) \end{pmatrix} := \begin{pmatrix} \Pr(Z(k) = 1) \\ \Pr(Z(k) = 2) \\ \vdots \\ \Pr(Z(k) = N) \end{pmatrix} \in [0, 1]^N, \quad (3.18)$$

der die Wahrscheinlichkeiten $\Pr(Z(k) = z)$ für alle $z = 1, 2, \dots, N$ enthält und damit die diskrete *Wahrscheinlichkeitsverteilung* der Zustände zur Zeit k repräsentiert.

Außerdem können die bedingten Zustandsübergangswahrscheinlichkeiten (3.13) in einer quadratischen *Übergangsmatrix*

$$\mathbf{F} = \begin{pmatrix} \mathcal{F}(1|1) & \mathcal{F}(1|2) & \dots & \mathcal{F}(1|N) \\ \mathcal{F}(2|1) & \mathcal{F}(2|2) & \dots & \mathcal{F}(2|N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{F}(N|1) & \mathcal{F}(N|2) & \dots & \mathcal{F}(N|N) \end{pmatrix} \in [0, 1]^{N \times N} \quad (3.19)$$

angeordnet werden, deren Elementen $\mathcal{F}(\bar{z}|z)$ durch die bedingten Wahrscheinlichkeiten

$$\mathcal{F}(\bar{z}|z) = \Pr(Z(k+1) = \bar{z} | Z(k) = z) \quad (3.20)$$

gegeben sind. Mit (3.20) und den Zustandsübergangswahrscheinlichkeiten (3.13) erhält man die Übergangsmatrix des autonomen stochastischen Automaten. Ein Element $\mathcal{F}(\bar{z}|z)$ der Übergangsmatrix \mathbf{F} repräsentiert demzufolge die bedingte Übergangswahrscheinlichkeit eines Zustandstransits von z nach \bar{z} , wobei der Zeilenindex dem Folgezustand \bar{z} und der Spaltenindex dem Zustand z entspricht.⁴ Damit beschreibt das Element $\mathcal{F}(1|2) = \Pr(Z(k+1) = 1 | Z(k) = 2)$ beispielsweise die bedingte Wahrscheinlichkeit dafür, dass der stochastische Prozess zur Zeit $k+1$ in den Zustand 1 übergeht, wenn er sich zur Zeit k im Zustand 2 befand.

Für die Übergangsmatrix \mathbf{F} gilt analog zu (3.6) und (3.7) der Zusammenhang

$$\forall \bar{z}, z \in \mathcal{Z} : \quad 0 \leq \mathcal{F}(\bar{z}|z) \leq 1 \quad \text{sowie} \quad \forall z \in \mathcal{Z} : \quad \sum_{\bar{z}=1}^N \mathcal{F}(\bar{z}|z) = 1,$$

was bedeutet, dass die Übergangsmatrix *spaltenstochastisch* ist und ausschließlich Elemente mit Werten zwischen Eins und Null enthält. Neben dem diskreten Wahrscheinlichkeitsvektor der Zustände (3.18), werden nun auch die initialen Zustandswahrscheinlichkeiten aus Gleichung (3.9) für alle $z \in \mathcal{Z}$ in einem Vektor

$$\mathbf{p}_0 = \begin{pmatrix} (p_0)_1 \\ (p_0)_2 \\ \vdots \\ (p_0)_N \end{pmatrix} := \begin{pmatrix} \Pr(Z(0) = 1) \\ \Pr(Z(0) = 2) \\ \vdots \\ \Pr(Z(0) = N) \end{pmatrix} \in [0, 1]^N \quad (3.21)$$

angeordnet, der die initiale Zustandswahrscheinlichkeitsverteilung zur Zeit $k = 0$ repräsentiert.

Sind nun die Übergangswahrscheinlichkeiten wie in der Übergangsmatrix $\mathbf{F} \in [0, 1]^{N \times N}$ angeordnet und die initialen Zustandswahrscheinlichkeiten zur Zeit $k = 0$ durch den Vektor der Startverteilung \mathbf{p}_0

⁴In mathematischen Publikationen ist diese Anordnung in der Regel umgekehrt, das heißt der Folgezustand entspricht dem Spaltenindex und der Zustand dem Zeilenindex.

gegeben, dann können die Zustandswahrscheinlichkeitsverteilungen mithilfe des Matrix-Vektor Produkts für beliebige Zeiten $k = 0, 1, 2, \dots$ berechnet werden:

$$\begin{aligned} k \geq 0 : \quad & \mathbf{p}_z(k+1) = \mathbf{F} \cdot \mathbf{p}_z(k) \in [0, 1]^N \\ k = 0 : \quad & \mathbf{p}_z(0) = \mathbf{p}_0 . \end{aligned} \quad (3.22)$$

Zur Zeit $k = 0$ wird Gleichung (3.22) mit dem Vektor der Startverteilung (3.21) initialisiert und man erhält den Vektor $\mathbf{p}_z(1)$ mit der Zustandswahrscheinlichkeitsverteilung zur Zeit $k = 1$. Durch rekursive Anwendung der Gleichung ergeben sich dann die Wahrscheinlichkeitsverteilungen der Zustände für beliebige Zeiten $k \in \mathbb{N}$. Das Zustandsübergangsverhalten stochastischer Automaten kann wie bei homogenen Markow-Ketten durch einen Übergangsgraphen, den man im Kontext der Automaten auch als Automatengraphen bezeichnet, visualisiert werden. Abschließend zu diesem einführenden Abschnitt zeigt ein einfaches Beispiel, wie die Verteilungen der Zustandswahrscheinlichkeiten mit homogenen Markow-Ketten, beziehungsweise durch autonome stochastische Automaten, berechnet werden können.

Beispiel 3.3 (Autonomer stochastischer Automat / Homogene Markow-Kette): *Gegeben sei ein Experiment, beschrieben durch fünf Räume (Abbildung 3.2, oben links) und eine Person, die sich zur Zeit $k = 0$ im Raum Eins befindet.*

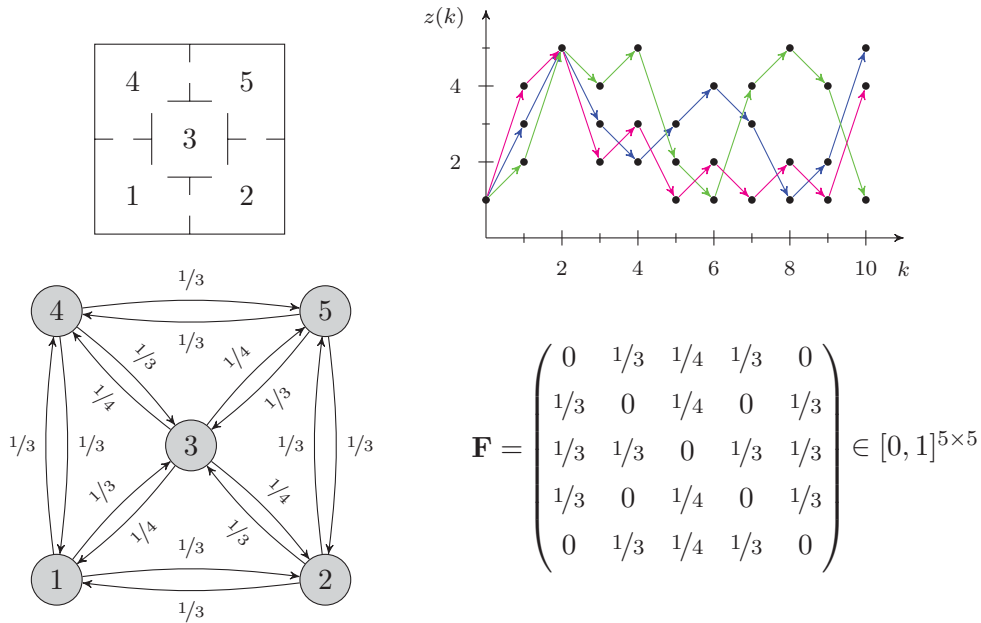


Abbildung 3.2: Oben links: Experiment, oben rechts: drei Realisierungen des Experiments, unten links: Automatengraph, unten rechts: Übergangsmatrix

Die Person entscheidet nun völlig zufällig in jedem weiteren Zeitschritt $k = 1, 2, \dots$ in welchen Raum sie als nächstes wechselt, wobei sie nicht im gleichen Raum verbleiben darf. Da die Zufallsentscheidung, in welchen Raum als nächstes gewechselt wird, in jedem Zeitschritt neu getroffen wird und nicht von der Vergangenheit abhängt, handelt es sich bei dem Experiment offensichtlich um einen homogenen Markow-Prozess. Der stochastische Prozess ist somit durch eine Folge von Zufallsvariablen

$$Z(0 \dots T) = (z(0), z(1), z(2), \dots, z(T))$$

definiert, deren Elemente $z(k)$ die Werte $z \in \mathcal{Z} = \{1, 2, 3, 4, 5\}$ annehmen können. Die Abbildung oben rechts zeigt drei mögliche Realisierungen des stochastischen Prozesses.

Möchte man nun die Wahrscheinlichkeit berechnen, mit der sich die Person zu beliebiger Zeit in irgendeinem Raum befindet, kann ein autonomer stochastischer Automat nach Definition 3.2 verwendet werden. Die Übergangswahrscheinlichkeiten (3.13) sind für den hier beschriebenen Prozess offensichtlich, denn unabhängig davon in welchem Raum man startet, betragen diese stets $1/3$ für den Übergang aus einem Eckraum in irgendeinen angrenzenden anderen Raum und $1/4$ für den Übergang aus dem mittleren Raum in einen der Eckräume. Dieses Übergangsverhalten ist unten links in Abbildung 3.2 in einem Automatengraphen veranschaulicht, während unten rechts die Übergangsmatrix $\mathbf{F} \in [0, 1]^{5 \times 5}$ mit den Elementen $\mathcal{F}(\bar{z} | z) = \Pr(Z(k+1) = \bar{z} | Z(k) = z)$ gegeben ist. Beispielsweise repräsentiert das Element in der fünften Zeile und ersten Spalte die Wahrscheinlichkeit $\mathcal{F}(5 | 1) = \Pr(Z(k+1) = 5 | Z(k) = 1) = 0$, was bedeutet, dass ein Zustandsübergang von Raum Eins nach Fünf innerhalb eines Zeitschrittes nicht möglich ist.

Die Berechnung der Zustandsverteilungen zu einer bestimmten Zeit k erfolgt nun mit Gleichung (3.22), wobei angenommen werden soll, dass sich die Person zur Zeit $k = 0$ im Raum Eins befindet. Damit ist die initiale Verteilung der Zustandswahrscheinlichkeiten mit $\Pr(Z(0) = 1) = 1$ gegeben, wodurch man in der vektoriellen Notation (3.21) einen Spaltenvektor $\mathbf{p}_0 = (1 \ 0 \ 0 \ 0 \ 0)^T$ erhält. Mit der Verteilung der Zustandswahrscheinlichkeiten zur Zeit $k = 0$ und der Gleichung (3.22) berechnet man die Zustandsverteilung zur Zeit $k = 1$:

$$\mathbf{p}_z(1) = \mathbf{F} \cdot \mathbf{p}_z(0) = (0 \ 0.333 \ 0.333 \ 0.333 \ 0)^T.$$

Analog ergibt sich die Wahrscheinlichkeitsverteilung der Zustände zur Zeit $k = 2$ aus der Verteilung zur Zeit $k = 1$:

$$\mathbf{p}_z(2) = \mathbf{F} \cdot \mathbf{p}_z(1) = (0.306 \ 0.083 \ 0.222 \ 0.083 \ 0.306)^T.$$

Die folgende Tabelle veranschaulicht die weitere zeitliche Entwicklung der Wahrscheinlichkeitsverteilungen.

Tabelle 3.1: Zustandswahrscheinlichkeitsverteilungen zur Zeit k

	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=10$	$k=50$	$k=100$
$\Pr(Z(k)=1)$	1	0	0.306	0.111	0.238	0.154	0.192	0.188	0.187
$\Pr(Z(k)=2)$	0	0.333	0.083	0.259	0.139	0.220	0.183	0.187	0.187
$\Pr(Z(k)=3)$	0	0.333	0.222	0.259	0.247	0.251	0.250	0.250	0.250
$\Pr(Z(k)=4)$	0	0.333	0.083	0.259	0.139	0.220	0.183	0.187	0.187
$\Pr(Z(k)=5)$	0	0	0.306	0.111	0.238	0.154	0.192	0.188	0.187

Wie die Tabelle zeigt, sind zur Zeit $k = 1$ die Zustände Zwei, Drei und Vier mit gleicher Wahrscheinlichkeit möglich. Zur Zeit $k = 2$ sind bereits alle Zustände möglich, wobei die Wahrscheinlichkeit, dass sich die Person im Raum Eins oder Fünf befindet am größten ist. Dies lässt sich anhand des Raumschemas in Abbildung 3.2 einfach nachvollziehen, denn es bestehen insgesamt drei Möglichkeiten, um innerhalb von zwei Zeitschritten in den Raum Eins oder Fünf zu gelangen, während jeweils nur eine Möglichkeit existiert, den Raum Vier oder Zwei zu erreichen.

Anmerkung 3.3 (Automatennotation): Die Angabe der initialen Wahrscheinlichkeitsverteilung der Zustände in der Automatennotation (3.11) kann auch vektoriell erfolgen:

$$\mathcal{A}_a = (\mathcal{Z}, \mathcal{F}, \mathbf{p}_0). \quad (3.23)$$

Da in dieser Arbeit die Berechnung der Zustandswahrscheinlichkeiten weniger durch elementweise Notationen wie (3.14) beschrieben wird, sondern vielmehr durch Produkte der linearen und multilinearen Algebra, wird im Folgenden die vektorielle Notation (3.23) bevorzugt verwendet.

3.3 Diskrete Stochastische Prozesse mit Ein- und Ausgabe

3.3.1 Definition

Abbildung 3.3 zeigt einen stochastischen Prozess mit Ein- und Ausgabe (stochastischer E/A Prozess), der eine Erweiterung des in Abschnitt 3.1.2 eingeführten einfachen stochastischen Prozesses darstellt.

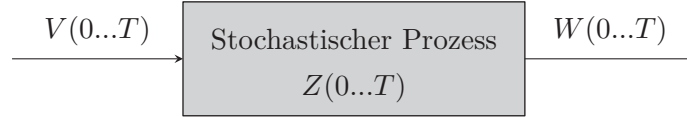


Abbildung 3.3: Stochastischer Prozess mit Eingabe und Ausgabe, nach [86, S. 366]

Betrachtet man den stochastischen E/A Prozess in obiger Abbildung wieder für Zeiten $k = 0, 1, 2, \dots$, so ist dieser nicht nur durch die Folge der Zufallsvariablen der Zustände $(Z(k))_{k \in \mathbb{N}}$ definiert, sondern auch durch die Folge der Zufallsvariablen der Ausgaben $(W(k))_{k \in \mathbb{N}}$ und der Eingaben $(V(k))_{k \in \mathbb{N}}$ [113, S. 18]. Die Zufallsvariablen der Eingaben können dabei die Eingabewerte $v \in \mathcal{V}$ annehmen und die Zufallsvariablen der Ausgaben die Ausgabewerte $w \in \mathcal{W}$. Dabei ist

$$\mathcal{V} := \{1, 2, \dots, M\} \quad (3.24)$$

die diskrete *Eingabemenge* und

$$\mathcal{W} := \{1, 2, \dots, Q\} \quad (3.25)$$

die diskrete *Ausgabemenge* des Prozesses. Der stochastische E/A Prozess generiert damit für eine zufällige Folge von Eingabewerten $(v(0), v(1), v(2), \dots)$, eine vom Zufall bestimmte Ausgabefolge $(w(0), w(1), w(2), \dots)$ und durchläuft eine zufällige Zustandsfolge $(z(0), z(1), z(2), \dots)$. Dies bezeichnet man als Realisierung des stochastischen E/A Prozesses, wobei Wiederholungen eines solchen Experiments, selbst für gleiche Eingabefolgen, zu unterschiedlichen Zustands- und Ausgabefolgen führen [11, S. 378]. Betrachtet man den Prozess wieder bis zu einem definierten Zeithorizont T , dann erzeugt dieser für eine Eingabefolge

$$V(0...T) = (v(0), v(1), v(2), \dots, v(T)) \in \mathcal{V}^{T+1}, \quad (3.26)$$

eine Ausgabefolge

$$W(0...T) = (w(0), w(1), w(2), \dots, w(T)) \in \mathcal{W}^{T+1} \quad (3.27)$$

und durchläuft dabei die Zustandsfolge $Z(0...T)$ aus Gleichung (3.2). Für die Ein- und Ausgabefolgen in (3.26) und (3.27) gelten analog zur Zustandsfolge die Anmerkungen 3.1 sowie 3.2.

Um das Verhalten eines solchen stochastischen E/A Prozesses zu beschreiben, wird wiederum vorausgesetzt, dass dieser die Markow-Eigenschaft besitzt, die im nächsten Abschnitt erläutert wird.

3.3.2 Homogene Markow-Ketten mit Ein- und Ausgabe

Ein stochastischer E/A Prozess besitzt die Markow-Eigenschaft, wenn für alle Zeiten $k \in \mathbb{N}$ und für alle Zustände $z(k+1), \dots, z(0) \in \mathcal{Z}$, Ausgaben $w(k), \dots, w(0) \in \mathcal{W}$ und Eingaben $v(k), \dots, v(0) \in \mathcal{V}$,

folgendes gilt [83, S. 322]:

$$\begin{aligned} & \Pr \left(\begin{array}{c} Z(k+1) = z(k+1), \\ W(k) = w(k) \end{array} \middle| \begin{array}{c} Z(k) = z(k), Z(k-1) = z(k-1), \dots, Z(0) = z(0), \\ V(k) = v(k), V(k-1) = v(k-1), \dots, V(0) = v(0) \end{array} \right) \\ &= \Pr \left(\begin{array}{c} Z(k+1) = z(k+1), \\ W(k) = w(k) \end{array} \middle| \begin{array}{c} Z(k) = z(k), \\ V(k) = v(k) \end{array} \right) \end{aligned} \quad (3.28)$$

Die Markow-Eigenschaft des stochastischen E/A Prozesses besagt demnach, dass die bedingte Wahrscheinlichkeit für das gemeinsame Auftreten eines Folgezustands $z(k+1)$ und einer Ausgabe $w(k)$, ausschließlich von der vorangegangenen Eingabe $v(k)$ und vom vorherigen Zustand $z(k)$ abhängt. Setzt man nun wieder voraus, dass die bedingten Übergangswahrscheinlichkeiten (3.28) des stochastischen Prozesses zeitinvariant sind, kann das Verhalten des stochastischen E/A Prozesses durch eine homogene Markow-Kette mit Ein- und Ausgabe beschrieben werden.

Definition 3.4 (Homogene Markow-Kette mit Ein- und Ausgabe [113, S. 18]): *Ein diskreter stochastischer Prozess, bei dem für alle $\bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V}$ und $k \in \mathbb{N}$*

$$\Pr \left(\begin{array}{c} Z(k+1) = \bar{z}, \\ W(k) = w \end{array} \middle| \begin{array}{c} Z(k) = z, \\ V(k) = v \end{array} \right) = \Pr \left(\begin{array}{c} Z(1) = \bar{z}, \\ W(0) = w \end{array} \middle| \begin{array}{c} Z(0) = z, \\ V(0) = v \end{array} \right) \quad (3.29)$$

gilt, wird als homogene Markow-Kette mit Ein- und Ausgabe bezeichnet.

Für detaillierte Informationen zur Herleitung und den zugrundeliegenden Annahmen der Beziehung (3.29), siehe beispielsweise [11, S. 379], [12, S. 537–538], [83, S. 321–322]. Ein stochastischer Prozess, der die Bedingung (3.29) erfüllt, wird als *homogener Markow Prozess mit Ein- und Ausgabe* bezeichnet [11, S. 379]. Im Folgenden wird ein solcher Prozess kürzer als *stochastischer E/A Prozess* bezeichnet.

Wie in Abschnitt (3.2) beschrieben wurde, kann ein autonomer stochastischer Automat als Modell eines einfachen stochastischen Prozesses verwendet und zur Berechnung der Zustandswahrscheinlichkeiten eingesetzt werden. Im nächsten Abschnitt wird gezeigt, wie ein stochastischer E/A Prozess durch einen stochastischen Automaten beschrieben werden kann und welche Fragestellungen der Automat als Modell beantworten soll. Dabei wird neben der Voraussetzung, dass der zu beschreibende Prozess die Markow-Eigenschaft (3.29) besitzt, außerdem angenommen, dass die Eingabe zur Zeit k , unabhängig von den vorangegangenen Zuständen und Eingaben auftritt [11, S. 379–380].

3.4 Allgemeiner stochastischer Automat

Die stochastischen Automaten bilden die modelltheoretische Grundlage für alle in dieser Arbeit betrachteten Systeme. In diesem Abschnitt wird gezeigt, wie stochastische Automaten als Modell stochastischer E/A Prozesse verwendet werden können. Diese Einführung bildet die Basis für die in Kapitel 4 beschriebene Modellierung realer technischer Systeme durch stochastische Automaten. Allgemeine theoretische Grundlagen zu stochastischen Automaten sind in [13] gegeben. Die Modellierung stochastischer E/A Prozesse durch stochastische Automaten findet sich beispielsweise in [11], [86], [87], [113].

3.4.1 Definiton

Im Gegensatz zum autonomen stochastischen Automaten (3.11) besitzt der allgemeine *stochastische Automat* nicht nur Zustände, sondern auch Eingaben und Ausgaben. Der stochastische Automat wird dabei wieder unter Berücksichtigung initialer Zustandswahrscheinlichkeiten definiert und dementsprechend als *initialer stochastischer Automat* bezeichnet.

Definition 3.5 (Initialer stochastischer Automat [11, S. 380]): *Das 5-Tupel*

$$\mathcal{A} = (\mathcal{Z}, \mathcal{V}, \mathcal{W}, \mathcal{L}, \mathbf{p}_0) , \quad (3.30)$$

mit der Eingabemenge \mathcal{V} , der Ausgabemenge \mathcal{W} , der Zustandsmenge \mathcal{Z} sowie der Verhaltensrelation \mathcal{L} und dem initialen Wahrscheinlichkeitsvektor der Zustände \mathbf{p}_0 , heißt initialer stochastischer Automat.

Die Eingabemenge \mathcal{V} des stochastischen Automaten wird auch als *Eingabealphabet* und die Ausgabemenge \mathcal{W} als *Ausgabealphabet* bezeichnet [69, S. 42]. Der stochastische Automat kann damit wie der stochastische E/A Prozess die Eingaben

$$v \in \mathcal{V} := \{1, 2, \dots, M\} , \quad (3.31)$$

die Ausgaben

$$w \in \mathcal{W} := \{1, 2, \dots, Q\} \quad (3.32)$$

und die Zustände

$$z \in \mathcal{Z} := \{1, 2, \dots, N\} \quad (3.33)$$

annehmen. Die Eingabe-, Ausgabe- und Zustandsmenge des stochastischen Automaten entsprechen somit den bereits in (3.1), (3.24) und (3.25) definierten Mengen. Die Variable N gibt dabei die Anzahl der Automatenzustände an, während Q und M die Anzahl der Automatenausgaben und -eingaben definieren. Der Vektor $\mathbf{p}_0 \in [0, 1]^N$ beschreibt die initialen Zustandswahrscheinlichkeiten und entspricht der bereits in (3.21) definierten Verteilung. Die in (3.30) auftretende Funktion

$$\mathcal{L} : \mathcal{Z} \times \mathcal{W} \times \mathcal{Z} \times \mathcal{V} \longrightarrow [0, 1] \quad (3.34)$$

beschreibt eine bedingte Wahrscheinlichkeitsverteilung und wird als *Verhaltensrelation* des stochastischen Automaten bezeichnet [86, S. 367], [113, S. 13] (s. a. Definition 3.2).

Definition 3.6 (Verhaltensrelation des stochastischen Automaten [69, S. 42]): *Die Verhaltensrelation des stochastischen Automaten*

$$\mathcal{L}(\bar{z}, w, z, v) = \Pr \left(\begin{array}{c} Z(k+1) = \bar{z}, \\ W(k) = w \end{array} \middle| \begin{array}{c} Z(k) = z, \\ V(k) = v \end{array} \right) \quad (3.35)$$

beschreibt für alle $\bar{z}, z \in \mathcal{Z}$, $w \in \mathcal{W}$, $v \in \mathcal{V}$ und $k \in \mathbb{N}$ die bedingte Wahrscheinlichkeit dafür, dass wenn sich der Automat im Zustand z befindet und die Eingabe v erhält, er in den Folgezustand \bar{z} übergeht und die Ausgabe w erzeugt.

Da die Verhaltensrelation $\mathcal{L}(\bar{z}, w, z, v)$ eine bedingte Wahrscheinlichkeit repräsentiert, wird in der Regel die Notation $\mathcal{L}(\bar{z}, w | z, v)$ verwendet [87, S. 378]. Die Verhaltensrelation (3.35) mit den bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ enthält ausschließlich Werte zwischen Null und Eins, womit folgende Beziehung gilt [11, S. 380]:

$$\forall \bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V} : 0 \leq \mathcal{L}(\bar{z}, w | z, v) \leq 1 , \quad (3.36)$$

$$\forall z \in \mathcal{Z}, v \in \mathcal{V} : \sum_{\bar{z}=1}^N \sum_{w=1}^Q \mathcal{L}(\bar{z}, w | z, v) = 1 . \quad (3.37)$$

Der initiale stochastische Automat \mathcal{A} mit der Verhaltensrelation $\mathcal{L}(\bar{z}, w | z, v)$ sowie den endlichen Eingabe-, Ausgabe- und Zustandsmengen \mathcal{V}, \mathcal{W} und \mathcal{Z} entspricht damit einer diskreten homogenen Markow-Kette mit Ein- und Ausgabe nach Definition 3.4 [113, S. 18], [11, S. 380]. Im Folgenden wird der stochastische Automat \mathcal{A} als homogenes Markow-Modell eines zugrundeliegenden stochastischen E/A Prozesses verstanden.

3.4.2 Stochastische Automaten mit besonderen Eigenschaften

Die folgende Übersicht stellt vier besondere Fälle des stochastischen Automaten vor und ist den Arbeiten [69, S. 43] und [113, S. 16] entnommen:

- *Automat ohne Eingabe:* Ein initialer stochastischer Automat ohne Eingabe ist definiert durch das 4-Tupel

$$(\mathcal{Z}, \mathcal{W}, \mathcal{L}(\bar{z}, w | z), \mathbf{p}_0) . \quad (3.38)$$

- *Automat ohne Ausgabe:* Ein initialer stochastischer Automat ohne Ausgabe ist definiert durch das 4-Tupel

$$(\mathcal{Z}, \mathcal{V}, \mathcal{F}(\bar{z} | z, v), \mathbf{p}_0) . \quad (3.39)$$

- *Automat ohne Ein- und Ausgabe:* Ein initialer stochastischer Automat ohne Ein- und Ausgabe ist definiert durch das 3-Tupel

$$(\mathcal{Z}, \mathcal{F}(\bar{z} | z), \mathbf{p}_0) . \quad (3.40)$$

Die oben aufgeführten Sonderfälle des stochastischen Automaten sind insofern interessant, als dass sie unter bestimmten Voraussetzungen aus dem stochastischen Automaten \mathcal{A} in Gleichung (3.30) hervorgehen. Ein stochastischer Automat (3.30) dessen Eingabemenge nur ein einziges Element besitzt oder dessen Eingabe für alle Zeiten k konstant ist, entspricht einem Automaten ohne Eingabe [113, S. 16]. Dies gilt entsprechend auch für den Automaten ohne Ausgabe. Im Falle des Automaten ohne Ein- und Ausgabe (3.40) beschreibt die Übergangsrelation $\mathcal{F}(\bar{z} | z)$ nur die bedingte Wahrscheinlichkeit eines Zustandsübergangs von z nach \bar{z} , was der Definition 3.2 des autonomen stochastischen Automaten entspricht.

Ebenso wie der in Abschnitt 3.2 beschriebene autonome stochastische Automat, besitzt auch der allgemeine stochastische Automat (3.30) eine Matrixrepräsentation, die im Folgenden erläutert wird.

3.4.3 Matrixrepräsentation

Wie bereits für den autonomen stochastischen Automaten in Abschnitt 3.2 beschrieben wurde, ist es sinnvoll, dessen Übergangswahrscheinlichkeiten (3.35) in einer Matrix zusammenzufassen, da sich dadurch die Berechnung der gesuchten Zustandswahrscheinlichkeiten durch das Matrix-Vektor Produkt realisieren lässt. Für den allgemeinen stochastischen Automaten (3.30), der hier als Modell eines stochastischen E/A Prozesses dient, gilt dies ebenso. Die bedingten Übergangswahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ der Verhaltensrelation (3.35) können dabei in einer *Blockmatrix* oder einem *Tensor* (s. Abschnitt 7.1) angeordnet werden, die dann als Basis für die weiteren Betrachtungen und Berechnungen dienen. An dieser Stelle soll zunächst die einfach verständliche Matrixrepräsentation des stochastischen Automaten eingeführt werden, die aus [69, S. 44] übernommen wird.

Verhaltensmatrix

Die Verhaltensmatrix des stochastischen Automaten ist eine Blockmatrix, die für jede Kombination einer Automateneingabe v und Automatenausgabe w , eine quadratische Matrix enthält. Für jedes Paar (w, v) von Ausgaben und Eingaben (E/A Paar) wird dabei eine Matrix der Form

$$\mathbf{L}(w, v) = \begin{pmatrix} \mathcal{L}(1, w | 1, v) & \mathcal{L}(1, w | 2, v) & \cdots & \mathcal{L}(1, w | N, v) \\ \mathcal{L}(2, w | 1, v) & \mathcal{L}(2, w | 2, v) & \cdots & \mathcal{L}(2, w | N, v) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}(N, w | 1, v) & \mathcal{L}(N, w | 2, v) & \cdots & \mathcal{L}(N, w | N, v) \end{pmatrix} \in [0, 1]^{N \times N} \quad (3.41)$$

erstellt, welche die bedingten Wahrscheinlichkeit $\mathcal{L}(\bar{z}, w | z, v)$ der Verhaltensrelation (3.35), für das gegebene E/A Paar enthält. Da der Automat eine Anzahl von R Ausgaben und M Eingaben besitzt, erhält man eine Anzahl von $R \cdot M$ quadratischer Matrizen der Größe $N \times N$. Diese können dann für jede Kombination von Eingaben v und Ausgaben w zu einer großen Matrix, der sogenannten *Verhaltensmatrix* des stochastischen Automaten

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}(1, 1) & \mathbf{L}(1, 2) & \cdots & \mathbf{L}(1, M) \\ \mathbf{L}(2, 1) & \mathbf{L}(2, 2) & \cdots & \mathbf{L}(2, M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}(R, 1) & \mathbf{L}(R, 2) & \cdots & \mathbf{L}(R, M) \end{pmatrix} \in [0, 1]^{N \cdot R \times N \cdot M}, \quad (3.42)$$

zusammengefasst werden.

Wie aus Gleichung (3.41) und (3.42) hervorgeht, werden als Indizes der Matrixelemente die Automatenzustände und Folgezustände $\bar{z}, z \in \mathcal{Z}$, die Eingaben $v \in \mathcal{V}$ sowie die Ausgaben $w \in \mathcal{W}$ verwendet. Damit ist zu jeder Zeit klar, dass ein Matrixelement die bedingte Wahrscheinlichkeit $\mathcal{L}(\bar{z}, w | z, v)$, für ebendiese Kombination von Ein-, Ausgaben und Zuständen repräsentiert. Im Folgenden wird dabei insbesondere im Kontext der Verhaltensmatrix, entweder konkret von den Eingaben v , Ausgaben w , Zuständen z und Folgezuständen \bar{z} gesprochen, oder kürzer – aber gleichbedeutend – einfach von den *Indizes* \bar{z}, w, z und v . Da diese Indizes grundsätzlich nur Werte aus den entsprechenden Mengen $\mathcal{V}, \mathcal{W}, \mathcal{Z}$ annehmen können, wird, solange dies nicht notwendig ist, auf die explizite Angabe dieser Zugehörigkeit verzichtet und beispielsweise für die Eingabe einfach v anstatt $v \in \mathcal{V}$ geschrieben.

Für die Verhaltensmatrix gelten die Beziehungen (3.36) und (3.37). Das bedeutet zum einen, dass die Werte aller Matrixelemente innerhalb des Einheitsintervalls $[0, 1]$ liegen und dass wenn man alle Elemente entlang der Indizes w und \bar{z} aufsummiert, die Summe für alle Kombination von Indizes v und z , eins ergeben muss.

3.4.4 Modellierungsziel

Der stochastische Automat \mathcal{A} in Gleichung (3.30) bildet die modelltheoretische Grundlage zur Beschreibung aller realen Gebäudesysteme, die in dieser Arbeit betrachtet werden. Wie bereits erwähnt, wird der stochastische Automat als ein homogenes Markow-Modell verstanden, mit dem das Verhalten des zugrundeliegenden stochastischen E/A Prozesses repräsentiert werden soll. Im Allgemeinen werden dabei an den stochastischen Automaten zwei unterschiedliche Fragestellungen adressiert, die als *Simulation* und *Beobachtung* bezeichnet werden. Da der stochastische Automat selbst ein stochastischer Prozess ist, werden die Begriffe des stochastischen Automaten und des stochastischen E/A Prozesses in der Literatur oft gleichbedeutend verwendet [s. 83, S. 322]. Es wird daher sowohl explizit von der Beobachtung und Simulation des stochastischen Prozesses *durch* einen stochastischen Automaten, oder einfach von der Simulation und Beobachtung stochastischer Automaten gesprochen. Im den Folgenden Absätzen wird deutlich, wie die Begriffe der Simulation und Beobachtung interpretiert werden und was darunter im Detail verstanden wird.

Simulation

Die *Simulation* stochastischer Automaten beschreibt die Berechnung der Zustands- und Ausgabewahrscheinlichkeiten für gegebene Eingabewahrscheinlichkeiten

$$\Pr(V(k) = v), \quad v \in \mathcal{V}. \quad (3.43)$$

In der Praxis erfolgt diese Berechnung häufig unter der Randbedingung, dass die Eingaben $v \in \mathcal{V}$ zu jeder Zeit k exakt bekannt und durch eine Folge $V(0 \dots T)$ wie in (3.26) gegeben sind [86, S. 366]. Wahrscheinlichkeitstheoretisch bedeutet die exakte Kenntnis der Eingabe nichts anderes, als dass die

Zufallsvariable $V(k)$ *fast sicher*⁵ den Wert $v(k)$ annimmt, womit für die Wahrscheinlichkeit dieser Eingabe folgendes gilt:

$$\Pr(V(k) = v(k)) = 1 . \quad (3.44)$$

Die Wahrscheinlichkeitsverteilung der Eingaben (3.43) hat damit nur für die durch die Eingabefolge $V(0...T)$ zur Zeit k gegebene Eingabe $v(k) \in \mathcal{V}$ den Wert Eins und für alle anderen möglichen Eingabewerte den Wert Null [87, S. 370–371]. Als Simulation bezeichnet man dann die Berechnung der bedingten Zustandswahrscheinlichkeiten

$$\Pr(Z(k+1) = \bar{z} | V(0...k)), \quad \bar{z} \in \mathcal{Z}$$

und der bedingten Ausgabewahrscheinlichkeiten

$$\Pr(W(k) = w | V(0...k)), \quad w \in \mathcal{W}$$

für Zeiten $k = 0, 1, 2, \dots, T$ [113, S. 21–22]. Die bedingten Wahrscheinlichkeiten der Zustände zur Zeit $k+1$ hängen dabei von der bis zur Zeit k durchlaufenen Eingabefolge $V(0...k)$ ab. Gleiches gilt für die Ausgabewahrscheinlichkeiten, die ebenfalls unter der Bedingung einer gegebenen Eingabefolge bestimmt werden. Für die Simulation des stochastischen Automaten oder Prozesses muss dabei ausschließlich die Eingabefolge und die im stochastischen Automatenmodell (3.30) implizierte initiale Zustandswahrscheinlichkeitsverteilung bekannt sein. Die folgende Abbildung veranschaulicht, wie der stochastische Automat \mathcal{A} als Modell zur Simulation des Eingabe- und Ausgabeverhalten eines zugrundeliegenden stochastischen Prozesses interpretiert werden kann.

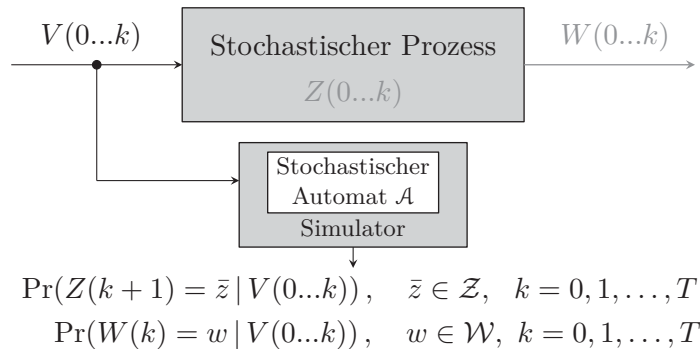


Abbildung 3.4: Simulation eines stochastischen E/A Prozesses durch einen stochastischen Automaten \mathcal{A}

Die Simulation stochastischer Automaten wird in dieser Arbeit nicht beschrieben. Die obige Beschreibung dient daher vor allem der Differenzierung zur im Folgenden erläuterten Beobachtung stochastischer Automaten, die die Grundlage zur Fehlerdetektion bildet.

Beobachtung

Unter der *Beobachtung* stochastischer Automaten versteht man die Bestimmung der Zustandswahrscheinlichkeiten, unter der Bedingung gegebener Eingabewahrscheinlichkeiten (3.43) und Ausgabewahrscheinlichkeiten

$$\Pr(W(k) = w), \quad w \in \mathcal{W} . \quad (3.45)$$

Die Berechnung der gesuchten Zustandswahrscheinlichkeiten erfolgt in der Regel nicht nur unter der Randbedingung exakt bekannter Eingaben (3.44), sondern man setzt auch für die Ausgaben voraus,

⁵In der Wahrscheinlichkeitstheorie werden Ereignisse, die mit einer Wahrscheinlichkeit von Eins eintreten, als *fast sicher* und Ereignisse, die eine Wahrscheinlichkeit von Null aufweisen, als *fast unmöglich* bezeichnet [115, S. 94].

dass diese durch eine eindeutige Folge $W(0...T)$ wie in (3.27) gegeben sind. Aus der exakten Kenntnis der Ausgabe zur Zeit k folgt dann wiederum, dass die Zufallsvariable $W(k)$ fast sicher den Wert $w(k)$ annimmt, womit

$$\Pr(W(k) = w(k)) = 1 \quad (3.46)$$

gilt. Auch für die Wahrscheinlichkeitsverteilung der Ausgaben (3.45) gilt damit, dass diese nur für die aus der Folge $W(0...T)$ zur Zeit k gegebene Ausgabe $w(k)$ den Wert Eins besitzt und für alle anderen möglichen Ausgaben den Wert Null. Ziel der Beobachtung ist dabei die Berechnung der bedingten Zustandswahrscheinlichkeiten

$$\Pr(Z(k) = z \mid V(0...k), W(0...k)), \quad z \in \mathcal{Z} \quad (3.47)$$

und

$$\Pr(Z(k+1) = \bar{z} \mid V(0...k), W(0...k)), \quad \bar{z} \in \mathcal{Z} \quad (3.48)$$

für alle Zeiten $k = 0, 1, 2, \dots, T$ [83, S. 333]. Im Gegensatz zur Simulation setzt die Beobachtung stochastischer Automaten oder Prozesse nicht nur die Kenntnis der Eingabefolge und der initialen Zustandswahrscheinlichkeiten voraus, sondern auch eine gegebene Ausgabefolge. Die Zustandswahrscheinlichkeiten zur Zeit k sind somit von der bis zu dieser Zeit durchlaufenen Ein- und Ausgabefolge abhängig. Die folgende Abbildung veranschaulicht das Prinzip der Beobachtung eines stochastischen Prozesses durch einen stochastischen Automaten \mathcal{A} . Beim Vergleich von Abbildung 3.5 und der qualitativen Modellvorstellung in Abbildung 1.7 wird bereits deutlich, wie der stochastische Automat zur Beobachtung realer Gebäudesysteme eingesetzt werden soll. An die Stelle des stochastischen Prozesses in Abbildung 3.5 tritt später ein reales System, dessen Verhalten mithilfe des stochastischen Automaten \mathcal{A} bewertet werden soll.

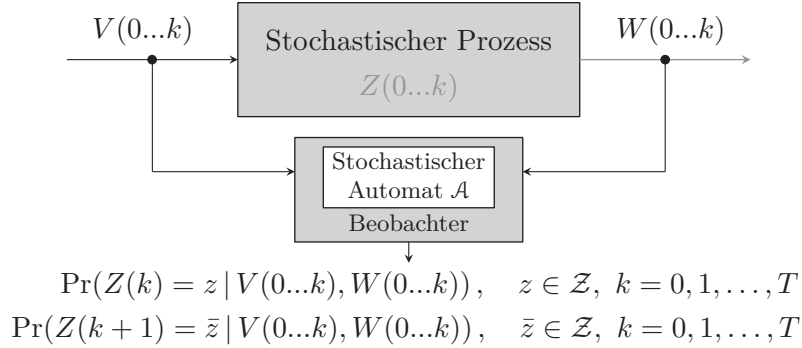


Abbildung 3.5: Beobachtung eines stochastischen E/A Prozesses durch einen stochastischen Automaten \mathcal{A}

Die Beobachtung stochastischer Automaten stellt eine klassische Methode der Prozessüberwachung dar, bei der der Zustand eines Prozesses anhand seines Ein- und Ausgabeverhaltens rekonstruiert werden soll. Die Zustandsbeobachtung bildet dabei auch die Grundlage für die in dieser Arbeit adressierte technische Fehlererkennung, die im nächsten Kapitel eingeführt wird. Die in diesem und auch im nächsten Kapitel vorausgesetzte Äquivalenz zwischen stochastischem Automaten und stochastischem Prozess ist hierbei insbesondere für die formale und wahrscheinlichkeitstheoretische Beschreibung der Zustandsbeobachtung und Fehlerdetektion notwendig. Der Grund dafür ist, dass der stochastische Automat \mathcal{A} ein Markow-Modell ist und damit die Zustandswahrscheinlichkeiten (3.47) und (3.48) eines zugrundeliegenden Prozesses nur dann exakt bestimmen kann, wenn der zu beschreibende Prozess ebenfalls die Markow-Eigenschaft besitzt. Das reale quantisierte System besitzt jedoch die Markow-Eigenschaft im Allgemeinen nicht und der stochastische Automat als Modell wird nur eine Näherung

der Zustandswahrscheinlichkeiten liefern und darüber hinaus sogar unzutreffende Lösungen (*engl.*: spurious solutions) generieren [11, S. 463, 468], [81, S. 279], [84, S. 33].

Wie die bei der Zustandsbeobachtung gesuchten Zustandswahrscheinlichkeiten allgemein und mithilfe der Matrixrepräsentation des stochastischen Automaten berechnet werden können, wird im nächsten Abschnitt erläutert. Das der Automat als Modell realer technischer Systeme, die in der Regel die Markow-Eigenschaft *nicht* besitzen, verwendet werden kann und welche Einschränkungen in diesem Fall gelten, wird in Kapitel 4 erläutert.

3.5 Beobachtung und Fehlererkennung

Die Zustandsbeobachtung stochastischer Automaten beschreibt die Berechnung der Zustandswahrscheinlichkeiten eines stochastischen Prozesses, unter der Bedingung einer gegebenen Ein- und Ausgabefolge (E/A Folge). Allgemein spricht man dabei auch vom sogenannten *aktuellen Zustandsbeobachtungsproblem*, welches folgendermaßen formuliert ist [113, S. 124]:

$$\left\| \begin{array}{l} \text{Gegeben: Initialer stochastischer Automat } \mathcal{A} = (\mathcal{Z}, \mathcal{V}, \mathcal{W}, \mathcal{L}, \mathbf{p}_0) \\ \text{Eingabefolge } V(0\dots T) \\ \text{Ausgabefolge } W(0\dots T) \\ \text{Gesucht: Zustandswahrscheinlichkeiten } \Pr(Z(k) = z | V(0\dots k), W(0\dots k)), z \in \mathcal{Z} \\ \text{für } k = 0, 1, 2, \dots, T \end{array} \right\|$$

Die Lösung des Zustandsbeobachtungsproblems erfolgt im Allgemeinen in zwei Schritten, die nachstehend erläutert werden.

3.5.1 Projektion und Prädiktion

Die gesuchten bedingten Zustandswahrscheinlichkeiten $\Pr(Z(k) = z | V(0\dots k), W(0\dots k))$, $z \in \mathcal{Z}$ repräsentieren die Wahrscheinlichkeit dafür, dass sich der Automat zur Zeit k im Zustand z befindet, unter der Bedingung, dass er bis zum Zeitpunkt k die Eingabefolge $V(0\dots k)$ und Ausgabefolge $W(0\dots k)$ durchlaufen hat. Der Gleichungen zur Lösung des Zustandsbeobachtungsproblems sind dabei wie folgt definiert [12, S. 581], [87, S. 513]:

$k \geq 0$:

$$\begin{aligned} \Pr(Z(k) = z | V(0\dots k), W(0\dots k)) & \\ &= \frac{\sum_{\bar{z}=1}^N \mathcal{L}(\bar{z}, w(k) | z, v(k)) \cdot \Pr(Z(k) = z | V(0\dots k-1), W(0\dots k-1))}{\sum_{z=1}^N \sum_{\bar{z}=1}^N \mathcal{L}(\bar{z}, w(k) | z, v(k)) \cdot \Pr(Z(k) = z | V(0\dots k-1), W(0\dots k-1))}, z \in \mathcal{Z} \end{aligned} \quad (3.49)$$

$k \geq 0$:

$$\begin{aligned} \Pr(Z(k+1) = \bar{z} | V(0\dots k), W(0\dots k)) & \\ &= \frac{\sum_{z=1}^N \mathcal{L}(\bar{z}, w(k) | z, v(k)) \cdot \Pr(Z(k) = z | V(0\dots k-1), W(0\dots k-1))}{\sum_{z=1}^N \sum_{\bar{z}=1}^N \mathcal{L}(\bar{z}, w(k) | z, v(k)) \cdot \Pr(Z(k) = z | V(0\dots k-1), W(0\dots k-1))}, \bar{z} \in \mathcal{Z} \end{aligned} \quad (3.50)$$

$k = 0$:

$$\Pr(Z(0) = z | v(-1), w(-1)) := (p_0)_z. \quad (3.51)$$

Eine ausführliche Herleitung obiger Gleichungen und deren Beweis findet sich in [83, S. 328–334] als auch in [113, S. 323–324].

In den Gleichungen (3.49) und (3.50) ist \mathcal{L} die Verhaltensrelation (3.35) und $(p_0)_z$ sind die initialen Zustandswahrscheinlichkeiten aus Gleichung (3.9). Für gegebene Ein- und Ausgabefolgen $V(0...T)$ und $W(0...T)$ ermöglichen die Beobachtergleichungen (3.49) und (3.50) die Berechnung der gesuchten Zustandswahrscheinlichkeiten für alle Zeiten $k = 0, 1, 2, \dots, T$. Setzt man beispielsweise einen Zeithorizont von $T = 1$ voraus, lässt sich diese Berechnung folgendermaßen beschreiben:

Zur Zeit $k = 0$ ist das E/A Paar $(v(0), w(0))$ gegeben und Gleichung (3.49) wird für diese E/A Kombination mit den Wahrscheinlichkeiten (3.51) initialisiert, wodurch man die aktuellen Zustandswahrscheinlichkeiten $\Pr(Z(0) = z | v(0), w(0)), z \in \mathcal{Z}$ erhält. Ebenso wird für $k = 0$ die Gleichung (3.51) in (3.50) eingesetzt und mit dem E/A Paar $(v(0), w(0))$ ergeben sich die Wahrscheinlichkeiten der Folgezustände $\Pr(Z(1) = \bar{z} | v(0), w(0)), \bar{z} \in \mathcal{Z}$. Dieser Berechnungsschritt wird auch als *Prädiktion* bezeichnet, da er für eine zur Zeit k gegebene E/A Kombination, eine Prognose der Zustandswahrscheinlichkeiten für die Zeit $k+1$ generiert [113, S. 130]. Für $k = 1$ und das E/A Paar $(v(1), w(1))$ setzt man die zum Zeitpunkt $k = 0$ mit Gleichung (3.50) berechneten Wahrscheinlichkeiten der Zustände wieder in Gleichung (3.49) ein und erhält mit $\Pr(Z(1) = z | V(0...1), W(0...1), z \in \mathcal{Z})$ die aktuellen Zustandswahrscheinlichkeiten. Diese Berechnung bezeichnet man als *Projektion*, die als eine Korrektur der zuvor prognostizierten Zustandswahrscheinlichkeiten interpretiert werden kann [11, S. 397]. Mit Gleichung (3.50) erhält man für $k = 1$ und die E/A Kombination $(v(1), w(1))$ wiederum eine Prognose der Zustandswahrscheinlichkeiten für die Zeit $k = 2$: $\Pr(Z(2) = \bar{z} | V(0...1), W(0...1), \bar{z} \in \mathcal{Z})$.

Es sei darauf hingewiesen, dass es sich bei der Notation der in (3.49) und (3.50) auftretenden bedingten Wahrscheinlichkeiten $\Pr(Z(k) = z | V(0...k-1), W(0...k-1))$ um eine verkürzte Schreibweise des Bedingungsteils handelt. Die vollständige Notation ist formal gegeben durch:

$$\begin{aligned} \Pr(Z(k) = z | V(0...k-1), W(0...k-1)) \\ := \Pr\left(Z(k) = z \left| \begin{array}{l} V(0)=v(0), V(1)=v(1), \dots, V(k-1)=v(k-1), \\ W(0)=w(0), W(1)=w(1), \dots, W(k-1)=w(k-1) \end{array} \right. \right). \end{aligned} \quad (3.52)$$

Gleiches gilt entsprechend für die Notation der anderen bedingten Wahrscheinlichkeiten in den Gleichungen (3.49) und (3.50). Weiterführende und ausführlichere Informationen zur Beobachtung stochastischer Automaten können der folgenden Literatur entnommen werden: [11], [12], [83], [87], [113].

Inkonsistente Ein- und Ausgabefolgen

Eine E/A Folge $(V(0...T), W(0...T))$ heißt *inkonsistent* mit dem Modell des stochastischen Automaten, wenn der in den Beobachtergleichungen (3.49) und (3.50) auftretende Nenner für irgendein E/A Paar $(v(k), (w(k)))$ dieser Folge Null wird [87, S. 513–514]. Führen in den Beobachtergleichungen hingegen alle E/A Paare $(v(k), (w(k)))$ einer gegebenen E/A Folge $(V(0...T), W(0...T))$ auf einen Nenner ungleich null, dann ist die E/A Folge *konsistent* mit dem Modell des stochastischen Automaten [87, S. 513–514]. Die Zustandswahrscheinlichkeiten (3.49) und (3.50) können somit nur für konsistente E/A Paare bestimmt werden, während für inkonsistente E/A Paare keine Zustände existieren, die der Automat annehmen kann.

Die Konsistenzbewertung der E/A Folgen bildet die Grundlage für die in dieser Arbeit adressierte Fehlerdetektion und wird in den folgenden Abschnitten weiter erläutert.⁶ Da die Beobachtergleichungen für die Projektion (3.49) und Prädiktion (3.50) den gleichen Nenner besitzen, ist es für die konsistenzbasierte Fehlererkennung ausreichend, lediglich den Prädiktionsschritt zu betrachten. Das Interesse gilt im Folgenden also weniger der Lösung des aktuellen Zustandsbeobachtungsproblems durch die Projektion, sondern vielmehr der Bereitstellung eines effizienten Algorithmus für die Fehlerdetektion. Im nächsten Abschnitt wird dabei zunächst gezeigt, wie sich die Beobachtergleichung für

⁶Eine Behandlung der in diesem Zusammenhang adressierten Begriffe der Beobachtbarkeit und Diagnostizierbarkeit stochastischer Automaten findet hier nicht statt. Siehe hierzu beispielsweise [11], [83], [84], [113].

die Prädiktion (3.50) in einer kompakten Form, basierend auf der *Matrixrepräsentation stochastischer Automaten* (s. Abschnitt 3.4.3), effizient realisieren lässt. Anschließend wird in Abschnitt 3.5.3 die konsistenzbasierte Fehlerdetektion eingeführt.

3.5.2 Beobachtergleichung

Auf Basis der Verhaltensmatrix $\mathbf{L} \in \mathbb{R}^{N \cdot Q \times N \cdot M}$ aus Gleichung (3.42) lässt sich nun die Beobachtergleichung für stochastische Automaten in Matrixrepräsentation formulieren. Sind die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ der Verhaltensrelation (3.35) nach Gleichung (3.41) in der Verhaltensmatrix (3.42) angeordnet, dann kann der Zähler der Beobachtergleichung für die Prädiktion (3.50) in Form des Matrix-Vektor Produkts

$$\mathbf{L}(w(k), v(k)) \cdot \mathbf{p}_z(k) \in [0, 1]^N \quad (3.53)$$

angegeben werden. Darin repräsentiert

$$\mathbf{p}_z(k) = \begin{pmatrix} p_z(k, 1) \\ p_z(k, 2) \\ \vdots \\ p_z(k, N) \end{pmatrix} := \begin{pmatrix} \Pr(Z(k) = 1 | V(0 \dots k-1), W(0 \dots k-1)) \\ \Pr(Z(k) = 2 | V(0 \dots k-1), W(0 \dots k-1)) \\ \vdots \\ \Pr(Z(k) = N | V(0 \dots k-1), W(0 \dots k-1)) \end{pmatrix} \in [0, 1]^N \quad (3.54)$$

den Vektor der diskreten Wahrscheinlichkeitsverteilung der Zustände zur Zeit k und

$$\begin{aligned} & \mathbf{L}(w(k), v(k)) \\ &= \begin{pmatrix} l(1, w(k), 1, v(k)) & l(1, w(k), 2, v(k)) & \dots & l(1, w(k), N, v(k)) \\ l(2, w(k), 1, v(k)) & l(2, w(k), 2, v(k)) & \dots & l(2, w(k), N, v(k)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N, w(k), 1, v(k)) & l(N, w(k), 2, v(k)) & \dots & l(N, w(k), N, v(k)) \end{pmatrix} \in [0, 1]^{N \times N}, \end{aligned} \quad (3.55)$$

ist die Matrix (3.41), die für das zur Zeit k gegebene E/A Paar $(w(k), v(k))$ ausgewählt wurde. Es kann somit in jedem Zeitschritt k und für alle möglichen Kombinationen von Automateingaben $v(k)$ und -ausgaben $w(k)$ eine Matrix $\mathbf{L}(w(k), v(k))$ aus der Block-Verhaltensmatrix (3.42) selektiert werden, deren Elemente die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ für ebendiese E/A Kombination repräsentieren. Da für die Verhaltensmatrix die Beziehungen (3.36) und (3.37) gelten, sind die einzelnen Blöcke $\mathbf{L}(w(k), v(k))$ in der Regel *sub-stochastisch*, was bedeutet, dass ihre Spaltensumme kleiner oder gleich Eins ist:

$$\forall z \in \mathcal{Z} : \sum_{\bar{z}=1}^N 0 \leq \mathcal{L}(\bar{z}, w(k), z, v(k)) \leq 1. \quad (3.56)$$

Wie man anhand des Nenners der Gleichung (3.50) erkennen kann, weist dieser die gleiche Struktur wie der Zähler auf, mit dem Unterschied, dass zusätzlich über alle Folgezustände $\bar{z} \in \mathcal{Z}$ summiert wird. Der Nenner stellt somit eine Normierung der mit dem Zähler berechneten Zustandswahrscheinlichkeiten dar. Diese Normierung muss dementsprechend auch für die Matrix-Vektor Darstellung der Beobachtergleichung berücksichtigt werden. Anhand von Gleichung (3.56) wird die Notwendigkeit dieses Schrittes deutlich, denn die Multiplikation einer substochastischen Matrix mit einem Wahrscheinlichkeitsvektor führt generell zu einem ebenfalls substochastischen Vektor. Das bedeutet, dass die Komponenten des mit (3.53) berechneten Produktvektors für $k \rightarrow \infty$ gegen null tendieren und letztlich numerisch nicht mehr berechenbar sind. Für die Matrix-Vektor Notation des Nenners von

Gleichung (3.50) bedeutet dies, dass alle Komponenten des aus dem Matrix-Vektor Produkt resultierenden Ergebnisvektors aufsummiert werden müssen. Da hier mit den bedingten Wahrscheinlichkeiten über ausschließlich nichtnegative Werte summiert wird, kann der Nenner der Beobachtergleichung (3.50) in Form der *Summennorm*⁷ notiert werden:

$$\|\mathbf{L}(w(k), v(k)) \cdot \mathbf{p}_z(k)\|_1 \in [0, 1] . \quad (3.57)$$

Mit den Gleichungen (3.53) und (3.57) lassen sich damit sowohl Zähler als auch Nenner der Beobachtergleichung (3.50) in eine kompakte Matrix-Vektor Notation überführen:

Definition 3.7 (Matrix-Beobachtergleichung): *Sind die E/A Folgen $(V(0...T), W(0...T))$ eines stochastischen Prozesses konsistent mit dem stochastischen Automaten \mathcal{A} , dann können die diskreten Wahrscheinlichkeitsverteilungen der Zustände mit*

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\mathbf{L}(w(k), v(k)) \cdot \mathbf{p}_z(k)}{\|\mathbf{L}(w(k), v(k)) \cdot \mathbf{p}_z(k)\|_1} \in [0, 1]^N , \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0 \end{aligned} \quad (3.58)$$

für alle diskreten Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden.

Es ist also offensichtlich, dass wenn man den Zähler und Nenner der Beobachtergleichung (3.58) in die elementweise Notation des Matrix-Vektor Produkts überführt, diese Gleichung exakt der Beobachtergleichung für die Prädiktion (3.50) entspricht. Die Beobachtergleichung (3.58) liefert mit dem Vektor $\mathbf{p}_z(k+1)$ eine Prognose der Zustandswahrscheinlichkeitsverteilung

$$\mathbf{p}_z(k+1) = \begin{pmatrix} p_z(k+1, 1) \\ p_z(k+1, 2) \\ \vdots \\ p_z(k+1, N) \end{pmatrix} := \begin{pmatrix} \Pr(Z(k+1)=1 \mid V(0...k), W(0...k)) \\ \Pr(Z(k+1)=2 \mid V(0...k), W(0...k)) \\ \vdots \\ \Pr(Z(k+1)=N \mid V(0...k), W(0...k)) \end{pmatrix} \in [0, 1]^N, \quad (3.59)$$

für den Zeitpunkt $k+1$. Der Vektor (3.59) stellt eine bedingte Wahrscheinlichkeitsverteilung dar, denn er repräsentiert die Zustandswahrscheinlichkeiten zur Zeit $k+1$, unter der Bedingung, dass bis zum Zeitpunkt k die Eingabefolge $V(0...k)$ und die Ausgabefolge $W(0...k)$ durchlaufen wurde. Initialisiert mit der Startverteilung \mathbf{p}_0 aus Gleichung (3.21), erhält man durch rekursive Anwendung der Beobachtergleichung (3.58) die Zustandswahrscheinlichkeitsvektoren für $k = 0, 1, 2, \dots, T$. Diese lassen sich dann in einer Vektorfolge zusammenfassen:

$$\mathbf{P}_z(0...T+1 \mid V(0...T), W(0...T)) = (\mathbf{p}_z(0), \mathbf{p}_z(1), \mathbf{p}_z(2), \dots, \mathbf{p}_z(T+1)) . \quad (3.60)$$

3.5.3 Fehlererkennung

Die Fehlererkennung mit stochastischen Automaten wurde bereits in früheren Arbeiten wie beispielsweise in [11], [69], [74], [83], [113] beschrieben, wobei in dieser Literatur neben der Fehlererkennung auch die Fehleridentifikation adressiert ist. Die Fehleridentifikation erfordert neben einem Nominalmodell des zu untersuchenden Prozesses auch Fehlermodelle, die das zu identifizierende anormale Verhalten des Prozesses beschreiben. Auch wenn in dieser Arbeit die Fehlererkennung im Mittelpunkt steht, soll darauf hingewiesen werden, dass die hier vorgestellten Algorithmen ebenso zur Fehleridentifikation verwendet werden können – vorausgesetzt es existieren entsprechende Fehlermodelle des zugrundeliegenden Prozesses. Die genaue Vorgehensweise zur Identifikation von Fehlern kann der oben aufgeführten Literatur entnommen werden.

⁷Die Summennorm oder 1-Norm eines Vektors $\mathbf{y} \in \mathbb{R}^J$ ist definiert als $\|\mathbf{y}\|_1 := \sum_{j=1}^J |y_j|$.

Konsistenzbewertung der Ein- und Ausgabefolgen

Wie bereits in Abschnitt 3.5.1 erwähnt wurde, basiert die Fehlererkennung auf der Bewertung der E/A Folgen hinsichtlich ihrer Konsistenz mit dem stochastischen Automaten. Eine wichtige Rolle hierbei spielen die Wahrscheinlichkeiten, mit denen der Automat eine E/A Folge durchlaufen kann. Wie diese berechnet werden, wird anhand der Beobachtergleichung (3.58) beschrieben.

Wahrscheinlichkeit der Ein- und Ausgabefolgen: Neben den Verteilungen der Zustandswahrscheinlichkeiten (3.60), liefert die Beobachtergleichung (3.58) auch die Wahrscheinlichkeit einer gegebenen E/A Folge, die mit

$$\Pr(W(0...T) \mid V(0...T)) . \quad (3.61)$$

bezeichnet wird [11, S. 386]. Dabei handelt es sich um die bedingte Wahrscheinlichkeit, dass der Automat die Ausgabefolge $W(0...T)$ erzeugt, wenn er die Eingabefolge $V(0...T)$ erhält. Die Berechnung dieser Wahrscheinlichkeit erfolgt anhand des Nenners der Beobachtergleichung (3.58), der für die weitere Betrachtung wie folgt abgekürzt wird:

$$b(k) := \|\mathbf{L}(w(k), v(k)) \cdot \mathbf{p}_z(k)\|_1 \in [0, 1] . \quad (3.62)$$

Die Beobachtergleichung (3.58) generiert damit für eine konsistente E/A Folge die Werte $b(k)$ für Zeiten $k = 0, 1, 2, \dots, T$, was zu einer Folge von Skalaren führt:

$$B(0...T \mid V(0...T), W(0...T)) = (b(0), b(1), b(2), \dots, b(T)) . \quad (3.63)$$

Diese Folge wird im Folgenden als *Verhaltensfolge* oder *Verhaltenssignal* bezeichnet. Die gesuchte Wahrscheinlichkeit einer E/A Folge (3.61) ist dann gegeben durch

$$\Pr(W(0...T) \mid V(0...T)) = \prod_{k=0}^T b(k) . \quad (3.64)$$

Die oben beschriebene Vorgehensweise zur Bestimmung der Wahrscheinlichkeit $\Pr(W(0...T) \mid V(0...T))$ ist eine direkte Schlussfolgerung aus den in [113, S. 20–22] beschriebenen Berechnungen. Siehe hierzu auch [92, S. 418]. Die Wahrscheinlichkeit (3.64) bildet die Basis zur Beschreibung des sogenannten *Verhaltens* des stochastischen Automaten.

Verhalten stochastischer Automaten: Auch wenn das Verhalten stochastischer Automaten prinzipiell zu den Grundlagen der Automatentheorie zählt, macht eine Einführung an dieser Stelle Sinn, da sich dieser Begriff am einfachsten anhand der Wahrscheinlichkeit $\Pr(W(0...T) \mid V(0...T))$ definieren lässt.

Definition 3.8 (Verhalten stochastischer Automaten [86, S. 373–374]): *Die Menge aller E/A Folgen $(V(0...T), W(0...T))$, die der stochastische Automat mit von Null verschiedener Wahrscheinlichkeit durchlaufen kann, wird als Verhalten*

$$\mathcal{B}^T = \{(V(0...T), W(0...T)) : \Pr(W(0...T) \mid V(0...T)) > 0\} \subseteq \mathcal{V}^{T+1} \times \mathcal{W}^{T+1} \quad (3.65)$$

des stochastischen Automaten bezeichnet.

Ob eine E/A Folge konsistent mit dem stochastischen Automaten \mathcal{A} ist und damit zum Verhalten \mathcal{B}^T gehört, hängt damit allgemein von der Wahrscheinlichkeit (3.64) ab, die jedoch ihrerseits wiederum von der initialen Verteilung der Zustandswahrscheinlichkeiten (3.21) des Automaten abhängt. So kann es beispielsweise sein, dass der stochastische Automat für eine bestimmte Eingabefolge

und initiale Zustandswahrscheinlichkeiten eine bestimmte Ausgabefolge mit einer Wahrscheinlichkeit $\Pr(W(0...T)|V(0...T)) > 0$ erzeugt, jedoch die gleiche E/A Folge mit einer anderen Startverteilung zu einer Wahrscheinlichkeit $\Pr(W(0...T)|V(0...T)) = 0$ führt. Das Verhalten \mathcal{B}^T kann somit entweder für eine bestimmte initiale Zustandsverteilung des stochastischen Automaten berechnet werden oder aber für alle möglichen Startwahrscheinlichkeiten [86, S. 373–374]. Im folgenden gilt das Interesse vor allem der Bewertung der E/A Folgen hinsichtlich ihrer Zugehörigkeit zur Menge \mathcal{B}^T . Da sich diese Zugehörigkeit aus der Wahrscheinlichkeit $\Pr(W(0...T)|V(0...T))$, beziehungsweise sogar direkt aus der Variablen $b(k)$ in (3.62) ergibt, ist es nicht notwendig und komplexitätstechnisch auch nicht wünschenswert, die Menge \mathcal{B}^T explizit zu berechnen. Vielmehr soll im folgenden geprüft werden, ob der initiale stochastische Automat \mathcal{A} , gegebene E/A Folgen mit von null verschiedener Wahrscheinlichkeit durchlaufen kann. Diese *Konsistenzprüfung* der E/A Folgen wird im folgenden Abschnitt erläutert. Weitere ausführliche Erläuterungen zum Verhalten stochastischer Automaten finden sich insbesondere in [11], [69].

Konsistenzprüfung der Ein- und Ausgabefolgen: Davon ausgehend, dass der stochastische Automat \mathcal{A} das nominale Verhalten des stochastischen Prozesses vollständig beschreibt, kann die Beobachtergleichung (3.58) für eine konsistenzbasierte Fehlererkennung verwendet werden. Diese Art der Fehlererkennung basiert auf der Analyse der gegebenen E/A Folgen $(V(0...T), W(0...T))$ des stochastischen E/A Prozesses hinsichtlich ihrer Konsistenz mit dem Modell des stochastischen Automaten \mathcal{A} [11, S. 387]. Entscheidend für diese Bewertung ist die Analyse des Nenners der Beobachtergleichung (3.58), der durch die Variable $b(k)$ aus (3.62) beschrieben wird (vgl. Abschnitt 3.5.1). Die Variable $b(k)$, beziehungsweise die Folge $B(0...T|V(0...T), W(0...T))$, bildet somit die Grundlage der Konsistenzbewertung der E/A Folgen und damit auch die Basis für die Fehlererkennung.

Eine E/A Folge $(V(0...T), W(0...T))$ mit den E/A Paaren $(v(k), w(k))$ ist *konsistent* mit dem stochastischen Automaten \mathcal{A} , wenn für die Elemente $b(k)$ der Verhaltensfolge B folgendes gilt:

$$\forall k : \quad b(k) > 0. \quad (3.66)$$

Aus der Beobachtergleichung (3.58) erhält man dann die Folge der Zustandswahrscheinlichkeitsverteilungen (3.60) sowie die Verhaltensfolge (3.63) und es gilt für die Wahrscheinlichkeit der E/A Folge $\Pr(W(0...T) | V(0...T)) > 0$. Die E/A Folge gehört damit zum Verhalten \mathcal{B}^T des stochastischen Automaten [11, S. 387]:

$$(V(0...T), W(0...T)) \in \mathcal{B}^T. \quad (3.67)$$

Eine E/A Folge $(V(0...T), W(0...T))$ mit E/A Paaren $(v(k), w(k))$ ist *inkonsistent* mit dem stochastischen Automaten \mathcal{A} , wenn für die Elemente $b(k)$ der Verhaltensfolge B folgendes gilt:

$$\exists k : \quad b(k) = 0. \quad (3.68)$$

In diesem Fall ist die Wahrscheinlichkeit der E/A Folge Null ($\Pr(W(0...T) | V(0...T)) = 0$) und diese gehört *nicht* zum Verhalten \mathcal{B}^T des stochastischen Automaten:

$$(V(0...T), W(0...T)) \notin \mathcal{B}^T. \quad (3.69)$$

Ein inkonsistentes E/A Paar, für das $b(k) = 0$ gilt, führt im Nenner der Beobachtergleichung (3.58) zu einer Division durch null, weshalb die Berechnung an diesem Punkt abgebrochen werden muss.

In der Praxis können unterschiedliche Gründe dazu führen, dass eine gemessene E/A Folge inkonsistent mit dem Modell des stochastischen Automaten ist. Die Ursachen reichen dabei von fehlerhaften Messungen der Ein- und Ausgaben, über äußere Störgrößen die auf den Prozess einwirken, hin zu Fehlern innerhalb des Prozesses selbst. Im folgenden Kapitel wird gezeigt, wie stochastische Automaten als qualitatives Modell realer Systeme verwendet werden können und wie sich die Fehlererkennung mit deren Hilfe realisieren lässt.

4

Qualitatives Modell

In diesem Kapitel werden die qualitativen Modelle eingeführt. Zunächst erfolgt in Abschnitt 4.1 eine kurze Einleitung bezüglich der historischen Hintergründe der qualitativen Modelle. Abschnitt 4.2 beschreibt die Grundlagen quantisierter Systeme, deren Verhalten anhand qualitativer Größen beschrieben wird. Außerdem wird erläutert, wie die aus den Gebäudeautomationssystemen verfügbaren Zeitreihen durch Partitionierung ihrer Signalmräume auf ihre qualitativen Werte abgebildet werden. Die formale Einführung der qualitativen Modelle und mit welchen Methoden diese gebildet werden können, wird in Abschnitt 4.3 gezeigt. Abschließend erfolgt im Abschnitt 4.4 die Einführung in die qualitative Beobachtung und die Definition von Algorithmen zur qualitativen Fehlererkennung.

4.1 Einleitung

Qualitative Modelle sind nichtdeterministische Modelle, die das Verhalten dynamischer Systeme anhand quantisierter Systemgrößen beschreiben und eine näherungsweise Repräsentation des Systemverhaltens bieten. Ziel der qualitativen Modellierung technischer Prozesse ist dabei eine zwar grobe, aber möglichst vollständige Beschreibung der Systemdynamik. Die Grundlage eines qualitativen Modells bildet ein stochastischer Automat, mit dem das nichtdeterministische qualitative Verhalten des zugrundeliegenden quantisierten Systems beschrieben wird. Die Entwicklung dieser Art von qualitativen Modellen geht zurück in die 1990er Jahre und wurde maßgeblich geprägt durch die Arbeiten von Lunze [76], [77], [78], [79], [80], [81]. Qualitative Modelle können dabei zur Lösung verschiedener Problemstellungen, die von der Simulation und Beobachtung über die Fehlererkennung und Diagnose bis hin zur Steuerung und Regelung technischer Systeme reichen, eingesetzt werden. In dieser Arbeit steht die *qualitative Beobachtung* im Mittelpunkt, die die Grundlage zur technischen Fehlererkennung bildet. Fundamentale Erkenntnisse auf diesem Gebiet wurden dabei insbesondere durch Lichtenberg et al. [68], Lichtenberg [69], Lichtenberg und Lunze [71], Lichtenberg und Steele [74], Lunze und Schröder [83], Lunze [84], Neidig [95] und Schröder [113] erzielt.

4.2 Quantisierung der Messgrößen

4.2.1 Quantisiertes System

Ein *quantisiertes System* ist ein dynamisches System in diskreter Zeit, dessen Eingänge, Ausgänge und Zustände nicht durch quantitative Größen beschrieben werden, sondern durch die quantisierten Werte der Systemgrößen. Die Abbildung 4.1 greift dabei die Vorstellung des schon in der Einleitung eingeführten Begriffs des quantisierten System auf.

Betrachtet man den in der Mitte von Abbildung 4.1 dargestellten zeitdiskreten Prozess über einen beliebigen Zeithorizont T , dann durchläuft dieser für eine gegebene quantitative Eingangsfolge

$$U(0...T) = (\mathbf{u}(0), \mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(T)) , \quad (4.1)$$

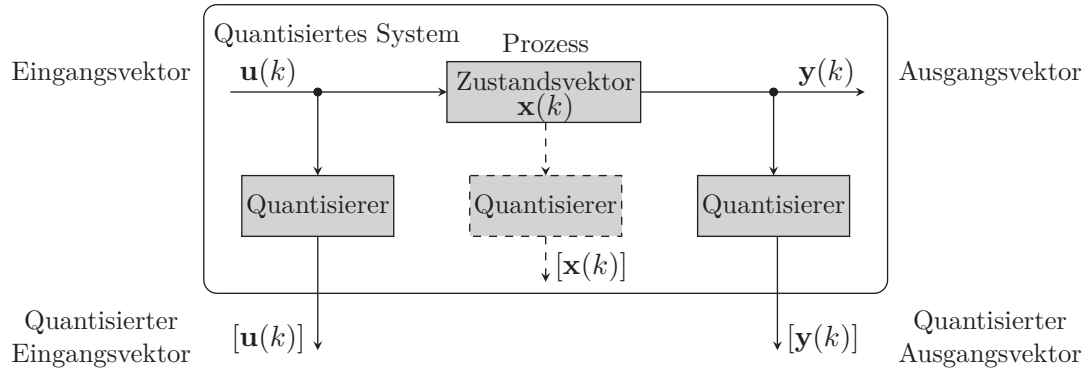


Abbildung 4.1: Quantisiertes System aus Sicht eines Beobachters, nach [113, S. 170]

eine quantitative Zustandsfolge

$$\mathbf{X}(0...T) = (\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)) \quad (4.2)$$

und generiert eine quantitative Ausgangsfolge

$$\mathbf{Y}(0...T) = (\mathbf{y}(0), \mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(T)) . \quad (4.3)$$

Diese Folgen stellen klassische Zeitreihen dar, die wie in Abschnitt 1.3 beschrieben, zu Vektoren zusammengefasst sind. Im Folgenden wird vorausgesetzt, dass diese quantitativen Folgen durch Messungen aus den Gebäudeautomationssystemen verfügbar sind und entsprechend zur qualitativen Modellbildung genutzt werden können. Da das qualitative Modell, beziehungsweise der diesem zugrundeliegende stochastische Automat, nur diskrete Größen verarbeiten kann, müssen die Folgen (4.1), (4.2) und (4.3) zunächst quantisiert werden. Das Interesse gilt im Folgenden somit den quantisierten Eingangsfolgen

$$[\mathbf{U}(0...T)] = ([\mathbf{u}(0)], [\mathbf{u}(1)], [\mathbf{u}(2)], \dots, [\mathbf{u}(T)]) , \quad (4.4)$$

Zustandsfolgen

$$[\mathbf{X}(0...T)] = ([\mathbf{x}(0)], [\mathbf{x}(1)], [\mathbf{x}(2)], \dots, [\mathbf{x}(T)]) \quad (4.5)$$

und Ausgangsfolgen

$$[\mathbf{Y}(0...T)] = ([\mathbf{y}(0)], [\mathbf{y}(1)], [\mathbf{y}(2)], \dots, [\mathbf{y}(T)]) . \quad (4.6)$$

Ein quantisiertes System wird somit ausschließlich durch die wert- und zeitdiskreten quantisierten Größen beschrieben und repräsentiert damit das *qualitative Verhalten* des zugrundeliegenden Prozesses [69, S. 4], welches durch ein qualitatives Modell beschrieben werden kann. Während die quantitativen Zustands- und Ausgangsfolgen $\mathbf{X}(0...T)$ und $\mathbf{Y}(0...T)$ des zeit- und wertkontinuierlichen Prozesses in Abbildung 4.1 für einen bestimmten initialen Zustand und eine gegebene Eingangsfolge $\mathbf{U}(0...T)$ eindeutig bestimmt sind, kann das quantisierte System für eine Folge quantisierter Eingangswerte $[\mathbf{U}(0...T)]$ und einen quantisierten Startzustand im Allgemeinen verschiedene quantisierte Zustandsfolgen $[\mathbf{X}(0...T)]$ und Ausgangsfolgen $[\mathbf{Y}(0...T)]$ durchlaufen [69, S. 11]. Das qualitative Verhalten des quantisierten Systems ist damit *nichtdeterministisch* [11, S. 458], [69, S. 11], [113, S. 40–41]. Zum Verhaltensbegriff des quantisierten Systems siehe auch Abschnitt 4.3.2. Bevor nun ein qualitatives Modell definiert werden kann, muss zuvor jedoch noch eine grundlegende Festlegung bezüglich der Quantisierung getroffen werden.

Das Prinzip der Quantisierung eines zeitdiskreten und wertkontinuierlichen Signals basiert auf der Partitionierung seines Wertebereichs und wurde bereits rechts in Abbildung 1.1 für ein einzelnes Signal dargestellt. Da die meisten realen Prozesse jedoch Mehrgrößensysteme sind und beispielsweise der Ausgangsvektor $\mathbf{y}(k) \in \mathbb{R}^q$ eine Anzahl von q Ausgangssignalen repräsentiert, muss eine Vorschrift definiert werden, die es erlaubt, alle reellen Signalwerte des Vektors auf einen einzelnen quantisierten Wert $[\mathbf{y}(k)]$ abzubilden. Die Quantisierung der Eingangs-, Ausgangs- und Zustandsvektoren erfolgt dabei durch eine *Partitionierung* des Eingangs-, Ausgangs- und Zustandsraumes und wird im nächsten Abschnitt erläutert.

Anmerkung 4.1 (Quantisierte Zustandsfolge): Zur *qualitativen Zustandsfolge* sei angemerkt, dass diese wieder eine Kurznotation der expliziten Schreibweise

$$[\mathbf{X}(0 \dots T)] = ([\mathbf{x}(0)] = z(0), [\mathbf{x}(1)] = z(1), [\mathbf{x}(2)] = z(2), \dots, [\mathbf{x}(T)] = z(T))$$

darstellt, wobei gleiches für die qualitativen Eingangs- und Ausgangsfolgen gilt.

4.2.2 Allgemeine Quantisierung

Die Beschreibung der Signalquantisierung erfolgt beispielhaft anhand des quantitativen Zustandsvektors $\mathbf{x} \in \mathbb{R}^n$ und gilt ebenso für die reellwertigen Eingänge $\mathbf{u} \in \mathbb{R}^m$ und Ausgänge $\mathbf{y} \in \mathbb{R}^q$.

Als *allgemeine Signalquantisierung* wird in dieser Arbeit eine Methode bezeichnet, die einen reellwertigen Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$ durch Partitionierung des *mehrdimensionalen Zustandsraumes* \mathbb{R}^n auf einen ganzzahligen quantisierten Wert $[\mathbf{x}]$ abbildet. Die Zustandsquantisierung kann dabei als eine Funktion

$$E_{\mathbf{x}} : \mathbb{R}^n \rightarrow \mathcal{X} \quad (4.7)$$

interpretiert werden, die den quantitativen n -dimensionalen Zustandsraum \mathbb{R}^n auf den sogenannten *qualitativen Zustandsraum* $\mathcal{X} := \{1, 2, \dots, N\}$ abbildet [72, S. 189], [70, S. 150]. Dies kann vereinfacht wie folgt interpretiert werden:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \xrightarrow[\text{Quantisierung}]{\text{Allgemeine}} [\mathbf{x}] \in \mathcal{X} . \quad (4.8)$$

Die Funktion (4.7) lässt dabei offen, nach welchem Schema der quantitative Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$ auf den quantisierten Wert $[\mathbf{x}] \in \mathcal{X}$ abgebildet wird. Diese Festlegung erfolgt durch die Vorgabe einer Partitionierung, die den mehrdimensionalen Zustandsraum \mathbb{R}^n in eine endliche Anzahl beschränkter und disjunkter Mengen $\mathcal{E}_x(z)$, $z \in \mathcal{X}$ unterteilt, für die folgendes gilt [82, S. 397]:

$$\bigcup_{z \in \mathcal{X}} \mathcal{E}_x(z) = \mathbb{R}^n , \quad (4.9)$$

$$\forall i, j \in \mathcal{X} : \mathcal{E}_x(i) \cap \mathcal{E}_x(j) = \emptyset \quad \text{für } i \neq j . \quad (4.10)$$

Gleichung (4.9) sagt dabei, dass die *Vereinigung* aller Mengen $\mathcal{E}_x(z)$, $z \in \mathcal{X}$, der Menge des quantitativen Zustandsraums \mathbb{R}^n entspricht. Gleichung (4.10) drückt die Forderung nach der Disjunktheit der Mengen untereinander aus. Das bedeutet, dass beispielsweise die *Schnittmenge* $\mathcal{E}_x(1) \cap \mathcal{E}_x(2)$ durch die leere Menge \emptyset gegeben ist, da die beiden Mengen keine gemeinsamen Elemente haben dürfen.

Die Mengen $\mathcal{E}_x(z)$, $z \in \mathcal{X}$ werden auch als Partitionierungen des Zustandsraums bezeichnet. Mit der Festlegung dieser Partitionierungen ist die Quantisierung des Zustands durch die Beziehung

$$[\mathbf{x}] = z \iff \mathbf{x} \in \mathcal{E}_x(z) \quad (4.11)$$

gegeben [82, S. 397]. Das bedeutet: Liegt der quantitative Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$ in der Partition $\mathcal{E}_x(z)$, dann besitzt der quantisierte Zustand $[\mathbf{x}]$ den qualitativen Wert $z \in \mathcal{X}$ [80, S. 1439]. Der Wert z wird dabei als *qualitativer Zustand* bezeichnet [69], [79], [80].

Am einfachsten lässt sich die Quantisierung anhand von Abbildung 4.2 nachvollziehen, was im folgenden Beispiel 4.1 erläutert ist.

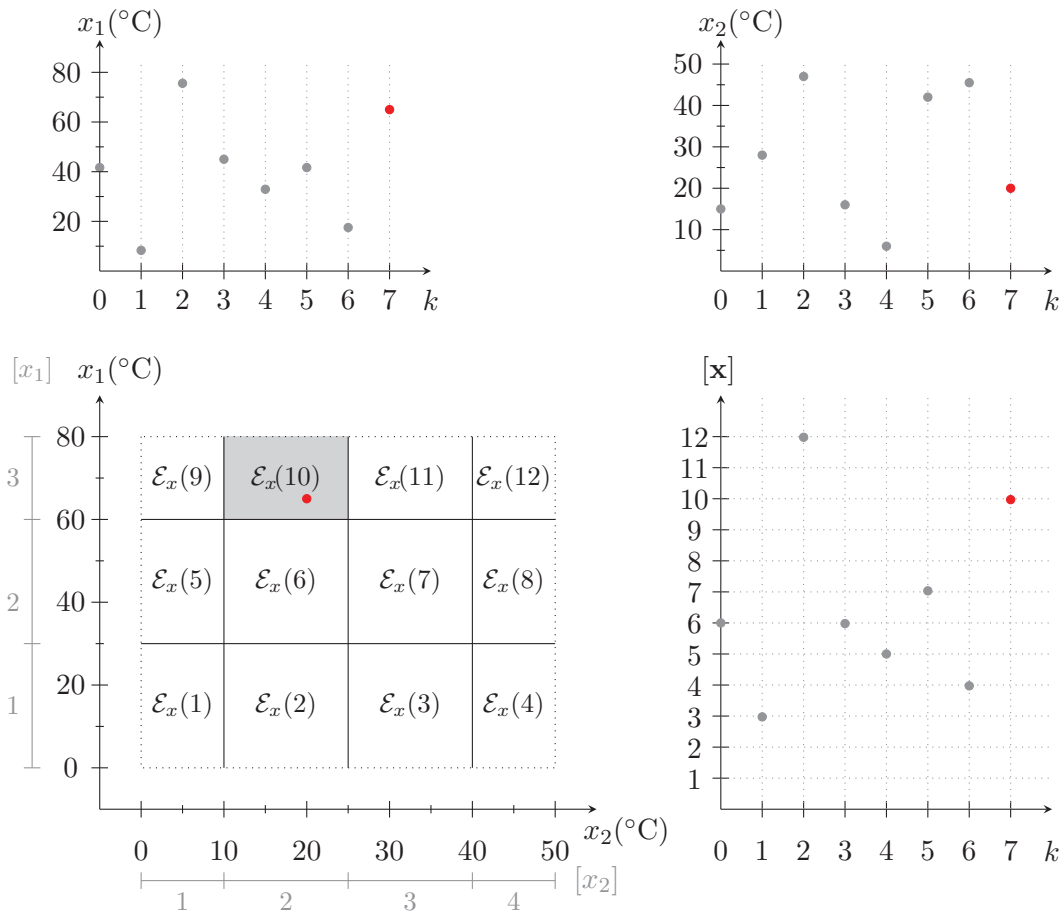


Abbildung 4.2: Allgemeine Quantisierung eines Zustandssignals $\mathbf{x}(k) \in \mathbb{R}^2$, nach [113, S. 39]

Beispiel 4.1 (Allgemeine Quantisierung): Gegeben ist ein Prozess mit dem quantitativem Zustandsraum \mathbb{R}^2 und den beiden reellwertigen Zuständen x_1, x_2 , die durch den Zustandsvektor $\mathbf{x} \in \mathbb{R}^2$ repräsentiert werden. Der quantitative Zustandsraum \mathbb{R}^2 soll nun auf den qualitativen Zustandsraum $\mathcal{X} = \{1, 2, \dots, 12\}$ mit $N = 12$ qualitativen Zuständen $z \in \mathcal{X}$ abgebildet werden. Im oberen Teil der Abbildung 4.2 sind die wertkontinuierlichen Signalwerte $x_1(k)$ und $x_2(k)$ des Zustandsvektors $\mathbf{x}(k) \in \mathbb{R}^2$ für acht Zeitschritte dargestellt. Der untere linke Teil der Abbildung zeigt eine mögliche rechtwinklige Partitionierung des Zustandsraums in 12 Bereiche, die entsprechend für $z = 1, 2, \dots, 12$ durch die Mengen $\mathcal{E}_x(z)$ definiert sind. Betrachtet man nun beispielsweise den zur Zeit $k = 7$ gegebenen Zustandsvektor $\mathbf{x} \in \mathbb{R}^2$ mit den Signalwerten $x_1(7) = 65$ °C und $x_2(7) = 20$ °C, dann kann anhand der Partitionierung einfach nachvollzogen werden, dass für diesen Zustandsvektor $\mathbf{x}(7) \in \mathcal{E}_x(10)$

gilt. Der Zustandsvektor $\mathbf{x}(7)$ besitzt damit den qualitativen Wert $[\mathbf{x}(7)] = 10$ (s. Gleichung (4.11)). Die Interpretation von Gleichung (4.8) ist dann durch

$$\mathbf{x}(7) = \begin{pmatrix} x_1(7) \\ x_2(7) \end{pmatrix} = \begin{pmatrix} 65 \\ 20 \end{pmatrix} \in \mathbb{R}^2 \xrightarrow[\text{Quantisierung}]{\text{Allgemeine}} [\mathbf{x}(7)] = 10 \in \mathcal{X}$$

gegeben. Aus der Betrachtung aller in der Abbildung dargestellten Zeitschritte erhält man die quantisierte Zustandsfolge

$$[\mathbf{X}(0...7)] = (6, 3, 12, 6, 5, 8, 6, 10) .$$

Analog zur oben beschriebenen Quantisierung des Zustands $\mathbf{x} \in \mathbb{R}^n$, werden die quantitativen Eingänge $\mathbf{u} \in \mathbb{R}^m$ und Ausgänge $\mathbf{y} \in \mathbb{R}^q$ durch die Partitionierung des quantitativen Eingangs- und Ausgangsraumes quantisiert. Dazu definiert man einen *qualitativen Eingangsraum* $\mathcal{U} := \{1, 2, \dots, M\}$ und einen *qualitativen Ausgangsraum* $\mathcal{Y} := \{1, 2, \dots, Q\}$, sodass für den quantisierten Eingangsvektor $[\mathbf{u}] \in \mathcal{U}$ und für den Ausgangsvektors $[\mathbf{y}] \in \mathcal{Y}$ gilt. Die Partitionierungsmengen der Ein- und Ausgänge werden dann durch $\mathcal{E}_u(v), v \in \mathcal{U}$ und $\mathcal{E}_y(w), w \in \mathcal{Y}$ definiert, wodurch analog zu (4.11) die folgenden Beziehungen gelten [69, S. 9]:

$$[\mathbf{u}] = v \iff \mathbf{u} \in \mathcal{E}_u(v) , \quad (4.12)$$

$$[\mathbf{y}] = w \iff \mathbf{y} \in \mathcal{E}_y(w) . \quad (4.13)$$

Die Werte v und w bezeichnet man entsprechend als *qualitativen Eingang* und *qualitativen Ausgang* [69, S. 9]. Ein sehr anschauliches und intuitives Beispiel zur Quantisierung findet sich in den Anwendungsbeispielen in Abschnitt 5.2.2.

Erfolgt die Partitionierung in ausschließlich rechtwinklige Partitionierungen wie in Abbildung 4.2, dann kann die allgemeine Quantisierung auch als das *komponentenweise Quantisieren* der einzelnen Zustandsgrößen interpretiert werden [72, S. 189]. Dies ist bei der Partitionierung des Zustandsraumes in Abbildung 4.2 durch die in der Farbe grau eingezeichneten einzelnen qualitativen Werte $[x_1]$ und $[x_2]$ angedeutet. Ein solcher zweidimensionaler Zustandsraum kann demnach einfach durch das getrennte Festlegen der Partitionierungen der Zustände x_1 und x_2 aufgespannt werden. Generell ist diese einfache Interpretation der Partitionierung eines mehrdimensionalen Zustandsraums jedoch nur möglich, solange dieser ausschließlich rechtwinklig partitioniert werden soll. Prinzipiell kann der Zustandsraum jedoch in beliebig komplex geformte Gebilde partitioniert werden, was beispielsweise für sogenannte kanonische Partitionierungen in [69, S. 87–94] beschrieben ist. In dieser Arbeit werden jedoch ausschließlich rechtwinklige Partitionierungen verwendet.

Ein wichtiger Aspekt bei der Quantisierung auf Basis der komponentenweisen Diskretisierung der Wertebereiche der einzelnen Zustandssignale, ist die Festlegung der Nummerierung der Mengen $\mathcal{E}_x(z)$. Dass beispielsweise in Abbildung 4.2 den qualitativen Zustandswerten $[x_1] = 3$ und $[x_2] = 2$ der qualitative Wert $[\mathbf{x}] = 10$ zugeordnet ist, liegt an der Wahl eines *Enumerationsschemas*, mit dem die Partitionierungen $\mathcal{E}_x(z)$ des zweidimensionalen Zustandsraumes \mathbb{R}^2 nummeriert wurden. Vor dem Hintergrund, dass der Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$ im Allgemeinen mehr als nur zwei Signale repräsentiert und damit der Zustandsraum \mathbb{R}^n nicht mehr virtualisierbar ist, muss für größere Systeme eine Regel aufgestellt werden, die den komponentenweise quantisierten Signalen $[x_i], i \in \{1, 2, \dots, n\}$, einen einzelnen qualitativen Zustandswert $[\mathbf{x}]$ im Zustandsraum \mathbb{R}^n zuordnet. Ein solches Enumerationsschema ist in Abschnitt 10.2.1 beschrieben. Wie die Grenzen der Partitionierungen in der Praxis festgelegt werden können, wird in Abschnitt 5.2.2, S. 77 erläutert.

4.3 Qualitatives Modell des quantisierten Systems

4.3.1 Modellbeschreibung

Die Grundlage des qualitativen Modells bildet der initiale stochastische Automat (3.30), dessen Eingabemenge \mathcal{V} , Ausgabemenge \mathcal{W} und Zustandsmenge \mathcal{Z} durch die Mengen qualitativer Eingänge, Ausgänge und Zustände definiert werden:

$$\mathcal{V} = \mathcal{U}, \quad \mathcal{W} = \mathcal{Y}, \quad \mathcal{Z} = \mathcal{X}. \quad (4.14)$$

Damit ist das qualitative Modell wie folgt definiert:

Definition 4.1 (Qualitatives Modell [69, S. 47]): *Der initiale stochastische Automat*

$$\tilde{\mathcal{A}} = (\mathcal{X}, \mathcal{U}, \mathcal{Y}, \tilde{\mathcal{L}}, \tilde{\mathbf{p}}_0) \quad (4.15)$$

mit der Verhaltensrelation

$$\tilde{\mathcal{L}}(\bar{z}, w | z, v) = \Pr \left(\begin{array}{c} [\mathbf{x}(k+1)] = \bar{z} \\ [\mathbf{y}(k)] = w \end{array} \middle| \begin{array}{c} [\mathbf{x}(k)] = z \\ [\mathbf{u}(k)] = v \end{array} \right), \quad (4.16)$$

wird als qualitatives Modell des quantisierten Systems bezeichnet.

Die Verhaltensrelation $\tilde{\mathcal{L}}$ des qualitativen Modells $\tilde{\mathcal{A}}$ beschreibt demnach die bedingte Wahrscheinlichkeit dafür, dass wenn sich das quantisierte System im qualitativen Zustand z befindet und den qualitativen Eingang v erhält, es den qualitativen Ausgang w erzeugt und in den qualitativen Folgezustand \bar{z} übergeht. Die bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}$ sollen dabei auch beim qualitativen Modell durch eine entsprechende Verhaltensmatrix repräsentiert werden, die wie folgt definiert ist.

Definition 4.2 (Verhaltensmatrix des qualitativen Modells): *Die Matrix*

$$\tilde{\mathbf{L}} = \begin{pmatrix} \tilde{\mathbf{L}}(1,1) & \tilde{\mathbf{L}}(1,2) & \cdots & \tilde{\mathbf{L}}(1,M) \\ \tilde{\mathbf{L}}(2,1) & \tilde{\mathbf{L}}(2,2) & \cdots & \tilde{\mathbf{L}}(2,M) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{L}}(R,1) & \tilde{\mathbf{L}}(R,2) & \cdots & \tilde{\mathbf{L}}(R,M) \end{pmatrix} \in [0,1]^{N \cdot R \times N \cdot M} \quad (4.17)$$

mit den Blöcken

$$\tilde{\mathbf{L}}(w, v) = \begin{pmatrix} \tilde{\mathcal{L}}(1, w | 1, v) & \tilde{\mathcal{L}}(1, w | 2, v) & \cdots & \tilde{\mathcal{L}}(1, w | N, v) \\ \tilde{\mathcal{L}}(2, w | 1, v) & \tilde{\mathcal{L}}(2, w | 2, v) & \cdots & \tilde{\mathcal{L}}(2, w | N, v) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{L}}(N, w | 1, v) & \tilde{\mathcal{L}}(N, w | 2, v) & \cdots & \tilde{\mathcal{L}}(N, w | N, v) \end{pmatrix} \in [0,1]^{N \times N} \quad (4.18)$$

heißt Verhaltensmatrix des qualitativen Modells.

Der außerdem in (4.15) auftretende initiale Vektor der Zustandswahrscheinlichkeitsverteilung zur Zeit $k = 0$ ist nun durch die Gleichverteilung

$$\tilde{\mathbf{p}}_0 := \begin{pmatrix} \Pr([\mathbf{x}(0)] = 1) \\ \Pr([\mathbf{x}(0)] = 2) \\ \vdots \\ \Pr([\mathbf{x}(0)] = N) \end{pmatrix} = \begin{pmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{pmatrix} \in [0,1]^N \quad (4.19)$$

gegeben und repräsentiert damit eine Schätzung der initialen qualitativen Zustandswahrscheinlichkeiten $\Pr([\mathbf{x}(0)] = z)$ für alle $z \in \mathcal{X}$. Gleichung (4.19) liegt somit die Annahme zugrunde, dass alle qualitativen Zustände zum Zeitpunkt $k = 0$ gleich wahrscheinlich sind. Ist der qualitative Startzustand jedoch bekannt, kann dieser auch entsprechend berücksichtigt werden (vgl. Beispiel 3.3).

Während der stochastische Automat \mathcal{A} in (3.30) ein exaktes Modell eines stochastischen Prozesses repräsentiert, stellt der stochastische Automat $\tilde{\mathcal{A}}$ in (4.15) nur ein approximiertes Modell des quantisierten Systems dar. Der Grund hierfür ist, dass das quantisierte System im Allgemeinen die Markow-Eigenschaft nicht besitzt [11, S. 463], [80]. Da bei quantisierten Systemen die Wahrscheinlichkeiten eines qualitativen Zustandsübergangs nicht nur vom aktuellen qualitativen Eingang und vom aktuellen qualitativen Zustand abhängen (vgl. Gleichung (3.28)), können die auf der Markow-Eigenschaft basierenden Übergangswahrscheinlichkeiten der Verhaltensrelation $\tilde{\mathcal{L}}$ das Verhalten eines realen Systems nur näherungsweise repräsentieren. Das qualitative Modell kann jedoch entsprechend erweitert werden, indem man anstelle eines Markow-Modells erster Ordnung, ein Modell höherer Ordnung verwendet. Das bedeutet, das qualitative Modell kann durch die Berücksichtigung vergangener Zustände und Eingänge entsprechend optimiert werden [69, S. 84–86]. In dieser Arbeit führt allerdings noch ein weiterer Aspekt dazu, dass das qualitative Modell $\tilde{\mathcal{A}}$ nur ein approximiertes Modell des zugrundeliegenden Prozesses ist. Die Ursache dafür liegt in der Art und Weise wie die Verhaltensrelation $\tilde{\mathcal{L}}$ ermittelt wird und hängt entsprechend mit der *Modellbildung* zusammen, die nachfolgend beschrieben wird.

4.3.2 Modellbildung

Qualitative Modelle können durch unterschiedliche Vorgehensweisen generiert werden. Bevor diese im Folgenden beschrieben werden, muss jedoch auf ein wichtiges Kriterium eingegangen werden, welches als *Vollständigkeit* des qualitativen Modells bezeichnet wird.

Vollständigkeit des qualitativen Modells

Das qualitative Modell soll idealerweise das *qualitative Verhalten* des quantisierten Systems vollständig beschreiben. Zum qualitativen Verhalten des quantisierten Systems gehören alle quantisierten E/A Folgen $([U(0...T)], [Y(0...T)])$, die das quantisierte System mit dem zugrundeliegenden zeitdiskreten und wertkontinuierlichen Prozess für einen beliebigen Zeithorizont T durchlaufen kann [95, S. 145]:

$$\mathcal{B}_q^T = \{[U(0...T)], [Y(0...T)]\} \subseteq \mathcal{U}^{T+1} \times \mathcal{Y}^{T+1}. \quad (4.20)$$

Ein qualitatives Modell mit dem Verhalten \mathcal{B}_m^T wird nun als vollständig bezeichnet, wenn es für alle Zeiten T , das qualitative Verhalten des quantisierten Systems \mathcal{B}_q^T einschließt [69, S. 18]:

$$\mathcal{B}_m^T \supseteq \mathcal{B}_q^T. \quad (4.21)$$

Anders formuliert bedeutet dies, dass das qualitative Modell vollständig ist, wenn dessen Verhalten \mathcal{B}_m^T alle E/A Folgen umfasst, die dem quantisierten System möglich sind. Dabei stellt das Verhalten des Modells \mathcal{B}_m^T im Allgemeinen eine echte Obermenge des qualitativen Verhaltens \mathcal{B}_q^T des quantisierten Systems dar, was dazu führt, dass E/A Folgen existieren die zwar zum Modellverhalten gehören, aber nicht zum Verhalten des quantisierten Systems [11, S. 468]. Solche E/A Folgen werden als *unzutreffende Lösungen* (engl. spurious solutions) bezeichnet, die ein charakteristisches Merkmal qualitativer Modelle darstellen und deren Ursprung auf die Tatsache zurückzuführen ist, dass das qualitative Modell ein Markow-Modell ist, mit dem jedoch das nicht markowsche Verhalten

eines zugrundeliegenden quantisierten Systems beschrieben wird [11, S. 463, 468], [81, S. 279], [84, S. 33]. Sehr anschauliche Beispiele bezüglich der unzutreffenden Lösungen finden sich in [69, S. 38–39] sowie in [11, S. 465–477], wobei letztere Quelle auch gut illustrierte Darstellungen und ausführliche Beschreibungen zur Vollständigkeit qualitativer Modelle bietet.

Methoden der Modellbildung

Abstraktion: Qualitative Modelle können aus quantitativen Modellen, wie dem Zustandsraummodell (2.3)–(2.5), durch *Abstraktion* generiert werden [71, S. 895]. Die Abstraktion des qualitativen Modells erfolgt dabei durch Verfahren, die in der Literatur als *Zellabbildungsmethode* [69, S. 50–53] oder *Point-Mapping* [113, S. 97–98] bekannt sind. Diese Methode kann im Allgemeinen nicht garantieren, dass das qualitative Modell ein vollständiges Modell des quantisierten Systems ist. Eine Methode, die dies sicherstellt, ist unter dem Namen *Hyperbox-Mapping* bekannt und ebenfalls in [113, S. 98–102] beschrieben. Beide Methoden haben gemeinsam, dass zur Berechnung des qualitativen Modells quantitative White-Box oder zumindest Grey-Box Simulationsmodelle vorhanden sein müssen. Die Abstraktion qualitativer Modelle spielt in dieser Arbeit keine Rolle, weshalb auf eine eingehende Beschreibung verzichtet wird. Detaillierte Ausführungen hierzu finden sich in den angegebenen Quellen.

Qualitative Identifikation linearer Systeme: Diese Methode zur Erzeugung qualitativer Modelle ist, wie der Name bereits sagt, nur für lineare Systeme geeignet. Außerdem liefert die Methode als Modell keinen stochastischen Automaten, sondern lediglich einen nichtdeterministischen Automaten, der allerdings ein vollständiges qualitatives Modell repräsentiert. Im Unterschied zur Abstraktion basiert diese Methode auf der Verwendung realer Messdaten des quantisierten Systems und benötigt demnach kein quantitatives Simulationsmodell. Da in dieser Arbeit ausschließlich reale Systeme untersucht werden, die im Allgemeinen kein lineares Verhalten zeigen, ist diese Methode eher weniger geeignet, um ein qualitatives Modell zu erzeugen. Eine Beschreibung des Verfahrens findet sich in [67], [68] und [69, S. 55–60].

Stochastische qualitative Identifikation: Die stochastische qualitative Identifikation ermöglicht die qualitative Modellbildung auf Basis realer Messdaten und ist sowohl für lineare als auch nichtlineare Systeme geeignet. Das Verfahren bildet in dieser Arbeit die Grundlage der qualitativen Modellgenerierung. Die folgende Beschreibung der Methode der stochastischen qualitativen Identifikation ist der Arbeit [69, S. 53–55] entnommen:

Als Erstes können die Elemente der quantisierten Eingangs-, Ausgangs- und Zustandsfolgen (4.4)–(4.6) für $k = 0, 1, 2, \dots, T - 1$ zu 4-Tupeln

$$[\Theta(k)] = ([\mathbf{x}(k+1)], [\mathbf{y}(k)], [\mathbf{x}(k)], [\mathbf{u}(k)]) \quad (4.22)$$

zusammen gefasst werden. Anschließend kann aus der jeweiligen Anzahl gleicher Tupel, die *absolute Häufigkeit*

$$H(\bar{z}, w, z, v) = \text{Anzahl aller Tupel mit } [\Theta(k)] = (\bar{z}, w, z, v) , \quad (4.23)$$

für alle Kombinationen von \bar{z}, w, z, v berechnet werden. Aus dieser absoluten Häufigkeit $H(\bar{z}, w, z, v)$ und der Anzahl aller Experimente oder Messungen T , lässt sich die *relative Häufigkeit*

$$\hat{h}(\bar{z}, w, z, v) = \frac{H(\bar{z}, w, z, v)}{T} \quad (4.24)$$

bestimmen, die eine Schätzung der *gemeinsamen* Wahrscheinlichkeit

$$\hat{h}(\bar{z}, w, z, v) \approx \Pr \left(\begin{array}{l} [\mathbf{x}(k+1)] = \bar{z}, \quad [\mathbf{x}(k)] = z, \\ [\mathbf{y}(k)] = w, \quad [\mathbf{u}(k)] = v \end{array} \right) \quad (4.25)$$

darstellt. Aus der gemeinsamen Wahrscheinlichkeit $\hat{h}(\bar{z}, w, z, v)$ kann eine Näherung der Randverteilung

$$\hat{h}(z, v) = \sum_{\bar{z}=1}^N \sum_{w=1}^Q \hat{h}(\bar{z}, w, z, v) \quad (4.26)$$

bestimmt werden, woraus dann eine Approximation der bedingten Wahrscheinlichkeit

$$\tilde{\mathcal{L}}(\bar{z}, w | z, v) = \begin{cases} \frac{\hat{h}(\bar{z}, w, z, v)}{\hat{h}(z, v)} & \text{wenn } \hat{h}(z, v) \neq 0 \\ 0 & \text{sonst} \end{cases} \quad (4.27)$$

berechnet werden kann (s. a. Beispiel 4.2).

Die Bestimmung der approximierten Verhaltensrelation $\tilde{\mathcal{L}}$ basiert letztlich auf dem *Gesetz der großen Zahl* [60, S. 69], nach dem sich mit steigender Anzahl von Experimenten T , die relativen Häufigkeiten $\hat{h}(\bar{z}, w, z, v)$ der tatsächlichen Wahrscheinlichkeit $\Pr([\mathbf{x}(k+1)] = \bar{z}, [\mathbf{y}(k)] = w | [\mathbf{x}(k)] = z, [\mathbf{u}(k)] = v)$ des Ereignisses annähern. Die Erstellung eines qualitativen Modells mithilfe der Gleichungen (4.22)–(4.27) wird auch als *Trainieren* des qualitativen Modells bezeichnet. Dabei ist offensichtlich, dass das erzeugte qualitative Modell stets nur die Verhaltensformen wiedergeben kann, die auch anhand der zugrundeliegenden Messdaten (4.4)–(4.6) identifiziert wurden. Da in der Praxis nicht davon ausgegangen werden kann, dass die Messdaten tatsächlich alle dem quantisierten System möglichen Verhaltensformen umfassen, wird das qualitative Modell $\tilde{\mathcal{A}}$ mit der Verhaltensrelation $\tilde{\mathcal{L}}$ in der Regel unvollständig sein, womit

$$\mathcal{B}_m^T \not\supseteq \mathcal{B}_q^T \quad (4.28)$$

gilt. Dabei ist zu Unterscheiden, ob das qualitative Modell nicht alle dem quantisierten System möglichen Verhaltensformen wiedergeben kann oder ob es auch strukturell unvollständig ist. Die strukturelle Unvollständigkeit bedeutet, dass die Bedingungen (3.36) und (3.37) für die Verhaltensrelation $\tilde{\mathcal{L}}$ nicht erfüllt sind. Das unvollständige qualitative Modelle zur Fehlererkennung eingesetzt werden können, wurde bereits in [91], [92], [93] gezeigt und wird auch anhand der Anwendungsbeispiele in Kapitel 5 deutlich.

Beispiel 4.2 (Stochastische qualitative Identifikation): *Für ein System mit $n = 2$ physikalischen Zuständen und $N = 12$ qualitativen Zuständen, ist die quantisierte Zustandsfolge*

$$[\mathbf{X}(0\dots 9)] = (6, 3, 12, 6, 5, 8, 4, 10, 6, 5) \quad (4.29)$$

aus Abbildung 4.2 gegeben. Die Folge wurde hier noch um zwei zusätzliche Werte erweitert, sodass diese für den Zeithorizont $T = 9$ gegeben ist. Außerdem besitzt das System $m = 1$ Eingangssignale und $q = 1$ Ausgangssignale, wobei die Anzahl qualitativer Eingänge $M = 2$ und die der Ausgänge $Q = 2$ beträgt. Die quantisierten Ein- und Ausgangsfolgen seien dabei durch

$$[\mathbf{U}(0\dots 9)] = (1, 2, 2, 1, 1, 1, 2, 1, 1, 1), \quad (4.30)$$

$$[\mathbf{Y}(0\dots 9)] = (1, 1, 1, 1, 1, 2, 2, 1, 1, 1) \quad (4.31)$$

gegeben. Die Folgen qualitativer Ein-, Ausgänge und Zustände können nun in einer Tabelle angeordnet werden, wobei jede Zeile ein Tupel $[\Theta(k)]$ nach Gleichung (4.22) repräsentiert. Wie die Tabelle 4.1 zeigt, erhält man den qualitativen Wert des Folgezustandes $[\mathbf{x}(k+1)] = \bar{z}$ entsprechend durch das Verschieben des Zustandes um einen Zeitschritt, weshalb die Länge der Tabelle $T - 1$ beträgt. Die absoluten Häufigkeiten der Tupel $[\Theta(k)]$ sind nun leicht abzählbar. Wie die Tabelle zeigt, besitzen alle Tupel eine absolute Häufigkeit von Eins, bis auf das Tupel $[\Theta] = (5, 1, 6, 1)$, welches zu den

Tabelle 4.1: Qualitative Tupel $[\Theta(k)]$

k	$[\mathbf{x}(k+1)] = \bar{z}$	$[\mathbf{y}(k)] = w$	$[\mathbf{x}(k)] = z$	$[\mathbf{u}(k)] = v$
0	3	1	6	1
1	12	1	3	2
2	6	1	12	2
3	5	1	6	1
4	8	1	5	1
5	4	2	8	1
6	10	2	4	2
7	6	2	10	1
8	5	1	6	1

Zeiten $k = 3$ und $k = 8$ auftritt und dessen absolute Häufigkeit durch $H(5, 1, 6, 1) = 2$ gegeben ist (vgl. Gleichung (4.23)). Für dieses Tupel wird nun exemplarisch die gesuchte bedingte Wahrscheinlichkeit $\tilde{\mathcal{L}}(5, 1, 6, 1)$ bestimmt: Für die relative Häufigkeit erhält man zunächst nach Gleichung (4.24) den Wert $\hat{h}(5, 1, 6, 1) = 2/9$, während alle anderen Tupel eine relative Häufigkeit von $1/9$ besitzen. Damit kann aus (4.26) die Randverteilung $\hat{h}(6, 1) = \hat{h}(3, 1, 6, 1) + \hat{h}(5, 1, 6, 1) = 1/9 + 2/9 = 1/3$ bestimmt werden. Zuletzt erhält man dann aus (4.27) eine Approximation der bedingten Wahrscheinlichkeit $\tilde{\mathcal{L}}(5, 1, 6, 1) = \hat{h}((5, 1, 6, 1))/\hat{h}(6, 1) = (2/9)/(1/3) = 2/3$. Analog verfährt man für alle anderen Tupel in der Tabelle.

Eine bei dieser Vorgehensweise für die qualitative Modellierung hilfreiche Eigenschaft ist, dass die Werte der Folgezustände in der Tabelle 4.1 auch um mehr als nur einen Wert verschoben werden können. Diese Verschiebung wird in dieser Arbeit als *Beobachterzeitschrittdauer* T_b des qualitativen Modells bezeichnet. Eine Verschiebung des Folgezustandes um zwei Zeitschritte in Tabelle 4.1, führt dann beispielsweise dazu, dass in der ersten Zeile das Tupel $[\Theta] = (12, 1, 6, 1)$ und in der zweiten Zeile das Tupel $[\Theta] = (6, 1, 3, 2)$ steht. Die dann berechneten Wahrscheinlichkeiten unterscheiden sich entsprechend und führen zu einer »qualitativeren« Repräsentation der Verhaltensrelation $\tilde{\mathcal{L}}$. Anschauliche Beispiele hierzu und zum Begriff der Beobachterzeitschrittdauer finden sich in Kapitel 5.

4.4 Qualitative Beobachtung und Fehlererkennung

In diesem Abschnitt wird erläutert, wie die Fehlererkennung in quantisierten Systemen anhand der qualitativen Beobachtung realisiert werden kann. Dazu wird zunächst eine einfache qualitative Beobachtergleichung, basierend auf Definition 3.7 eingeführt. Außerdem werden für die Anwendung wichtige Eigenschaften des Verhaltenssignals erläutert sowie ein Algorithmus zur qualitativen Fehlererkennung definiert.

4.4.1 Modellierungsziel

Ziel der qualitativen Modellierung des quantisierten Systems durch den stochastischen Automaten $\tilde{\mathcal{A}}$, ist die sogenannte *qualitative Beobachtung*, die entsprechend auf der Beobachtung stochastischer Automaten beruht. Allgemein formuliert versteht man unter der *qualitativen Beobachtung* die Bestimmung der qualitativen Zustandswahrscheinlichkeiten des quantisierten Systems, unter der Bedingung, dass die Wahrscheinlichkeiten des qualitativen Eingangs

$$\Pr([\mathbf{u}(k)] = v), \quad v \in \mathcal{U} \quad (4.32)$$

und des qualitativen Ausgangs

$$\Pr([\mathbf{y}(k)] = w), \quad w \in \mathcal{Y} \quad (4.33)$$

gegeben sind. Die Berechnung der qualitativen Zustandswahrscheinlichkeiten erfolgt jedoch im Allgemeinen unter der Annahme, dass die qualitativen Ein- und Ausgangswerte zu jeder Zeit k exakt bekannt sind, was bedeutet, dass die Wahrscheinlichkeit mit der die quantisierte Eingangsgröße $[\mathbf{u}(k)]$ den qualitativen Wert $v(k) \in \mathcal{U}$ und die quantisierte Ausgangsgröße $[\mathbf{y}(k)]$ den qualitativen Wert $w(k) \in \mathcal{Y}$ annimmt, durch

$$\Pr([\mathbf{u}(k)] = v(k)) = 1 \quad (4.34)$$

und

$$\Pr([\mathbf{y}(k)] = w(k)) = 1 \quad (4.35)$$

gegeben ist (vgl. Abschnitt 3.4.4). Die Zustandswahrscheinlichkeiten werden somit in der Regel für sicher bekannte quantisierte Eingangsfolgen $[\mathbf{U}(0...T)]$ und Ausgangsfolgen $[\mathbf{Y}(0...T)]$ bestimmt. Gesucht ist damit die bedingte Wahrscheinlichkeit

$$\Pr([\mathbf{x}(k+1)] = \bar{z} \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)]), \quad \bar{z} \in \mathcal{X}. \quad (4.36)$$

Abbildung 4.3 visualisiert das Konzept der qualitativen Zustandsbeobachtung. Dabei wird davon ausgegangen, dass der qualitative Wert $[\mathbf{u}(k)] = v$ des Eingangsvektors $\mathbf{u}(k)$ und der qualitative Wert $[\mathbf{y}(k)] = w$ des Ausgangsvektors $\mathbf{y}(k)$ als Eingangsgrößen für das qualitative Modell $\tilde{\mathcal{A}}$ verfügbar sind. Der qualitative Beobachter generiert dann, solange es für die qualitativen Ein- und Ausgangswerte einen qualitativen Folgezustand $[\mathbf{x}(k+1)] = \bar{z}$ durchlaufen kann, die Zustandswahrscheinlichkeiten (4.36) für $k = 0, 1, 2, \dots, T$. Im folgenden Abschnitt wird erläutert wie die gesuchten

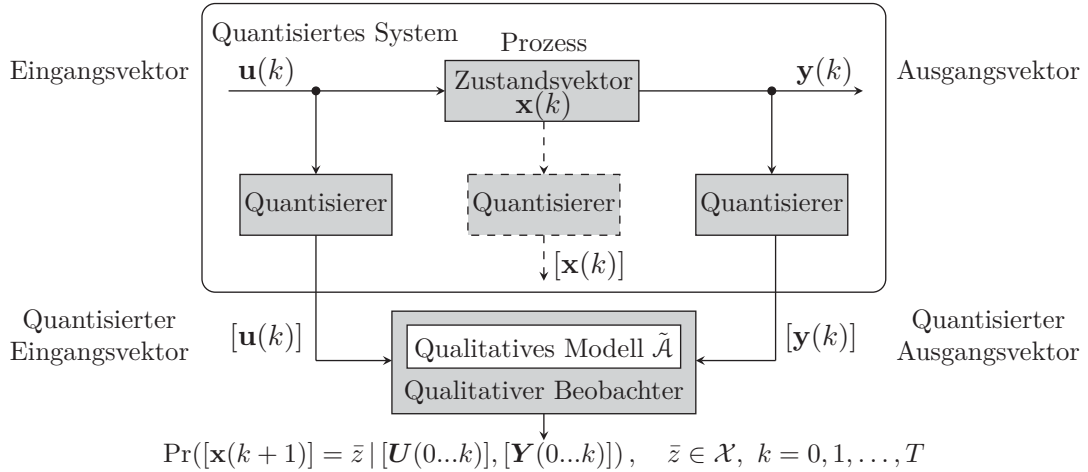


Abbildung 4.3: Qualitative Beobachtung, nach [113, S. 170]

Zustandswahrscheinlichkeiten durch eine einfache Beobachtergleichung berechnet werden können.

4.4.2 Qualitative Beobachtergleichung

Die Definition der Beobachtergleichung für die Matrixrepräsentation qualitativer Modelle, erfolgt analog zu Definition 3.7.

Definition 4.3 (Qualitative Matrix-Beobachtergleichung): Sind die quantisierten Eingangs- und Ausgangsfolgen $([U(0...T)], [Y(0...T)])$ des quantisierten Systems konsistent mit dem qualitativen Modell \tilde{A} , dann kann eine Approximation der Wahrscheinlichkeitsverteilungen der qualitativen Zustände mit

$$\begin{aligned} k \geq 0 : \quad \tilde{\mathbf{p}}_z(k+1) &= \frac{\tilde{\mathbf{L}}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)}{\|\tilde{\mathbf{L}}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1} \in [0, 1]^N, \\ k = 0 : \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0 \end{aligned} \quad (4.37)$$

für alle Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden. Der qualitative Eingang $v(k)$ und Ausgang $w(k)$ sind dabei in jedem Zeitschritt k durch die quantisierte Eingangsgröße $[\mathbf{u}(k)] = v(k)$ und Ausgangsgröße $[\mathbf{y}(k)] = w(k)$ gegeben.

Der Wahrscheinlichkeitsvektor der qualitativen Zustände zur Zeit $k+1$ ist durch

$$\tilde{\mathbf{p}}_z(k+1) = \begin{pmatrix} \Pr([\mathbf{x}(k+1)] = 1 \mid [U(0...k)], [Y(0...k)]) \\ \Pr([\mathbf{x}(k+1)] = 2 \mid [U(0...k)], [Y(0...k)]) \\ \vdots \\ \Pr([\mathbf{x}(k+1)] = N \mid [U(0...k)], [Y(0...k)]) \end{pmatrix} \in [0, 1]^N \quad (4.38)$$

gegeben und repräsentiert eine Schätzung der bedingten Wahrscheinlichkeitsverteilung

$$\Pr([\mathbf{x}(k+1)] = \bar{z} \mid [U(0...k)], [Y(0...k)]), \quad \bar{z} \in \mathcal{X}, \quad (4.39)$$

die wiederum eine Kurznotation der expliziten Schreibweise

$$\begin{aligned} \Pr([\mathbf{x}(k+1)] = \bar{z} \mid [U(0...k)], [Y(0...k)]) \\ = \Pr\left([\mathbf{x}(k+1)] = \bar{z} \mid \begin{array}{l} [\mathbf{u}(0)] = v(0), [\mathbf{u}(1)] = v(1), \dots, [\mathbf{u}(k)] = v(k), \\ [\mathbf{y}(0)] = w(0), [\mathbf{y}(1)] = w(1), \dots, [\mathbf{y}(k)] = w(k) \end{array} \right) \end{aligned} \quad (4.40)$$

darstellen (vgl. Gleichung (3.59)). Gleichung (4.40) repräsentiert dabei die bedingte Wahrscheinlichkeit dafür, dass sich das quantisierte System zum Zeitpunkt $k+1$ im qualitativen Zustand $\bar{z} \in \mathcal{X}$ befindet, wenn es bis zur Zeit k die quantisierten E/A Folgen $([U(0...k)], [Y(0...k)])$ durchlaufen hat. Aufgrund der bereits beschriebenen approximativen Eigenschaften des qualitativen Modells, beschreibt der Vektor $\tilde{\mathbf{p}}_z(k+1)$ jedoch lediglich eine Näherung dieser Wahrscheinlichkeit [11, S. 482]. Der Vektor $\tilde{\mathbf{p}}_z(k)$ in (4.37) repräsentiert analog zu Gleichung (3.54) die Wahrscheinlichkeitsverteilung der qualitativen Zustände $\Pr([\mathbf{x}(k)] = z \mid [U(0...k-1)], [Y(0...k-1)]), z \in \mathcal{X}$ zum Zeitpunkt k .

Wendet man die qualitative Beobachtergleichung (4.37) auf bis zu einem Zeithorizont T gegebene quantisierte E/A Folgen $([U(0...T)], [Y(0...T)])$ an, erhält man eine Folge von Wahrscheinlichkeitsvektoren

$$\tilde{\mathbf{P}}_z(0...T+1 \mid [U(0...T)], [Y(0...T)]) = (\tilde{\mathbf{p}}_z(0), \tilde{\mathbf{p}}_z(1), \tilde{\mathbf{p}}_z(2), \dots, \tilde{\mathbf{p}}_z(T+1)), \quad (4.41)$$

welche die approximierten Wahrscheinlichkeitsverteilungen der qualitativen Zustände bis zum Zeithorizont $T+1$ beschreibt. Außerdem wird der Nenner der qualitativen Beobachtergleichung (4.37) erneut mit

$$\tilde{b}(k) := \|\tilde{\mathbf{L}}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1 \in [0, 1] \quad (4.42)$$

abgekürzt, womit man wie in Gleichung (3.63), das für die Fehlererkennung wichtige Verhaltenssignal

$$\tilde{B}(0...T \mid [U(0...T)], [Y(0...T)]) = (\tilde{b}(0), \tilde{b}(1), \tilde{b}(2), \dots, \tilde{b}(T)) \quad (4.43)$$

des qualitativen Modells bilden kann.

Konsistenzprüfung der quantisierten Ein- und Ausgangsfolgen

Die Bewertung, ob die quantisierten E/A Folgen $([U(0...T)], [Y(0...T)])$ konsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$ sind, erfolgt analog zur Beschreibung in Abschnitt 3.5.3.

Eine qualitative E/A Folge $([U(0...T)], [Y(0...T)])$ mit den E/A Paaren $([u(k)], [y(k)])$ ist *konsistent* mit dem qualitativen Modell $\tilde{\mathcal{A}}$, wenn für die Elemente $\tilde{b}(k)$ der Verhaltensfolge \tilde{B} folgendes gilt:

$$\forall k : \quad \tilde{b}(k) > 0. \quad (4.44)$$

Die qualitative E/A Folge gehört damit zum Verhalten \mathcal{B}_m^T des qualitativen Modells $\tilde{\mathcal{A}}$ und es gilt entsprechend $(([U(0...T)], [Y(0...T)]) \in \mathcal{B}_m^T$. Folglich ist eine qualitative E/A Folge $([U(0...T)], [Y(0...T)])$ *inkonsistent* mit dem qualitativen Modell $\tilde{\mathcal{A}}$, wenn

$$\exists k : \quad \tilde{b}(k) = 0 \quad (4.45)$$

und damit $([U(0...T)], [Y(0...T)]) \notin \mathcal{B}_m^T$ gilt.

4.4.3 Qualitative Fehlererkennung

Unter der Voraussetzung, dass das qualitative Modell das Nominalverhalten des zugrundeliegenden quantisierten Systems beschreibt, können Inkonsistenzen in den quantisierten Eingangs- und Ausgangsfolgen als Fehler oder Anomalie interpretiert werden. Das Signal \tilde{B} bildet damit die Basis für die Fehlererkennung in technischen Systemen, wobei jedoch im Gegensatz zu den theoretischen Betrachtungen in Abschnitt 3.5.3, weitere Anforderungen an den Algorithmus zur Fehlererkennung als auch an die Interpretierbarkeit des Verhaltenssignals gestellt werden:

1. **Erkennung transienter Fehler:** Im Allgemeinen treten an technischen Anlagen nicht nur Fehler auf, die zu einem vollständigen Ausfall der Anlage führen, sondern auch solche, die beispielsweise wiederkehrend zu irgendwelchen Zeiten auftreten und deren Dauer womöglich zeitlich begrenzt ist. Der Beobachteralgorithmus soll dementsprechend in der Lage sein, solche transienten Fehler zu detektieren. Das bedeutet, dass die Beobachtergleichung (4.37) bei einem inkonsistenten E/A Paar nicht gestoppt werden darf, sondern dieser Fall entsprechend im Algorithmus berücksichtigt werden muss. Eine einfache Lösung hierfür ist, im Falle eines inkonsistenten E/A Paares $\tilde{b}(k) = 0$ die Beobachtergleichungen solange mit der initialen Gleichverteilung der Zustände (4.19) neu zu starten, bis wieder $\tilde{b}(k) > 0$ gilt. Die wird im Algorithmus zur Fehlererkennung im Abschnitt 4.4.4 entsprechend berücksichtigt.
2. **Einstellbare Sensitivität:** In der Regel sollte es vermieden werden, bereits einzelne oder einige wenige inkonsistente E/A Paare direkt als Fehler zu klassifizieren und diese dem Betreiber oder Nutzer der entsprechenden technischen Anlage zu melden. Der Algorithmus zur Fehlererkennung soll dementsprechend in seiner Sensitivität justierbar sein, damit in Abhängigkeit des zugrundeliegenden Prozess, den Abtastzeiten und den Nutzerbedürfnissen, eingestellt werden kann, nach wie vielen inkonsistenten E/A Paaren tatsächlich eine Fehlermeldung generiert oder visualisiert werden soll. Diese Anforderung lässt sich durch die Einführung eines einfachen ungewichteten gleitenden Mittelwertes

$$\tilde{b}_m(k) = \frac{1}{\phi} \sum_{j=0}^{\min(k, \phi-1)} \tilde{b}(k-j) \in [0, 1] \quad (4.46)$$

realisieren. Darin ist ϕ die *Fensterbreite*, also die einstellbare Anzahl zurückliegender Zeitschritte aus denen der aktuelle Mittelwert $\tilde{b}_m(k)$ gebildet wird. Aus (4.46) kann dann für Zeiten $k = 0, 1, 2, \dots, T$ die Folge

$$\tilde{B}_m(0...T) = (\tilde{b}_m(0), \tilde{b}_m(1), \tilde{b}_m(2), \dots, \tilde{b}_m(T)) \quad (4.47)$$

gebildet werden.

3. **Gute Interpretierbarkeit des Verhaltenssignals:** Da es sich bei dem Signal \tilde{B}_m um ein stochastisches Signal handelt, schwanken dessen Werte aufgrund von mehr oder weniger wahrscheinlichen E/A Paaren zwischen Null und Eins. Zur Verbesserung der Interpretierbarkeit wird ein einfacher Schwellwert $\tau \in [0, 1)$ eingeführt, der die Signalwerte $\tilde{b}_m(k)$ für $k = 0, 1, 2, \dots, T$ mit

$$\tilde{b}_b(k) = \begin{cases} 1 & \text{wenn } \tilde{b}_m(k) > \tau, \\ 0 & \text{sonst,} \end{cases} \quad (4.48)$$

auf die booleschen Werte $\tilde{b}_b(k) \in \{0, 1\}$ abbildet. Dabei bedeutet $\tilde{b}_b(k) = 0$, dass ein Fehler oder ein nicht nominaler Betriebszustand vorliegt und $\tilde{b}_b(k) = 1$, dass sich das zugrundeliegende System im Nominalbetrieb befindet. Im Folgenden gilt solange nicht anders erwähnt $\tau = 0$, da nur tatsächlich inkonsistente E/A Paare auch als Fehler klassifiziert werden sollen. Dem Schwellwert τ kommt in dieser Arbeit noch eine weitere Bedeutung zu, die in Abschnitt 9.1.3 erläutert wird. Wendet man Gleichung (4.48) nun für $k = 0, 1, 2, \dots, T$ auf die Signalwerte $\tilde{b}_m(k)$ an, erhält man die boolesche Folge

$$\tilde{B}_b(0...T) = (\tilde{b}_b(0), \tilde{b}_b(1), \tilde{b}_b(2), \dots, \tilde{b}_b(T)) . \quad (4.49)$$

Unter Einbeziehung dieser Erweiterungen kann abschließend ein qualitativer Beobachteralgorithmus zur Fehlererkennung definiert werden.

4.4.4 Algorithmus zur qualitativen Fehlererkennung

Algorithmus 4.1: Algorithmus zur qualitativen Fehlererkennung (Matrixrepräsentation)

Gegeben: Qualitatives Modell \tilde{A} , Gl. (4.15) mit Verhaltensmatrix \tilde{L} , Gl. (4.17)
 Quantisierte Eingangsfolge $[U(0...T)]$ mit Elementen $[u(k)] = v(k)$, Gl. (4.4)
 Quantisierte Ausgangsfolge $[Y(0...T)]$ mit Elementen $[y(k)] = w(k)$, Gl. (4.6)

Initialisiere: $k = 0$, $\tilde{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$

Iteriere:

- 1 **solange** $0 \leq k \leq T$
- 2 **berechne** $\tilde{L}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)$ Gl. (4.37)
- 3 **berechne** $\|\tilde{L}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1 =: \tilde{b}(k)$, Gl. (4.37)
- 4 **berechne** $\tilde{b}_m(k)$, Gl. (4.46) und $\tilde{b}_b(k)$, Gl. (4.48)
- 5 **wenn** $\tilde{b}(k) = 0$ **dann**
- 6 **setze** $\tilde{\mathbf{p}}_z(k+1) = \tilde{\mathbf{p}}_0$
- 7 **setze** $k := k+1$ und fahre fort mit Schritt 1
- 8 **sonst**
- 9 **berechne** $\tilde{\mathbf{p}}_z(k+1)$, Gl. (4.37)
- 10 **setze** $k := k+1$ und fahre fort mit Schritt 1

Ergebnis: Folge $\tilde{\mathbf{P}}_z(0...T+1 | [U(0...T)], [Y(0...T)])$, Gl. (4.41)
 Folge $\tilde{B}(0...T | [U(0...T)], [Y(0...T)])$, Gl. (4.43)
 Folge $\tilde{B}_m(0...T)$, Gl. (4.47)
 Folge $\tilde{B}_b(0...T)$, Gl. (4.49)

Der Beobachteralgorithmus 4.1 läuft wie folgt ab: Die Initialisierung erfolgt zur Zeit $k = 0$ mit der Startverteilung $\tilde{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$. Wenn nun die Eingangsbedingung in Schritt 1 erfüllt ist, wird in Schritt 2

zunächst der Zähler der Beobachtergleichung (4.37) berechnet. In Schritt 3 erfolgt die Bestimmung von $\tilde{b}(k)$ auf Basis des Nenners der Beobachtergleichung. Im vierten Schritt wird der gleitende Mittelwert $\tilde{b}_m(k)$ sowie der boolesche Wert $\tilde{b}_b(k)$ bestimmt. Anschließend folgt in Schritt 5 die Prüfung, ob das E/A Paar $(v(k), w(k))$ inkonsistent ist. Trifft dies zu, wird dem Zustandsvektor $\tilde{\mathbf{p}}_z(k+1)$ für diesen Zeitschritt k die initiale Startverteilung $\tilde{\mathbf{p}}_0$ zugewiesen (s. a. Anmerkung 4.2). Anschließend wird der Zeitindex mit $k := k + 1$ um einen Wert verschoben und die Prozedur startet wieder in Schritt 1. Durch die Schritte 6 und 7 wird damit sichergestellt, dass ein inkonsistentes E/A Paar in der Beobachtergleichung (4.37) nicht zu einer Division durch Null und damit zum Abbruch des Algorithmus führt. Gilt in Schritt 5 hingegen $\tilde{b}(k) > 0$, dann ist das E/A Paar konsistent mit dem Modell und die Berechnung des Vektors $\mathbf{p}_z(k+1)$ erfolgt in Schritt 9 mit der Beobachtergleichung (4.37). Anschließend wird der Zeitindex mit $k := k + 1$ um einen Wert verschoben und die Prozedur startet wieder in Schritt 1. Die Ausführung dieser Schleife erfolgt solange bis $k = T$ gilt und alle E/A Paare prozessiert wurden. Als Ergebnis des Beobachteralgorithmus erhält man die Folge der Zustandsvektoren $\tilde{\mathbf{P}}_z$ sowie die Verhaltenssignale \tilde{B} , \tilde{B}_m und \tilde{B}_b .

Es sei auch hier angemerkt, dass der Beobachteralgorithmus nicht nur auf bereits gegebene E/A Folgen angewandt werden kann, sondern auch auf Messdaten die für $k = 0, 1, 2, \dots$ kontinuierlich auflaufen. Dazu ersetzt man den Ausdruck in Schritt 1 einfach mit »messe $[\mathbf{u}(k)]$ und $[\mathbf{y}(k)]$ « wobei wieder $[\mathbf{u}(k)] = v(k)$ und $[\mathbf{y}(k)] = w(k)$ gilt.

Anmerkung 4.2 (Speicherung der Vektorfolgen $\tilde{\mathbf{P}}_z$): *Im Falle eines inkonsistenten E/A Paares wird durch die notwendige Reinitialisierung des Algorithmus in Schritt 6 dem Vektor $\tilde{\mathbf{p}}_z(k+1)$ die Gleichverteilung $\tilde{\mathbf{p}}_0$ zugewiesen. Entsprechend tritt diese Gleichverteilung auch an der betreffenden Stelle in der Vektorfolge $\tilde{\mathbf{P}}_z$ auf, was formal betrachtet nicht korrekt ist, da für ein inkonsistentes E/A Paar auch keine Zustandsübergangswahrscheinlichkeiten existieren. Dieser Umstand kann für $\tilde{b}(k) = 0$ nach Schritt 5 berücksichtigt werden, indem man mit*

$$\tilde{\mathbf{p}}_z(k+1) = \mathbf{0} \in \mathbb{R}^N,$$

den Nullvektor $\mathbf{0} \in \mathbb{R}^N$ anstelle der Gleichverteilung $\tilde{\mathbf{p}}_0$ in der Vektorfolge $\tilde{\mathbf{P}}_z$ speichert. Wohlge-merkt ändert sich hierbei nichts an der Reinitialisierung des Algorithmus, die wie vorgesehen mit der Gleichverteilung in Schritt 6 erfolgt. Allgemein kann dieser zusätzliche Schritt als optional angesehen werden und macht nur Sinn, wenn man an den Wahrscheinlichkeitsverteilungen der Folge $\tilde{\mathbf{P}}_z$ interessiert ist und diese, wie auch in dieser Arbeit realisiert, visualisieren möchte. Soll hingegen lediglich das Verhaltenssignal B hinsichtlich konsistenter E/A Paare analysiert werden, ist es nicht notwendig diesen Zwischenschritt einzuführen.

5

Anwendung qualitativer Modelle in der Gebäudeautomation

Die Einleitung in Abschnitt 5.1 fasst grundlegende Überlegungen und beachtenswerte Informationen für die qualitative Modellbildung zusammen. Abschnitt 5.2 zeigt den Weg zum qualitativen Modell, beginnend mit der Auswahl der Messdaten und deren Quantisierung über die stochastische qualitative Identifikation hin zur Beobachtung und Fehlererkennung. In Abschnitt 5.3 wird die qualitative Fehlererkennung anhand eines Erhitzers einer RLT-Anlage beschrieben. Ein qualitatives Modell zur Fehlererkennung in Heizkreisen und thermischen Speichern ist in den Abschnitten 5.4 und 5.5 erläutert. Das Kapitel schließt mit einem Zwischenfazit in Abschnitt 5.6.

5.1 Einleitung

Die Vorgehensweise bei der Erstellung qualitativer Modelle zur Fehlererkennung in gebäudetechnischen Anlagen lässt sich in vier Schritten beschreiben:

1. Auswahl des Systems und der Trainingsdaten, die als Eingangs-, Zustands- und Ausgangsgrößen des qualitativen Modells dienen sollen.
2. Festlegung der Partitionierungen der Signalmräume der Systemgrößen für die Quantisierung.
3. Modellbildung mithilfe der Methode der stochastischen qualitativen Identifikation und Erstellung der Verhaltensmatrix des stochastischen Automaten.
4. Anwendung des Beobachteralgorithmus zur Fehlererkennung.

Insbesondere der erste Punkt stellt häufig eine Herausforderung dar, da nicht immer einfach entschieden werden kann, ob die verfügbaren Messdaten einer Anlage den Nominalbetrieb beschreiben. Die Messdaten müssen also zuvor mithilfe von Expertenwissen gesichtet werden, damit sichergestellt ist, dass diese nicht bereits fehlerhafte Betriebszustände enthalten. Wie bei allen Black-Box Modellen führt das Trainieren von Fehlern dazu, dass diese später in der Anwendung vom Modell nicht mehr als solche erkannt werden können. Wie die Erfahrung zeigt, sind Gebäudebetreiber und Facility-Manager oftmals sehr gut über den Betriebszustand ihrer Anlagen informiert und wissen, ob diese innerhalb ihrer Standardparameter arbeiten. Es empfiehlt sich daher, dieses Wissen bei der Auswahl der Trainingsdaten entsprechend zu nutzen.

Bei Punkt zwei, der Festlegung der Partitionierungen, gilt es Folgendes zu beachten: In desto mehr qualitative Werte man die Messwerte der Eingangs-, Ausgangs- und Zustandsgrößen quantisiert, desto präziser wird das qualitative Modell und desto größer wird sein Speicherbedarf. Dabei ist es grundsätzlich nicht zielführend, allzu exakte qualitative Modelle zu generieren, da dies dem qualitativen Modellierungsprinzip widerspricht. Temperaturen oder Volumenströme auf zehn oder mehr qualitative Intervalle abzubilden, sollte daher die Ausnahme darstellen. Die praktische Erfahrung zeigt, dass bei solchen gebäudetypischen Größen meist eine Anzahl von drei bis sechs qualitativen Werten pro Systemgröße als solider Mittelweg zwischen Modellpräzision und Speicherbedarf angesehen werden kann. Es gilt außerdem zu beachten, dass in natürlicher Form vorliegende diskrete

Größen wie Schaltsignale (An/Aus), bereits durch zwei qualitative Werte $\{0, 1\}$ gegeben sind. Diese müssen also nicht quantisiert werden.

Punkt drei betrifft die Modellbildung. Diese läuft für alle beliebigen Anzahlen von Eingangs-, Zustands- und Ausgangsgrößen identisch ab. Die Betrachtung unterschiedlich komplexer Systeme, mit mehr oder weniger Systemgrößen, hat dabei ebenso wenig Einfluss auf die Prozedur der Modellbildung, wie die Anzahl deren qualitativer Werte. Das gleiche gilt auch für den Beobachteralgorithmus und damit den vierten Punkt in obiger Auszählung. Diese Eigenschaften stellen einen erheblichen Vorteil gegenüber einer klassischen White-Box Modellbildung dar, bei der mit jeder zusätzlich zu beschreibenden Systemgröße auch neue Gleichungssysteme aufgestellt werden müssen.

Im Folgenden wird anhand eines einführenden Beispiels die gesamte Prozedur der Modellbildung und Anwendung beschrieben. Aufgrund der bereits in Tabelle 1.1 aufgezeigten Problematik des großen Speicherbedarfs der Modelle, kann dies nur für ein sehr kleines Beispiel mit wenigen Systemgrößen realisiert werden. Mit Messdaten einer gebäudetechnischen Anlage lässt sich ein derart einfaches Beispiel, an dem alle Schritte der oben aufgeführten Punkte sinnvoll beschreibbar sind, kaum realisieren. Messdaten weisen insbesondere in der Gebäudetechnik oftmals viele Datenlücken auf und sind äußeren Einflüssen unterworfen, die zu Rauschen in den Daten führen. Dies stellt grundsätzlich kein Problem dar, allerdings bewegen sich sinnvolle qualitative Modelle gebäudetechnischer Systeme bereits in einer Größenordnung für die sich die grundlegenden Berechnungsschritte nicht mehr exemplarisch zeigen lassen. Aus diesem Grund werden im Folgenden die Daten eines MODELICA[®] Simulationsmodells eines Erhitzers einer RLT-Anlage verwendet, um die notwendigen Schritte zur qualitativen Modellbildung zu erläutern. Es sei an dieser Stelle ausdrücklich erwähnt, dass sich durch die Verwendung von Simulationsdaten anstelle von Messdaten nichts an der Vorgehensweise bei der Modellbildung ändert. Im Anschluss an dieses einführende Beispiel werden qualitative Modelle realer Gebäudesysteme vorgestellt und die Ergebnisse erläutert.

5.2 Der Weg zum qualitativen Modell

Die für dieses Beispiel verwendeten Daten wurden mithilfe eines MODELICA[®] Simulationsmodells erzeugt, welches im Rahmen einer Masterarbeit [110] am Fraunhofer ISE entstand und deren Ziel die Modellierung von RLT-Anlagen für das EU-Projekt CASCADE [21] war. Dabei wurden sowohl die Anlagenkomponenten typischer RLT-Anlagen sowie mögliche Fehler simuliert, wozu die vom Deutschen Zentrum für Luft- und Raumfahrt (DLR) entwickelte MODELICA[®] FAULT TRIGGERING LIBRARY [121] verwendet wurde. Die Existenz einer Modellbibliothek, die eigens dafür entwickelt wurde, das Verhalten technischer Prozesse unter Einwirkung von Fehlern zu beschreiben, zeigt den Stellenwert der Fehlererkennung und Diagnose. Zwar ist die Relevanz von Fehlersimulationen in sicherheitsrelevanten Bereichen wie der Luft- und Raumfahrttechnik höher einzuschätzen als in der Gebäudeautomation, jedoch führt auch hier die zunehmende Digitalisierung und Systemkomplexität zu ähnlichen Problemstellungen. Umso komplexer die Gebäudesysteme werden, desto schwieriger wird es auch, deren Verhalten unter Einwirkung von Fehlern zu verstehen. Dabei bieten Fehlersimulationen auch die Möglichkeit, ein einheitliches Testumfeld für den Vergleich unterschiedlicher Methoden zur Fehlererkennung zu schaffen. Methodenvergleiche anhand simulierter Fehler sind Gegenstand aktueller Forschung im Bereich der Fehlererkennung in der Gebäudeautomation und werden am Fraunhofer ISE untersucht, wobei Ergebnisse diesbezüglich noch ausstehen. Während die Simulation von Nominal- und Fehlerbetrieb im folgenden Beispiel trivial erscheint, kann dies bei komplexeren Systemen eine durchaus sinnvolle Maßnahme sein.

5.2.1 Systembeschreibung und Trainingsdaten

Abbildung 5.1 zeigt eine Prinzipdarstellung des MODELICA[®] Modells des Erhitzers einer RLT-Anlage. Die Luft tritt mit der Temperatur T_{L_E} und dem Volumenstrom \dot{V}_L in den Erhitzer ein und verlässt

diesen mit der Temperatur T_{L_A} und gleichem Volumenstrom. Erhitzt wird die Luft durch Heißwasser, welches mit der Temperatur T_{W_E} und dem Volumenstrom \dot{V}_W in den Erhitzer eintritt und diesen mit der Temperatur T_{W_A} verlässt. Außerdem verfügt der Erhitzer über ein Dreiwegeventil, welches den Wasservolumenstrom regelt und dessen Stellsignal $S_V \in [0, 1]$ reelle Werte zwischen Null und Eins annehmen kann sowie über eine Pumpe mit dem Signal $S_P \in \{0, 1\}$, die entweder An oder Aus sein kann. Zusätzlich wird vom MODELICA® Modell die Außentemperatur T_A berücksichtigt.

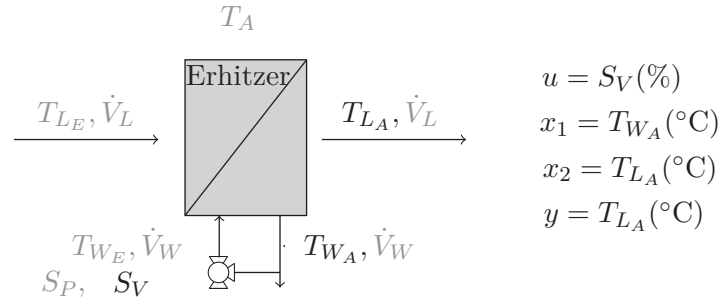


Abbildung 5.1: Erhitzer einer RLT-Anlage mit typischen Systemgrößen. Die zur qualitativen Modellbildung berücksichtigten Größen sind in Schwarz dargestellt.

Ziel ist es nun, ein qualitatives Modell des Erhitzers in Abbildung 5.1 zu erstellen. Als Eingangsgrößen kommen die Wasser- und Luft Eintrittstemperatur sowie deren Volumenströme, die Ventilstellung und das Pumpensignal in Frage. Als Ausgangsgrößen bieten sich die Wasser- und Luftaustrittstemperatur an. Bezüglich der Zustandsgröße(n) des Systems soll die übliche Annahme gelten, dass diese wie bei realen Systemen nicht messbar sind. Wie bereits in der Einleitung erwähnt, werden die Zustandsgrößen jedoch zur qualitativen Modellbildung benötigt, da diese die Dynamik des Systems repräsentieren und darüber entscheiden, wie die Ein- auf die Ausgangsgrößen abgebildet werden. Eine übliche Vorgehensweise in einem solchen Fall ist es, eine oder mehrere der messbaren Ausgangsgrößen des Systems, auch als Zustandsgröße(n) zu betrachten [66, S. 4].

Eingangs-, Zustands- und Ausgangsgrößen des qualitativen Modells

Da es bezüglich der Komplexität des qualitativen Modells und dessen Erläuterung nicht möglich ist, alle theoretisch möglichen Eingangs-, Zustands- und Ausgangsgrößen des Erhitzers zu berücksichtigen, werden nur die in Abbildung 5.1 in der Farbe Schwarz dargestellten Größen zur qualitativen Modellbildung verwendet. Dies spiegelt insofern auch die Realität wieder, als dass in der Praxis nur selten alle theoretisch wünschenswerten Systemgrößen auch als Messdaten verfügbar sind. Für das qualitative Modell wird damit festgelegt, dass die Eingangsgröße durch die Ventilstellung gegeben ist:

$$u = S_V .$$

Alternativ könnte man auch eine der anderen möglichen Eingangsgrößen verwenden. Als Zustandsgrößen des qualitativen Modells werden die in Frage kommenden Ausgangsgrößen, also die Luft- und Wasseraustrittstemperatur gewählt:

$$\begin{aligned} x_1 &= T_{W_A} , \\ x_2 &= T_{L_A} . \end{aligned}$$

Grundsätzlich könnte man nun diese beiden Größen auch als Ausgangsgrößen des qualitativen Modells festlegen. Um die Komplexität des Modells niedrig zu halten, wird hier jedoch nur die Luftaustrittstemperatur als Ausgangsgröße berücksichtigt, sodass

$$y = T_{L_A}$$

ist. Damit sind die Eingangs-, Zustands- und Ausgangsgrößen des qualitativen Modells vereinbart und es kann ein entsprechender Zeitbereich als Trainingsdatensatz für das qualitative Modell festgelegt werden.

Trainingsdaten des qualitativen Modells

In Abbildung 5.2 sind die Signale der ausgewählten Eingangs-, Zustands- und Ausgangsgrößen des Erhitzers für einen Zeitraum von zwei Monaten dargestellt. Die Signale der Eingangsgröße V_S , der Zustandsgrößen T_{W_A} und T_{L_A} und der Ausgangsgröße T_{L_A} wurden dabei mit einer für die Gebäudetechnik üblichen Abtastzeit von 900 s (15-Minutendaten) erzeugt. Es handelt sich also um klassische Zeitreihendaten. Dieser Zeitbereich von zwei Monaten stellt den Nominalbetrieb des Erhitzers dar und dient im Folgenden als Trainingsdatensatz für das qualitative Modell.

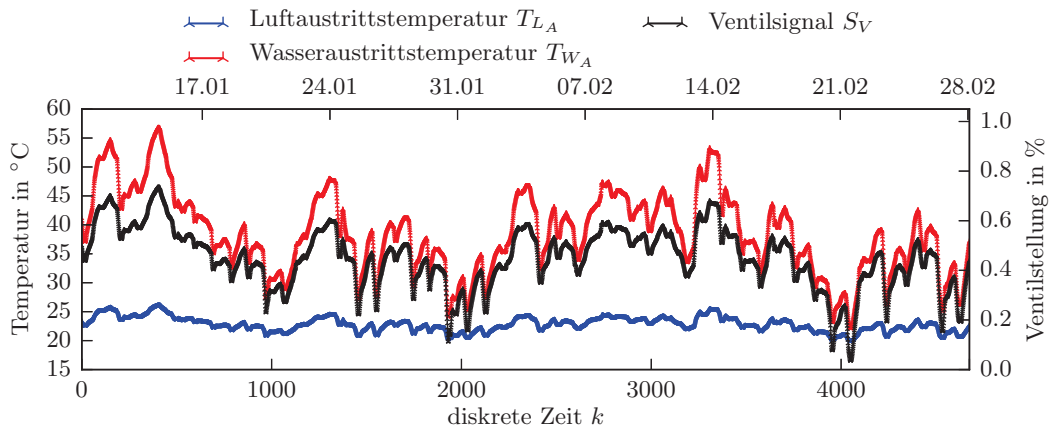


Abbildung 5.2: Eingangs-, Zustands und Ausgangssignale des Erhitzers. Abtastzeit $T_a = 900\text{s}$ (15-Minutendaten).

Da hier nur eine Eingangsgröße $u = V_S$ berücksichtigt wird, sind deren Signalwerte $u(k)$ für alle Zeitschritte $k = 0, 1, 2, \dots, T$ in einer skalaren Eingangsfolge

$$U(0..T) = (u(0), u(1), u(2), \dots, u(T)) \quad (5.1)$$

gegeben. Gleiches gilt für die Ausgangsgröße $y = T_{L_A}$, deren Werte $y(k)$ durch die skalare Ausgangsfolge

$$Y(0..T) = (y(0), y(1), y(2), \dots, y(T)) \quad (5.2)$$

repräsentiert werden. Aufgrund der Tatsache, dass das qualitative Modell über zwei Zustandsgrößen $x_1 = T_{W_A}$ und $x_2 = T_{L_A}$ verfügt, bilden deren Signalwerte $x_1(k)$ und $x_2(k)$ eine vektorielle Zustandsfolge

$$\begin{aligned} \mathbf{X}(0..T) &= (\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)) \\ &= \left(\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}, \begin{pmatrix} x_1(1) \\ x_2(1) \end{pmatrix}, \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix}, \dots, \begin{pmatrix} x_1(T) \\ x_2(T) \end{pmatrix} \right), \end{aligned} \quad (5.3)$$

welche die beiden Zustandsgrößen zusammenfasst. Die reellwertigen Folgen der Eingangsgröße (5.1), der Ausgangsgröße (5.2) und der Zustände (5.3) besitzen eine Länge von $T = 4675$ Werten (s. Abbildung 5.2). Im nächsten Schritt erfolgt nun die Partitionierung und Quantisierung dieser Signale.

5.2.2 Partitionierung der Signlräume und Quantisierung

Die Quantisierung des Eingangs $u(k) \in [0, 1]$, der Zustände $\mathbf{x}(k) \in \mathbb{R}^2$ und des Ausgangs $y(k) \in \mathbb{R}$ erfolgt nun durch die Partitionierung der Signlräume, mit der die Wertebereiche der Signale diskretisiert werden. Für die in Abbildung 5.2 dargestellten Signale der Eingangs-, Zustands- und Ausgangsgrößen wurden dabei die in Tabelle 5.1 dargestellten Partitionierungen gewählt. Diese sollen zunächst als gegeben angenommen werden, um das Prinzip der Quantisierung anschaulich zu erläutern. Welche unterschiedlichen Möglichkeiten es gibt, diese Partitionen anhand der Trainingsdaten festzulegen, wird am Ende dieses Abschnitts auf Seite 77 erläutert.

Tabelle 5.1: Partitionen der Eingangs-, Zustands- und Ausgangsgrößen

$[u], [x], [y]$	$u, (S_V)$	$x_1, (T_{W_A})$	$x_2, (T_{L_A})$	$y, (T_{L_A})$
1	$(-\infty, 0.453]$	$(-\infty, 38.065]$	$(-\infty, 22.281]$	$(-\infty, 22.281]$
2	$(0.453, \infty)$	$(38.065, \infty)$	$(22.281, 23.335]$	$(22.281, 23.335]$
3	–	–	$(23.335, \infty)$	$(23.335, \infty)$

Die zweite Spalte in Tabelle 5.1 zeigt beispielsweise, nach welchem Schema der reellwertigen Eingangsgröße u , der qualitative Wert $[u] = 1$ oder $[u] = 2$ zugewiesen wird: Besitzt der reellwertige Eingang $u(k)$ einen Wert der kleiner oder gleich 0.453 ist, dann wird ihm der qualitative Wert $[u(k)] = 1$ zugewiesen. Ist der Wert der Ventilsignals $u(k)$ hingegen größer als 0.453, dann ist der qualitative Wert durch $[u(k)] = 2$ gegeben. Gleiches gilt analog für die Zustandsgrößen und die Ausgangsgröße, deren Partitionierungen und die zugehörigen qualitativen Werte ebenfalls in der Tabelle dargestellt sind.

Anhand der Abbildung 5.3, welche die Signalverläufe der Eingangs-, Ausgangs- und Zustandsgrößen über die ersten zwölf Zeitschritte ($k = 0, \dots, 11$) zeigt, kann die Quantisierung anschaulich nachvollzogen werden.

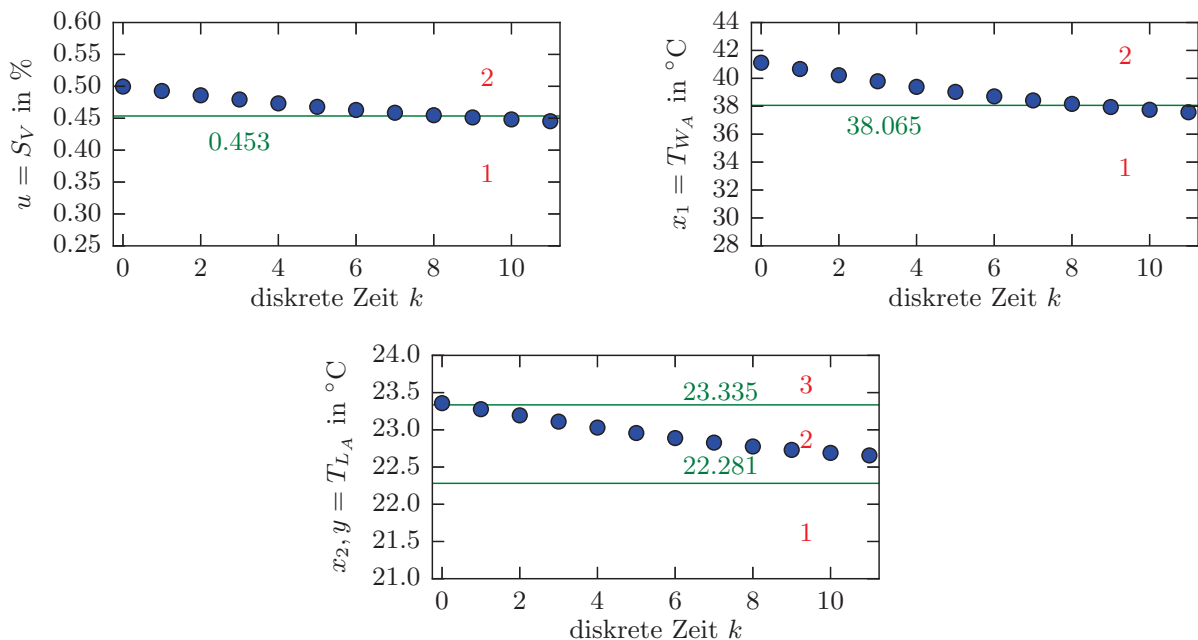


Abbildung 5.3: Die ersten zwölf Zeitschritte der Eingangs-, Zustands- und Ausgangsgrößen des Erhitzers. Die Grenzen der Partitionen sind in Grün dargestellt und deren Nummerierung in Rot. Die Nummer der Partitionierung entspricht den qualitativen Werten der quantisierten Systemgrößen.

Wie Abbildung 5.3 zeigt, sind die jeweiligen Grenzen der Partitionen aus Tabelle 5.1 in Grün eingezeichnet. Die Nummer der jeweiligen Partition ist in Rot dargestellt und entspricht dem qualitativen Wert der quantisierten Systemgröße. Mithilfe der Partitionierungen kann die in Abbildung 5.3 oben links dargestellte reellwertige Eingangsfolge

$$\begin{aligned} U(0...11) &= (u(0), u(1), u(2), \dots, u(11)) \\ &= (0.499, 0.492, 0.486, 0.479, 0.473, 0.467, 0.463, 0.458, 0.454, 0.451, 0.448, 0.445) \end{aligned}$$

auf die quantisierte Eingangsfolge

$$[U(0...11)] = ([u(0)], [u(1)], [u(2)], \dots, [u(11)]) = (2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1) \quad (5.4)$$

abgebildet werden. Zur Zeit $k = 2$ ist der reelle Signalwert des Ventilsignals etwa durch $u(2) = 0.486$ gegeben und liegt damit in der Partition mit der Nummer Zwei. Der quantisierte Signalwert $[u(k)]$ besitzt damit den qualitativen Wert Zwei und es ist $[u(k)] = 2$. Gleiches gilt für die im unteren Teil von Abbildung 5.3 dargestellte Ausgangsfolge

$$\begin{aligned} Y(0...11) &= (y(0), y(1), y(2), \dots, y(11)) \\ &= (23.354, 23.273, 23.189, 23.105, 23.025, 22.949, \\ &\quad 22.879, 22.820, 22.768, 22.724, 22.687, 22.655) , \end{aligned}$$

für die man mit der eingezeichneten Partitionierung die quantisierte Ausgangsfolge

$$[Y(0...11)] = ([y(0)], [y(1)], [y(2)], \dots, [y(11)]) = (3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2) \quad (5.5)$$

erhält. Ebenso wie dies oben für die ersten zwölf Zeitschritte dargestellt ist, verfährt man mit den restlichen Werten der Ein- und Ausgangsfolgen bis zum Zeithorizont $T = 4675$, womit die Quantisierung der Ein- und Ausgangswerte des Erhitzers abgeschlossen ist.

Für die Zustandsgrößen des Erhitzers, die Wasseraustrittstemperatur $x_1 = T_{WA}$ und die Luftaustrittstemperatur $x_2 = T_{LA}$, sind die Partitionierungen ebenfalls in Abbildung 5.3 eingezeichnet. Für die ersten zwölf Zeitschritte $k = 0, 1, \dots, 11$ bilden diese somit die reellwertige vektorielle Zustandsfolge

$$\begin{aligned} \mathbf{X}(0...11) &= (\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(11)) \\ &= \left(\begin{pmatrix} 41.107 \\ 23.354 \end{pmatrix}, \begin{pmatrix} 40.658 \\ 23.273 \end{pmatrix}, \begin{pmatrix} 40.210 \\ 23.189 \end{pmatrix}, \begin{pmatrix} 39.774 \\ 23.105 \end{pmatrix}, \begin{pmatrix} 39.369 \\ 23.025 \end{pmatrix}, \begin{pmatrix} 39.000 \\ 22.949 \end{pmatrix}, \right. \\ &\quad \left. \begin{pmatrix} 38.675 \\ 22.879 \end{pmatrix}, \begin{pmatrix} 38.390 \\ 22.820 \end{pmatrix}, \begin{pmatrix} 38.145 \\ 22.768 \end{pmatrix}, \begin{pmatrix} 37.929 \\ 22.724 \end{pmatrix}, \begin{pmatrix} 37.740 \\ 22.687 \end{pmatrix}, \begin{pmatrix} 37.572 \\ 22.655 \end{pmatrix} \right) . \end{aligned} \quad (5.6)$$

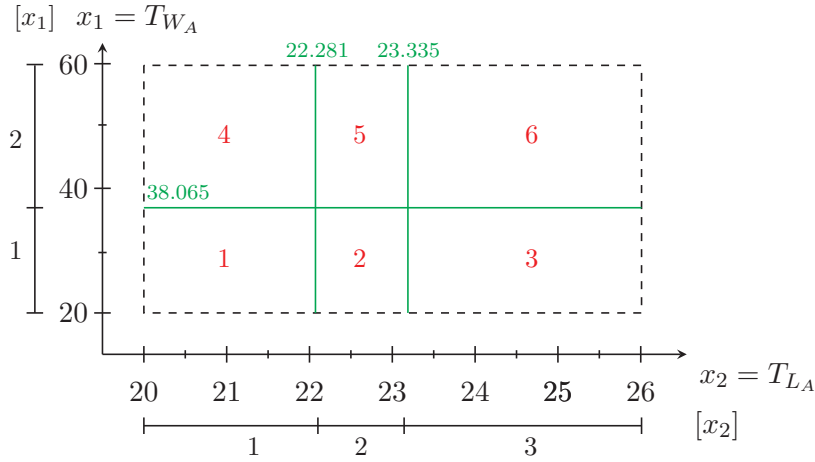
Da der vektorielle Zustand \mathbf{x} zwei Zustandsgrößen x_1 und x_2 repräsentiert, die aber beide auf einen *einzigsten* qualitativen Wert abgebildet werden sollen, benötigt man ein Enumerationsschema für die Quantisierung. Dieses ist in Abbildung 5.4 dargestellt.

Betrachtet man beispielsweise den reellwertigen Zustandsvektor zur Zeit $k = 2$

$$\mathbf{x}(2) = \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix} = \begin{pmatrix} 40.210 \\ 23.189 \end{pmatrix}$$

und trägt diesen in den zweidimensionalen Zustandsraum in Abbildung 5.4 ein, dann ist klar, dass dieser zur Partition mit der Nummer fünf gehört. Der quantisierte Zustandsvektor $[\mathbf{x}(2)]$ besitzt damit zur Zeit $k = 2$ den qualitativen Wert $[\mathbf{x}(2)] = 5$. Mithilfe der Partitionierung des Zustandsraumes erhält man ebenso alle anderen qualitativen Werte der reellwertigen Zustandsfolge (5.6) und damit die quantisierte Zustandsfolge

$$[\mathbf{X}(0...11)] = ([\mathbf{x}(0)], [\mathbf{x}(1)], [\mathbf{x}(2)], \dots, [\mathbf{x}(11)]) = (6, 5, 5, 5, 5, 5, 5, 5, 5, 2, 2, 2) . \quad (5.7)$$

Abbildung 5.4: Partitionierung des zweidimensionalen Zustandsraumes \mathbb{R}^2

Der Vorteil solcher rechtwinkligen Partitionierungen wie in Abbildung 5.4 besteht darin, dass ein direkter Zusammenhang zwischen dem qualitativen Wert des Zustandsvektors und den qualitativen Werten seiner Komponenten besteht. Betrachtet man den Zustandsvektor $\mathbf{x}(2)$, dann kann man aus dessen qualitativem Wert $[\mathbf{x}(2)] = 5$ unmittelbar schließen, dass die einzelnen quantisierten Zustandsgrößen jeweils den qualitativen Wert zwei besitzen: $[x_1(k)] = 2$ und $[x_2(k)] = 2$ (s. Abbildung 5.4). Gleiches gilt auch im Umkehrschluss. An dieser Stelle wird deutlich, dass die Quantisierung von vektoriellen Eingangs, Zustands oder Ausgangsgrößen, die mehr als zwei Größen repräsentieren, deutlich komplexer wird. Repräsentiert beispielsweise der Zustandsvektor nicht nur zwei sondern fünf Zustandsgrößen ($\mathbf{x} \in \mathbb{R}^5$), dann besitzt dieser somit den fünfdimensionalen Zustandsraum \mathbb{R}^5 , der entsprechend partitioniert werden muss. Dies ist grafisch nicht mehr darstellbar und man benötigt eine Gleichung, die den reellwertigen Zustandsvektoren einen eindeutigen und ganzzahligen qualitativen Wert zuweist. Eine solche Gleichung ist in Abschnitt 10.2.1 beschrieben.

Ein Punkt auf den bisher nicht eingegangen wurde, ist, wie die Partitionsgrenzen in Tabelle 5.1 und Abbildung 5.4 anhand der Trainingsdaten festgelegt werden. Dies wird im nächsten Absatz erläutert.

Festlegung der Partitionsgrenzen

Zur Festlegung der Partitionsgrenzen anhand der Trainingsdaten der Eingangs, Ausgangs- und Zustandsgrößen existieren unterschiedliche Möglichkeiten. Um dies zu erläutern wird ein beliebiges zeitdiskretes Signal $S(0 \dots T) = (s(0), s(1), \dots, s(T))$ mit Signalwerten $s(k), k = 0, 1, \dots, T$ betrachtet. Es stellt sich nun die Frage, wie man die Grenzen der Partitionen sinnvoll festlegt, denn von diesen hängt letztlich ab, welche qualitativen Werte den quantisierten Signalwerten $[s(k)]$ zugewiesen werden. Grundsätzlich können die Grenzen der Partitionen völlig frei gewählt, äquidistant eingeteilt oder anhand von Quantilen ermittelt werden. Für eine ganzzahlige Anzahl von $J \geq 2$ Partitionen werden äquidistante Partitionsgrenzen $s_a^{(i)}$ für $i = 1, 2, \dots, J - 1$ mit

$$s_a^{(i)} := s_{\min} + \frac{i}{J} (s_{\max} - s_{\min}) \quad (5.8)$$

$$\text{und } s_{\min} := \min_{k \in \{0, \dots, T\}} s(k), \quad s_{\max} := \max_{k \in \{0, \dots, T\}} s(k) \quad (5.9)$$

berechnet. Für $J = 2$ Partitionen erhält man mit den Gleichungen (5.8) und (5.9) eine einzige Partitionsgrenze $s_a^{(1)}$, die exakt zwischen dem maximalen s_{\max} und minimalen Signalwert s_{\min} liegt und

somit den arithmetischen Mittelwert dieser beiden Größen repräsentiert. Für $J \geq 3$ Partitionen besitzen die Grenzen $s_a^{(i)}$ der Intervalle stets den gleichen Abstand zueinander – sie sind also äquidistant. Bei einer auf Quantilen basierenden Partitionierung wird aus den Elementen $s(0), s(1), \dots, s(T)$ des Signals $S(0 \dots T)$ zunächst die geordnete Stichprobe

$$(s_1, s_2, \dots, s_K) \quad \text{mit} \quad s_1 \leq s_2 \leq \dots \leq s_K \quad (5.10)$$

gebildet, wobei $s_1 = s_{\min}$ und $s_K = s_{\max}$ nach Gleichung (5.9) gilt. Für eine ganzzahlige Anzahl von $J \geq 2$ Partitionen werden die Partitions Grenzen $s_q^{(p)}$ für $p = \frac{1}{J}, \frac{2}{J}, \dots, \frac{J-1}{J}$ auf Basis *empirischer* p -Quantile

$$s_q^{(p)} := \begin{cases} 0.5(s_r + s_{r+1}) & \text{wenn } r = pK \in \mathbb{N}, \\ s_{\lfloor r+1 \rfloor} & \text{wenn } r = pK \notin \mathbb{N}, \end{cases} \quad (5.11)$$

berechnet, welche die Stichprobe derart aufteilen, dass mindestens $(1-p) \cdot 100\%$ der Stichprobenwerte (5.10) größer oder gleich $s_q^{(p)}$ und mindestens $p \cdot 100\%$ der Werte kleiner oder gleich $s_q^{(p)}$ sind [40, S. 32]. Dabei bezeichnet der Ausdruck $\lfloor l \rfloor := \max\{h \in \mathbb{Z} : h \leq l\}$ eine Abrundungsfunktion, die einen reellen Wert $l \in \mathbb{R}$ auf die größte ganze Zahl $h \in \mathbb{Z}$ abrundet, die kleiner oder gleich l ist – wie beispielsweise $\lfloor 3.17 \rfloor = 3$ oder $\lfloor 2.79 \rfloor = 2$ [40, S. 32]. Mit Gleichung (5.11) können die Quantile $s_q^{(p)}$ der Stichprobe (5.10) grundsätzlich auch für beliebige Werte $0 < p < 1$ bestimmt werden. Die Grenzen der Partitionen in Tabelle 5.1 wurden auf Basis von Quantilen erstellt.

Betrachtet man nun beispielsweise das Signal in Abbildung 5.5, dessen Wertebereich in zwei Intervalle oder Partitionen aufgeteilt wird, so wird der Unterschied zwischen einer äquidistanten und einer auf Quantilen basierenden Bestimmung der Partitions Grenze deutlich.

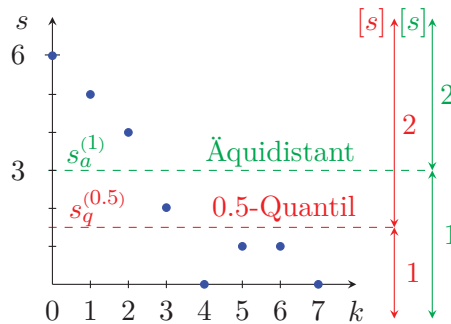


Abbildung 5.5: Äquidistante und auf Quantilen basierende Partitionierung

Versteht man die in Abbildung 5.5 in Blau dargestellten Signalwerte $s(k)$ als Trainingsdaten, dann führt eine äquidistante Partitionierung zu der in grün eingezeichneten Partitions Grenze $s_a^{(1)}$. Diese berechnet man mit den Gleichungen (5.8) und (5.9), wobei die Anzahl der Partitionen $J = 2$ beträgt und der minimale und maximale Signalwert durch $s_{\min} = 0$ und $s_{\max} = 6$ gegeben ist. Damit erhält man die Partitions Grenze mit $s_a^{(1)} = 0 + 1/2(6 - 0) = 3$. Signalwerten $s(k)$, die exakt auf der Partitions Grenze liegen, wird in dieser Arbeit stets der kleinere qualitative Wert zugeordnet. Es wird somit allen Werten $s(k)$, die kleiner oder gleich drei sind und im Intervall $(-\infty, 3]$ liegen, der qualitative Wert $[s(k)] = 1$ zugewiesen und Werten die zum Intervall $(3, \infty)$ gehören und damit größer drei sind, der qualitative Wert $[s(k)] = 2$. Auch wenn die physikalischen Signalwerte $s(k)$ in der Regel nur Werte aus einem begrenzten Wertebereich annehmen können, werden die anhand der Partitions Grenze gebildeten Intervalle somit als links-, bzw. rechtsseitig unbeschränkte Intervalle definiert. Jede weitere Beschränkung würde im Sinne der qualitativen Modellierung formal eine weitere Partition bedeuten, der ein entsprechender qualitativer Wert zugewiesen werden müsste.

Bei einer auf Quantilen basierenden Partitionierung des Signalwertebereichs in $J = 2$ Intervalle, ist die Partitions-grenze durch $p = 1/J = 0.5$ definiert. Dies entspricht dem 0.5-Quantil, beziehungsweise dem Median. Um die Lage der Partitions-grenze zu bestimmen, bildet man aus den Werten $s(k)$ in Abbildung 5.5 die geordnete Stichprobe (5.10), also

$$(s_1 = 0, s_2 = 0, s_3 = 1, s_4 = 1, s_5 = 2, s_6 = 4, s_7 = 5, s_8 = 6) .$$

Dann erhält man mit Gleichung (5.11) für $p = 0.5$ und $K = 8$ Messwerte den Wert $r = 0.5 \cdot 8 = 4 \in \mathbb{N}$ und damit die Partitions-grenze $s_q^{(0.5)} = 0.5(s_4 + s_5) = 0.5(1 + 2) = 1.5$, die in Abbildung 5.5 in Rot eingezeichnet ist. Demzufolge wird allen Signalwerten $s(k)$, die kleiner oder gleich 1.5 und damit im Intervall $(-\infty, 1.5]$ liegen, der qualitative Wert $[s(k)] = 1$ zugewiesen und denen die größer 1.5 sind und sich im Intervall $(1.5, \infty)$ befinden, der qualitative Wert $[s(k)] = 2$.

Wird ein Signal in mehr als nur $J = 2$ zwei Bereiche partitioniert, verfährt man analog zur obigen Beschreibung. Bei einer Einteilung des Signalwertebereichs in beispielsweise $J = 4$ Intervalle, erhält man mit den Gleichungen (5.8) und (5.9) die Partitions-grenzen $s_a^{(1)} = 1.5$, $s_a^{(2)} = 3$, $s_a^{(3)} = 4.5$ und die Intervalle $(-\infty, 1.5]$, $(1.5, 3]$, $(3, 4.5]$ und $(4.5, \infty)$. Bei der Verwendung von Quantilen werden diese für $p = 1/4$, $p = 2/4$ und $p = 3/4$ bestimmt. Es handelt sich somit um das 0.25, das 0.5 und das 0.75-Quantil. Mit Gleichung (5.11) erhält man deren Lage und damit die Partitions-grenzen bei $s_q^{(0.25)} = 0.5$, $s_q^{(0.5)} = 1.5$ und $s_q^{(0.75)} = 4.5$. An dieser Stelle soll abschließend noch darauf hingewiesen werden, dass bei der Verwendung von Quantilen zur Bestimmung der Partitions-grenzen, bei speziellen Verteilungen der Stichprobe, Probleme auftreten können. Für $J = 4$ Partitionen und die geordnete Stichprobe $(s_1 = 0, s_2 = 1, s_3 = 1, s_4 = 2, s_5 = 4)$ fallen beispielsweise das 0.25 und 0.5-Quantil zusammen, sodass $s_q^{(0.25)} = s_q^{(0.5)} = 1$ und $s_q^{(0.75)} = 2$ ist. Es ist also nicht möglich vier Partitionen zu bilden. Abhilfe kann dann die Wahl einer äquidistanten Partitionierung $s_a^{(1)} = 1$, $s_a^{(2)} = 2$, $s_a^{(3)} = 3$ oder die Reduktion auf $J = 3$ Partitionen schaffen, mit der man $s_q^{(0.33)} = 1$ und $s_q^{(0.66)} = 2$ erhält.

Hat man die Partitions-grenzen festgelegt und alle reellwertigen Signalwerte des Erhitzers quantisiert, sodass diese in Form der quantisierten Eingangsfolgen (5.4), Ausgangsfolgen (5.5) und Zustandsfolgen (5.7) für den *gesamten* Trainingsdatensatz vorliegen, dann können diese zur qualitativen Modellbildung verwendet werden.

5.2.3 Qualitative Modellbildung

Die qualitative Modellbildung erfolgt anhand der quantisierten Eingangs-, Ausgangs- und Zustandsfolgen. Da es nicht möglich ist, diesen Prozess für alle Werte des gesamten Trainingsdatensatzes zu zeigen, erfolgt diese Erläuterung anhand der ersten zwölf Werte der quantisierten Eingangs- (5.4), Ausgangs- (5.5) und Zustandsfolgen (5.7).

Im Folgenden ist es zielführend, anstelle von quantisierten Eingängen $[u(k)]$, Zuständen $[\mathbf{x}(k)]$ und Ausgängen $[y(k)]$, auch von den qualitativen Eingängen v , den qualitativen Zuständen z und den qualitativen Ausgängen w zu sprechen. Denn dies sind diejenigen Größen, mit denen das qualitative Modell, beziehungsweise der diesem zuzugliedernde stochastische Automat »arbeitet«. Dabei gilt der Zusammenhang

$$\begin{aligned} [u(k)] &= v \in \mathcal{U} \quad \text{mit} \quad \mathcal{U} := \{1, 2, \dots, M\} , \\ [\mathbf{x}(k)] &= z \in \mathcal{X} \quad \text{mit} \quad \mathcal{X} := \{1, 2, \dots, N\} , \\ [y(k)] &= w \in \mathcal{Y} \quad \text{mit} \quad \mathcal{Y} := \{1, 2, \dots, Q\} . \end{aligned}$$

Der qualitative Eingang v kann wie in Abbildung 5.3 dargestellt, lediglich zwei verschiedene Werte annehmen: den Wert Eins oder Zwei. Der qualitative Eingangsraum ist damit durch die Menge $\mathcal{U} := \{1, 2\}$ gegeben und die Anzahl der qualitativen Eingänge ist $M = 2$. Ebenso kann man Abbildung 5.3 entnehmen, dass der qualitative Ausgang w drei unterschiedliche Werte besitzen kann,

womit der qualitative Ausgangsraum durch die Menge $\mathcal{Y} := \{1, 2, 3\}$ definiert ist. Die Anzahl qualitativer Ausgänge ist $Q = 3$. Der qualitative Zustand z kann, wie aus der Zustandsraumpartitionierung in Abbildung 5.4 hervorgeht, sechs verschiedene Werte annehmen. Der qualitative Zustandsraum ist entsprechend durch die Menge $\mathcal{X} := \{1, \dots, 6\}$ gegeben und die Anzahl qualitativer Zustände ist $N = 6$.

Die Modellbildung erfolgt nun anhand der in Abschnitt 4.3.2 beschriebenen Methode der stochastischen qualitativen Identifikation.

Stochastische qualitative Identifikation

In einem ersten Schritt werden die qualitativen Werte der Eingänge, Ausgänge und Zustände, die durch die quantisierten Eingangsfolgen (5.4), Ausgangsfolgen (5.5) und Zustandsfolgen (5.7) gegeben sind, in die Tabelle 5.2 eingetragen.

Tabelle 5.2: Qualitative Tupel

k	$[\mathbf{x}(k+1)] = \bar{z}$	$[\mathbf{y}(k)] = w$	$[\mathbf{x}(k)] = z$	$[\mathbf{u}(k)] = v$
0	5	3	6	2
1	5	2	5	2
2	5	2	5	2
3	5	2	5	2
4	5	2	5	2
5	5	2	5	2
6	5	2	5	2
7	5	2	5	2
8	2	2	5	2
9	2	2	2	1
10	2	2	2	1

Den Folgezustand $[\mathbf{x}(k+1)] = \bar{z}$ in der ersten Spalte erhält man dadurch, dass man den Zustand $[\mathbf{x}(k)] = z$ um eine Zeile nach oben verschiebt. Die Beobachterzeitschrittdauer beträgt damit $T_b = 1$. Die Tabelle besitzt aufgrund dieser Verschiebung einen Zeitschritt weniger als die Folgen der quantisierten Eingänge, Ausgänge und Zustände. Jede Zeile der Tabelle bildet nun ein Tupel, bestehend aus qualitativem Folgezustand \bar{z} , Ausgang w , Zustand z und Eingang v . Als nächstes kann man die absoluten Häufigkeiten $H(\bar{z}, w, z, v)$ dieser Tupel auszählen. Das Tupel (5, 2, 5, 2) tritt zum Beispiel sieben mal in Tabelle 5.2 auf, womit für dessen absolute Häufigkeit

$$H(5, 2, 5, 2) = 7$$

gilt. Teilt man diesen Wert basierend auf Gleichung (4.24) durch die Gesamtzahl aller Tupel, die hier $T = 11$ beträgt, erhält man die relative Häufigkeit

$$\hat{h}(5, 2, 5, 2) = \frac{H(5, 2, 5, 2)}{T} = 7/11.$$

Ebenso verfährt man für alle anderen Tupel in Tabelle 5.2, deren absolute und relative Häufigkeiten in der Tabelle 5.3 zusammengefasst sind.

Auf Basis der relativen Häufigkeiten $\hat{h}(\bar{z}, w, z, v)$ in Tabelle 5.3 kann nun die Randverteilung $\hat{h}(z, v)$ nach Gleichung (4.26) berechnet werden. Diese erhält man, indem man die relativen Häufigkeiten aller Tupel mit identischem qualitativen Zustand z und gleichem qualitativen Eingang v aufsummiert.

Tabelle 5.3: Absolute Häufigkeiten $H(\bar{z}, w, z, v)$ und relative Häufigkeiten $\hat{h}(\bar{z}, w, z, v)$

\bar{z}	w	z	v	$H(\bar{z}, w, z, v)$	$\hat{h}(\bar{z}, w, z, v)$
5	3	6	2	1	$1/11$
5	2	5	2	7	$7/11$
2	2	5	2	1	$1/11$
2	2	2	1	2	$2/11$

Für den qualitativen Zustand $z = 6$ und den qualitativen Eingang $v = 2$ erhält man somit die Randverteilung

$$\hat{h}(6, 2) = \sum_{\bar{z}=1}^6 \sum_{w=1}^3 \hat{h}(\bar{z}, w, 6, 2) = \hat{h}(5, 3, 6, 2) = 1/11 .$$

Der qualitative Zustand $z = 2$ und der qualitative Eingang $v = 2$ führen zur Randverteilung

$$\begin{aligned} \hat{h}(5, 2) &= \sum_{\bar{z}=1}^6 \sum_{w=1}^3 \hat{h}(\bar{z}, w, 5, 2) \\ &= \hat{h}(5, 2, 5, 2) + \hat{h}(2, 2, 5, 2) = 7/11 + 1/11 = 8/11 . \end{aligned}$$

Für $z = 2$ und $v = 1$ ergibt sich

$$\hat{h}(2, 1) = 2/11 .$$

Damit kann Tabelle 5.3 um die berechneten Randverteilungen erweitert werden, was in Tabelle 5.4 dargestellt ist.

Tabelle 5.4: Absolute Häufigkeiten $H(\bar{z}, w, z, v)$, relative Häufigkeiten $\hat{h}(\bar{z}, w, z, v)$, Randverteilung $\hat{h}(z, v)$ und bedingte Wahrscheinlichkeit $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$

\bar{z}	w	z	v	$H(\bar{z}, w, z, v)$	$\hat{h}(\bar{z}, w, z, v)$	$\hat{h}(z, v)$	$\tilde{\mathcal{L}}(\bar{z}, w z, v)$
5	3	6	2	1	$1/11$	$1/11$	1
5	2	5	2	7	$7/11$	$8/11$	$7/8$
2	2	5	2	1	$1/11$	$8/11$	$1/8$
2	2	2	1	2	$2/11$	$2/11$	1

Für Randverteilungen $\hat{h}(z, v) \neq 0$ erhält man nach Gleichung (4.27) die gesuchte Approximation der bedingten Übergangswahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$, die ebenfalls in Tabelle 5.4 eingetragen sind. Die bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$ ergeben sich dadurch, dass man die relativen Häufigkeiten $\hat{h}(\bar{z}, w, z, v)$ durch die entsprechende Randverteilung $\hat{h}(z, v)$ dividiert. Für die dritte Zeile in Tabelle 5.4 ergibt sich beispielsweise die bedingte Wahrscheinlichkeit

$$\tilde{\mathcal{L}}(2, 2 | 5, 2) = \frac{\hat{h}(2, 2, 5, 2)}{\hat{h}(5, 2)} = \frac{1/11}{8/11} = 1/8 . \quad (5.12)$$

Dies ist somit die Wahrscheinlichkeit dafür, dass das System in den qualitativen Folgezustand $\bar{z} = 2$ wechselt und den qualitativen Ausgang $w = 2$ ausgibt, unter der Bedingung, dass es sich im qualitativen Zustand $z = 5$ befindet und den qualitativen Eingang $v = 2$ erhält. Analog zu Gleichung (5.12)

berechnet man alle weiteren bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$ die in der Tabelle 5.4 abgebildet sind. Damit ist das qualitative Modell des Erhitzers durch den stochastischen Automaten

$$\tilde{\mathcal{A}} = (\mathcal{X}, \mathcal{U}, \mathcal{Y}, \tilde{\mathcal{L}}, \tilde{\mathbf{p}}_0) \quad (5.13)$$

definiert. Dabei dient die Menge der qualitativen Eingänge \mathcal{U} als Eingabealphabet des Automaten und die qualitative Ausgangsmenge \mathcal{Y} als Ausgabealphabet. Die qualitative Zustandsmenge \mathcal{X} repräsentiert ebenso die Zustandsmenge des Automaten. Die Approximation der Verhaltensrelation $\tilde{\mathcal{L}}$ des stochastischen Automaten ist entsprechend durch die letzte Spalte der Tabelle 5.4 gegeben. Der initiale Zustandsvektor $\mathbf{p}_0 \in [0, 1]^6$ des Automaten hat die Länge $N = 6$, da das qualitative Modell des Erhitzers sechs Zustände besitzt.

Verhaltensmatrix

Um nun das qualitative Modell (5.13) zur Beobachtung und Fehlererkennung nutzen zu können, bildet man zunächst die Verhaltensmatrix aus den Gleichungen (4.17) und (4.18). Da das qualitative Erhitzermodells $M=2$ Eingänge, $N=6$ Zustände und $Q=3$ Ausgänge besitzt, ergibt sich eine Verhaltensmatrix $\tilde{\mathbf{L}} \in [0, 1]^{NR \times NM}$ der Dimension

$$\tilde{\mathbf{L}} = \begin{pmatrix} \tilde{\mathbf{L}}(1, 1) & \tilde{\mathbf{L}}(1, 2) \\ \tilde{\mathbf{L}}(2, 1) & \tilde{\mathbf{L}}(2, 2) \\ \tilde{\mathbf{L}}(3, 1) & \tilde{\mathbf{L}}(3, 2) \end{pmatrix} \in [0, 1]^{6 \cdot 3 \times 6 \cdot 2}, \quad (5.14)$$

die sich aus den Blöcken $\tilde{\mathbf{L}}(w, v) \in [0, 1]^{6 \times 6}$ zusammensetzt. Jede Matrix $\tilde{\mathbf{L}}(w, v)$ besitzt dabei die Form

$$\mathbf{L}(w, v) = \begin{pmatrix} \tilde{\mathcal{L}}(1, w | 1, v) & \tilde{\mathcal{L}}(1, w | 2, v) & \cdots & \tilde{\mathcal{L}}(1, w | 6, v) \\ \tilde{\mathcal{L}}(2, w | 1, v) & \tilde{\mathcal{L}}(2, w | 2, v) & \cdots & \tilde{\mathcal{L}}(2, w | 6, v) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathcal{L}}(6, w | 1, v) & \tilde{\mathcal{L}}(6, w | 2, v) & \cdots & \tilde{\mathcal{L}}(6, w | 6, v) \end{pmatrix} \in [0, 1]^{6 \times 6}$$

und die Elemente repräsentieren die bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$ aus Tabelle 5.4. Dies lässt sich nun einfach veranschaulichen. In der ersten Zeile der Tabelle 5.4 gilt für den qualitativen Folgezustand $\bar{z} = 5$, der qualitative Ausgang besitzt den Wert $w = 3$, der Zustand ist $z = 6$ und der Eingang ist $v = 1$. Die bedingte Wahrscheinlichkeit dieser Kombination ist hiernach

$$\tilde{\mathcal{L}}(5, 3 | 6, 1) = 1.$$

Da die Werte der qualitativen Folgezustände \bar{z} , Ausgänge w , Zustände z und Eingänge v den Indizes der Verhaltensmatrix entsprechen, steht diese Wahrscheinlichkeit in der Matrix $\tilde{\mathbf{L}}(3, 1)$. Die Position innerhalb dieser Matrix ist durch den Folgezustand $\bar{z} = 5$, der die Zeilennummer angibt und den Zustand $z = 6$, der die Spaltennummer angibt, bestimmt. Die Wahrscheinlichkeit $\tilde{\mathcal{L}}(5, 3 | 6, 1) = 1$ steht somit in der fünften Zeile und der sechsten Spalte der Matrix $\tilde{\mathbf{L}}(3, 1)$, was in der Verhaltensmatrix in Gleichung (5.15) in der Farbe Rot hervorgehoben ist. Für die zweite Zeile der Tabelle 5.4 erhält man

$$\tilde{\mathcal{L}}(5, 2 | 5, 2) = 7/8$$

Diese Wahrscheinlichkeit steht in der Matrix $\tilde{\mathbf{L}}(2, 2)$ in der fünften Zeile und der fünften Spalte. Die weiteren bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$ aus Tabelle 5.4 sind ebenfalls in Rot in die

entsprechenden Blöcke der Verhaltensmatrix eingetragen.

$$\begin{aligned}
 \tilde{\mathbf{L}}(1,1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(1,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
 \tilde{\mathbf{L}}(2,1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(2,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1/8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{7/8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
 \tilde{\mathbf{L}}(3,1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(3,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
 \end{aligned} \tag{5.15}$$

Die in der Matrix (5.15) dargestellten Blöcke der Verhaltensmatrix (5.14) enthalten dabei nur diejenigen Wahrscheinlichkeiten, die anhand der ersten elf Zeitschritte des Trainingsdatensatzes des Erhitzers ermittelt wurden. Die gesamte Verhaltensmatrix, für den kompletten Trainingsdatensatz von $T = 4675$ Werten, ist in Gleichung (5.16) dargestellt.

$$\begin{aligned}
 \tilde{\mathbf{L}}(1,1) &= \begin{pmatrix} 0.987 & 0 & 0 & 0 & 0 & 0 \\ 0.013 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(1,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
 \tilde{\mathbf{L}}(2,1) &= \begin{pmatrix} 0 & 0.027 & 0 & 0 & 0 & 0 \\ 0 & 0.954 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.019 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(2,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.021 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.960 & 0 \\ 0 & 0 & 0 & 0 & 0.019 & 0 \end{pmatrix}, \\
 \tilde{\mathbf{L}}(3,1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(3,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0.99 \end{pmatrix}.
 \end{aligned} \tag{5.16}$$

An dieser Verhaltensmatrix (5.16) lassen sich einige grundlegende Eigenschaften ablesen:

1. Wie man erkennen kann, sind die Matrizen nur äußerst spärlich besetzt. Bei einem Speicherbedarf von

$$\gamma = N \cdot M \cdot N \cdot Q = 6 \cdot 2 \cdot 6 \cdot 3 = 216 \tag{5.17}$$

Werten, sind nur zwölf von diesen ungleich null.

2. Die Wahrscheinlichkeit dafür, dass das System innerhalb eines Zeitschrittes im gleichen Zustand verbleibt ist sehr hoch. In der Matrix $\tilde{\mathbf{L}}(3, 2)$ steht beispielsweise unten rechts die bedingte Wahrscheinlichkeit $\tilde{\mathcal{L}}(6, 3 | 6, 2) = 0.99$. Dies ist die Wahrscheinlichkeit dafür, dass wenn das System den qualitativen Eingang $v = 2$ erhält und sich im Zustand $z = 6$ befindet, es den Ausgang $w = 3$ erzeugt und den Folgezustand $\bar{z} = 1$ annimmt. Es verbleibt in diesem Fall mit 99 %-iger Wahrscheinlichkeit im gleichen Zustand. Diese Eigenschaft des Systems lässt sich einfach daran erkennen, indem man die Wahrscheinlichkeiten auf der Diagonalen der Matrizen betrachtet. Da die Matrizen quadratisch sind, gilt für die Indizes hier stets $\bar{z} = z$. Große Werte auf der Diagonalen bedeuten somit immer eine hohe Wahrscheinlichkeit dafür, dass das System während eines Zeitschrittes im gleichen Zustand verbleibt. Damit geht einher, dass oftmals nur wenige andere Zustandsübergänge überhaupt möglich sind und dass diese mit nur sehr niedrigen Wahrscheinlichkeiten bewertet sind. Dies ist an der Verhaltensmatrix (5.16) gut zu erkennen. Grundsätzlich ist eine Verhaltensmatrix mit einer solchen Struktur nicht wünschenswert, da diese das Verhalten des Systems nur wenig »qualitativ« und fast schon deterministisch beschreibt. Diese Eigenschaft der Verhaltensmatrix hängt mit der gewählten Beobachterzeitschrittdauer von $T_b = 1$ zusammen und soll hier nächst hingenommen werden. In Abschnitt 5.2.6 wird gezeigt, wie die Wahl einer größeren Beobachterzeitschrittdauer dieses Problem verringern kann.
3. An der Struktur der Verhaltensmatrix (5.16) ist unmittelbar ersichtlich, dass das qualitative Modell strukturell unvollständig ist, denn die Bedingungen (3.36) und (3.37) sind offensichtlich nicht erfüllt. Bei einem strukturell vollständigen Modell müssten die Spaltensummen über jeweils alle drei untereinander stehenden Matrizen Eins ergeben. Dies ist beispielsweise für die Matrizen $\tilde{\mathbf{L}}(1, 1)$, $\tilde{\mathbf{L}}(2, 1)$ und $\tilde{\mathbf{L}}(3, 1)$ nur für die Spalten Eins, Zwei und Fünf gegeben. Dies bedeutet, dass das quantisierte System, unabhängig vom Wert der qualitativen Ein- und Ausgänge, über den gesamten Zeitbereich des Trainingsdatensatzes nie im qualitativen Zustand $z = 3$ und $z = 4$ war. Der reellwertige Zustandsvektor $\mathbf{x}(k)$ lag somit nie in den in Abbildung 5.4 dargestellten Partitionen drei und vier. Diese Tatsache lässt sich häufig nicht verhindern. Man kann jedoch versuchen, durch Variation der Partitions Grenzen oder deren Anzahl, dem entgegenzuwirken. Es sei an dieser Stelle darauf hingewiesen, dass selbst strukturell vollständige qualitative Modelle nicht notwendigerweise vollständig im Sinne von Gleichung (4.21) sind. Vollständige qualitative Modelle wird man in der Regel jedoch nur durch Abstraktion erhalten (vgl. Abschnitt 4.3.2). Alternativ könnte man auch das hier verwendete MODELICA[®] Modell nutzen und mit diesem eine große Anzahl von Simulation mit Variationen der Eingangsgrößen durchführen, um möglichst alle dem System möglichen Verhaltensformen abzubilden. Dies ist jedoch in der Realität und den Gebäudemessdaten nicht möglich und man wird in der Regel nur unvollständige Modelle erhalten, die stets nur das Eingangs-Ausgangs-Verhalten repräsentieren können, welches durch die verfügbaren Trainingsdaten beschrieben ist. Von Seiten der Theorie aus betrachtet, ist die Unvollständigkeit des Modells zwar unangenehm, jedoch wird man auch mit keinem anderen Black-Box Modell das vollständige Eingangs-Ausgangs-Verhalten eines System beschreiben können, solange nicht die entsprechenden Trainingsdaten zur Verfügung stehen. Das unvollständige qualitative Modelle in der Praxis dennoch zur Fehlererkennung eingesetzt werden können, wird im nächsten Absatz beschrieben.

Abschließend zur Modellbildung sind in Tabelle 5.5 die grundlegenden Modelleigenschaften zusammengefasst.

5.2.4 Qualitative Beobachtung

Dieser Abschnitt zeigt anhand einfach nachvollziehbarer Berechnungen die Realisierung der qualitativen Beobachtung, basierend auf der Beobachtergleichung (4.37). Ziel der qualitativen Beobachtung ist

Tabelle 5.5: Modellparameter

System	Erhitzer einer RLT-Anlage
Abtastzeit der Daten	$T_a = 900 \text{ s}$ (15-Minutendaten)
Eingangsgrößen Anzahl	$u = S_V$ $m = 1$
Zustandsgrößen Anzahl	$x_1 = T_{WA}, x_2 = T_{LA}$ $n = 2$
Ausgangsgrößen Anzahl	$y = T_{LA}$ $q = 1$
Anzahl qual. Eingänge	$M = 2$
Anzahl qual. Zustände	$N = 6$ ($N_1 = 2, N_2 = 3$)
Anzahl qual. Ausgänge	$Q = 3$
Speicherbedarf	$\gamma = 6 \cdot 3 \cdot 6 \cdot 2 = 216$ Werte
Trainingsdaten	2 Monate
Beobachterzeitschrittdauer	$T_b = 1, (T_b = 3, 6 \text{ in Abschnitt 5.2.6})$

es, anhand der quantisierten Eingangsfolgen $[U(0...T)]$ und Ausgangsfolgen $[Y(0...T)]$ des Erhitzers, die qualitativen Zustandswahrscheinlichkeiten

$$\Pr([\mathbf{x}(k+1)] = \bar{z} \mid [U(0...k)], [Y(0...k)]) \quad (5.18)$$

für alle sechs Folgezustände $\bar{z} \in \mathcal{X} := \{1, 2, \dots, 6\}$ und alle Zeiten $k = 0, 1, \dots, T$ zu bestimmen. Das Konzept der qualitativen Beobachtung des Erhitzers ist in Abbildung 5.6 visualisiert.

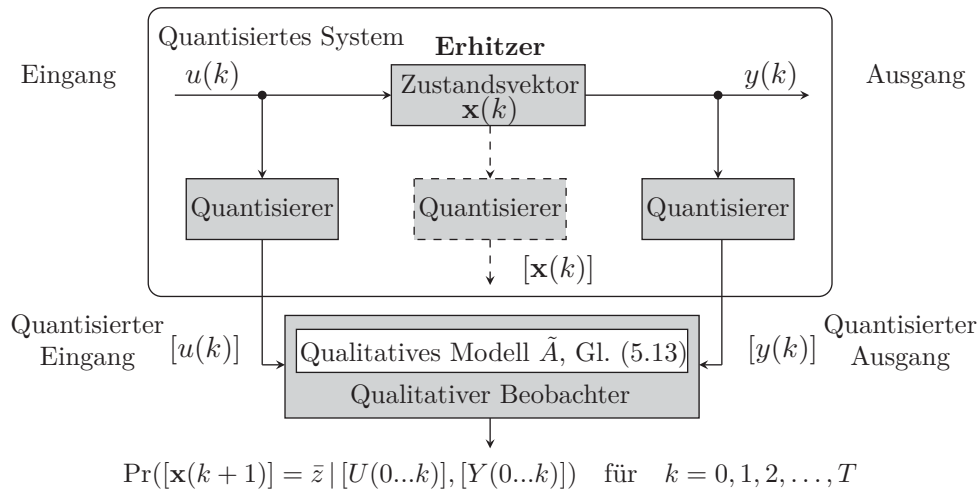


Abbildung 5.6: Qualitative Beobachtung des Erhitzers

Um die Berechnung der gesuchten Zustandswahrscheinlichkeiten zu beschreiben, werden die quantisierten Eingangsfolgen (5.4) und Ausgangsfolgen (5.5) des Erhitzers für zehn Zeitschritte, also für einen Zeithorizont von $T = 9$ betrachtet:

$$\begin{aligned} [U(0...9)] &= (2, 2, 2, 2, 2, 2, 2, 2, 2, 1) , \\ [Y(0...9)] &= (3, 2, 2, 2, 2, 2, 2, 2, 2, 2) . \end{aligned} \quad (5.19)$$

Mit den quantisierten Ein- und Ausgangsfolgen (5.19) sind die qualitativen Eingänge $v(k)$ und Ausgänge $w(k)$ des qualitativen Erhitzermodells für $k = 0, 1, \dots, 9$ durch die Beziehung $[u(k)] = v(k)$

und $[y(k)] = w(k)$ gegeben. Bevor nun die Beobachtung realisiert werden kann, muss noch der initiale Startvektor $\tilde{\mathbf{p}}_0 \in [0, 1]^N$ definiert werden. Dabei soll eine Gleichverteilung über die $N = 6$ qualitativen Zustände angenommen werden:

$$\tilde{\mathbf{p}}_0 = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix} \in [0, 1]^6 .$$

Mit der Startbedingung $\tilde{\mathbf{p}}_z(0) = \tilde{\mathbf{p}}_0$, der Verhaltensmatrix (5.15) und den quantisierten Folgen der Ein- und Ausgänge (5.19) kann die Beobachtergleichung

$$\tilde{\mathbf{p}}_z(k+1) = \frac{\tilde{\mathbf{L}}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)}{\|\tilde{\mathbf{L}}(w(k), v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1} \in [0, 1]^N$$

aus Definition 4.3 initialisiert werden.

Beobachtung zum Zeitpunkt $k = 0$

Zur Zeit $k = 0$ berechnet man mit dem qualitativen Eingang $v(0) = 3$ und dem qualitativen Ausgang $w(0) = 2$ den Zähler der Beobachtergleichung mit dem Matrix-Vektor Produkt

$$\tilde{\mathbf{L}}(3, 2) \cdot \tilde{\mathbf{p}}_z(0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0.99 \end{pmatrix} \cdot \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.001\bar{6} \\ 0.16\bar{5} \end{pmatrix} . \quad (5.20)$$

Für den gegebenen Eingang $v(0) = 3$ und Ausgang $w(0) = 2$ wird somit die Matrix $\tilde{\mathbf{L}}(3, 2)$ der Verhaltensmatrix (5.16) selektiert und mit dem Startvektor multipliziert. Der Nenner der Beobachtergleichung ist durch die Summennorm

$$\|\tilde{\mathbf{L}}(3, 2) \cdot \tilde{\mathbf{p}}_z(0)\|_1 = 0.16\bar{6} \quad (5.21)$$

gegeben. Diesen erhält man durch einfaches Aufsummieren aller Werte des Ergebnisvektors in (5.20). Anschließend ergibt sich der Vektor der qualitativen Zustandswahrscheinlichkeiten zur Zeit $k = 1$

$$\tilde{\mathbf{p}}_z(1) = \frac{\tilde{\mathbf{L}}(3, 2) \cdot \tilde{\mathbf{p}}_z(0)}{\|\tilde{\mathbf{L}}(3, 2) \cdot \tilde{\mathbf{p}}_z(0)\|_1} = \frac{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.001\bar{6} \\ 0.16\bar{5} \end{pmatrix}}{0.16\bar{6}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.01 \\ 0.99 \end{pmatrix} . \quad (5.22)$$

Den für die spätere Fehlererkennung und zur Bildung des Verhaltenssignals \tilde{B} aus Gleichung (4.43) wichtigen Wert $b(k)$ hat man damit bereits in Gleichung (5.21) bestimmt, denn es ist

$$\tilde{b}(0) = \|\tilde{\mathbf{L}}(3, 2) \cdot \tilde{\mathbf{p}}_z(0)\|_1 = 0.16\bar{6} .$$

Beobachtung zum Zeitpunkt $k = 1$

Zur Zeit $k = 1$ erhält man mit dem qualitativen Eingang $v(1) = 2$ und Ausgang $w(1) = 2$ sowie dem zuvor bestimmten Zustandswahrscheinlichkeitsvektor (5.22), für den Zähler der Beobachtergleichung

$$\tilde{\mathbf{L}}(2, 2) \cdot \tilde{\mathbf{p}}_z(1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.021 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.960 & 0 \\ 0 & 0 & 0 & 0 & 0.019 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.01 \\ 0.99 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.00021 \\ 0 \\ 0 \\ 0.0096 \\ 0.00019 \end{pmatrix}$$

und für den Nenner

$$\|\tilde{\mathbf{L}}(2, 2) \cdot \tilde{\mathbf{p}}_z(1)\|_1 = 0.01. \quad (5.23)$$

Damit ergibt sich der Vektor der Zustandswahrscheinlichkeiten zur Zeit $k = 2$ mit

$$\tilde{\mathbf{p}}_z(2) = \frac{\tilde{\mathbf{L}}(2, 2) \cdot \tilde{\mathbf{p}}_z(1)}{\|\tilde{\mathbf{L}}(2, 2) \cdot \tilde{\mathbf{p}}_z(1)\|_1} = \frac{\begin{pmatrix} 0 \\ 0.00021 \\ 0 \\ 0 \\ 0.0096 \\ 0.00019 \end{pmatrix}}{0.01} = \begin{pmatrix} 0 \\ 0.021 \\ 0 \\ 0 \\ 0.96 \\ 0.019 \end{pmatrix}.$$

Aus dem bereits berechneten Wert des Nenners (5.23) folgt wieder

$$\tilde{b}(1) = \|\tilde{\mathbf{L}}_{22} \cdot \tilde{\mathbf{p}}_z(1)\|_1 = 0.01.$$

So wie für diese ersten zwei Zeitschritte der Folgen (5.19) beschrieben, verfährt man auch für alle weiteren Zeitschritte. Die Ergebnisse der qualitativen Beobachtung sind dabei in der Tabelle 5.6 dargestellt.

Tabelle 5.6: Qualitative Zustandswahrscheinlichkeiten $\Pr([\mathbf{x}(k)] = z \mid [U(0 \dots k-1)], [Y(0 \dots k-1)])^1$

z	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$
1	0.166	0	0	0	0	0	0	0	0	0	0.001
2	0.166	0	0.021	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.999
3	0.166	0	0	0	0	0	0	0	0	0	0
4	0.166	0	0	0	0	0	0	0	0	0	0
5	0.166	0.01	0.96	0.961	0.961	0.961	0.961	0.961	0.961	0.961	0
6	0.166	0.99	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0
$\tilde{b}(k)$	0.166	0.01	0.981	0.981	0.981	0.981	0.981	0.981	0.981	0.981	-

Die Tabelle 5.6 enthält damit die gesuchten qualitativen Zustandswahrscheinlichkeiten aus Gleichung (5.18) für die Zeiten $k = 0, \dots, 10$. Es sei hier nochmals erwähnt, dass es sich dabei um

¹Zum Zeitpunkt $k = 0$ sind die qualitativen Zustandswahrscheinlichkeiten durch die initiale Startverteilung $\tilde{\mathbf{p}}_0$ definiert, sodass $\Pr([\mathbf{x}(0)] = z \mid [u(-1)], [y(-1)]) = \Pr([\mathbf{x}(0)] = z)$, $\forall z$ gilt. Siehe hierzu Gleichung (4.19) und Seite 66.

bedingte Wahrscheinlichkeiten handelt. Den beispielsweise zur Zeit $k = 2$ gegebenen Wahrscheinlichkeiten liegt die Verteilung

$$\mathbf{p}_z(2) = \begin{pmatrix} \Pr([\mathbf{x}(2)] = 1 \mid [U(0...1)] = (2, 2), [Y(0...1)] = (3, 2)) \\ \Pr([\mathbf{x}(2)] = 2 \mid [U(0...1)] = (2, 2), [Y(0...1)] = (3, 2)) \\ \vdots \\ \Pr([\mathbf{x}(2)] = 6 \mid [U(0...1)] = (2, 2), [Y(0...1)] = (3, 2)) \end{pmatrix}$$

zugrunde. Es handelt sich somit um die Zustandswahrscheinlichkeiten zur Zeit $k = 2$, jedoch unter der Bedingung, dass bis zur Zeit $k = 1$ die quantisierte Eingangsfolge $[U(0...1)] = (2, 2)$ und Ausgangsfolge $[Y(0...1)] = (3, 2)$ durchlaufen wurde. Dies gilt entsprechend für alle Zeitschritte.

Deutlich anschaulicher als in Form einer Tabelle ist die Visualisierung der Zustandswahrscheinlichkeiten. Abbildung 5.7 zeigt die Zustandswahrscheinlichkeiten umgerechnet auf die beiden Zustandsgrößen des Erhitzers $x_1 = T_{WA}$ und $x_2 = T_{LA}$. Die blauen Punkte entsprechen den reellen Signalwerten der beiden Zustandsgrößen aus Abbildung 5.3. In Grün sind wieder die Grenzen der Partitionen aus Tabelle 5.1 eingezeichnet, während die roten Ziffern den qualitativen Zuständen entsprechen. Die im Hintergrund dargestellten Graustufen repräsentieren die Wahrscheinlichkeiten der qualitativen Zustände der Wasser- und Luftaustrittstemperatur des Erhitzers. Die jeweils rechts der Abbildungen dargestellte Skala gibt an, durch welche Farbe die Wahrscheinlichkeiten der qualitativen Zustände repräsentiert werden. Schwarz bedeutet eine Wahrscheinlichkeit von Eins und Weiß eine Wahrscheinlichkeit von Null.

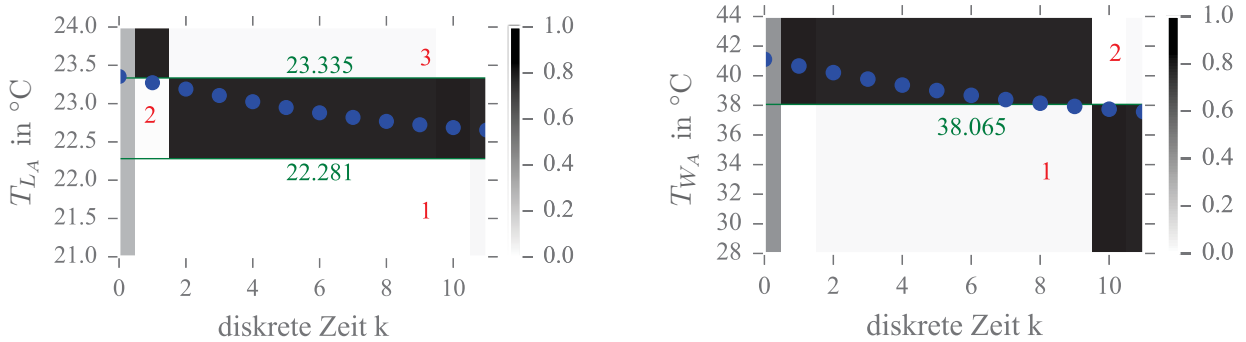


Abbildung 5.7: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} und der Wasseraustrittstemperatur T_{WA}

Um die Zustandswahrscheinlichkeiten des Erhitzers aus Tabelle 5.6 wie in Abbildung 5.7 für beide Zustände darstellen zu können, müssen die Wahrscheinlichkeiten

$$\Pr([x_1(k)] = z_1 \mid [U(0...k-1)], [Y(0...k-1)])$$

und

$$\Pr([x_2(k)] = z_2 \mid [U(0...k-1)], [Y(0...k-1)])$$

für $k = 0, 1, \dots, 10$ bestimmt werden. Dies sind die bedingten Wahrscheinlichkeiten dafür, dass die quantisierten Zustandsgrößen $[x_1(k)]$ und $[x_2(k)]$ die qualitativen Zustände z_1 und z_2 annehmen. Diese Wahrscheinlichkeiten erhält man, indem man die gemeinsamen Zustandswahrscheinlichkeiten aus Tabelle 5.6 entlang der Achsen des Enumerationsschemas der Zustandsraumpartitionierung in Abbildung (5.4) aufsummiert. So erhält man beispielsweise die Wahrscheinlichkeit dafür, dass die Luftaustrittstemperatur zur Zeit $k = 2$ den qualitativen Wert $[x_2(2)] = 2$ annimmt, indem man in der Spalte $k = 2$ in Tabelle 5.6 die Wahrscheinlichkeiten in der zweiten und fünften Zeile aufsummiert. Die Ergebnisse dieser Berechnungen sind in den Tabellen 5.7 und 5.8 dargestellt. Diese enthalten somit die qualitativen Zustandswahrscheinlichkeiten, die in Abbildung 5.7 im Hintergrund abgebildet sind.

Tabelle 5.7: Qualitative Zustandswahrscheinlichkeiten $\Pr([x_1(k)] = z_1 | [U(0 \dots k-1)], [Y(0 \dots k-1)])$

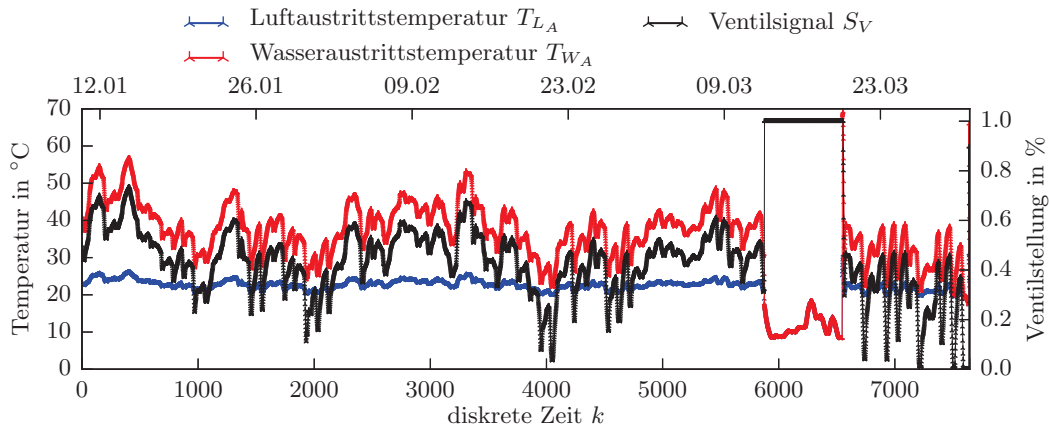
z_1	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
1	0.5	0	0.021	0.02	0.02	0.02	0.02	0.02	0.02	0.02	1
2	0.5	1	0.979	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0

Tabelle 5.8: Qualitative Zustandswahrscheinlichkeiten $\Pr([x_2(k)] = z_2 | [U(0 \dots k-1)], [Y(0 \dots k-1)])$

z_2	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$
1	0.333	0	0	0	0	0	0	0	0	0	0
2	0.333	0.01	0.981	0.981	0.981	0.981	0.981	0.981	0.981	0.981	1
3	0.333	0.99	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0.019	0

5.2.5 Qualitative Fehlererkennung

Um die Fehlererkennung mit qualitativen Modellen am Beispiel des Erhitzers zeigen zu können, wurde mit der MODELICA[®] FAULT TRIGGERING LIBRARY [121] ein typischer Fehler simuliert, der in Abbildung 5.8 dargestellt ist.

Abbildung 5.8: Signale des Erhitzers. Abtastzeit $T_a = 900\text{s}$ (15-Minutendaten).

Simuliert wurde ein Ausfall der Erhitzerpumpe. In diesem Fall wird der Erhitzer nicht mehr mit Heißwasser versorgt und das Dreiwegeventil öffnet vollständig, um dem Problem der zu geringen Luftaustrittstemperatur mit einem höheren Wasservolumenstrom entgegenzuwirken. Da der Durchfluss Null ist, nähern sich die Temperaturen der Wasseraustrittstemperatur und Luftaustrittstemperatur entsprechend an. Ziel ist es nun, diesen Fehler mit dem qualitativen Erhitzermodell zu erkennen, was unmittelbar durch die Beobachtergleichung (4.37) und dem daraus resultierenden Verhaltenssignal \tilde{B} aus Gleichung (4.43) realisiert werden kann. Die Abbildungen 5.9 zeigen das Ergebnis der qualitativen Beobachtung für den Beginn des Fehlers.

Wie man in der Abbildung 5.9 erkennt, kann das qualitative Modell für den Zeitpunkt $k = 5876$ keine positiven Zustandswahrscheinlichkeiten mehr bestimmen, denn der Hintergrund in den Abbildungen ist vollständig weiß. Obwohl der reelle Wert einen Zeitschritt früher ($k = 5875$) in den gleichen Partitionen mit der Nummer Eins liegt und diese mit einer hohen Wahrscheinlichkeit bewertet sind, ist die Wahrscheinlichkeit für diese Zustände zum Zeitpunkt $k = 5876$ Null. Dies liegt daran, dass das zum Zeitpunkt $k = 5875$ gegebene Paar der qualitativen Ein- und Ausgänge inkonsistent mit dem qualitativen Modell ist. An den quantisierten Eingangs- und Ausgangsfolgen des Erhitzers lässt sich

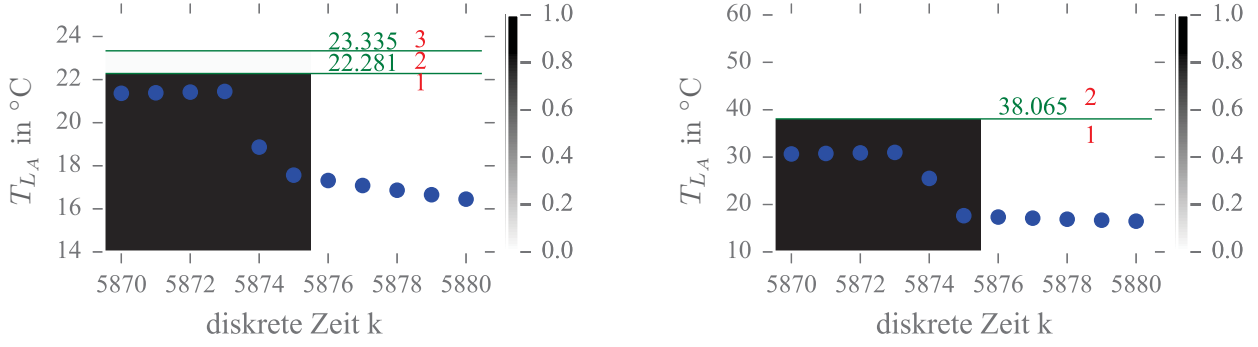


Abbildung 5.9: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} und der Wasseraustrittstemperatur T_{WA} für den Beginn des Fehlerbereichs

dies leicht erläutern:

$$\begin{aligned} [U(5870...5879)] &= (1, 1, 1, 1, 1, 2, 2, 2, 2, 2) , \\ [Y(5870...5879)] &= (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) . \end{aligned} \quad (5.24)$$

Zum Zeitpunkt $k = 5875$ haben die quantisierten Ein- und Ausgangsgrößen die qualitativen Werte $[u(5875)] = 2$ und $[y(5875)] = 1$. Mit der Beobachtergleichung (7.25) wird somit zu diesem Zeitpunkt der Block $\tilde{L}(1,2)$ der Verhaltensmatrix (5.16) selektiert. In dieser Matrix sind jedoch alle Werte Null, weshalb auch die Multiplikation mit dem Zustandsvektor $\tilde{\mathbf{p}}_z(5875)$ nur zu einem Folgezustandsvektor $\tilde{\mathbf{p}}_z(5876) = \mathbf{0}$ führen kann, der ebenfalls ausschließlich Nullen enthält. Diese E/A Kombination trat offensichtlich während des Trainingsintervalls niemals auf. Man kann an dieser Stelle auch bereits anhand der Partitionen des quantisierten Eingangs und Ausgangs in Abbildung 5.3 intuitiv nachvollziehen, dass diese E/A Kombination nicht nominal sein kann, denn sie würde bedeuten, dass der reelle Wert des Ventilsignals in der Partition zwei liegt und dass das Ventil weiter als 45.4% geöffnet ist, während die Luftaustrittstemperatur T_{LA} in der Partition eins liegt und damit eine Temperatur unter 22.8°C besitzt.

Die Tabelle 5.9 zeigt die Zustandswahrscheinlichkeiten für den Beginn des Fehlerbereichs. Wie man sehen kann, ist der Wert des Verhaltenssignals zum Zeitpunkt $k = 5875$ bereits null $\tilde{b}(5875) = 0$, da das E/A Paar ($[u(5875)] = 2, [y(5875)] = 1$) zu diesem Zeitpunkt inkonsistent mit dem qualitativen Modell ist. Die Elemente $\tilde{b}(k)$ des Verhaltenssignals \tilde{B} aus Gleichung (4.43) sind ebenfalls in der Tabelle 5.9 dargestellt und bilden die Folge

$$\tilde{B}(5872, \dots, 5877) = (0.987, 0.987, 0.987, 0, 0, 0) .$$

Tabelle 5.9: Qualitative Zustandswahrscheinlichkeiten für $[\mathbf{x}(k) = z]$

z	$k = 5872$	$k = 5873$	$k = 5874$	$k = 5875$	$k = 5876$	$k = 5877$	$k = 5878$
1	0.987	0.987	0.987	0.987	0	0	0
2	0.013	0.013	0.013	0.013	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
$\tilde{b}(k)$	0.987	0.987	0.987	0	0	0	-

Die nachfolgende Abbildung 5.10 zeigt den gesamten Bereich in dem der Fehler auftrat anhand der qualitativen Zustandswahrscheinlichkeiten der Wasseraustrittstemperatur T_{WA} und der Luftaustrittstemperatur T_{LA} . Zusätzlich ist das Verhaltenssignal \tilde{B} darunter abgebildet. Die Abbildungen 5.10 können damit als das Ergebnis des qualitativen Beobachteralgorithmus 4.1 interpretiert werden.

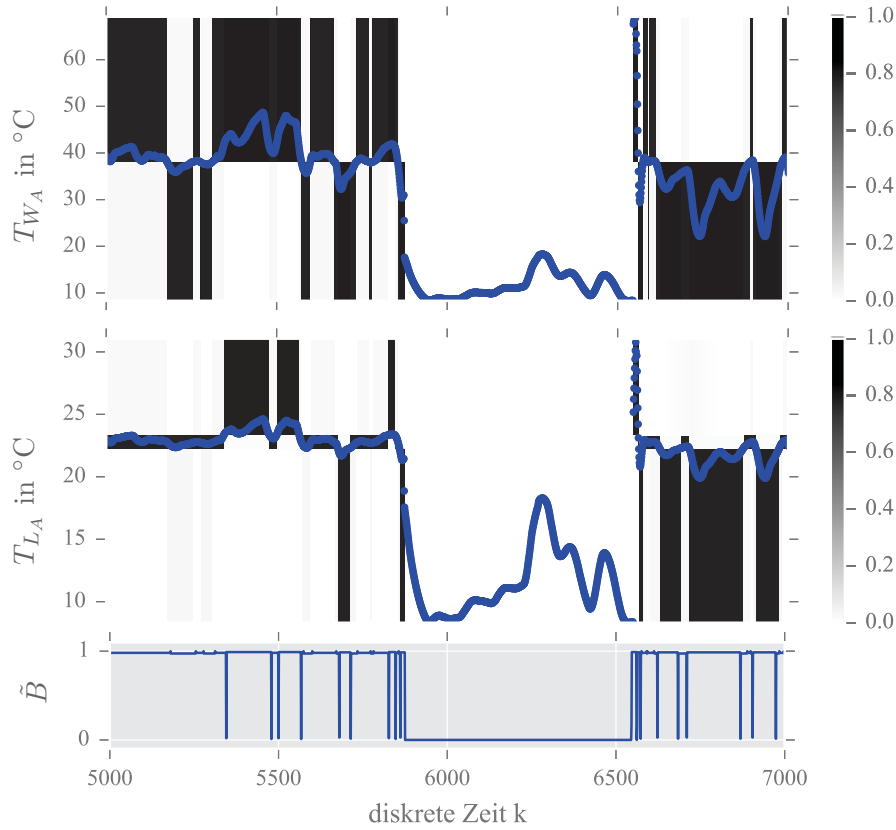


Abbildung 5.10: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{L_A} und der Wasseraustrittstemperatur T_{W_A} sowie das Verhaltenssignal \tilde{B} .

Wie man in Abbildung 5.10 erkennen kann, bietet das Verhaltenssignal grundsätzlich eine übersichtlichere Möglichkeit zur Bewertung des Verhaltens des Erhitzers. Das Verhaltenssignal ist stets größer null, solange die gegebenen Ein- und Ausgangspaare des Erhitzers mit dem qualitativen Modell konsistent sind und gleich null, wenn diese inkonsistent mit dem Modell sind. Da das Verhaltenssignal \tilde{B} ein stochastisches Signal ist, welches aufgrund von mehr oder weniger wahrscheinlichen Ein- und Ausgängen zwischen Null und Eins schwankt, kann dieses durch Anwendung des gleitenden Mittelwertes aus Gleichung (4.46) geglättet werden. Über die Fensterbreite des gleitenden Mittelwertes ϕ kann somit eingestellt werden, nach wie vielen Zeitschritten das Signal \tilde{B}_m aus Gleichung (4.47) im Fehlerfall tatsächlich den Wert Null annimmt. Eine Fensterbreite von $\phi = 1$ bedeutet dabei, dass kein Mittelwert gebildet wird und damit $\tilde{B}_m = \tilde{B}$ ist. Nach der Mittelwertbildung wird das Signal \tilde{B}_m mit Gleichung (4.48) auf die booleschen Werte $\{0, 1\}$ abgebildet, sodass sich das einfach zu interpretierende Signal \tilde{B}_b aus Gleichung (4.49) ergibt. Dieses besitzt im Nominalfall den Wert Eins und bei inkonsistenten E/A Paaren, beziehungsweise einem Fehler, den Wert Null. Der Schwellwert τ in Gleichung (4.48), der darüber entscheidet, ab welchem Wert den gemittelten Signalwerten $\tilde{b}_m(k)$ der Wert Null oder Eins zugewiesen wird, sollte auf den Wert Null ($\tau = 0$) gesetzt werden. Für weitere Erläuterungen siehe Abschnitt 4.4.3. In der Abbildung 5.11 ist sowohl das Verhaltenssignal \tilde{B} (oben) als auch das boolesche Verhaltenssignal \tilde{B}_b für den entsprechenden Zeitbereich abgebildet.

Das Verhaltenssignal \tilde{B}_b wurde hier mit einem gleitenden Mittelwert mit einer Fensterbreite von $\phi = 2$ aus dem Signal \tilde{B} erzeugt. Der Schwellwert τ wurde auf einen Wert von Null gesetzt, sodass nur tatsächlich inkonsistente E/A Paare auch als solche bewertet werden. Damit erhält man das eindeutige und einfach zu interpretierende Signal \tilde{B}_b im unteren Teil der Abbildung 5.11.

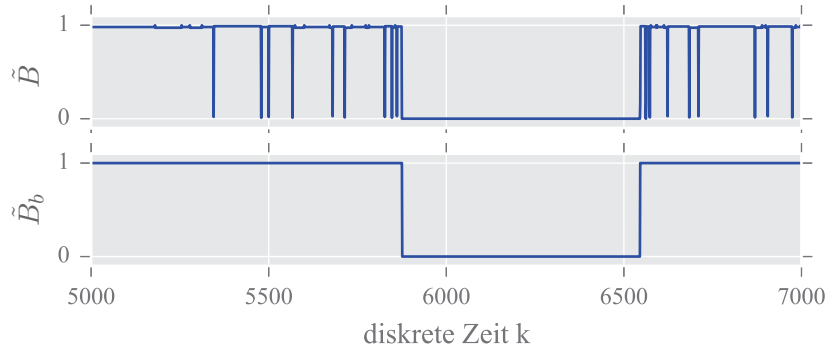


Abbildung 5.11: Verhaltenssignal \tilde{B} (oben) und \tilde{B}_b (unten) ($\phi = 2, \tau = 0$)

Bewertung der Ergebnisse

Wie bereits erwähnt besitzt die Verhaltensmatrix (5.15) eine Struktur, die sehr hohe Wahrscheinlichkeiten dafür vorsieht, dass das System stets im gleichen Zustand verbleibt. Dies ist auch an den qualitativen Zustandswahrscheinlichkeiten in den Abbildungen 5.7 und 5.10 sowie in der Tabelle 5.6 erkennbar. In den erwähnten Abbildungen sind die vom qualitativen Modell berechneten Zustandswahrscheinlichkeiten fast ausnahmslos schwarz dargestellt und demzufolge stets mit sehr hohen Wahrscheinlichkeiten bewertet. Die Verteilungen der Zustandswahrscheinlichkeiten ändern sich nur dann merklich, wenn sich die Ein- und Ausgänge ändern. Das qualitative Modell bewertet dabei stets einen einzelnen qualitativen Zustand mit einer sehr hohen Wahrscheinlichkeit, während die Wahrscheinlichkeiten der anderen Zustände sehr niedrig oder Null sind. Dieses nahezu deterministische Modellverhalten ist grundsätzlich nicht ideal, da solche Modelle in Bezug auf die Fehlererkennung dazu neigen, bereits bei geringen Abweichungen vom trainierten Nominalverhalten, Fehler auszugeben. Die Ursache für dieses Verhalten liegt hier an der geringen Anzahl qualitativer Eingänge, Ausgänge und Zustände als auch an der Beobachterzeitschrittdauer von nur einem Zeitschritt ($T_b = 1$). Diese Restriktionen sind dem Umstand geschuldet, dass sich die grundlegenden Berechnungen zur Erstellung, Anwendung und Fehlererkennung mit qualitativen Modellen nur für solch kleine Beispiele zeigen lassen. Dieses Beispiel zeigt dabei allerdings auch, dass selbst mit einem nicht idealen und dazu noch unvollständigen qualitativen Modell, die Fehlererkennung in technischen Systemen möglich ist. Im nächsten Abschnitt wird anhand des Erhitzers beispielhaft gezeigt, dass sich mit etwas anderen Modellparametern deutlich bessere Ergebnisse erzielen lassen.

5.2.6 Variation der Partitionierungen und der Beobachterzeitschrittdauer

Dieser Abschnitt zeigt beispielhaft, wie man das qualitative Modell des Erhitzers optimieren kann, um die im Abschnitt zuvor angesprochenen Probleme zu minimieren. Dazu wird sowohl die Beobachterzeitschrittdauer T_b als auch die Anzahl der Eingangs-, Ausgangs- und Zustandsgrößen sowie die Anzahl der Partitionierungen geändert.

Variation der Beobachterzeitschrittdauer

Die Abbildung 5.12 zeigt die Resultate des qualitativen Modells für verschiedene Beobachterzeitschrittdauern T_b . Alle anderen Modellparameter blieben dabei unverändert und entsprechen denjenigen aus Tabelle 5.5.

Wie man an den qualitativen Zustandswahrscheinlichkeiten erkennen kann, nehmen mit zunehmender Beobachterzeitschrittdauer die grauen Bereiche in den Abbildungen zu. Die Erhöhung der Beobach-

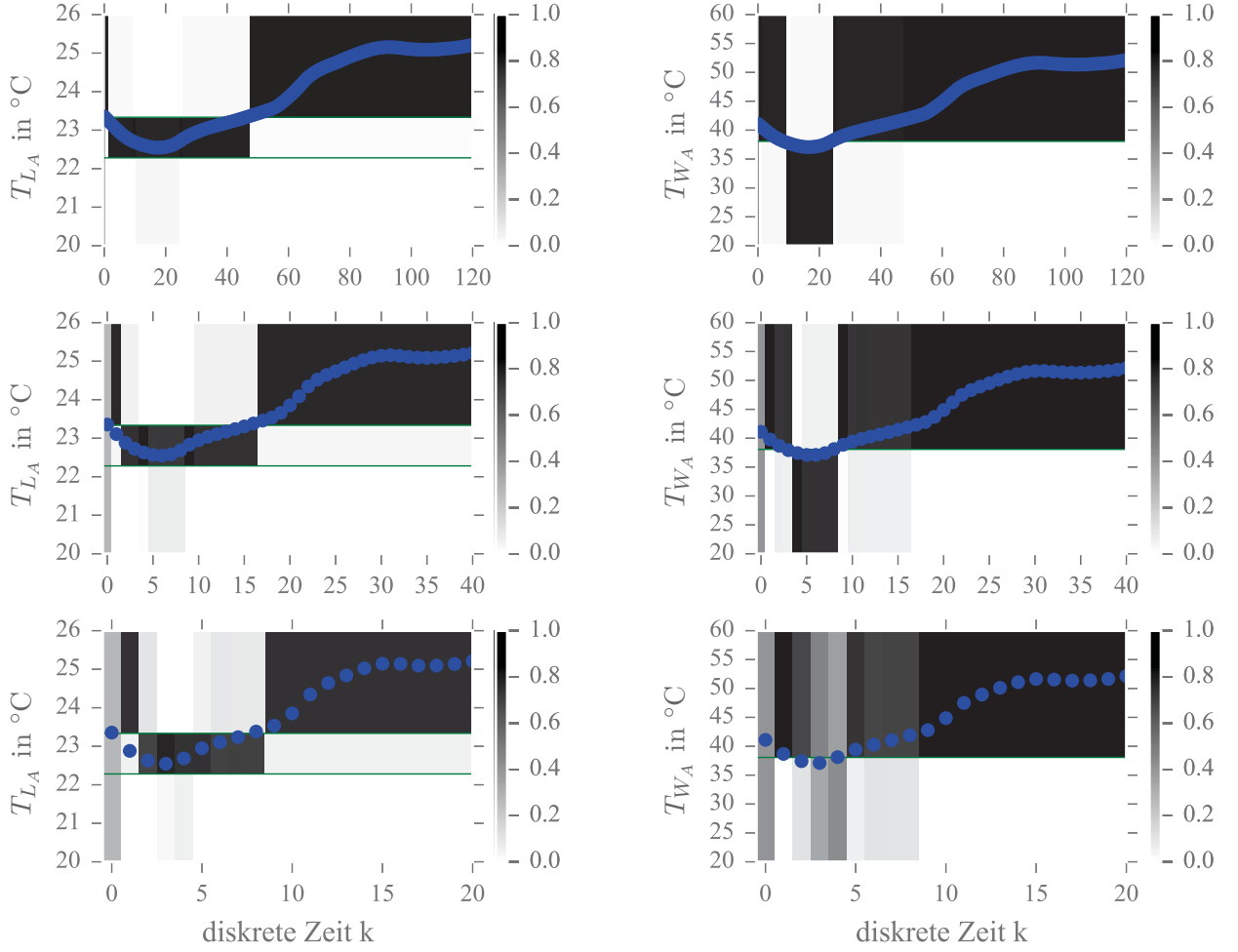


Abbildung 5.12: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{L_A} und der Wasseraustrittstemperatur T_{W_A} für eine Beobachterzeitschrittdauer von $T_b = 1$ (1. Reihe), $T_b = 3$ (2. Reihe), $T_b = 6$ (3. Reihe)

terzeitschrittdauer führt zu einer breiteren Verteilung der Zustandswahrscheinlichkeiten, sodass zu einem Zeitpunkt k in der Regel mehr qualitative Zustände mit einer positiven Zustandswahrscheinlichkeit bewertet werden. Die Anzahl der Nicht-Nullelemente in der Verhaltensmatrix steigt somit an. Dies zeigt auch der Vergleich der Verhaltensmatrix (5.25), die unter Berücksichtigung einer Beobachterzeitschrittdauer von $T_b = 6$ erstellt wurde, mit derjenigen aus Gleichung (5.16), die eine Beobachterzeitschrittdauer von $T_b = 1$ zugrunde lag. Dabei gilt es zu beachten, dass sich durch die Veränderung der Beobachterzeitschrittdauer die Zeitachse k ändert. In der oberen Reihe der Abbildung 5.12 beträgt die Beobachterzeitschrittdauer $T_b = 1$ und ein Zeitschritt k entspricht der Abtastzeit der Messdaten und beträgt damit 900 s (15 Minuten). Bei einer Beobachterzeitschrittdauer von $T_b = 3$ entspricht ein Zeitschritt k einem Zeitraum von $3 \cdot 900 \text{ s} = 45$ Minuten. In der unteren Reihe sind es $6 \cdot 900 \text{ s} = 90$ Minuten. Alle Abbildungen zeigen damit den gleichen Zeitbereich von 30 Stunden.

$$\tilde{\mathbf{L}}(1,1) = \begin{pmatrix} 0.921 & 0 & 0 & 0 & 0 & 0 \\ 0.079 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}}(1,2) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (5.25)$$

$$\begin{aligned}
\tilde{\mathbf{L}}(2,1) &= \begin{pmatrix} 0 & 0.155 & 0 & 0 & 0 & 0 \\ 0 & 0.721 & 0 & 0 & 0.333 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.124 & 0 & 0 & 0.667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(2,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0.128 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.756 & 0 \\ 0 & 0 & 0 & 0 & 0.112 & 0 \end{pmatrix}, \\
\tilde{\mathbf{L}}(3,1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, & \tilde{\mathbf{L}}(3,2) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.055 \\ 0 & 0 & 0 & 0 & 0 & 0.944 \end{pmatrix}.
\end{aligned}$$

Variation der Partitionierungen und der Beobachterzeitschrittdauer

Das im Folgenden beschriebene qualitative Modell des Erhitzers wurde mit den in Tabelle 5.10 gegebenen Parametern identifiziert.

Tabelle 5.10: Modellparameter

System	Erhitzer einer RLT-Anlage aus Abbildung 5.1
Abtastzeit der Daten	$T_a = 900$ s (15-Minutendaten)
Eingangsgrößen Anzahl	$u_1 = S_V, u_2 = T_{W_E}, u_3 = T_{L_E}$ $m = 3$
Zustandsgrößen Anzahl	$x = T_{L_A}$ $n = 1$
Ausgangsgrößen Anzahl	$y = T_{W_A}$ $q = 1$
Anzahl qual. Eingänge	$M = 64$ ($M_1 = 4, M_2 = 4, M_3 = 4$)
Anzahl qual. Zustände	$N = 6$
Anzahl qual. Ausgänge	$Q = 4$
Speicherbedarf	$\gamma = 6 \cdot 4 \cdot 6 \cdot 64 = 9216$ Werte
Trainingssdaten	60 Tage
Beobachterzeitschrittdauer	$T_b = 1, T_b = 3, T_b = 6, T_b = 12$

Im Unterschied zum einfachen qualitativen Erhitzermodell mit den Parametern aus Tabelle 5.5, werden nun drei Eingangsgrößen berücksichtigt: Die Ventilstellung S_V , die Wassereintrittstemperatur T_{W_E} und die Luft Eintrittstemperatur T_{L_E} . Als Zustand dient die Luftaustrittstemperatur T_{L_A} und die Wasseraustrittstemperatur T_{W_A} stellt die Ausgangsgröße dar. Das Modell besitzt $M = 64$ qualitative Eingänge, $N = 6$ qualitative Zustände und $Q = 4$ qualitative Ausgänge, womit der Speicherbedarf der Verhaltensmatrix bereits 9216 Werte beträgt. Die Abbildung 5.13 zeigt die qualitativen Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{L_A} für verschiedene Beobachterzeitschrittdauern.

Wie man an der Abbildung 5.13 oben links erkennen kann, sind die Zustandswahrscheinlichkeiten bereits für eine Beobachterzeitschrittdauer von $T_b = 1$ deutlich »unschärfer« als in Abbildung 5.12. Dies liegt daran, dass die Zustandsgröße nun über mehr qualitative Zustände ($N = 6$) verfügt und zudem nicht mehr identisch mit einer der Ausgangsgrößen ist. Zudem tragen die zusätzlichen Eingangsgrößen dazu bei, dass das qualitative Modell die dynamischen Eigenschaften des Erhitzer besser abbilden kann. Dabei wird auch an diesem Beispiel deutlich, dass die Erhöhung der Beobachterzeitschrittdauer zu einer unschärferen Repräsentation der qualitativen Zustandswahrscheinlichkeiten führt.

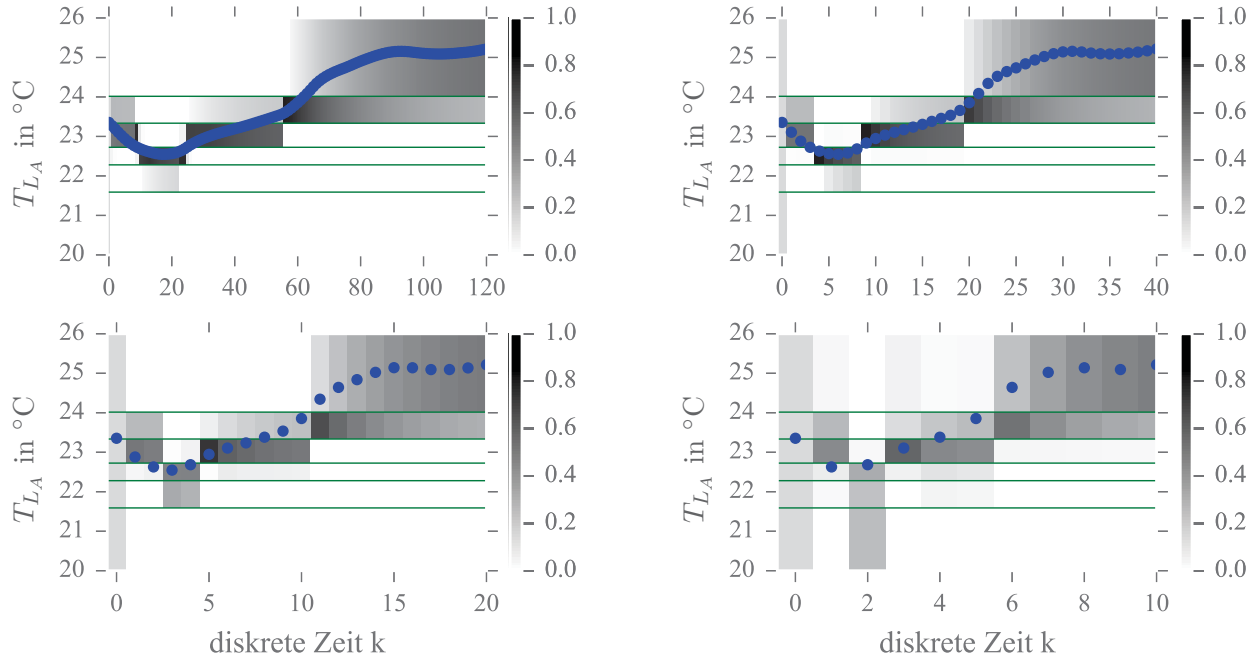


Abbildung 5.13: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} für eine Beobachterzeitschrittdauer von $T_b = 1$ (oben links), $T_b = 3$ (oben rechts), $T_b = 6$ (unten links) und $T_b = 12$ (unten rechts)

An diesem Beispiel lässt sich zudem die qualitative Systemdynamik sehr gut erkennen. Dazu können die qualitativen Zustandswahrscheinlichkeiten in Abbildung 5.14, die unter Verwendung einer Beobachterzeitschrittdauer von $T_b = 12$ bestimmt wurden, betrachtet werden.

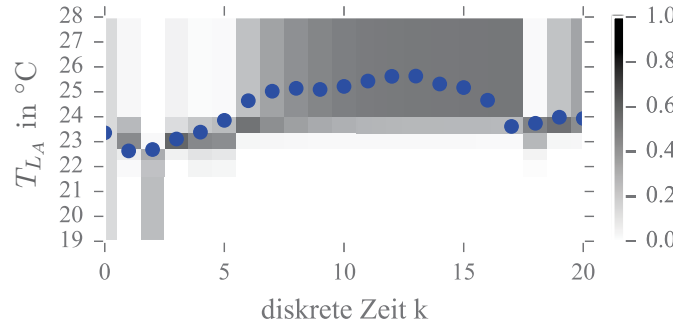


Abbildung 5.14: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} für eine Beobachterzeitschrittdauer von $T_b = 12$

Während des Zeitraumes $k = 5$ bis $k = 15$ ändern sich die qualitativen Werte der quantisierten Ein- und Ausgänge des Erhitzers nicht und sind für diesen Zeitbereich durch die Folgen

$$\begin{aligned} [U(5, \dots, 15)] &= (61, 61, 61, 61, 61, 61, 61, 61, 61, 61, 61) , \\ [Y(5, \dots, 15)] &= (4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4) \end{aligned}$$

gegeben. Trotz der Tatsache, dass die qualitativen Ein- und Ausgänge über diesen Zeitraum konstant sind, ändern sich jedoch die Zustandswahrscheinlichkeiten innerhalb dieses Zeitbereichs (s. Abbil-

dung 5.14). Dies zeigt die Dynamik des qualitativen Modells. In einem solchen Fall würde jedes *statische* Black-Boxmodell auf die konstanten Modelleingaben stets die gleiche Modellausgabe generieren. Dass die Fehlererkennung auch mit diesem qualitativen Modell möglich ist, zeigt Abbildung 5.15 abschließend anhand der qualitativen Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} und des Verhaltenssignals \tilde{B}_b .

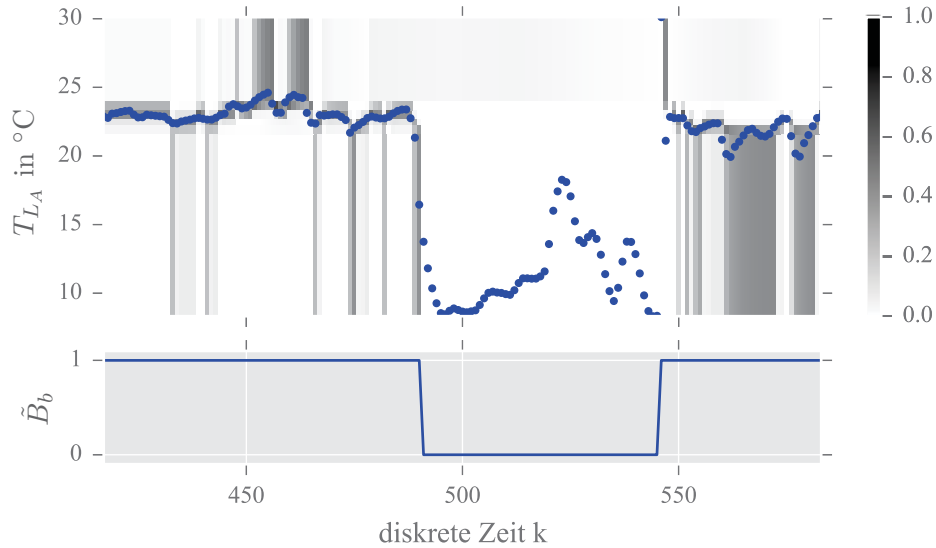


Abbildung 5.15: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} für eine Beobachterzeitschrittdauer $T_b = 12$ und das Verhaltenssignal \tilde{B}_b ($\phi = 2, \tau = 0$)

5.3 Fehlererkennung in einem Erhitzer

RLT-Anlagen gehören, wie bereits in der Einleitung erwähnt, zu den größten Energieverbrauchern in Gebäuden. Fehler in den Erhitzern, Kühlern oder Wärmerückgewinnungssystemen führen oftmals zu höheren Energiekosten aber auch zu thermischen Komforteinbußen in den Gebäuden. Die Fehlererkennung in diesen Systemen besitzt damit eine besonders hohe Relevanz. Während des EU-Projekts CASCADE [21] wurden am Fraunhofer ISE sowohl regelbasierte Systeme zur Fehlererkennung in RLT-Anlagen entwickelt als auch erste Anwendungstests mit qualitativen Modellen zur Fehlererkennung durchgeführt. Die Realisierung der qualitativen Fehlererkennung für den nachfolgend eingeführten RLT-Erhitzer wurde auch in [91] beschrieben.

5.3.1 Systembeschreibung

Vom Erhitzer der betreffenden RLT-Anlage sind die in der Abbildung 5.16 eingezeichneten Signale verfügbar. Diese wurden entsprechend zur qualitativen Modellbildung verwendet. Die weiteren Parameter des qualitativen Modells für den Erhitzer sind in der Tabelle 5.11 dargestellt.

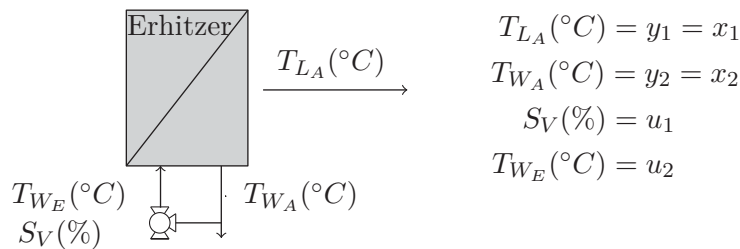


Abbildung 5.16: Erhitzer einer RLT-Anlage

Tabelle 5.11: Modellparameter

System	Erhitzer einer RLT-Anlage aus Abbildung 5.16
Abtastzeit der Daten	$T_a = 900$ s (15-Minutendaten)
Eingangsgrößen Anzahl	$u_1 = S_V, u_2 = T_{W_E}$ $m = 2$
Zustandsgrößen Anzahl	$x_1 = T_{L_A}, x_2 = T_{W_A}$ $n = 2$
Ausgangsgrößen Anzahl	$y_1 = T_{L_A}, y_2 = T_{W_A}$ $q = 2$
Anzahl qual. Eingänge	$M = 36$ ($M_1 = 6, M_2 = 6$)
Anzahl qual. Zustände	$N = 36$ ($N_1 = 6, N_2 = 6$)
Anzahl qual. Ausgänge	$Q = 36$ ($Q_1 = 6, Q_2 = 6$)
Speicherbedarf	$\gamma = 36 \cdot 36 \cdot 36 \cdot 36 = 1\,679\,616$ Werte
Trainingsdaten	57 Tage
Beobachterzeitschrittdauer	$T_b = 2$

Wie die Tabelle 5.11 zeigt, wurden die Ventilstellung S_V und die Wassereintrittstemperatur T_{W_E} als Eingangsgrößen für das Modell verwendet. Die Zustände und die Ausgänge sind identisch und durch die Luftaustrittstemperatur T_{L_A} sowie die Wasseraustrittstemperatur T_{W_A} gegeben. Das Modell besitzt dabei $M = 36$ qualitative Eingänge, $N = 36$ qualitative Zustände und $Q = 36$ qualitative Ausgänge, womit der Speicherbedarf der Verhaltensmatrix erheblich ansteigt und bei 1 679 616 Werten liegt. Die Beobachterzeitschrittdauer wurde bei der Modellbildung auf $T_b = 2$ Zeitschritte festgelegt, sodass bei der Beobachtung ein diskreter Zeitschritt k , einem Zeitraum von 1800 s, beziehungsweise einer halben Stunde entspricht. Die Abbildung 5.17 zeigt die mit der Beobachtergleichung (4.37) berechneten Zustandswahrscheinlichkeitsverteilungen für die Luftaustrittstemperatur T_{L_A} und die Wasseraustrittstemperatur T_{W_A} des Erhitzers über die gesamte Trainingsperiode des Modells.

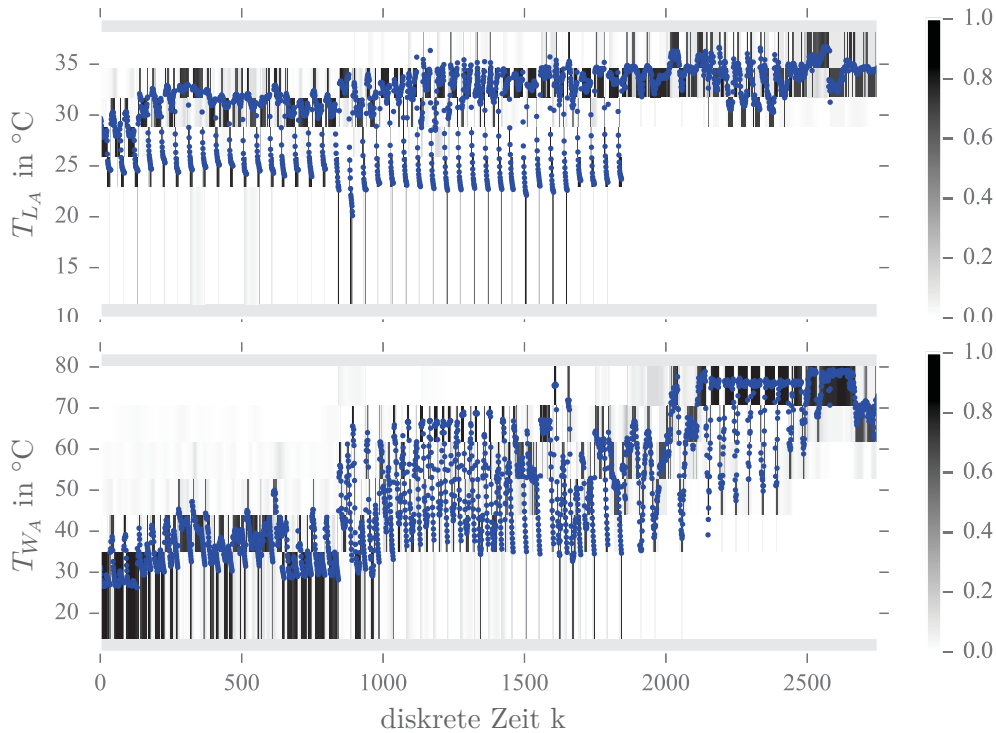


Abbildung 5.17: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{L_A} und der Wasseraustrittstemperatur T_{W_A} sowie deren reelle Messwerte (blaue Punkte)

Man kann an der Abbildung 5.17 bereits erkennen, dass dieses qualitative Modell des realen Systems für die relativ kurze Beobachterzeitschrittdauer von $T_b = 2$ schon recht nichtdeterministisch ist. Die abgebildeten Wahrscheinlichkeitsverteilungen der Zustände weisen unterschiedliche Grauschattierungen auf. Damit sind zu jedem Zeitpunkt k stets mehrere qualitative Zustände möglich und die Gefahr, dass das Modell stets nur einen Zustand als sehr wahrscheinlich bewertet und alle anderen Zustände keine oder sehr geringe Wahrscheinlichkeiten besitzen, kann als niedrig erachtet werden. Noch deutlicher erkennt man dies, wenn ein kürzerer Ausschnitt des Verlaufs der qualitativen Zustandswahrscheinlichkeiten betrachtet wird (s. Abbildung 5.18).

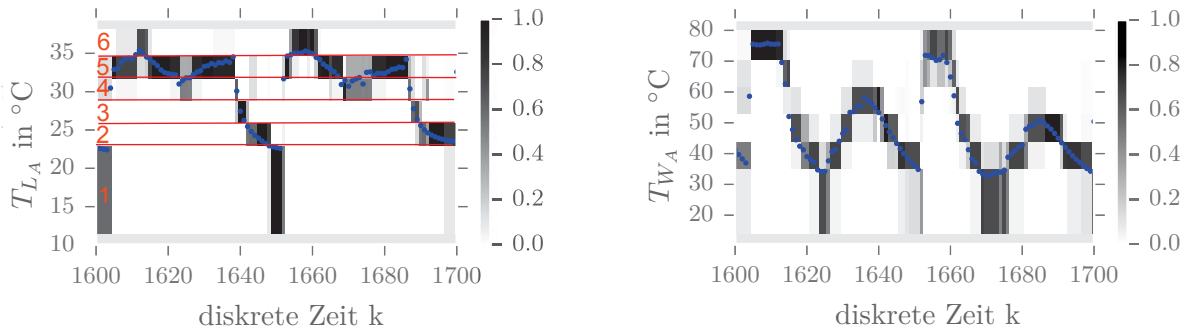


Abbildung 5.18: Qualitative Zustandswahrscheinlichkeiten der Luftaustrittstemperatur T_{LA} (mit eingezeichneten Partitionen) und der Wasseraustrittstemperatur T_{WA}

Das in diesem Beispiel schon eine Beobachterzeitschrittdauer von nur zwei Zeitschritten eine gute Wahl ist, liegt an der Verwendung realer Messdaten. Diese unterliegen natürlichen äußeren Einflüssen und zeigen kein derart idealisiertes Verhalten wie die im Beispiel in Abschnitt 5.2 verwendeten Simulationsdaten des MODELICA[®] Modells.

5.3.2 Qualitative Fehlererkennung

Nach dem Trainieren des qualitativen Modells kann dieses entsprechend zur Bewertung neuer Messdaten des Erhitzers eingesetzt werden, wozu der einfache Beobachteralgorithmus 4.1 verwendet wird. Die beiden folgenden Abbildungen 5.19 zeigen dabei einen Zeitraum von etwas mehr als 5 Tagen im Monat Dezember, in dem die Detektion einer Anomalie im Verhalten des Erhitzers sichtbar ist.

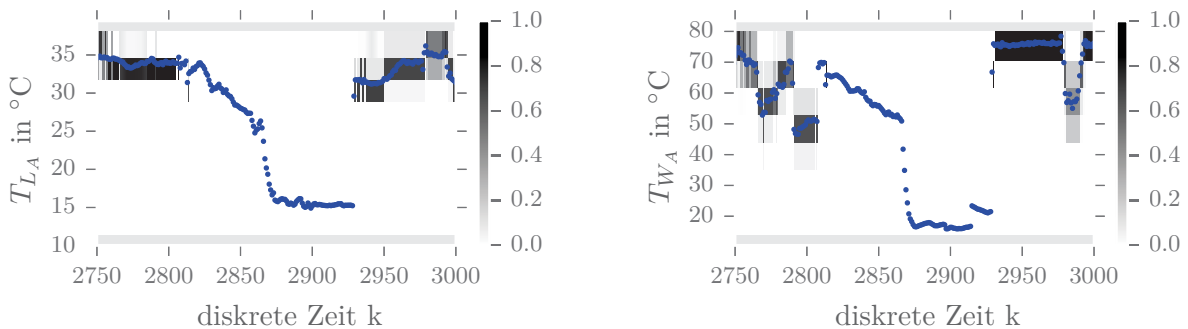


Abbildung 5.19: Anomalie in der Wasser- und Luftaustrittstemperatur. Darstellung über ca. 5 Tage des Monats Dezember.

Wie man in den Abbildungen erkennt, kann das qualitative Modell offensichtlich für den diskreten Zeitbereich $\approx [2800 \leq k \leq 2930]$ keine positiven Zustandswahrscheinlichkeiten berechnen, was bedeutet, dass die gemessenen quantisierten Ein- und Ausgangsfolgen inkonsistent mit dem qualitativen Modell sind. Deutlich erkennbar ist auch der Neustart des Beobachteralgorithmus, nachdem der vom

Modell als nicht nominaler Betrieb klassifizierte Zustand wieder verlassen wurde. Im hier dargelegten Fall lässt sich leicht erkennen, dass sowohl die Luftaustrittstemperatur als auch die Wasseraustrittstemperatur für den Heizfall außergewöhnlich niedrig sind. Das Dreiwegeventil des Erhitzers ist während dieser Zeit vollständig geöffnet und in der Wasservorlauftemperatur treten mehrere Schwankungen auf, die alleine jedoch nicht als Ursache in Frage kommen. Am nächstliegenden scheint hier ein Problem mit dem wasserseitigen Durchfluss zu sein, was wiederum verschiedene Gründe haben kann. Die Ursache des Problems konnte in diesem Fall nicht geklärt werden. An dieser Stelle sei darauf hingewiesen, dass der Grund für die vom Modell nicht berechenbaren Zustandswahrscheinlichkeiten nicht die niedrigen Luft- und Wasseraustrittstemperaturen sind, sondern die Kombinationen der gemessenen quantisierten Ein- und Ausgänge. Wie man in der Abbildung erkennen kann, wird die Anomalie bereits detektiert bevor sich die Luft- und Wassertemperaturen in einem sehr niedrigen Bereich befinden.

Die beiden Abbildungen 5.20 zeigen das Verhaltenssignal \tilde{B} aus der Gleichung (7.28) und \tilde{B}_b aus (4.49). Dabei wurde der Schwellwert τ in (4.48) auf Null gesetzt und der gleitende Mittelwert (4.46) nicht verwendet. Diese Konfiguration bedeutet, dass alle mit dem Modell inkonsistenten E/A Folgen detektiert und visualisiert werden.



Abbildung 5.20: Verhaltenssignal \tilde{B} (links) und \tilde{B}_b (rechts) ($\phi = 1, \tau = 0$)

5.4 Fehlererkennung in einem Heizkreis

Im Rahmen des BMWi Projekts OBSERVE [98] wurde ein qualitatives Modell zur Fehlererkennung in Heizkreisen entwickelt, welches einem der Projektpartner zu Erprobungszwecken zur Verfügung gestellt wurde. Die folgenden Ergebnisse wurden auch im Endbericht [108] dieses Projekts veröffentlicht.

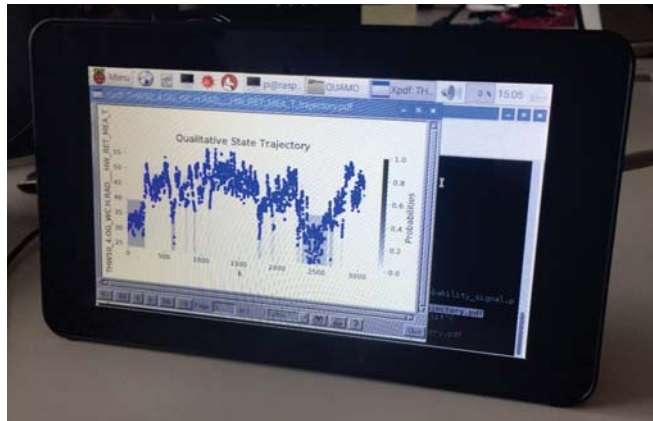


Abbildung 5.21: Qualitatives Modell installiert auf einem RASPBERRY PI®

Das im Folgenden beschriebene qualitative Modell wurde auf einem Minicomputer des Typs RASPBERRY PI® integriert. Solche dezentralen und flexiblen Lösungen gewinnen zunehmend an Relevanz, da sie die Fehlererkennung unabhängig von Herstellern und den bestehenden Gebäudeautomations-systemen machen. Abbildung 5.21 zeigt beispielhaft ein qualitatives Modell, installiert auf einem RASPBERRY PI®.

5.4.1 Systembeschreibung

Die Abbildung 5.22 zeigt die Systemvorstellung des Heizkreises mit den entsprechenden Eingangs-, Zustands- und Ausgangsgrößen. Bei dem Gebäude handelt es sich um ein mittelgroßes Nichtwohngebäude in Deutschland.

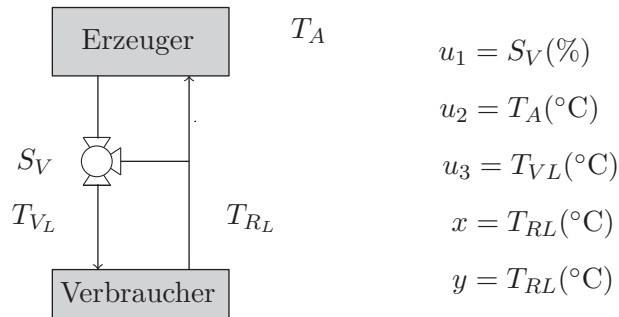


Abbildung 5.22: Einfacher Heizkreis

In der Tabelle 5.12 sind die Parameter des qualitativen Modells dargestellt. Als Eingangsgrößen für das qualitative Modell wurden die Ventilstellung S_V , die Vorlauftemperatur T_{VL} und die Außentemperatur T_A verwendet. Als Zustand sowie als Ausgang dient die Rücklauftemperatur T_{RL} .

Tabelle 5.12: Modellparameter

System	Heizkreis aus Abbildung 5.22
Abtastzeit der Daten	$T_a = 60$ s (Minutendaten)
Eingangsgrößen Anzahl	$u_1 = S_V, u_2 = T_A, u_3 = T_{VL}$ $m = 3$
Zustandsgrößen Anzahl	$x = T_{RL}$ $n = 1$
Ausgangsgrößen Anzahl	$y = T_{RL}$ $q = 1$
Anzahl qual. Eingänge	$M = 216$ ($M_1 = 6, M_2 = 6, M_3 = 6$)
Anzahl qual. Zustände	$N = 6$
Anzahl qual. Ausgänge	$Q = 6$
Speicherbedarf	$\gamma = 6 \cdot 6 \cdot 6 \cdot 216 = 46\,656$ Werte
Trainingssdaten	250 Tage
Beobachterzeitschrittdauer	$T_b = 1$

5.4.2 Qualitative Fehlererkennung

Abbildung 5.23 zeigt die qualitativen Zustandswahrscheinlichkeiten der Rücklauftemperatur T_{RL} des Heizkreises für einen Zeitraum von fünf Stunden während der Heizperiode im Januar 2018. Da die Beobachterzeitschrittdauer hier nur einen Zeitschritt beträgt ($T_b = 1$) und die zugrundeliegenden Daten Minutendaten sind, entspricht ein Zeitschritt k einer Minute. Der in der Abbildung sichtbare Fehler dauerte dabei 67 Minuten. Verursacht wurde diese Betriebsstörung des Heizkreises dadurch, dass in diesem Zeitraum Öl für den Heizkessel geliefert wurde und die Anlage während dieser Zeit abgeschaltet war.

In Abbildung 5.23 zeigt sich wieder, dass der Fehler nicht anhand der Rücklauftemperatur detektiert wird, sondern auf Basis des Eingangs-Ausgangs-Verhaltens des Heizkreises. In der Abbildung wird ein Bereich als fehlerhaft detektiert, indem sich die Rücklauftemperatur nicht ungewöhnlich verhält.

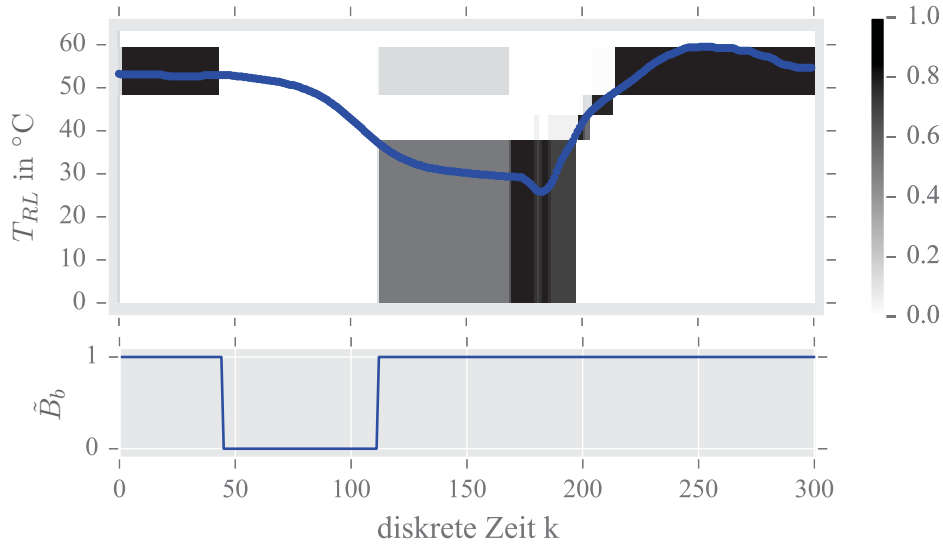


Abbildung 5.23: Qualitative Zustandswahrscheinlichkeiten der Rücklauftemperatur T_{RL} und das Verhaltenssignal \tilde{B}_b ($\phi = 2, \tau = 0$)

Diese sinkt nach dem Fehlerbereich sogar weiter ab, was zeigt, dass der Fehler nicht am Verhalten der Zustandsgröße detektiert wird. Einen solchen Fehler würde man durch eine reine Grenzwertüberwachung der Rücklauftemperatur nicht detektieren können.

Abbildung 5.24 zeigt ein weiteres Beispiel, in dem ein sehr kurzer Fehler von nur zwei mal einer Minute im Monat Februar 2018 auftritt. Deutlich zu sehen ist hier ein Einbruch der Rücklauftemperatur des Heizkreises. Diese kurze Betriebsstörung wurde durch das Nachfüllen von Kaltwasser in den Heizkreis verursacht. Detektiert wurde die Veränderung wieder anhand des sich ändernden Eingangs-Ausgangs-Verhaltens des Systems. In Abbildung 5.24 wird deutlich, dass sich auch sehr kurz auftretende Fehler detektieren lassen, wenn die Sensitivität durch einen niedrigen Wert von $\phi = 1$ entsprechend hoch eingestellt ist.

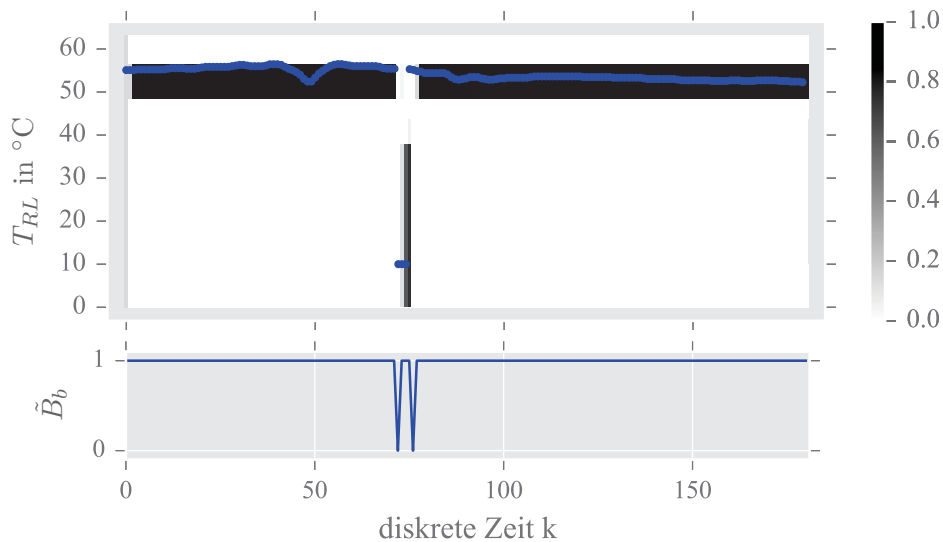


Abbildung 5.24: Qualitative Zustandswahrscheinlichkeiten der Rücklauftemperatur T_{RL} und das Verhaltenssignal \tilde{B}_b ($\phi = 1, \tau = 0$)

Während der Anwendungsphase des qualitativen Heizkreismodells wurde im Rahmen des Forschungsprojekts am gleichen Heizkreis auch eine modellprädiktive Regelung getestet. Diese beeinflusste ent-

sprechend das Verhalten des Heizkreises, was ebenfalls vom qualitativen Modell detektiert wurde. Dies ist in Abbildung 5.25 für die ersten drei Wochen des Monats Dezember 2017 dargestellt. Durch die veränderte Regelstrategie des Heizkreises ändert sich das dynamische Verhalten des Gesamtsystems, was zu vielen kurzzeitig auftretenden Fehlern führt.

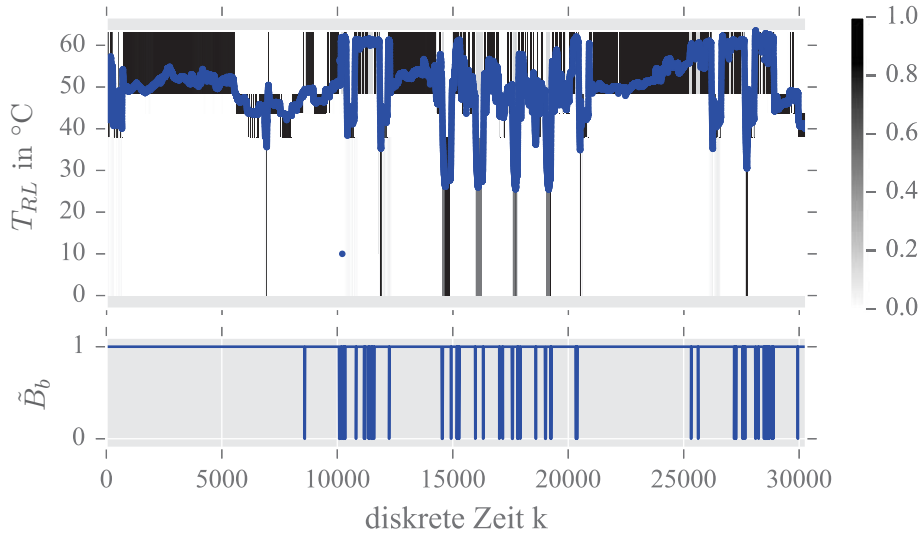


Abbildung 5.25: Qualitative Zustandswahrscheinlichkeiten der Rücklauftemperatur T_{RL} und das Verhaltenssignal \tilde{B}_b ($\phi = 2, \tau = 0$)

Nach der Testphase der modellprädiktiven Regelung des Heizkreises wurde im Januar 2018 wieder die ursprüngliche Regelstrategie übernommen. Die Abbildung 5.26 zeigt das Ergebnis der qualitativen Fehlerdetektion für die letzten beiden Wochen im Januar, in dem nur zwei kurze Inkonsistenzen auftreten, denen sich jedoch keine bestimmten Fehler zuordnen ließen. Solche kurzen Inkonsistenzen können bei Bedarf durch die Erhöhung der Fensterbreite ϕ des gleitenden Mittelwertes eliminiert werden. Hierbei gilt es stets abzuwägen, ab wie vielen zusammenhängenden Fehlern tatsächlich ein Fehler im Verhaltenssignal visualisiert werden soll. In der Praxis wird man wegen eines »Fehlers«, der nur wenige Minuten andauert, kaum eine Fehlermeldung generieren oder den Gebäudebetreiber informieren. Hier sind also stets die Bedürfnisse der Gebäudebetreiber entsprechend zu berücksichtigen.

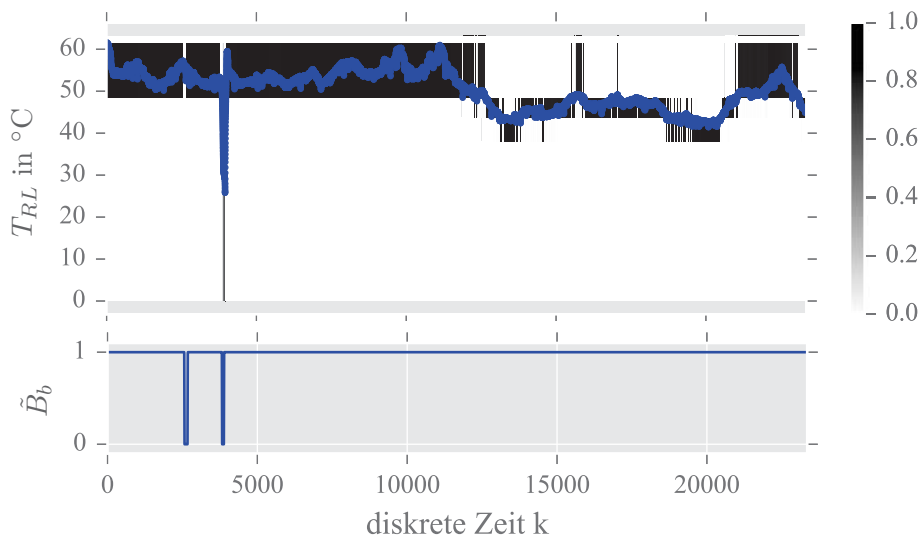


Abbildung 5.26: Qualitative Zustandswahrscheinlichkeiten der Rücklauftemperatur T_{RL} und das Verhaltenssignal \tilde{B}_b ($\phi = 2, \tau = 0$)

5.5 Fehlererkennung in einem thermischen Speicher

Während des EU-Projekts TOPAS [118] wurde ein qualitatives Modell eines thermischen Speichers erstellt, um dessen Betriebsverhalten zu überwachen.

5.5.1 Systembeschreibung

Ein Prinzipbild, welches die verfügbaren Sensoren des Speichers zeigt, ist in Abbildung 5.27 dargestellt. Der Speicher verfügt über fünf Temperatursensoren, die als Zustandsgrößen des qualitativen Modells dienen. Die Speichervorlauftemperatur zu den Verbrauchern wird als Eingangsgröße und die Rücklauftemperatur als Ausgangsgröße verwendet. Die Einstellungen und Parametrierungen für das qualitative Modell sind in der Tabelle 5.13 zusammengefasst.

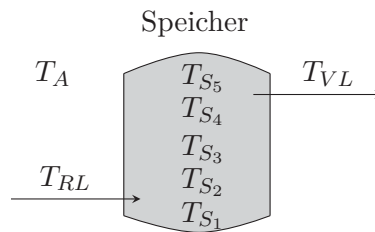


Abbildung 5.27: Thermischer Speicher

Wie Tabelle 5.13 zeigt, beträgt der Speicherbedarf des qualitativen Modells aufgrund der vielen berücksichtigten Zustandsgrößen über 67 Millionen Werte. Der reelle Speicherbedarf für das qualitative Modell liegt hier bei über 500 Megabyte. Dieses Beispiel zeigt die Notwendigkeit dafür, den Speicherbedarf qualitativer Modelle zu reduzieren. Ein direkter Einsatz dieses qualitativen Modells auf einem GLT-Rechner, einer DDC oder einem Mikrocomputer scheint bei diesem Modell nicht mehr praktikabel. Dabei muss zudem bedacht werden, dass es sich nur um ein einziges System eines Gebäudes handelt. Gebäude besitzen jedoch eine Vielzahl von dynamischen Systemen, die es hinsichtlich ihres Verhaltens zu überwachen gilt. Wenn der Speicherbedarf eines einzigen qualitativen Modells bereits derart hoch ist, scheint es eher unwahrscheinlich, auf den meist in ihrer Rechenleistung begrenzten Gebäudeautomationssystemen, mehrere qualitative Modelle zur Überwachung multipler Systeme einzusetzen. Eine Lösung für diese Problem wird in Kapitel 9 beschrieben.

Tabelle 5.13: Modellparameter

System	Thermischer Speicher aus Abbildung 5.27
Abtastzeit der Daten	$T_a = 900 \text{ s}$ (15-Minutendaten)
Eingangsgrößen Anzahl	$u_1 = T_{VL}, u_2 = T_A$ $m = 2$
Zustandsgrößen Anzahl	$x_1 = T_{S_1}, x_2 = T_{S_2}, x_3 = T_{S_3}, x_4 = T_{S_4}, x_5 = T_{S_5}$ $n = 5$
Ausgangsgrößen Anzahl	$y = T_{RL}$ $q = 1$
Anzahl qual. Eingänge	$M = 16$ ($M_1 = 4, M_2 = 4$)
Anzahl qual. Zustände	$N = 1024$ ($N_1 = 4, N_2 = 4, N_3 = 4, N_4 = 4, N_5 = 4$)
Anzahl qual. Ausgänge	$Q = 4$
Speicherbedarf	$\gamma = 1024 \cdot 4 \cdot 1024 \cdot 16 = 67\,108\,864$ Werte
Trainingsdaten	79 Tage
Beobachterzeitschrittdauer	$T_b = 1$

5.5.2 Qualitative Fehlererkennung

Die Abbildung 5.28 zeigt die qualitativen Zustandswahrscheinlichkeiten über einen Zeitraum von 37 Tagen im Frühjahr 2017. Deutlich zu erkennen sind die geschichteten Temperaturniveaus des Speichers und die vom qualitativen Modell berechneten Zustandswahrscheinlichkeiten. Auch wird wieder deutlich, dass das qualitative Modell das Speicherverhalten offenbar gut approximiert, denn die realen Messwerte liegen stets in den dunkleren Bereichen, die mit einer hohen Zustandswahrscheinlichkeit bewertet sind.

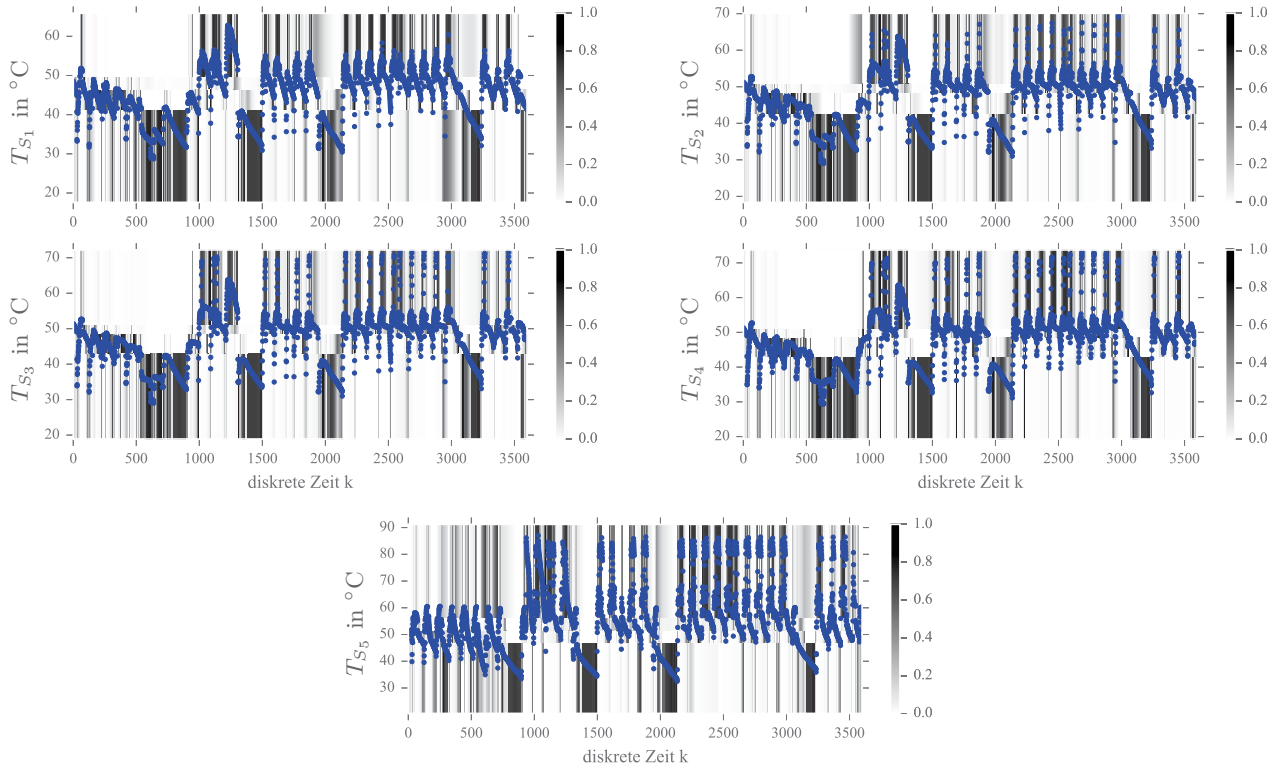


Abbildung 5.28: Qualitative Zustandswahrscheinlichkeiten der Speichertemperaturen, dargestellt über 37 Tage des Trainingszeitraums

In Abbildung 5.29 ist wiederum das Verhaltenssignale \tilde{B} über einen Zeitraum von eineinhalb Jahren dargestellt.

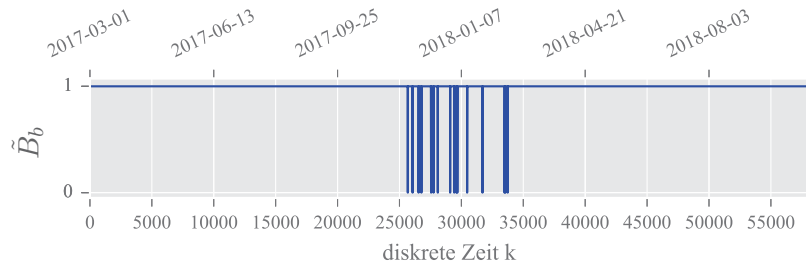


Abbildung 5.29: Verhaltenssignal \tilde{B}_b ($\phi = 4, \tau = 0$)

Wie man sieht, werden im Zeitraum um Weihnachten 2017 mehrere Anomalien detektiert, während über den restlichen Zeitraum keine Fehler erkannt wurden. Die zur Weihnachtszeit auftretenden Inkonsistenzen des Modells mit den Messdaten sind dabei vermutlich auf ein verändertes Nutzungsprofil

des Gebäudes, beziehungsweise des Speichers zurückzuführen. Bei dem Gebäude handelt es sich um eine Universität und der Bereich mit den detektierten Anomalien entspricht den Weihnachtsferien und der vorlesungsfreien Zeit. In diesem Zeitraum wird weniger Warmwasser gezapft als während des Semester und die Temperaturschichtungen des Speichers sind schärfer, was sich auch im Eingangs-Ausgangs-Verhalten zeigt. Es handelt sich hierbei demnach nicht um einen klassischen Fehler im System, sondern lediglich um eine nutzungsspezifische Abweichung vom durch die Trainingsdaten festgelegten Nominalbetrieb.

5.6 Zwischenfazit

Wie dieses Kapitel zeigt, können qualitative Modelle zur Fehlererkennung in verschiedenen Gebäudesystemen erfolgreich verwendet werden, vorausgesetzt man verfügt über entsprechende Messdaten, mit denen sich ein Nominalverhalten der Systeme trainieren lässt. Dieser Punkt stellt bei der Anwendung solcher Modelle sicherlich eine Herausforderung dar, da die Datenvorprozessierung mitunter zeitaufwändig ist. Bei der Verwendung von Gebäudemessdaten zum Trainieren des qualitativen Modells, ist es entsprechend hilfreich, wenn diese von hoher Qualität und mit kurzen Abtastzeiten verfügbar sind. Existieren Lücken in den Messdaten, muss darauf geachtet werden, dass die sich aus diesen Lücken ergebenden nicht-kausalen Zusammenhänge bei der qualitativen Modellbildung nicht mittrainiert werden. Über die notwendige Anzahl von Messdaten, die man benötigt, um ein plausibles qualitatives Modell zu identifizieren, können grundsätzlich keine pauschalen Aussagen getroffen werden. Letztendlich hängt dies davon ab, wie dynamisch das Verhalten der Gebäudesysteme ist und ob die verfügbaren Messdaten ausreichen, dieses Verhalten möglichst vollständig abzubilden. Bei Gebäudeprozessen die stark von der Jahreszeit abhängig sind, sollten möglichst immer Daten aller Jahreszeiten trainiert werden, damit das qualitative Modell »lernt«, wie sich beispielsweise ein Heizkreis im Sommer und im Winter verhält.

Ist man im Besitz geeigneter Messdaten, braucht es nur wenige Informationen um ein qualitatives Modell zu trainieren. Die einstellbaren Modellparameter beschränken sich im Wesentlichen auf die Anzahl der Partitionierungen, die man für die Systemgrößen festlegt und die damit über die Modellpräzision und den benötigten Speicherbedarf entscheiden. Auch die Beobachterzeitschrittdauer des qualitativen Modells kann variiert werden, um ein Modell zu finden, das hinreichend nichtdeterministisch ist und das Systemverhalten damit nicht zu präzise abbildet. Wie dieses Kapitel außerdem zeigte, lassen sich die Ergebnisse bezüglich der Bewertung, ob sich ein System nominal oder fehlerhaft verhält, sehr gut und eindeutig interpretierbar darstellen. Mithilfe der in Abschnitt 4.4.3 eingeführten und erläuterten Methoden zur Weiterverarbeitung des Verhaltenssignals \hat{B} , wird dem Gebäudebetreiber mit dem einstellbaren gleitenden Mittelwert ϕ ein Werkzeug zur Verfügung gestellt, mit dem er selbst entscheiden kann, ab wie vielen fehlerhaften Zeitschritten er informiert werden möchte.

Im Vergleich zu den Methoden des Stands der Technik der Fehlererkennung in der Gebäudeautomation, ergeben sich aus der Verwendung qualitativer Modelle zahlreiche Vorteile. Während die aktuellen Methoden lediglich auf der Überwachung von Grenzwerten und dem visuellen Auswerten von Messdaten beruhen, bieten qualitative Modelle die Möglichkeit, das Systemverhalten automatisiert und dynamisch zu überwachen. Zusätzlich erhält man durch die Darstellung der qualitativen Zustandswahrscheinlichkeiten eine anschauliche Interpretation des qualitativen Systemverhaltens. Durch die auf Grundlage des Beobachteralgorithmus 4.1 automatisierten Abläufe entfällt ein manuelles Sichten der Gebäudemessdaten und der Gebäudetreiber wird entsprechend entlastet.

Leider besitzen die qualitativen Modelle auch Nachteile, die es zu erwähnen gilt. Aufgrund der hier verwendeten und auf Messdaten basierenden Modellbildung, unterliegen die qualitativen Modelle den gleichen Nachteilen, wie alle anderen Black-Box Modelle. Werden bereits bei der Modellidentifikation fehlerhafte Betriebszustände einer Anlage trainiert, dann können diese später nicht mehr erkannt

werden und das Modell bewertet einen womöglich fehlerhaften Zeitbereich als nominal. Abhilfe kann hier nur eine sorgfältige Prüfung und Auswahl der Trainingsdaten schaffen, was häufig mit einem großen Zeitaufwand verbunden ist und dennoch nicht mit hundertprozentiger Sicherheit gewährleisten kann, tatsächlich nur fehlerfreie Daten zu erhalten. Ein weiterer Nachteil ist der überaus große Speicherbedarf der qualitativen Modelle, der es unmöglich macht, Modelle von großen Systemen mit vielen Systemgrößen oder von mehreren kleinen Systemen, effizient auf den Gebäudeleitrechnern oder auch auf dezentralen Minicomputern zu speichern. Eine Lösung für dieses Problems wird in den folgenden Kapiteln beschrieben. Die Abbildung 5.30 veranschaulicht ergänzend zu Tabelle 1.1 das Komplexitätsproblem stochastischer Automaten und qualitativer Modelle.

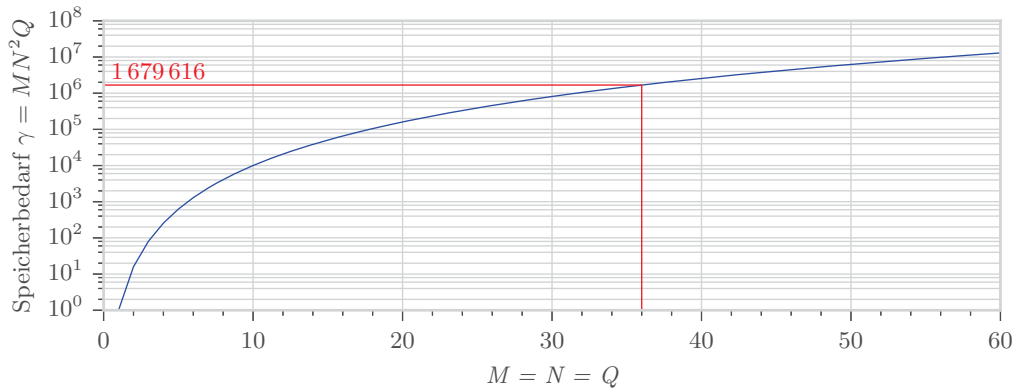


Abbildung 5.30: Speicherbedarf γ qualitativer Modelle für eine gleiche Anzahl qualitativer Eingänge M , Zustände N und Ausgänge Q

In der Abbildung 5.30 ist der Speicherbedarf des realen Erhitzers aus Abschnitt 5.3 mit den Modellparametern aus Tabelle 5.11 eingezeichnet. Der Erhitzer dient in den folgenden Kapiteln als Grundlage, um die darin vorgestellte Methode der *Tensordekomposition* zur Speicherreduktion stochastischer Automaten und qualitativer Modelle zu demonstrieren. Das folgende Kapitel führt dabei zunächst die benötigten Grundlagen der Tensoralgebra ein.

6

Lineare und multilineare Algebra

Dieses Kapitel beinhaltet eine Einführung in die lineare und multilineare Algebra und stellt grundlegende mathematische Produkte und Definitionen von Vektoren, Matrizen und Tensoren vor. In Abschnitt 6.1 erfolgt zunächst eine kurze Einleitung bezüglich der Darstellung von Matrizen, Vektoren und Tensoren, während in Abschnitt 6.2 die Tensoren und ihre Eigenschaften beschrieben werden. Grundlegende Operationen wie die Matrixisierung und Vektorisierung von Tensoren werden in Abschnitt 6.3 erläutert. Anschließend erfolgt in Abschnitt 6.4 eine ausführliche Beschreibung der in dieser Arbeit benötigten Tensor-, Matrix- und Vektorprodukte. Abschließend werden in Abschnitt 6.5 spezielle Tensoren eingeführt und deren Eigenschaften beschrieben.

6.1 Grundlagen

6.1.1 Notationen

Tabelle 6.1 zeigt die im Folgenden verwendeten Symbolkonventionen für Skalare, Vektoren, Matrizen und Tensoren. Für Tensoren sind außerdem die erweiterten Darstellungskonventionen in Tabelle 6.2 zu beachten.

Tabelle 6.1: Darstellungskonvention für Vektoren, Matrizen und Tensoren

Bedeutung	Konvention
Skalare: klein und kursiv, z. B.	$x \in \mathbb{R}$
Vektoren: klein und fett, z. B.	$\mathbf{x} \in \mathbb{R}^I$
Vektorkomponente:	$x_i \in \mathbb{R}$
Matrizen: groß und fett, z. B.	$\mathbf{X} \in \mathbb{R}^{I \times J}$
Matrixspalte, Spaltenvektor	$\mathbf{x}_{:j} \in \mathbb{R}^I$
Matrixzeile, Zeilenvektor	$\mathbf{x}_{i:} \in \mathbb{R}^J$
Matrixelement:	$x_{ij} \in \mathbb{R}$
Tensoren: groß und serifenlos, z. B.	$\mathbf{X} \in \mathbb{R}^{I \times J \times K}$
Tensorscheibe:	$\mathbf{X}_{:j:} \in \mathbb{R}^{I \times K}$
Tensorfiber:	$\mathbf{x}_{i:k} \in \mathbb{R}^J$
Tensorelement:	$x_{ijk} \in \mathbb{R}$

6.1.2 Vektoren und Matrizen

Ein *Vektor*

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_I \end{pmatrix} \in \mathbb{R}^I, \quad \mathbf{x}^\top = (x_1 \ x_2 \ \cdots \ x_I) \in \mathbb{R}^I \quad (6.1)$$

mit *Komponenten* $x_i \in \mathbb{R}$ und Indizes $i \in \{1, 2, \dots, I\}$ ist ein Element des I -dimensionalen reellen Vektorraums \mathbb{R}^I . Dabei stellt \mathbf{x}^\top den zu \mathbf{x} transponierten Vektor dar. Eine *Matrix* mit I Zeilen und J Spalten

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1} & x_{I2} & \cdots & x_{IJ} \end{pmatrix} \in \mathbb{R}^{I \times J} \quad (6.2)$$

ist eine tabellarische Anordnung von *Elementen* und wird als $I \times J$ Matrix bezeichnet. Hierbei ist $x_{ij} \in \mathbb{R}$ das Element der i -ten Zeile und der j -ten Spalte, wobei für die Indizes $i \in \{1, 2, \dots, I\}$ und $j \in \{1, 2, \dots, J\}$ gilt. I und J sind die Dimensionen der Matrix.

Index-Operator »:«

Wählt man etwa aus der Matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ in (6.2) die j -te Spalte aus, erhält man einen Spaltenvektor, der wie folgt notiert wird:

$$\mathbf{x}_{:j} = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{Ij} \end{pmatrix} \in \mathbb{R}^I. \quad (6.3)$$

Der *Index-Operator* »:« beschreibt somit die Auswahl *aller* Elemente die der jeweilige Index annehmen kann [56, S. 457].

6.2 Tensoren

Im Allgemeinen existieren in Abhängigkeit des Anwendungsgebiets verschiedene Definitionen von Tensoren. In Abgrenzung zur Mathematik und Physik, wo häufig Tensorfelder, Spannungstensoren oder Energie-Impuls-Tensoren behandelt werden, bezieht sich die Definition eines Tensors hier auf die Repräsentation mehrdimensionaler Datenstrukturen [56, S. 455]. In diesem Kontext werden Tensoren durch mehrdimensionale Arrays (N -dimensionale Arrays) repräsentiert [4, S. 635], [56, S. 455]. Entsprechende Literatur zu dieser Interpretation von Tensoren findet sich in [4], [25], [48], [56], [57]. Abbildung 6.1 zeigt einleitend die Vorstellung von Vektoren, Matrizen und eines 3-dimensionalen Tensors.

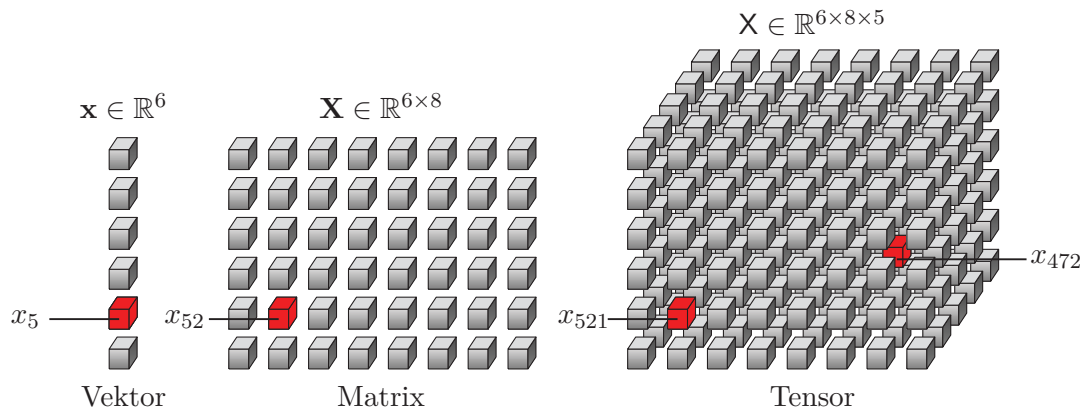


Abbildung 6.1: Vektor, Matrix und Tensor, nach [25, S. 29]

6.2.1 Definition

Ein Tensor ist wie folgt definiert.

Definition 6.1 (Tensor [25, S. 28]): *Ein Tensor N -ter Ordnung*

$$\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \quad (6.4)$$

ist ein N -dimensionales Array, dessen Elemente $x_{i_1 i_2 \dots i_N}$ mit $i_n \in \{1, 2, \dots, I_n\}$ für $n = 1, 2, \dots, N$ indiziert sind.

Die *Ordnung* oder *Stufe* eines Tensors (6.4) entspricht der Anzahl seiner *Dimensionen* oder *Modi* [25, S. 28]. Gleichbedeutend kann man formulieren, dass die Stufe oder Ordnung der Anzahl an Indizes entspricht, die man benötigt, um ein einzelnes Element des Tensors zu selektieren. Abbildung 6.2 zeigt dies beispielhaft anhand eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.

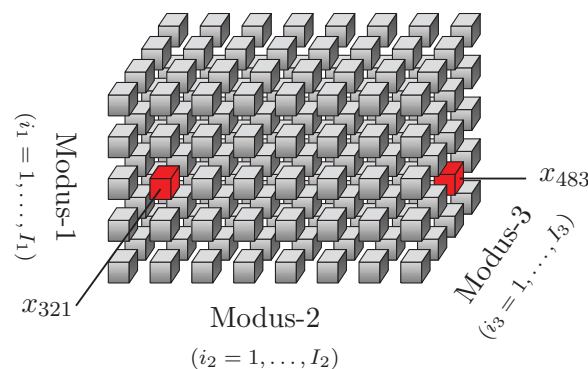


Abbildung 6.2: Elemente $x_{i_1 i_2 i_3}$ eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ mit $I_1 = 5$, $I_2 = 8$ und $I_3 = 4$, nach [25, S. 29]

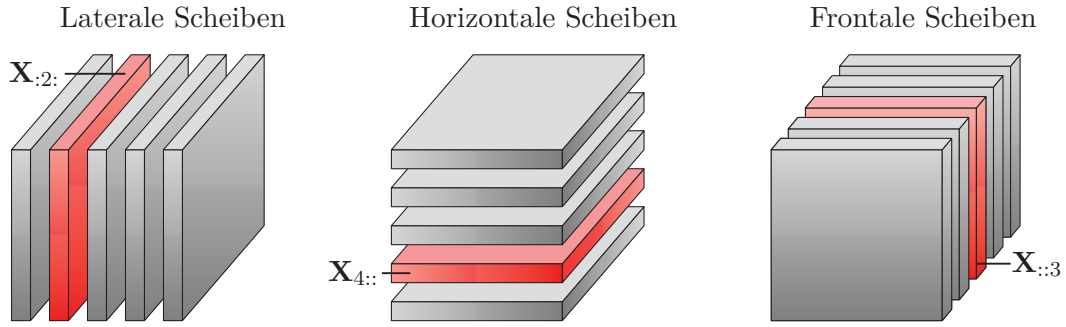
In Abhängigkeit der Ordnung eines Tensors existieren Analogien zu Skalaren, Vektoren und Matrizen. So ist beispielsweise ein Skalar $x \in \mathbb{R}$ ein Tensor der Ordnung null, ein Vektor $\mathbf{x} \in \mathbb{R}^I$ ist ein Tensor erster Ordnung und eine Matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ ist ein Tensor zweiter Ordnung [4, S. 637]. Ab einer Ordnung von drei oder größer spricht man von Tensoren *höherer* oder N -ter Ordnung [56, S. 455].

Fixiert man bei einem Tensor mindestens einen Index, dann erhält man einen *Subtensor* dessen Ordnung um die Anzahl der fixierten Indizes geringer ist. Für den in Abbildung 6.2 dargestellten Tensor dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ führt beispielsweise die Fixierung des ersten Index i_1 zu einer Matrix $\mathbf{X}_{i_1::} \in \mathbb{R}^{I_2 \times I_3}$, die auch als *Tensorscheibe* bezeichnet wird [56, S. 457].

6.2.2 Tensorscheiben

Definition 6.2 (Tensorscheibe [25, S. 29]): *Fixiert man bei einem Tensor höherer Ordnung ($N \geq 3$) alle Indizes bis auf zwei, dann erhält man eine Tensorscheibe, beziehungsweise eine Matrix.*

Definition 6.2 kann am Beispiel eines Tensor dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ leicht veranschaulicht werden. Wie Abbildung 6.3 zeigt, erhält man je nachdem welcher der drei Indizes i_1, i_2, i_3 fixiert wird, laterale $\mathbf{X}_{:i_2:} \in \mathbb{R}^{I_1 \times I_3}$, horizontale $\mathbf{X}_{i_1::} \in \mathbb{R}^{I_2 \times I_3}$ oder frontale $\mathbf{X}_{::i_3} \in \mathbb{R}^{I_1 \times I_2}$ Scheiben [56, S. 457]. Neben den Tensorscheiben, die das Äquivalent einer Matrix darstellen, können sogenannte *Tensorfibern* als Analogie zu Vektoren definiert werden.

Abbildung 6.3: Tensorscheiben eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, nach [25, S. 31]

6.2.3 Tensorfibern

Definition 6.3 (Tensorfiber [25, S. 29]): *Fixiert man bei einem Tensor höherer Ordnung ($N \geq 2$) alle Indizes bis auf einen, dann erhält man eine Tensorfiber, beziehungsweise einen Vektor.*

Auch die Tensorfibern werden anhand ihrer Orientierung unterschieden. Bei einem wie in Abbildung 6.4 dargestellten Tensor dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, unterscheidet man anhand seiner Modi, zwischen Modus-1 $\mathbf{x}_{:i_2 i_3} \in \mathbb{R}^{I_1}$, Modus-2 $\mathbf{x}_{i_1 : i_3} \in \mathbb{R}^{I_2}$ und Modus-3 $\mathbf{x}_{i_1 i_2 :} \in \mathbb{R}^{I_3}$ Fibern [56, S. 458]. Werden Berechnungen mit den Tensor-Fibern angestellt, werden diese unabhängig von ihrer Orientierung als Spaltenvektoren interpretiert [25, S. 31], [56, S. 457].

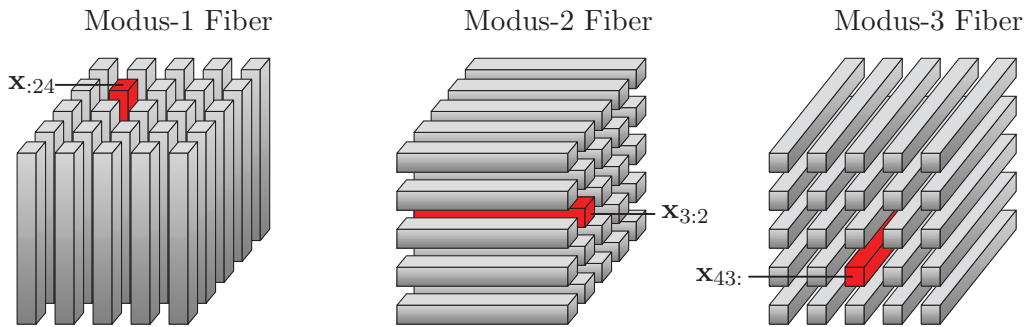
Abbildung 6.4: Tensorfibern eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, nach [25, S. 31]

Tabelle 6.2 gibt einen abschließenden Überblick bezüglich der bestehenden Analogien zwischen Tensoren N -ter Ordnung und deren Subtensoren, die man durch Fixierung eines oder mehrerer Indizes erhält.

Tabelle 6.2: Indexnotation für Tensoren, nach [23, S. 146], [5, S. 207–208]

Darstellung	Bezeichnung
$\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$	Tensor N -ter Ordnung
$x_{i_1 i_2 \dots i_N} \in \mathbb{R}$	(i_1, i_2, \dots, i_N) -tes Element von \mathbf{X} , Subtensor der
$x(i_1, i_2, \dots, i_N) \in \mathbb{R}$	Ordnung null, Skalar
$\mathbf{x}_{:i_2 \dots i_N} \in \mathbb{R}^{I_1}$	Tensorfiber von \mathbf{X} , Subtensor erster Ordnung, Vektor
$\mathbf{x}(:, i_2, \dots, i_N) \in \mathbb{R}^{I_1}$	
$\mathbf{X}_{::i_3 \dots i_N} \in \mathbb{R}^{I_1 \times I_2}$	Tensorscheibe von \mathbf{X} , Subtensor zweiter Ordnung, Matrix
$\mathbf{X}(:, :, i_3, \dots, i_N) \in \mathbb{R}^{I_1 \times I_2}$	
$\mathbf{X}_{:::i_4 \dots i_N} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$	Subtensor dritter Ordnung
$\mathbf{X}(:, :, :, i_4, \dots, i_N) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$	

Neben der Festlegung der Bezeichnung verschiedener Subtensoren zeigt die Tabelle 6.2 in der linken Spalte eine alternative Notation zur Angabe der Tensorindizes. Die Indizes eines Tensors können demnach entweder tiefgestellt oder kommasepariert in runden Klammern angegeben werden. Das heißt, die Selektion einer Modus-2 Fiber eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ kann anstatt durch $\mathbf{x}_{i_1:i_3} \in \mathbb{R}^{I_2}$ auch durch die Notation $\mathbf{x}(i_1, :, i_3) \in \mathbb{R}^{I_2}$ ausgedrückt werden. Während erstere Notation sehr kompakt ist, bietet die zweite Darstellung Vorteile in Bezug auf die Lesbarkeit. Dies gilt vor allem dann, wenn die entsprechenden Tensoren viele Dimensionen besitzen oder die Indizes zeitabhängig sind, sodass ein zusätzlicher Zeitindex angehängt werden muss. Die beiden Darstellungsmöglichkeiten werden im Folgenden in Abhängigkeit des jeweiligen Kontexts und der Interpretierbarkeit äquivalent verwendet.

6.3 Tensortransformationen

Tensoren können in Matrizen und Vektoren transformiert werden, indem man ihre Elemente in einer entsprechenden zweidimensionalen oder eindimensionalen Struktur anordnet. Das Umordnen der Elemente eines Tensors höherer Ordnung in eine Matrixform wird als *Matrizisierung* (*engl.*: *Matricization*) und die Transformation in einen Vektor als *Vektorisierung* (*engl.*: *Vectorization*) bezeichnet [48, S. 109–111]. Hierbei existieren für beide Transformationen feste Regeln, die definieren in welcher Reihenfolge die Elemente eines Tensors in der jeweiligen Zielstruktur angeordnet werden. Diese Festlegung erfolgt durch einen sogenannten *Multiindex*, der in Abschnitt 6.3.1 eingeführt wird. Neben der Matrizisierung und Vektorisierung von Tensoren können umgekehrt auch Matrizen und Vektoren in Tensoren transformiert werden, was man als *Tensorisierung* (*engl.*: *Tensorization*) bezeichnet [22, S. 277]. Eine Übersicht, die den Zusammenhang zwischen allen oben genannten Transformationen zeigt, befindet sich in [22, S. 274]. Die Tensorisierung findet in dieser Arbeit keine Anwendung. Die Vektorisierung und Matrizisierung werden in den Abschnitten 6.3.2 und 6.3.3 erläutert.

6.3.1 Multiindex

Definition 6.4 (Multiindex [63, S. 601]): *Indizes $j_n \in \{1, 2, \dots, J_n\}$, die für $n = 1, 2, \dots, N$ gegeben sind, werden durch den Multiindex*

$$\begin{aligned} \overline{j_1 j_2 \dots j_N} &= j_N + (j_{N-1} - 1)J_N + (j_{N-2} - 1)J_{N-1}J_N + \dots \\ &\quad + (j_2 - 1)J_3 J_4 \dots J_N + (j_1 - 1)J_2 J_3 J_4 \dots J_N \in \mathbb{N} \end{aligned} \quad (6.5)$$

auf einen ganzzahligen skalaren Wert abgebildet.

Der in Definition 6.4 festgelegte Multiindex entspricht der sogenannten *big-endian* Konvention [63, S. 601]. Durch diese Definition wird sowohl für die Matrizisierung als auch für die Vektorisierung festgelegt, an welcher Position die Elemente eines Tensors in einer Matrix oder dem Vektor angeordnet werden. Neben der *big-endian* Konvention, die ihren Ursprung in der Programmiersprache C hat und standardmäßig in PYTHONTM zum Einsatz kommt, existiert die *little-endian* Konvention, die beispielsweise in MATLAB[®] und FORTRAN verwendet wird [22, S. 274–275]. Beide Konventionen sind unter anderem in [22, S. 274], [64, S. 924] beschrieben und führen bei der Matrizisierung und Vektorisierung zu unterschiedlichen Anordnungen der Tensorelemente in der jeweiligen Zielstruktur (s. Anmerkung 6.1). Grundsätzlich ist es dabei unerheblich, für welche der beide Varianten man sich entscheidet, solange man diese konsistent verwendet [56, S. 460]. In der Praxis erweist es sich jedoch als intuitiver und weniger fehleranfällig, die Konvention zu verwenden, die der zugrundeliegenden Programmiersprache entspricht (s. a. Abschnitt 6.3.2).

6.3.2 Matrizisierung

Bei der Matrizisierung eines Tensors wird im Wesentlichen zwischen zwei verschiedenen Methoden unterschieden, die in der Literatur als *Modus- n Matrizisierung* (engl.: mode- n matricization) [22], [56], [63] und *kanonische Modus- n Matrizisierung* (engl.: mode- n canonical matricization) [22] bekannt sind. Die kanonische Modus- n Matrizisierung wird auch als *n -te kanonische Matrizisierung* (engl.: n -th canonical matricization) [63] oder Modus- $(1, 2, \dots, N)$ Matrizisierung [62], [64] bezeichnet.

Modus- n Matrizisierung

Ordnet man die Modus- n Fibern eines Tensors als Spaltenvektoren einer Matrix an, spricht man von einer Modus- n Matrizisierung (s. Abbildung 6.5).

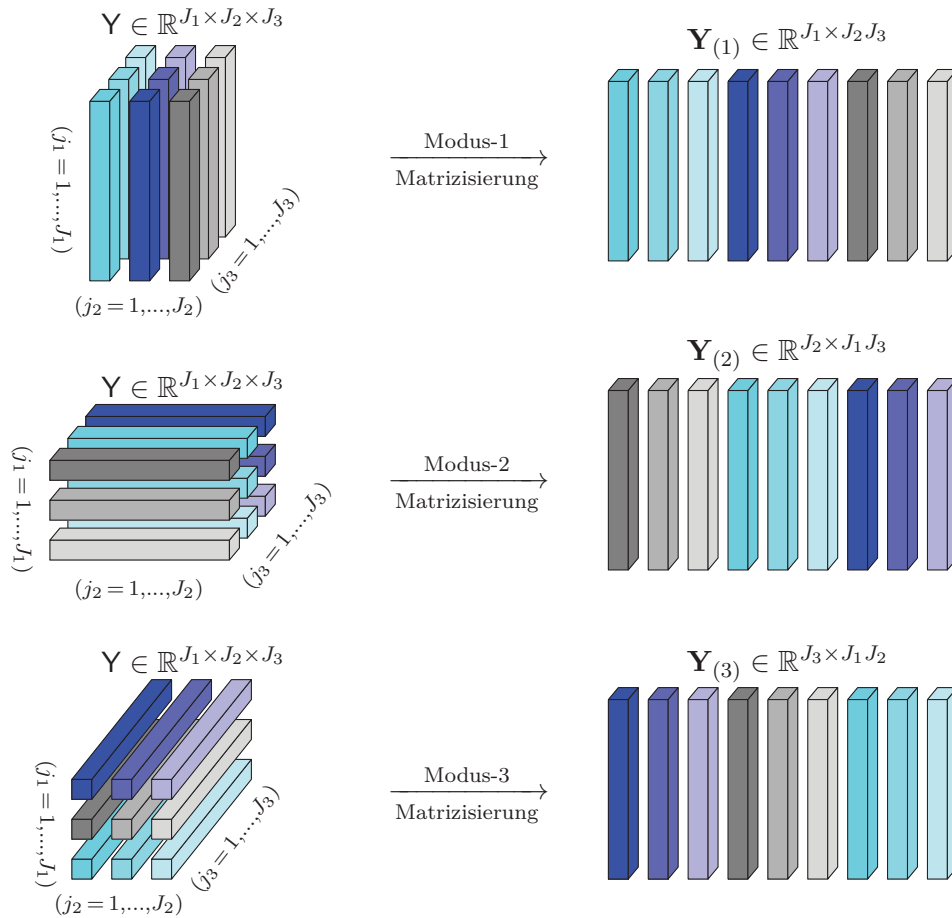


Abbildung 6.5: Modus- n Matrizisierung

Die in Abbildung 6.5 dargestellten Neuaneordnungen der Modus- n Fibern des Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ ergeben sich aus der Festlegung des Multiindex (6.5) sowie der folgenden Definition der Modus- n Matrizisierung.

Definition 6.5 (Modus- n Matrizisierung [22, S. 275]): Die Matrizisierung eines Tensors N -ter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ entlang eines definierten Modus $n \in \{1, \dots, N\}$ ergibt eine Matrix

$$\mathbf{Y}_{(n)} \in \mathbb{R}^{J_n \times J_1 J_2 \dots J_{n-1} J_{n+1} \dots J_N} \quad (6.6)$$

mit Elementen

$$(\mathbf{Y}_{(n)})_{j_n \overline{j_1 \dots j_{n-1} j_{n+1} \dots j_N}} = y(j_1, j_2, \dots, j_N). \quad (6.7)$$

Die Notation (6.7) bedeutet, dass das Tensorelement $y(j_1, j_2, \dots, j_N)$ durch die Matrizisierung nun an der Position $(j_n, \overline{j_1 \dots j_{n-1} j_{n+1} \dots j_N})$ in der Matrix $\mathbf{Y}_{(n)}$ steht. Die Berechnung des Multiindex in (6.7) kann mit

$$\begin{aligned} \overline{j_1 \dots j_{n-1} j_{n+1} \dots j_N} &= j_N + (j_{N-1} - 1)J_N + \dots + (j_{n+1} - 1)J_{n+2} \dots J_N \\ &\quad + (j_{n-1} - 1)J_{n+1} \dots J_N + \dots + (j_1 - 1)J_2 J_{n-1} J_{n+1} \dots J_N \end{aligned} \quad (6.8)$$

auch explizit angegeben werden (vgl. Definition 6.4).

Beispiel 6.1 (Modus-1 Matrizisierung): Gegeben sei der in Abbildung 6.6 dargestellte Tensor dritter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ mit den Elementen $y_{j_1 j_2 j_3}$ sowie den Dimensionsgrößen $J_1 = 2$, $J_2 = 3$ und $J_3 = 2$.

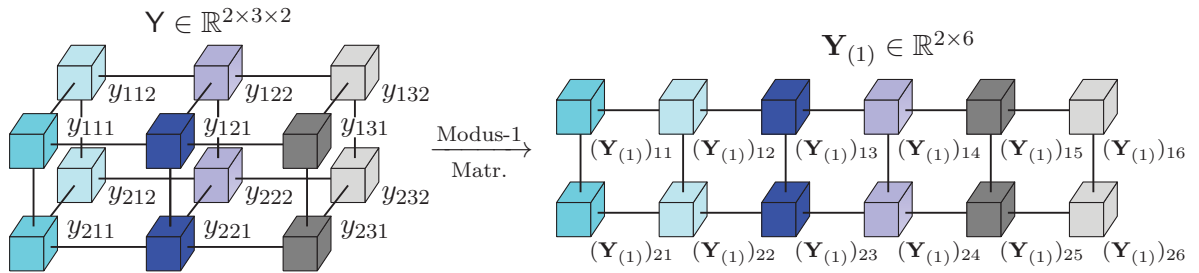


Abbildung 6.6: Modus-1 Matrizisierung

Die in Abbildung 6.6 dargestellte Modus-1 Matrizisierung des Tensors $\mathbf{Y} \in \mathbb{R}^{2 \times 3 \times 2}$ führt unter Verwendung von Gleichung (6.7) zu der Matrix $\mathbf{Y}_{(1)} \in \mathbb{R}^{2 \times 6}$ (vgl. Abbildung 6.5). Dabei vereinfacht sich die Gleichung (6.7) für den dreidimensionalen Fall ($N = 3$) und die Modus-1 Matrizisierung ($n = 1$) zu

$$(\mathbf{Y}_{(1)})_{j_1 \overline{j_2 j_3}} = y_{j_1 j_2 j_3} \quad \text{mit} \quad \overline{j_2 j_3} = j_3 + (j_2 - 1) \cdot J_3.$$

Für ein Tensorelement y_{132} ergibt sich aus dem Multiindex $\overline{32} = 2 + (3 - 1) \cdot 2 = 6$ und $j_1 = 1$ die Beziehung

$$(\mathbf{Y}_{(1)})_{16} = y_{132}.$$

Anmerkung 6.1 (Modus- n Matrizisierung): Wie man in Abbildung 6.5 sieht, werden bei der Modus- n Matrizisierung die Modus- n Fibern eines Tensors als Spaltenvektoren einer Matrix angeordnet. Die Reihenfolge in der diese Umordnung geschieht, wird bestimmt durch die in Definition 6.4 festgelegte big-endian Konvention des Multiindex. Diese Art der Matrizisierung geht zurück auf die ursprünglich in [28, S. 1255–1256] eingeführte Definition der Matrizisierung. Eine alternative Art der Matrizisierung, die zu einer anderen Anordnung der Modus- n Fibern führt, wird in [48, S. 109–111] beschrieben. Die dort dargestellte Anordnung der Modus- n Fibern entspricht derjenigen, die man unter Verwendung der little-endian Konvention des Multiindex erhalten würde. Beide Varianten der Matrizisierung sind in [4, S. 647] anschaulich gegenübergestellt.

Kanonische Modus- n Matrizisierung

Wie die folgende Abbildung zeigt, werden bei der kanonischen Modus- n Matrizisierung, die Modus- n Fibern des Ursprungstensors nicht nur nebeneinander, sondern auch untereinander angeordnet.

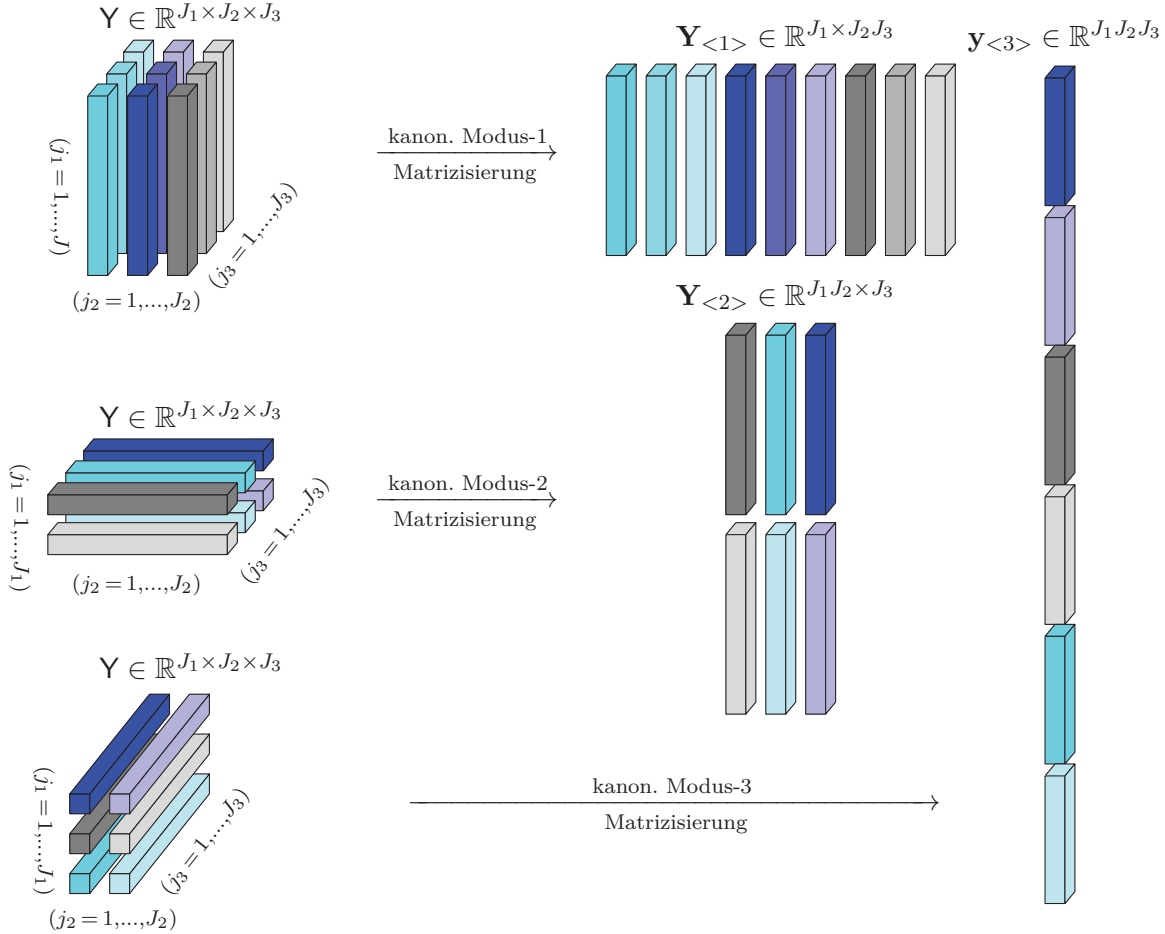


Abbildung 6.7: Kanonische Modus- n Matrizisierung

Definition 6.6 (Kanonische Modus- n Matrizisierung [22, S. 275]): Die kanonische Modus- n Matrizisierung eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ entlang eines definierten Modus $n \in \{1, \dots, N\}$ ergibt eine Matrix

$$\mathbf{Y}_{<n>} \in \mathbb{R}^{J_1 J_2 \dots J_n \times J_{n+1} J_{n+2} \dots J_N} \quad (6.9)$$

mit Elementen

$$(\mathbf{Y}_{<n>})_{\overline{j_1 j_2 \dots j_n} \overline{j_{n+1} j_{n+2} \dots j_N}} = y(j_1, j_2, \dots, j_N). \quad (6.10)$$

Wie ein Vergleich der Abbildungen 6.5 und 6.7 zeigt, ist die kanonische Modus-1 Matrizisierung mit der Modus-1 Matrizisierung identisch, womit folgendes gilt:

$$\mathbf{Y}_{<1>} \equiv \mathbf{Y}_{(1)}. \quad (6.11)$$

In Abbildung 6.7 wird außerdem deutlich, dass wenn man einen Tensor N -ter Ordnung entlang des letzten, also des N -ten Modus matriziert, das Ergebnis ein Vektor ist. Diese Operation ist identisch mit der Vektorisierung eines Tensors, die im nächsten Abschnitt beschrieben wird.

6.3.3 Vektorisierung

Definition 6.7 (Vektorisierung [63, S. 601]): Die Vektorisierung eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ der Ordnung N ist definiert als die N -te kanonische Matrizierung

$$\text{vec}(\mathbf{Y}) \equiv \mathbf{y}_{\langle N \rangle} \in \mathbb{R}^{J_1 J_2 \dots J_N} \quad (6.12)$$

und ergibt einen Vektor mit Elementen

$$(\text{vec}(\mathbf{Y}))_{\overline{j_1 j_2 \dots j_N}} = y(j_1, j_2, \dots, j_N) . \quad (6.13)$$

6.4 Tensor- und Matrixprodukte

Die folgende Tabelle 6.3 gibt einen Überblick bezüglich der in diesem Abschnitt eingeführten Tensor- und Matrixprodukte. Dabei sei angemerkt, dass alle im Folgenden beschriebenen Produkte in entsprechenden Toolboxes in MATLAB[®] und PYTHON[™] zur Verfügung stehen. Eine Übersicht dazu findet sich in Tabelle 8.1 in Kapitel 8.

Tabelle 6.3: Tensor- und Matrixprodukte

Symbol	Bedeutung
\cdot	Matrixprodukt, Matrix-Vektor Produkt
\otimes	Hadamard-Produkt
\otimes	Kronecker-Produkt
\circ	Dyadisches Produkt
$\bar{\times}_{(n)}$	Modus- n Tensor-Vektor Produkt
\odot	Kontrahiertes Modus- $(P 1)$ Tensorprodukt
$\langle \cdot \cdot \rangle$	Kontrahiertes Tensorprodukt

6.4.1 Hadamard-Produkt

Das *Hadamard-Produkt* ist das elementweise Produkt zweier oder mehrerer dimensionsgleicher Tensoren.

Definition 6.8 (Hadamard-Produkt): Das *Hadamard-Produkt* zweier Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ und $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ ist ein Tensor

$$\mathbf{Z} = \mathbf{X} \otimes \mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N} \quad (6.14)$$

mit Elementen

$$z(j_1, j_2, \dots, j_N) = x(j_1, j_2, \dots, j_N) \cdot y(j_1, j_2, \dots, j_N) . \quad (6.15)$$

Sind mehrere Tensoren N -ter Ordnung $\mathbf{X}^{(k)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ für $k = 1, \dots, K$ gegeben, dann kann ihr Hadamard-Produkt kompakt notiert werden:

$$\mathbf{Z} = \mathbf{X}^{(1)} \otimes \mathbf{X}^{(2)} \otimes \dots \otimes \mathbf{X}^{(K)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N} \quad (6.16)$$

$$= \bigotimes_{k=1}^K \mathbf{X}^{(k)} . \quad (6.17)$$

Das Hadamard-Produkt ist ebenso auf dimensionsgleiche Matrizen und Vektoren anwendbar.

6.4.2 Dyadisches Vektorprodukt

Das *dyadische Vektorprodukt* wird auch als äußeres Vektorprodukt (*engl.: vector outer product*) bezeichnet und stellt einen Sonderfall des allgemeinen dyadischen Produkts (*engl.: outer product*) dar, welches auch auf Tensoren beliebiger Ordnung anwendbar ist [25, S. 34].

Definition 6.9 (Dyadisches Vektorprodukt [5, S. 208], [25, S. 34–35]): Das dyadische Produkt von N Vektoren, die durch $\mathbf{x}^{(n)} \in \mathbb{R}^{J_n}$ für $n = 1, 2, \dots, N$ gegeben sind, ist ein Tensor N -ter Ordnung

$$\mathbf{Y} = \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \dots \circ \mathbf{x}^{(N)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N} \quad (6.18)$$

$$= \bigcirc_{n=1}^N \mathbf{x}^{(n)} \quad (6.19)$$

mit Elementen

$$y(j_1, j_2, \dots, j_N) = x_{j_1}^{(1)} \cdot x_{j_2}^{(2)} \cdot \dots \cdot x_{j_N}^{(N)} \quad (6.20)$$

und Indizes $j_n \in \{1, 2, \dots, J_n\}$.

Abbildung 6.9 visualisiert anschaulich das dyadische Vektor-Produkt dreier Vektoren $\mathbf{x}^{(1)} \in \mathbb{R}^I$, $\mathbf{x}^{(2)} \in \mathbb{R}^J$ und $\mathbf{x}^{(3)} \in \mathbb{R}^K$, was zu einem Produkttensor dritter Ordnung $\mathbf{Y} \in \mathbb{R}^{I \times J \times K}$ führt.

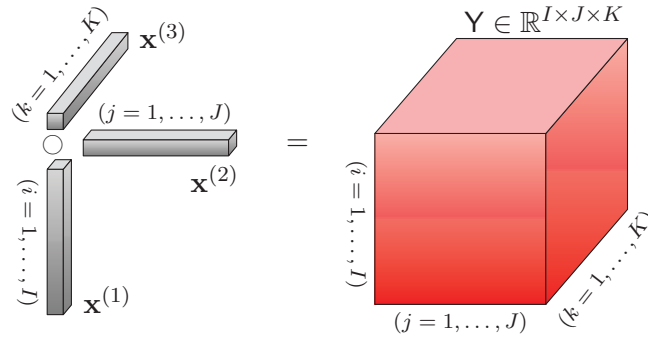


Abbildung 6.8: Generische Darstellung des dyadischen Vektorprodukts dreier Vektoren $\mathbf{x}^{(1)} \in \mathbb{R}^I$, $\mathbf{x}^{(2)} \in \mathbb{R}^J$ und $\mathbf{x}^{(3)} \in \mathbb{R}^K$

Beispiel 6.2 (Dyadisches Vektorprodukt): Gegeben seien die drei Vektoren $\mathbf{x}^{(1)} \in \mathbb{R}^I$, $\mathbf{x}^{(2)} \in \mathbb{R}^J$ und $\mathbf{x}^{(3)} \in \mathbb{R}^K$ mit Komponenten $x_i^{(1)}$, $x_j^{(2)}$, $x_k^{(3)}$. Wie in Abbildung 6.8 dargestellt, führt das dyadische Produkt dieser Vektoren zu einem Tensor dritter Ordnung $\mathbf{Y} \in \mathbb{R}^{I \times J \times K}$ mit Elementen y_{ijk} . Sei nun die Länge der Vektoren durch $I = J = 3$ und $K = 2$ gegeben, ergibt sich die elementweise Darstellung des dyadischen Produkts in Abbildung 6.9.

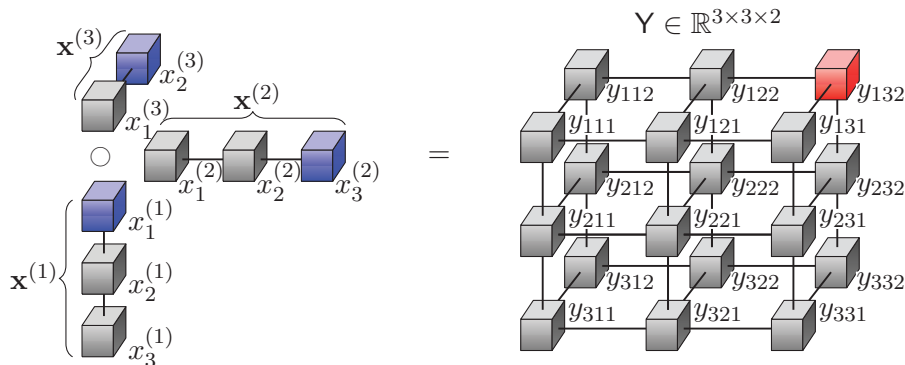


Abbildung 6.9: Dyadisches Produkt dreier Vektoren $\mathbf{x}^{(1)} \in \mathbb{R}^3$, $\mathbf{x}^{(2)} \in \mathbb{R}^3$ und $\mathbf{x}^{(3)} \in \mathbb{R}^2$

Das in rot eingezeichnete Element y_{132} des Produkttensors $\mathbf{Y} \in \mathbb{R}^{3 \times 3 \times 2}$ erhält man durch Multiplikation der blauen Komponenten der drei Vektoren. Dieser Zusammenhang ergibt sich aus Gleichung (6.20), welche die elementweise Berechnung des dyadischen Vektorprodukts beschreibt und für das Beispiel folgende Form annimmt:

$$y_{ijk} = x_i^{(1)} \cdot x_j^{(2)} \cdot x_k^{(3)} .$$

6.4.3 Kronecker-Produkt

Definition 6.10 (Kronecker-Produkt [25, S. 35]): Das Kronecker-Produkt der Matrizen $\mathbf{X} \in \mathbb{R}^{I \times J}$ und $\mathbf{Y} \in \mathbb{R}^{K \times L}$ ist eine Matrix

$$\mathbf{Z} = \mathbf{X} \otimes \mathbf{Y} = \begin{pmatrix} x_{11}\mathbf{Y} & x_{12}\mathbf{Y} & \dots & x_{1J}\mathbf{Y} \\ x_{21}\mathbf{Y} & x_{22}\mathbf{Y} & \dots & x_{2J}\mathbf{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1}\mathbf{Y} & x_{I2}\mathbf{Y} & \dots & x_{IJ}\mathbf{Y} \end{pmatrix} \in \mathbb{R}^{(IK) \times (JL)} . \quad (6.21)$$

Wie aus (6.21) ersichtlich wird, lässt sich das Kronecker-Produkt auch auf Matrizen mit unterschiedlicher Spalten- und Zeilenanzahl anwenden. In dieser Arbeit findet das Kronecker-Produkt ausschließlich Anwendung in der Multiplikation von Vektoren. Die Definition des Kronecker-Produkts von Vektoren folgt aus der allgemeinen Definition 6.10.

Definition 6.11 (Kronecker-Produkt von Vektoren): Das Kronecker-Produkt der Vektoren $\mathbf{x} \in \mathbb{R}^I$ und $\mathbf{y} \in \mathbb{R}^J$ ergibt einen Vektor

$$\mathbf{z} = \mathbf{x} \otimes \mathbf{y} = \begin{pmatrix} x_1 y_1 \\ x_1 y_2 \\ \vdots \\ x_1 y_J \\ x_2 y_1 \\ x_2 y_2 \\ \vdots \\ x_I y_J \end{pmatrix} \in \mathbb{R}^{IJ} . \quad (6.22)$$

Sind nicht nur zwei, sondern N Vektoren $\mathbf{x}^{(n)} \in \mathbb{R}^{I_n}$ für $n = 1, 2, \dots, N$ gegeben, dann ist ihr Kronecker-Produkt durch den Vektor

$$\mathbf{z} = \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \dots \otimes \mathbf{x}^{(N)} \in \mathbb{R}^{I_1 I_2 \dots I_N} \quad (6.23)$$

$$= \bigotimes_{n=1}^N \mathbf{x}^{(n)} \quad (6.24)$$

gegeben.

Das Kronecker-Produkt kann auch mithilfe des Multiindex definiert werden [64, S. 924]. Sind wie in Gleichung (6.23) die Vektoren $\mathbf{x}^{(n)} \in \mathbb{R}^{I_n}$ für $n = 1, 2, \dots, N$ gegeben und besitzen die Komponenten $x_{i_n}^{(n)}, i_n \in \{1, 2, \dots, I_n\}$, dann ist

$$z_{\overline{i_1 i_2 \dots i_N}} = x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_N}^{(N)} \quad (6.25)$$

die $(i_1 i_2 \dots i_N)$ -te Komponente des Vektors $\mathbf{z} \in \mathbb{R}^{I_1 I_2 \dots I_N}$. Man schreibt implizit auch

$$\left(\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \dots \otimes \mathbf{x}^{(N)} \right)_{i_1 i_2 \dots i_N} = x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_N}^{(N)}, \quad (6.26)$$

wobei der Ausdruck auf der linken Seite die Komponente des Produktvektors darstellt [64, S. 924]. Wird wie in dieser Arbeit die big-endian Konvention des Multiindex nach Definition 6.4 verwendet, dann gilt außerdem [22, S. 275]:

$$\text{vec} \left(\bigcirc_{n=1}^N \mathbf{x}^{(n)} \right) \equiv \bigotimes_{n=1}^N \mathbf{x}^{(n)} \in \mathbb{R}^{I_1 I_2 \dots I_N}. \quad (6.27)$$

6.4.4 Modus- n Tensor-Vektor Produkt

Das *Modus- n Tensor-Vektor Produkt* beschreibt die Multiplikation eines Tensors N -ter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ entlang eines beliebigen Modus $n \in \{1, 2, \dots, N\}$ mit einem Vektor $\mathbf{y} \in \mathbb{R}^{I_n}$ [56, S. 461]. Dabei muss die Anzahl der Elemente entlang des n -ten Tensor Modus mit der Anzahl der Vektorkomponenten übereinstimmen. Das Ergebnis des Modus- n Tensor-Vektor Produkts ist ein Tensor der Ordnung $N - 1$, da der n -te Modus entfällt [4, S. 641–642] (s. Abbildung 6.10).

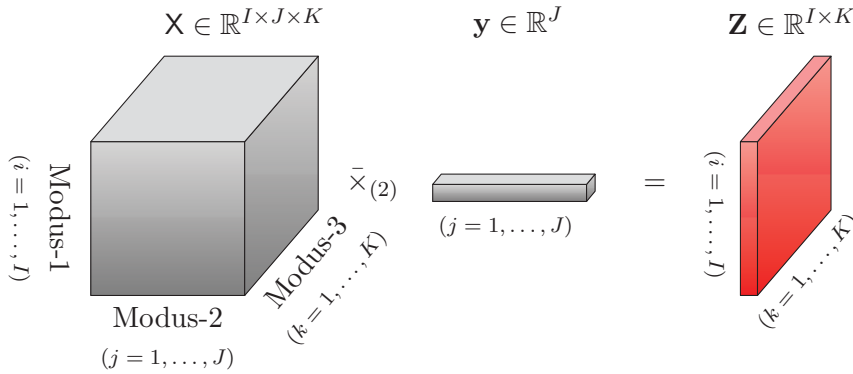


Abbildung 6.10: Generische Darstellung des Modus-2 Tensor-Vektor Produkt eines Tensors dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ mit einem Vektor $\mathbf{y} \in \mathbb{R}^J$

Definition 6.12 (Modus- n Tensor-Vektor Produkt [56, S. 461]): Das *Modus- n Produkt* eines Tensors $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ und eines Vektors $\mathbf{y} \in \mathbb{R}^{I_n}$ ist ein Tensor

$$\mathbf{Z} = \mathbf{X} \bar{\times}_{(n)} \mathbf{y} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N} \quad (6.28)$$

mit Elementen

$$z(i_1, \dots, i_{n-1}, i_{n+1}, \dots, i_N) = \sum_{i_n=1}^{I_n} x(i_1, i_2, \dots, i_N) \cdot y_{i_n}. \quad (6.29)$$

Das folgende Beispiel zeigt die elementweise Berechnung des Modus-2 Tensor-Vektor Produkts.

Beispiel 6.3 (Modus-2 Tensor-Vektor Produkt): Neben der generischen Darstellung der Modus-2 Multiplikation in Abbildung 6.10, kann auch die Berechnung der Elemente des Produkttensors grafisch demonstriert werden. Als Beispiel dient wieder ein Tensor dritter Ordnung $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ mit Elementen x_{ijk} sowie ein Vektor $\mathbf{y} \in \mathbb{R}^J$ mit den Komponenten y_j . Für die Dimensionsgrößen gilt $I = K = 2$

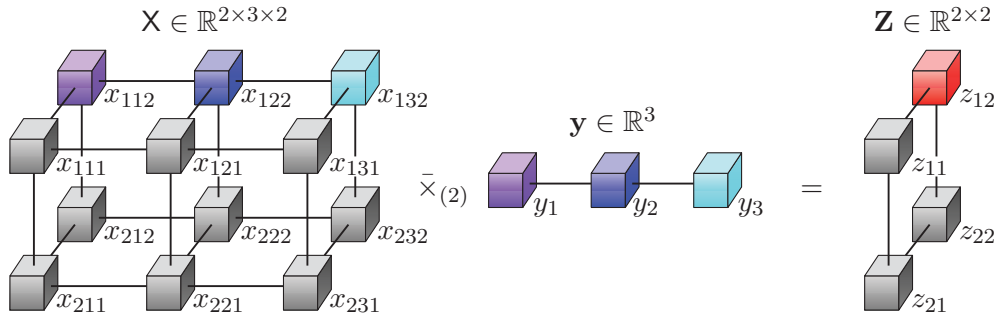


Abbildung 6.11: Modus-2 Tensor-Vektor Produkt eines Tensor $\mathbf{X} \in \mathbb{R}^{2 \times 3 \times 2}$ mit einem Vektor $\mathbf{y} \in \mathbb{R}^3$

und $J = 3$. Abbildung 6.11 zeigt die Berechnung der Elemente z_{ik} des Ergebnistensors $\mathbf{Z} \in \mathbb{R}^{I \times K}$. Um zum Beispiel das rote Element z_{12} zu berechnen, werden alle gleichfarbigen Tensorelemente und Vektorkomponenten miteinander multipliziert und die resultierenden Produkte aufsummiert.

Für das Beispiel in Abbildung (6.11) nimmt die Gleichung (6.29) die Form

$$z_{ik} = \sum_{j=1}^J x_{ijk} \cdot y_j$$

an und das Element z_{12} berechnet sich mit

$$z_{12} = \sum_{j=1}^3 x_{1j2} \cdot y_j = x_{112} \cdot y_1 + x_{122} \cdot y_2 + x_{132} \cdot y_3.$$

Das Modus- n Tensor-Vektor Produkt gehört zu den sogenannten *kontrahierten* Tensorprodukten und kann als Spezialfall des allgemeinen kontrahierten Tensorprodukts interpretiert werden (s. Abschnitt 6.4.6).

Anmerkung 6.2 (Vektor Orientierung): In den Abbildungen 6.10 und 6.11 wurde der Vektor $\mathbf{y} \in \mathbb{R}^J$ horizontal eingezeichnet, um den Zusammenhang zwischen den korrespondierenden Dimensionen der Länge J , entlang derer das Modus-2 Tensor-Vektor Produkt gebildet wird, hervorzuheben. Dabei handelt es sich um eine rein grafische Interpretation, denn mathematisch wird der im Modus- n Tensor-Vektor Produkt (6.28) auftretende Vektor $\mathbf{y} \in \mathbb{R}^{I_n}$, unabhängig vom Modus entlang dessen das Produkt gebildet wird, als Spaltenvektor interpretiert (vgl. Definition 6.9 und [56, S. 461]). Eine andere grafische Darstellung des Modus- n Tensor-Vektor Produkts findet sich beispielsweise in [25, S. 38]. Diese Anmerkung zur grafischen Interpretation der Orientierung von Vektoren gilt entsprechend auch für die Illustration des dyadischen Produkts in den Abbildungen 6.8 und 6.9.

6.4.5 Kontrahiertes Modus- $(P | 1)$ Tensorprodukt

Das *kontrahierte Modus- $(P | 1)$ Tensorprodukt* beschreibt die Multiplikation eines Tensors P -ter Ordnung $\mathbf{X} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_P}$ entlang des letzten Modus (Modus- P), mit dem ersten Modus (Modus-1) eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_P \times K_2 \times \dots \times K_S}$ [62, S. 996]. Vereinfacht wird dann von einer Modus- $(P | 1)$ *Kontraktion* gesprochen. Der P -te Modus des Tensors \mathbf{X} und der erste Modus des Tensors \mathbf{Y} müssen dabei dimensionsgleich sein, also die gleiche Anzahl von Elementen besitzen. Alternativ wird dieses Produkt auch als kontrahiertes Modus-1 Produkt bezeichnet [63, S. 600–601]. Anstelle des hier verwendeten Symbols \odot sind in der Literatur auch die Symbole \bullet [62], $[63]$ und \times^1 [64] üblich.

Definition 6.13 (Kontrahiertes Modus-($P|1$) Tensorprodukt [62, S. 996]): Das kontrahierte Modus-($P|1$) Produkt zweier Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_P}$ und $\mathbf{Y} \in \mathbb{R}^{J_P \times K_2 \times \dots \times K_S}$ ergibt einen Tensor

$$\mathbf{Z} = \mathbf{X} \odot \mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_{P-1} \times K_2 \times \dots \times K_S} \quad (6.30)$$

mit Elementen

$$z(j_1, j_2, \dots, j_{P-1}, k_2, \dots, k_S) = \sum_{j_P=1}^{J_P} x(j_1, j_2, \dots, j_P) \cdot y(j_P, k_2, \dots, k_S). \quad (6.31)$$

Abbildung 6.12 zeigt die Modus-($2|1$) Kontraktion eines Tensors zweiter Ordnung $\mathbf{X} \in \mathbb{R}^{I \times J}$ mit einem dreidimensionalen Tensor $\mathbf{Y} \in \mathbb{R}^{J \times K \times L}$. Es wird also der letzte Modus (Modus-2) der Matrix \mathbf{X} , mit dem ersten Modus (Modus-1) des Tensors \mathbf{Y} kontrahiert.

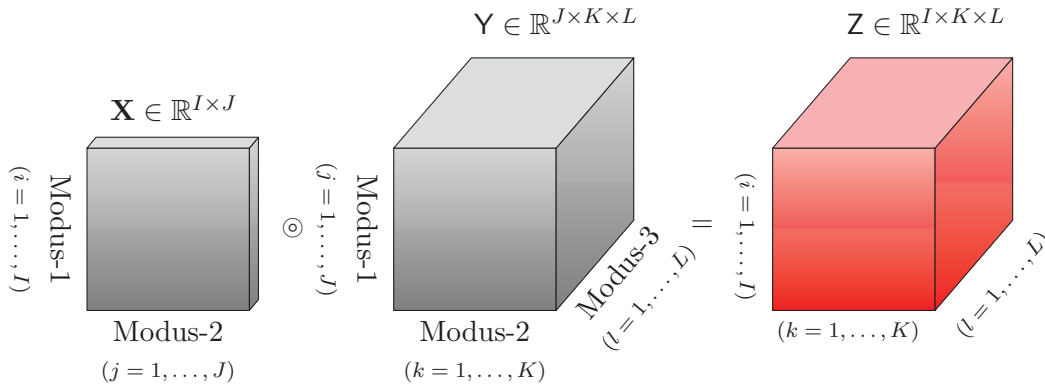


Abbildung 6.12: Generische Darstellung der Modus-($2|1$) Kontraktion eines Tensors zweiter Ordnung $\mathbf{X} \in \mathbb{R}^{I \times J}$ mit einem Tensor $\mathbf{Y} \in \mathbb{R}^{J \times K \times L}$

Beispiel 6.4 (Kontrahiertes Modus-($2|1$) Tensorprodukt): Legt man für die in Abbildung 6.12 dargestellten Tensoren $\mathbf{X} \in \mathbb{R}^{I \times J}$ und $\mathbf{Y} \in \mathbb{R}^{J \times K \times L}$ die Dimensionsgrößen beispielsweise durch $I = K = 3$ und $J = L = 2$ fest, dann ergibt sich aus ihrer Modus-($2|1$) Kontraktion der in Abbildung 6.13 dargestellte Produkttensor $\mathbf{Z} \in \mathbb{R}^{3 \times 3 \times 2}$. Abbildung 6.13 zeigt beispielhaft die Berechnung eines Tensorelements z_{132} , welches man dadurch erhält, dass man die gleichfarbigen Elemente der Tensoren \mathbf{X} und \mathbf{Y} multipliziert und deren Produkte aufsummiert.

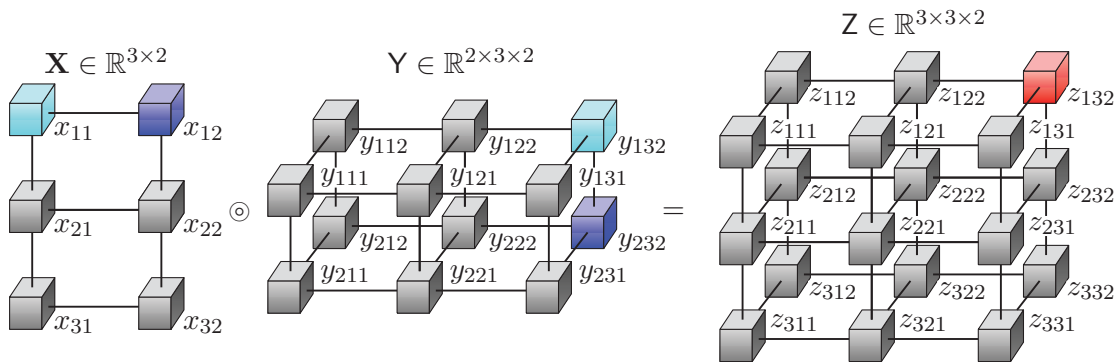


Abbildung 6.13: Modus-($2|1$) Kontraktion eines Tensors zweiter Ordnung $\mathbf{X} \in \mathbb{R}^{3 \times 2}$ mit einem Tensor dritter Ordnung $\mathbf{Y} \in \mathbb{R}^{2 \times 3 \times 2}$

Für das Beispiel in Abbildung 6.13 geht Gleichung (6.31) in die Form

$$z_{ikl} = \sum_{j=1}^J x_{ij} \cdot y_{jkl}$$

über und das Element z_{132} berechnet sich wie folgt:

$$z_{132} = \sum_{j=1}^2 x_{ij} \cdot y_{jkl} = x_{11} \cdot y_{132} + x_{12} \cdot y_{232} .$$

Auch das Modus- $(P | 1)$ Tensorprodukt kann als Sonderfall des kontrahierten Tensorprodukts interpretiert werden, welches im nächsten Abschnitt erläutert wird.

6.4.6 Kontrahiertes Tensorprodukt

Das *kontrahierte Tensorprodukt* stellt eine Verallgemeinerung des Tensor-Vektorprodukts [4, S. 643], aber auch des kontrahierten Modus- $(P | 1)$ Produkts dar. Während bei der Modus- $(P | 1)$ Kontraktion die Modi entlang derer das Produkt gebildet wird, fest definiert sind, besteht beim kontrahierten Tensorprodukt Freiheit in der Auswahl und Anzahl der entsprechenden Tensormodi. Eine grundlegende Voraussetzung für die Anwendbarkeit des Produkts ist aber auch hier, dass alle Modi entlang derer die Tensoren kontrahiert werden, die gleiche Anzahl an Elementen besitzen. In der Literatur wird das kontrahierte Tensorprodukt häufig als Modus- $(1, \dots, P | 1, \dots, P)$ Kontraktion definiert, bei der zwei Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times \dots \times J_P \times K_1 \times \dots \times K_S}$ und $\mathbf{Y} \in \mathbb{R}^{J_1 \times \dots \times J_P \times I_1 \times \dots \times I_L}$ entlang der ersten P Modi kontrahiert werden [4, S. 643–644], [25, S. 40–41].

Definition 6.14 (Kontrahiertes Tensorprodukt [25, S. 40–41]): *Das kontrahierte Tensorprodukt der Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times \dots \times J_P \times K_1 \times \dots \times K_S}$ und $\mathbf{Y} \in \mathbb{R}^{J_1 \times \dots \times J_P \times I_1 \times \dots \times I_L}$ entlang der ersten P Modi ist definiert durch einen Tensor*

$$\mathbf{Z} = \langle \mathbf{X} | \mathbf{Y} \rangle_{(1, \dots, P | 1, \dots, P)} \in \mathbb{R}^{K_1 \times \dots \times K_S \times I_1 \times \dots \times I_L} \quad (6.32)$$

mit Elementen

$$z(k_1, \dots, k_S, i_1, \dots, i_L) = \sum_{j_1=1}^{J_1} \dots \sum_{j_P=1}^{J_P} x(j_1, \dots, j_P, k_1, \dots, k_S) \cdot y(j_1, \dots, j_P, i_1, \dots, i_L) . \quad (6.33)$$

Das Ergebnis des kontrahierten Tensorprodukts ist ein Tensor, dessen Ordnung und Dimensionsgrößen sich aus den verbleibenden Modi der beiden multiplizierten Tensoren zusammensetzen (s. Abbildung 6.14).

Wie bereits erwähnt, ist das kontrahierte Tensorprodukt nicht auf die Multiplikation entlang der jeweils ersten P Modi beschränkt, sondern kann entlang beliebiger dimensionsgleicher Modi gebildet werden. Abbildung 6.14 zeigt beispielsweise in der oberen Zeile eine Modus- $(1, 2 | 1, 2)$ Kontraktion nach Definition 6.14. Die mittlere Zeile der Abbildung veranschaulicht hingegen eine Modus- $(1, 3 | 2, 3)$ Kontraktion zweier Tensoren dritter Ordnung, während in der unteren Zeile eine Modus- $(2, 3 | 1, 2)$ Kontraktion eines dreidimensionalen Tensors mit einer Matrix dargestellt ist.

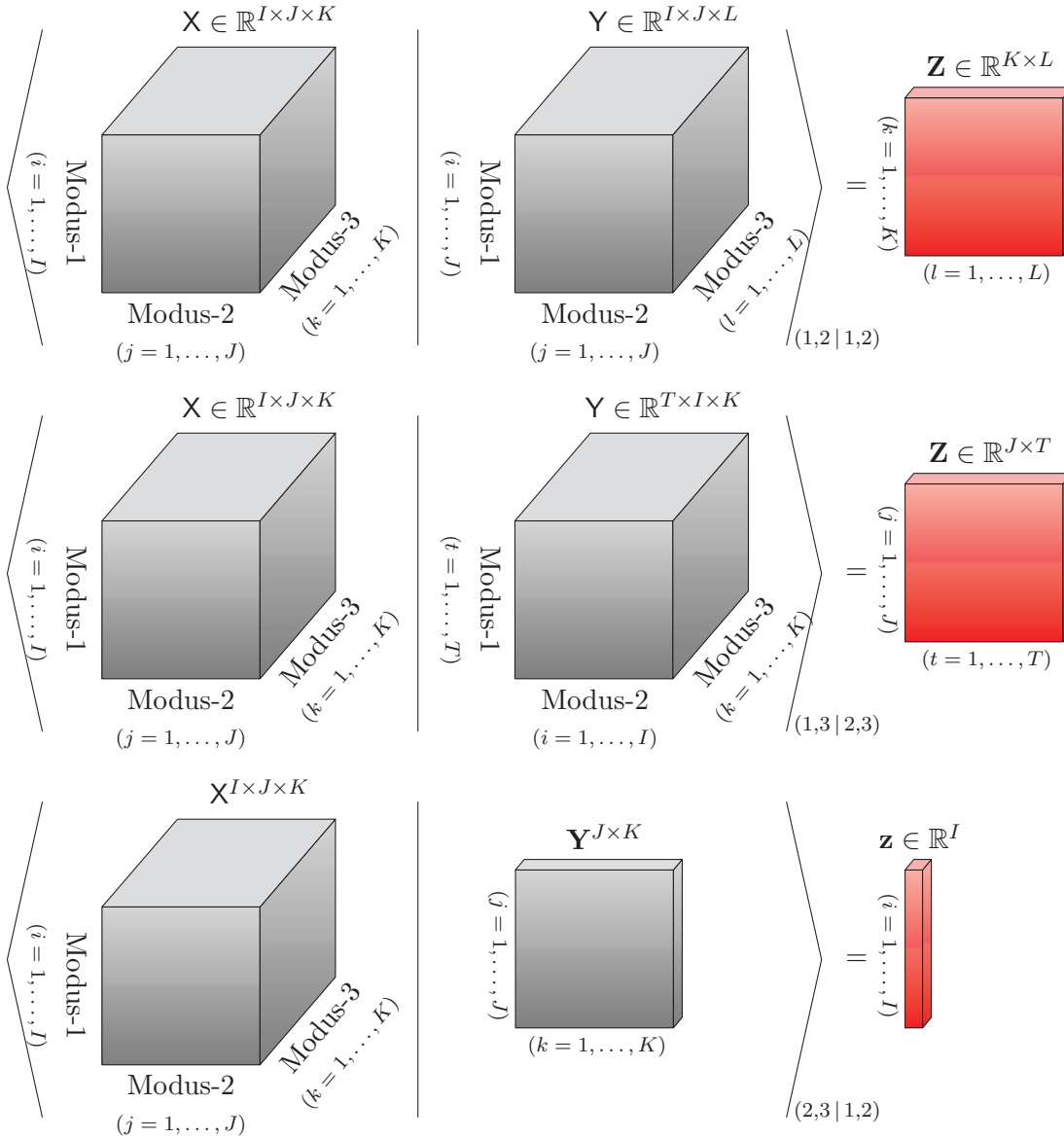


Abbildung 6.14: Generische Darstellung verschiedener kontrahierter Tensorprodukte

Das nachfolgende Beispiel 6.5 zeigt die elementweise Berechnung des kontrahierten Tensorprodukts.

Beispiel 6.5 (Kontrahiertes Tensorprodukt): Gegeben ist der Tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ mit den Elementen x_{ijk} sowie der Tensor $\mathbf{Y} \in \mathbb{R}^{T \times I \times K}$ mit Elementen y_{tik} . Für die Größen der Dimensionen gilt $I = J = 3$ und $T = K = 2$. Abbildung 6.15 veranschaulicht die Berechnung einer Modus-(1,3 | 2,3) Kontraktion der beiden Tensoren in Anlehnung an die mittlere Zeile in Abbildung 6.14. Als Ergebnis dieser Berechnung erhält man den Produkttensor zweiter Ordnung $\mathbf{Z} \in \mathbb{R}^{3 \times 2}$ mit Elementen z_{jt} . Das Element z_{11} erhält man durch Multiplikation der Elemente der lateralen Scheibe $\mathbf{X}_{:1}$ mit denen gleicher Farbe der horizontalen Scheibe $\mathbf{Y}_{1::}$ und anschließendes Aufsummieren der Produkte.

Da für die Berechnung der Elemente des Produkttensors die Summe von Produkten stets über diejenigen Modi gebildet wird entlang derer die Tensoren kontrahiert werden, geht Gleichung (6.33) in die Form

$$z_{jt} = \sum_{i=1}^I \sum_{k=1}^K x_{ijk} \cdot y_{tik}$$

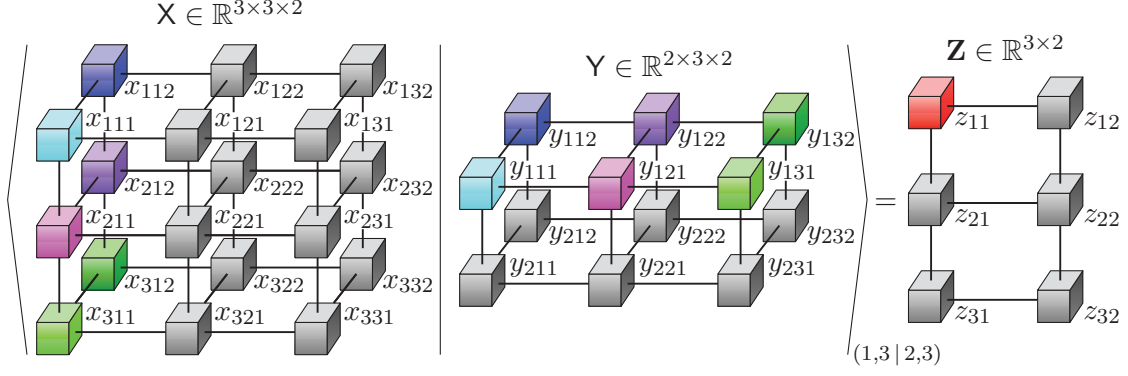


Abbildung 6.15: Modus-(1,3|2,3) Kontraktion zweier Tensoren dritter Ordnung $\mathbf{X} \in \mathbb{R}^{3 \times 3 \times 2}$ und $\mathbf{Y} \in \mathbb{R}^{2 \times 3 \times 2}$

über und das Element z_{11} kann wie folgt bestimmt werden:

$$\begin{aligned} z_{11} &= \sum_{i=1}^3 \sum_{k=1}^2 x_{ijk} \cdot y_{tik} \\ &= x_{111} \cdot y_{111} + x_{112} \cdot y_{112} + x_{211} \cdot y_{121} + x_{212} \cdot y_{122} + x_{311} \cdot y_{131} + x_{312} \cdot y_{132} . \end{aligned}$$

Wie bereits erwähnt, stellt das kontrahierte Tensorprodukt eine Generalisierung des Modus-($P|1$) Produkts und des Modus- n Tensor-Vektor Produkts dar. So gilt beispielsweise für die in Abbildung 6.10 dargestellte Modus-2 Tensor-Vektor Multiplikation die äquivalente Beziehung

$$\langle \mathbf{X} | \mathbf{y} \rangle_{(2|1)} \equiv \mathbf{X} \bar{\times}_{(2)} \mathbf{y} \quad (6.34)$$

und für die Modus-(2|1) Kontraktion in Abbildung 6.12 der Zusammenhang

$$\langle \mathbf{X} | \mathbf{Y} \rangle_{(2|1)} \equiv \mathbf{X} \odot \mathbf{Y} . \quad (6.35)$$

Generell könnte man also sowohl das Modus- n Tensor-Vektor Produkt als auch das kontrahierte Modus-($P|1$) Produkt durch ein entsprechendes kontrahiertes Tensorprodukt ersetzen. Zugunsten der Übersichtlichkeit wird in der Regel jedoch, wie auch in dieser Arbeit, auf diese Möglichkeit verzichtet.

Im Folgenden findet das kontrahierte Tensorprodukt oftmals als Kontraktion zweier Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_P \times K_1 \times K_2 \times \dots \times K_S}$ und $\mathbf{Y} \in \mathbb{R}^{K_1 \times K_2 \times \dots \times K_S}$ entlang der *letzten* S Modi des Tensors \mathbf{X} mit *allen* S Modi des Tensors \mathbf{Y} Anwendung. Dies wird als Modus- $(P+1, \dots, P+S|1, \dots, S)$ Kontraktion bezeichnet und folgenderweise definiert:

Definition 6.15 (Modus- $(P+1, \dots, P+S|1, \dots, S)$ Tensorkontraktion):

Die Modus- $(P+1, \dots, P+S|1, \dots, S)$ Kontraktion zweier Tensoren $\mathbf{X} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_P \times K_1 \times K_2 \times \dots \times K_S}$ und $\mathbf{Y} \in \mathbb{R}^{K_1 \times K_2 \times \dots \times K_S}$ ist definiert durch den Tensor

$$\mathbf{Z} = \langle \mathbf{X} | \mathbf{Y} \rangle \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_P} \quad (6.36)$$

mit den Elementen

$$z(j_1, j_2, \dots, j_P) = \sum_{k_1=1}^{K_1} \dots \sum_{k_S=1}^{K_S} x(j_1, j_2, \dots, j_P, k_1, k_2, \dots, k_S) \cdot y(k_1, k_2, \dots, k_S) . \quad (6.37)$$

Da das kontrahierte Tensorprodukt in dieser Arbeit größtenteils in der Konfiguration nach Definition 6.15 verwendet wird, kann auf die explizite Angabe der Modi, entlang derer das Produkt gebildet wird, verzichtet werden. Ein kontrahiertes Tensorprodukt nach Definition 6.15 ist beispielhaft in der untersten Zeile von Abbildung 6.14 dargestellt, denn die Kontraktion erfolgt hier über die letzten beiden Modi des Tensors $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ und entlang aller Modi der Matrix $\mathbf{Y} \in \mathbb{R}^{J \times K}$. Eine solche Tensorkontraktion wird im Folgenden also nicht mehr explizit durch $\langle \mathbf{X} | \mathbf{Y} \rangle_{(2,3|1,2)} = \mathbf{z} \in \mathbb{R}^I$ angegeben, sondern schlicht durch $\langle \mathbf{X} | \mathbf{Y} \rangle = \mathbf{z} \in \mathbb{R}^I$ ausgedrückt.

6.5 Spezielle Tensoren und Tensorrang

Dieser Abschnitt führt einige grundlegende Begriffe und Definitionen ein, die im weiteren Verlauf dieser Arbeit benötigt werden. Dabei wird zunächst ein spezieller Tensor eingeführt, den man als *Rang-1 Tensor* bezeichnet. Dieser Tensor bildet die Grundlage für die Einführung des Begriffes des *Tensorrangs*. Abschließend wird der sogenannte *Kruskal-Tensor* definiert, dem eine grundlegende Bedeutung in dieser Arbeit zukommt.

6.5.1 Rang-1 Tensor

Zum Begriff des *Rangs* sei vorab angemerkt, dass dieser je nach Anwendungsgebiet auch die Ordnung eines Tensors bezeichnet. Im hier verwendeten Kontext ist dieser jedoch stets als das Äquivalent zum Begriff des Rangs einer Matrix zu verstehen [56, S. 464].

Definition 6.16 (Rang-1 Tensor [56, S. 458]): *Ein Tensor N -ter Ordnung $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ist ein Rang-1 Tensor, wenn er sich als dyadisches Produkt von N Vektoren darstellen lässt:*

$$\mathbf{X} = \bigcirc_{n=1}^N \mathbf{y}^{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}, \quad (6.38)$$

mit $\mathbf{y}^{(n)} \in \mathbb{R}^{I_n}$ für $n = 1, 2, \dots, N$.

Ein Tensor \mathbf{X} , der wie in (6.38) durch das dyadische Produkt von Vektoren gebildet wird, besitzt stets den Rang Eins. Man schreibt in diesem Fall $\text{rang}(\mathbf{X}) = 1$. Bei den beispielhaft in den Abbildungen 6.8 und 6.9 dargestellten Tensoren, handelt es sich demzufolge um Rang-1 Tensoren.

6.5.2 Tensorrang

Basierend auf Definition 6.16, kann der Rang eines Tensors wie folgt definiert werden:

Definition 6.17 (Tensorrang [25, S. 41]): *Der Rang eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ ist definiert als die kleinstmögliche Anzahl R von Rang-1 Tensoren $\mathbf{Z}^{(r)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$, für deren Summe*

$$\sum_{r=1}^R \mathbf{Z}^{(r)} = \mathbf{Y} \quad (6.39)$$

gilt.

Die Berechnung des Rangs eines Tensors ist im allgemeinen nur für Tensoren mit niedriger Ordnung und geringer Dimensionsgrößen möglich [56, S. 464–467]. Grund dafür ist, dass die Bestimmung des Tensorrangs eines beliebigen Tensors zur Klasse der NP-schweren Probleme zählt, was eine algorithmische Berechnung in Polynomialzeit ausschließt [38], [41].

Für die in dieser Arbeit verwendeten Tensoren wird aufgrund ihrer Größe und Komplexität, eine Berechnung des Tensorrangs *nicht* möglich sein. Dennoch wird dieser Begriff insbesondere in Kapitel 8 benötigt, um grundlegende Eigenschaften von Algorithmen und Optimierungsproblemen zu erläutern.

6.5.3 Kruskal-Tensor

Die Bezeichnung *Kruskal-Tensor* geht zurück auf den amerikanischen Mathematiker J. B. Kruskal, wobei die Namensgebung für diese spezielle Darstellung eines Tensors insbesondere durch die Arbeiten [5], [55] geprägt wurde. Die Definition eines Kruskal-Tensors ist wie folgt:

Definition 6.18 (Kruskal-Tensor [5, S. 222]): *Ein Kruskal-Tensor*

$$\llbracket \boldsymbol{\lambda}; \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(N)} \rrbracket \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}, \quad (6.40)$$

bestehend aus einem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}^R$ und Faktormatrizen $\mathbf{Y}^{(n)} \in \mathbb{R}^{I_n \times R}$ mit Spaltenvektoren $\mathbf{y}_{:,r}^{(n)} \in \mathbb{R}^{I_n}$ für $n = 1, 2, \dots, N$, ist definiert als Summe von R Rang-1 Tensoren:

$$\llbracket \boldsymbol{\lambda}; \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(N)} \rrbracket \equiv \sum_{r=1}^R \lambda_r \bigcirc_{n=1}^N \mathbf{y}_{:,r}^{(n)} = \mathbf{K} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}. \quad (6.41)$$

Das Symbol $\llbracket \cdot \rrbracket$ wird auch als *Kruskal-Operator* [55] bezeichnet, da es den zwischen den beiden Klammern stehenden Ausdruck mit einer definierten mathematischen Rechenvorschrift verknüpft. Die Darstellung eines Kruskal-Tensors (6.40) kann demzufolge als eine Kurznotation der in der Mitte von Gleichung (6.41) explizit notierten Summe dyadischer Produkte (Rang-1 Tensoren) interpretiert werden [55]. Die auf der linken Seite und in der Mitte stehenden Ausdrücke in Gleichung (6.41) werden dabei auch unabhängig voneinander als Kruskal-Tensor oder als Kruskalrepräsentation des Tensors \mathbf{K} bezeichnet [5, S. 222]. Die (optionale) Gewichtung der einzelnen Rang-1 Tensoren durch die Komponenten λ_r des Vektors $\boldsymbol{\lambda} \in \mathbb{R}^R$ ermöglicht die Normierung der Spaltenvektoren $\mathbf{y}_{:,r}^{(n)} \in \mathbb{R}^{I_n}$ der Faktormatrizen auf eine Länge von Eins, was für einige Anwendungen hilfreich ist [56, S. 464]. Abschließend sei angemerkt, dass ein Tensor \mathbf{K} , der aus einer Summe von R Rang-1 Tensoren gebildet wird, nicht zwangsläufig den Rang R besitzt, sondern allgemein nur $\text{rang}(\mathbf{K}) \leq R$ gilt.

Aus der Definition des Kruskal-Tensors lassen sich zwei wichtige Spezialfälle ableiten:

1. Besitzen alle Komponenten des Gewichtungsvektors $\boldsymbol{\lambda} \in \mathbb{R}^R$ den Wert Eins, sprich es gilt $\boldsymbol{\lambda} = \mathbf{1} \in \{1\}^R$, dann ist

$$\llbracket \mathbf{1}; \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(N)} \rrbracket = \llbracket \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(N)} \rrbracket. \quad (6.42)$$

Der Gewichtungsvektor entfällt somit in der Notation des Kruskal-Tensors.

2. Besitzen die Matrizen $\mathbf{Y}^{(n)} \in \mathbb{R}^{I_n \times R}$ eines Kruskal-Tensors $\llbracket \mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \dots, \mathbf{Y}^{(N)} \rrbracket$ nur eine Spalte, sprich der Kruskal-Tensor enthält nur die Vektoren $\mathbf{y}^{(n)} \in \mathbb{R}^{I_n}$ für $n = 1, 2, \dots, N$, dann ist

$$\llbracket \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(N)} \rrbracket \equiv \bigcirc_{n=1}^N \mathbf{y}^{(n)} = \mathbf{K} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \quad (6.43)$$

ein Rang-1 Tensor nach Definition 6.16. Ein Kruskal-Tensor wie $\llbracket \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(N)} \rrbracket$, wird im Folgenden als *Kruskal-Rang-1 Tensor* bezeichnet.

Dem Kruskal-Tensor nach Definition 6.18 kommt in dieser Arbeit eine grundlegende Bedeutung zu. Auf der einen Seite wird diese Tensordarstellung verwendet, um Rang-1 Tensoren wie in Gleichung (6.43) zu repräsentieren und zum anderen stellt die Kruskalrepräsentation eines Tensors das Ergebnis der sogenannten *CP-Tensordekomposition* dar, die eine grundlegende Rolle in Bezug auf die Reduktion des Speicherbedarfs qualitativer Modelle einnimmt und in Kapitel 8 erläutert wird.

7

Qualitative Modelle in Tensorrepräsentation

Dieses Kapitel führt die Tensorrepräsentation stochastischer Automaten und qualitativer Modelle ein, welche die Grundlage der Modellreduktion durch die Tensordekomposition in Kapitel 8 bildet. Zusätzlich wird eine Beobachtergleichung eingeführt, die es ermöglicht, Unsicherheiten in den Ein- und Ausgaben des Automaten oder des qualitativen Modells zu berücksichtigen. Dies ist für den stochastischen Automaten in Abschnitt 7.1 und für die qualitativen Modelle in Abschnitt 7.2 beschrieben.

7.1 Tensorrepräsentation stochastischer Automaten

Eine Alternative zu der in Abschnitt 3.4.3 vorgestellten Matrixrepräsentation stochastischer Automaten nach Definition 3.5, bietet die sogenannte *Tensorrepräsentation*. Hierbei werden die bedingten Wahrscheinlichkeiten der Verhaltensrelation $\mathcal{L}(\bar{z}, w | z, v)$ aus Gleichung (3.35) nicht in einer Matrix, sondern in einem Tensor vierter Ordnung gespeichert.

7.1.1 Verhaltenstensor

Der Verhaltenstensor ist ein Tensor vierter Ordnung und repräsentiert für alle Zustände $\bar{z}, z \in \mathcal{Z}$, Eingaben $v \in \mathcal{V}$ und Ausgaben $w \in \mathcal{W}$ die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ der Verhaltensrelation (3.35) des stochastischen Automaten \mathcal{A} aus Gleichung (3.30). Die Dimensionen des Verhaltenstensors sind entsprechend durch die Anzahl der Automateneingaben M , die Zustandsanzahl N und die Anzahl der Ausgaben Q bestimmt.

Definition 7.1 (Verhaltenstensor [92, S. 417]): *Der Tensor*

$$\mathbf{L} \in [0, 1]^{N \times Q \times N \times M} \quad (7.1)$$

für dessen Elemente

$$l(\bar{z}, w, z, v) = \mathcal{L}(\bar{z}, w | z, v) \quad \forall \bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V} \quad (7.2)$$

gilt, heißt Verhaltenstensor.

Wie aus Definition 7.1 hervorgeht, werden als Indizes der Tensorelemente die Automatenzustände und Folgezustände $\bar{z}, z \in \mathcal{Z}$, die Eingaben $v \in \mathcal{V}$ sowie die Ausgaben $w \in \mathcal{W}$ verwendet. Damit ist klar, dass ein Tensorelement $l(\bar{z}, w, z, v)$ die bedingte Wahrscheinlichkeit $\mathcal{L}(\bar{z}, w | z, v)$ für diese Kombination von Ein-, Ausgaben und Zuständen repräsentiert. Für den Verhaltenstensor gelten durch Gleichung (7.2) wiederum die Beziehungen (3.36) und (3.37).

Mithilfe der Tensorrepräsentation des stochastischen Automaten kann nun zunächst eine einfache Beobachtergleichung definiert werden, welche exakt die gleiche Struktur wie die Matrix-Beobachtergleichung aus Definition (3.7) besitzt.

7.1.2 Beobachtergleichung

Auf Basis des Verhaltenstensors $\mathbf{L} \in \mathbb{R}^{N \times Q \times N \times M}$ aus Gleichung (7.1) lässt sich nun die Beobachtergleichung für stochastische Automaten in Tensorrepräsentation formulieren. Sind die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ der Verhaltensrelation (3.35) nach Definition 7.1 im Verhaltenstensor (7.1) angeordnet, dann kann der Zähler der Beobachtergleichung für die Prädiktion (3.50) wieder in Form des Matrix-Vektor Produkts

$$\mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k) \in [0, 1]^N \quad (7.3)$$

angegeben werden. Darin repräsentiert

$$\mathbf{p}_z(k) = \begin{pmatrix} p_z(k, 1) \\ p_z(k, 2) \\ \vdots \\ p_z(k, N) \end{pmatrix} := \begin{pmatrix} \Pr(Z(k) = 1 | V(0 \dots k-1), W(0 \dots k-1)) \\ \Pr(Z(k) = 2 | V(0 \dots k-1), W(0 \dots k-1)) \\ \vdots \\ \Pr(Z(k) = N | V(0 \dots k-1), W(0 \dots k-1)) \end{pmatrix} \in [0, 1]^N$$

wieder den Vektor der diskreten Wahrscheinlichkeitsverteilung der Zustände zur Zeit k aus Gleichung (3.54) und

$$\begin{aligned} & \mathbf{L}(:, w(k), :, v(k)) \\ &= \begin{pmatrix} l(1, w(k), 1, v(k)) & l(1, w(k), 2, v(k)) & \dots & l(1, w(k), N, v(k)) \\ l(2, w(k), 1, v(k)) & l(2, w(k), 2, v(k)) & \dots & l(2, w(k), N, v(k)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N, w(k), 1, v(k)) & l(N, w(k), 2, v(k)) & \dots & l(N, w(k), N, v(k)) \end{pmatrix} \in [0, 1]^{N \times N}, \end{aligned} \quad (7.4)$$

einen quadratischen Subtensor zweiter Ordnung, beziehungsweise eine Matrix. Der Subtensor (7.4) stellt formal also eine Tensorscheibe dar (vgl. Definition 6.2, Tabelle 6.2), die man durch Fixieren der Indizes w und v des Verhaltenstensors \mathbf{L} erhält. Es kann somit in jedem Zeitschritt k und für alle möglichen Kombinationen von Automateingaben $v(k)$ und -ausgaben $w(k)$ ein Subtensor des Verhaltenstensors selektiert werden, dessen Elemente $l(\bar{z}, w, z, v)$ die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$ für das entsprechende E/A Paar repräsentieren. Die Gesamtzahl der Subtensoren (7.4) des Verhaltenstensors entspricht dem Produkt $Q \cdot M$ und ergibt sich aus den Anzahlen Q und M der Automatausgaben und -eingaben. Da für den Verhaltenstensor die Beziehungen (3.36) und (3.37) gelten, sind die Subtensoren (7.4) *sub-stochastisch*, wodurch ihre Spaltensumme in der Regel kleiner Eins ist. Es gilt damit wiederum die Gleichung (3.56). Ebenso wie bei der Matrix-Beobachtergleichung (3.58) erhält man den an die Tensorrepräsentation des stochastischen Automaten angepassten Nenner der Beobachtergleichung (3.50) durch die Summennorm:

$$\|\mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k)\|_1 \in [0, 1]. \quad (7.5)$$

Mit den Gleichungen (7.3) und (7.5) folgt die Definition der Beobachtergleichung für stochastische Automaten in Tensorrepräsentation.

Definition 7.2 (Tensor-Beobachtergleichung): *Sind die E/A Folgen $(V(0 \dots T), W(0 \dots T))$ eines stochastischen Prozesses konsistent mit dem stochastischen Automaten \mathcal{A} , dann können die diskreten Wahrscheinlichkeitsverteilungen der Zustände mit*

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k)}{\|\mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k)\|_1} \in [0, 1]^N, \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0 \end{aligned} \quad (7.6)$$

für alle diskreten Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden.

Vergleicht man die Matrix-Beobachtergleichung in Definition 3.7 mit der Tensor-Beobachtergleichung in Definition 7.2, stellt man fest, dass diese strukturell identisch sind. Es ändert sich lediglich die Art und Weise, wie die für eine gegebene Eingabe $v(k)$ und Ausgabe $w(k)$ selektierten Submatrizen indiziert werden. Während bei der Verhaltensmatrix ein Block $\mathbf{L}(w(k), v(k)) \in [0, 1]^{N \times N}$ für ein gegebenes E/A Paar $(w(k), v(k))$ ausgewählt wird, wird in der Tensorrepräsentation ein Subtensor $\mathbf{L}(:, w(k), :, v(k)) \in [0, 1]^{N \times N}$ selektiert. Initialisiert mit der Startverteilung \mathbf{p}_0 aus (3.21), erhält man durch rekursive Anwendung der Beobachtergleichung die Zustandsvektoren $\mathbf{p}_z(k+1)$ für alle diskreten Zeiten $k = 0, 1, 2, \dots, T$. Diese lassen sich wiederum in der Vektorfolge

$$\mathbf{P}_z(0 \dots T+1 | V(0 \dots T), W(0 \dots T)) = (\mathbf{p}_z(0), \mathbf{p}_z(1), \mathbf{p}_z(2), \dots, \mathbf{p}_z(T+1)) .$$

zusammenfassen, die identisch mit Gleichung (3.60) ist. Ebenso erhält man aus dem Nenner der Tensor-Beobachtergleichung den Wert

$$b(k) := \|\mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k)\|_1 \in [0, 1]$$

und damit die als Verhaltenssignal bezeichnete Folge

$$B(0 \dots T | V(0 \dots T), W(0 \dots T)) = (b(0), b(1), b(2), \dots, b(T)) ,$$

aus Gleichung (3.63). Für die Fehlererkennung und Konsistenzbewertung der Ein- und Ausgabefolgen gelten damit alle in den Abschnitten 3.5.2 und 3.5.3 erläuterten Beziehungen.

7.1.3 Erweiterte Beobachtergleichung

Der Beobachtergleichung (7.6) liegt, wie auch der Beobachtergleichung für die Prädiktion (3.50), die Annahme zugrunde, dass die Eingaben v und Ausgaben w exakt bekannt und durch die Folgen $V(0 \dots T)$ und $W(0 \dots T)$ gegeben sind. Es wird also angenommen, dass diese mit einer Wahrscheinlichkeit von Eins gegeben sind und damit die Gleichungen (3.44) und (3.46) gelten. Die Berücksichtigung unsicherer oder nicht exakt bekannter Ein- und Ausgaben ist dementsprechend weder mit Gleichung (7.6) noch mit Gleichung (3.50) möglich. Die generelle Grundlage zur Berücksichtigung von Unsicherheiten basiert dabei auf einer vektoriellen Repräsentation der Ein- und Ausgaben, was im Folgenden erläutert wird.

Ein- und Ausgabewahrscheinlichkeitsvektoren

Die Wahrscheinlichkeiten der Ausgaben $\Pr(W(k) = w)$ und Eingaben $\Pr(V(k) = v)$ können zunächst für alle $w \in \mathcal{W}$ und $v \in \mathcal{V}$ in Form von Wahrscheinlichkeitsvektoren notiert werden:

$$\mathbf{p}_v(k) = \begin{pmatrix} p_v(k, 1) \\ p_v(k, 2) \\ \vdots \\ p_v(k, M) \end{pmatrix} := \begin{pmatrix} \Pr(V(k) = 1) \\ \Pr(V(k) = 2) \\ \vdots \\ \Pr(V(k) = M) \end{pmatrix} \in [0, 1]^M , \quad (7.7)$$

$$\mathbf{p}_w(k) = \begin{pmatrix} p_w(k, 1) \\ p_w(k, 2) \\ \vdots \\ p_w(k, M) \end{pmatrix} := \begin{pmatrix} \Pr(W(k) = 1) \\ \Pr(W(k) = 2) \\ \vdots \\ \Pr(W(k) = Q) \end{pmatrix} \in [0, 1]^Q . \quad (7.8)$$

Der Eingabewahrscheinlichkeitsvektor $\mathbf{p}_v(k)$ und der Ausgabewahrscheinlichkeitsvektor $\mathbf{p}_w(k)$ beschreiben somit die diskreten Wahrscheinlichkeitsverteilungen der Automaten-eingaben und -ausgaben

zur Zeit k . Wie auch bei den skalaren E/A Folgen $(V(0...T), W(0...T))$ wird außerdem eine Folge von Eingabewahrscheinlichkeitsvektoren

$$\mathbf{P}_v(0...T) = (\mathbf{p}_v(0), \mathbf{p}_v(1), \mathbf{p}_v(2), \dots, \mathbf{p}_v(T)) \quad (7.9)$$

und Ausgabewahrscheinlichkeitsvektoren

$$\mathbf{P}_w(0...T) = (\mathbf{p}_w(0), \mathbf{p}_w(1), \mathbf{p}_w(2), \dots, \mathbf{p}_w(T)) \quad (7.10)$$

definiert. Die Folgen (7.9) und (7.10) sollen dabei im Weiteren sowohl exakt bekannte E/A Folgen $V(0...T), W(0...T)$ repräsentieren, als auch solche, die mit Unsicherheiten behaftet sind. Zum Begriff der Ein- und Ausgabefolgen siehe auch Anmerkung 7.1.

Exakte Ein- und Ausgabewahrscheinlichkeitsvektoren: Es wird nun zunächst davon ausgegangen, dass die E/A Folgen $(V(0...T), W(0...T))$ exakt bekannt sind und durch eine Folge von Eingabewahrscheinlichkeitsvektoren (7.9) und Ausgabewahrscheinlichkeitsvektoren (7.10) repräsentiert werden. Dies ist leicht umsetzbar, denn sind die Eingaben $v \in \mathcal{V}$ und Ausgaben $w \in \mathcal{W}$ zu jeder Zeit k durch die exakten Folgen $(V(0...T), W(0...T))$ gegeben, dann sind die Wahrscheinlichkeiten der Ein- und Ausgaben durch $\Pr(W(k)=w(k))=1$ und $\Pr(V(k)=v(k))=1$ bestimmt (vgl. (3.44), (3.46)). In diesem Fall besitzen die Wahrscheinlichkeitsvektoren der Eingaben (7.7) und Ausgaben (7.8) jeweils nur an der $v(k)$ -ten und $w(k)$ -ten Position den Wert Eins, während alle anderen Komponenten den Wert Null aufweisen. Für den Eingabewahrscheinlichkeitsvektor gilt dann $\mathbf{p}_v(k) \in \{0, 1\}^M$ und für den Ausgabewahrscheinlichkeitsvektor $\mathbf{p}_w(k) \in \{0, 1\}^Q$. Ein- und Ausgabevektoren mit dieser Eigenschaft werden als *exakte Ein- und Ausgabewahrscheinlichkeitsvektoren* bezeichnet. Besitzen die E/A Folgen $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ ausschließlich solche Elemente, werden diese ebenfalls als exakt bezeichnet. Die Folgen $\mathbf{P}_v(0...T)$ und $\mathbf{P}_w(0...T)$ stellen in diesem Fall lediglich eine vektorielle Repräsentation der exakten Ein- und Ausgabefolgen $V(0...T)$ und $W(0...T)$ dar.

Beispiel 7.1 (Exakte Eingabewahrscheinlichkeitsvektoren): Für einen Zeithorizont von $T = 4$ sei die exakte Eingabefolge $V(0...4) = (2, 1, 3, 1, 2)$ gegeben und die Anzahl der Automateneingaben durch $M = 3$ festgelegt, sodass für die Eingaben $v(k) \in \mathcal{V} := \{1, 2, 3\}$ gilt. Da zum Zeitpunkt $k = 0$ die Automateneingabe durch $v(0) = 2$ gegeben ist, ergibt sich ein Eingabewahrscheinlichkeitsvektor dessen zweite Komponente den Wert Eins besitzt:

$$\mathbf{p}_v(0) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \in \{0, 1\}^3.$$

Für die gesamte skalare Eingabefolge $V(0...4)$ erhält man somit die Folge exakter Eingabewahrscheinlichkeitsvektoren

$$\mathbf{P}_v(0...4) = \left(\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right).$$

Auf Basis der Wahrscheinlichkeitsvektoren der Ein- und Ausgaben in (7.7) und (7.8), lässt sich nun eine Beobachtergleichung auf der Grundlage des kontrahierten Tensorprodukts formulieren.

Anmerkung 7.1 (Zum Begriff der E/A Folgen): In der Literatur wird unter dem Begriff einer Ein- und Ausgabefolge in der Regel die Darstellung der Ein- und Ausgaben in Form der Skalarfolgen $V(0...T)$ und $W(0...T)$ verstanden. Im Weiteren werden diese häufig als Folgen von Wahrscheinlichkeitsvektoren $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ dargestellt, da sich hierdurch mehr Freiheiten ergeben und beispielsweise Unsicherheiten berücksichtigt werden können. Im Folgenden können also mit dem

Begriff einer E/A Folge sowohl die Wahrscheinlichkeitsvektoren der Ein- und Ausgaben als auch die skalare Darstellung gemeint sein, wobei aus dem Kontext stets klar sein wird, um welche Repräsentation es sich handelt. Während wahrscheinlichkeitstheoretische Beschreibungen in der Regel anhand der skalaren Darstellung erfolgen, werden in Algorithmen tendenziell die Wahrscheinlichkeitsvektoren der Ein- und Ausgaben verwendet.

Beobachtung und Fehlererkennung mit dem kontrahierten Tensorprodukt

Es kann nun gezeigt werden, dass für sichere Ein- und Ausgabewahrscheinlichkeitsvektoren folgende Proposition gilt.

Proposition 7.1: Für exakte Ein- und Ausgabewahrscheinlichkeitsvektoren gilt

$$\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle = \mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k) \in [0, 1]^N, \quad (7.11)$$

mit $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in \{0, 1\}^Q$ sowie $\mathbf{p}_v(k) \in \{0, 1\}^M$.

Beweis: Der Beweis von Proposition 7.1 findet sich im Anhang B.1.

In (7.11) steht auf der linken Seite das kontrahierte Tensorprodukt (6.36), bestehend aus dem Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ und einem Kruskal-Rang-1 Tensor dritter Ordnung

$$\llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket = \mathbf{p}_w(k) \circ \mathbf{p}_z(k) \circ \mathbf{p}_v(k) \in [0, 1]^{Q \times N \times M} \quad (7.12)$$

nach Gleichung (6.43). Die Tensorkontraktion in (7.11) erfolgt dementsprechend entlang der Modi 2, 3 und 4 des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ und über alle Modi des aus Gleichung (7.12) resultierenden Produktensors, was zu einem Vektor der Länge N führt. Aus Proposition 7.1 ergibt sich unmittelbar folgendes Lemma.

Lemma 7.1: Durch Einsetzen von Gleichung (7.11) in Gleichung (7.6) erhält man die Beobachtergleichung auf Grundlage des kontrahierten Tensorprodukts:

$$k \geq 0 : \quad \mathbf{p}_z(k+1) = \frac{\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle}{\left\| \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle \right\|_1} \in [0, 1]^N, \quad (7.13)$$

$$k = 0 : \quad \mathbf{p}_z(0) = \mathbf{p}_0.$$

Dabei ist $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in \{0, 1\}^Q$ sowie $\mathbf{p}_v(k) \in \{0, 1\}^M$.

Sind die E/A Folgen $(V(0...T), W(0...T))$ gegeben und durch die exakten Folgen von Wahrscheinlichkeitsvektoren $\mathbf{P}_v(0...T)$ und $\mathbf{P}_w(0...T)$ repräsentiert, dann generiert die Beobachtergleichung (7.13) eine Folge von Zustandswahrscheinlichkeitsvektoren (3.60) deren Verteilungen für $k = 0, 1, 2, \dots, T$ durch (3.59) gegeben sind. Für gleiche initiale Zustandswahrscheinlichkeiten sowie *exakt* bekannte Folgen von Ein- und Ausgaben liefert die Beobachtergleichung (7.13) das gleiche Ergebnis wie die Beobachtergleichung (7.6). Diese Aussage kann dabei als eine Minimalforderung angesehen werden, die auch für alle weiteren in dieser Arbeit vorgestellten Beobachtergleichungen gelten muss. Die Gleichung (7.13) ist wie alle bisher vorgestellten Beobachtergleichungen nur dann anwendbar, wenn die E/A Folgen konsistent mit dem Modell \mathcal{A} aus Definition 3.5 sind.

Aufgrund der Tatsache, dass in der Beobachtergleichung in Lemma 7.1 die Ein- und Ausgaben vektoriell repräsentiert werden, ermöglicht diese Gleichung auch die Berücksichtigung von Unsicherheiten und wird im nächsten Absatz erläutert.

Berücksichtigung unsicherer Ein- und Ausgaben: Beispiel 7.2 zeigt, wie Unsicherheiten in den E/A Folgen $(W(0...T), V(0...T))$ mithilfe ihrer wahrscheinlichkeitstheoretischen Repräsentation $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ berücksichtigt werden können.

Beispiel 7.2 (Unsichere Eingabewahrscheinlichkeitsvektoren): *Ein stochastischer Prozess mit $M = 3$ Eingaben und Eingabemenge $\mathcal{V} := \{1, 2, 3\}$ generiert für einen Zeithorizont $T = 4$ eine Eingabefolge $V(0...4) = (2, 1, 3, 1, 2)$. Diese Eingaben sollen vom Modell \mathcal{A} jedoch nur mit 80-prozentiger Sicherheit angenommen werden. Eine Möglichkeit, diese Unsicherheit zu berücksichtigen, ist dann beispielsweise gegeben durch die Folge von Eingabewahrscheinlichkeitsvektoren*

$$\mathbf{P}_v(0...4) = \left(\begin{pmatrix} 0.1 \\ 0.8 \\ 0.1 \end{pmatrix}, \begin{pmatrix} 0.8 \\ 0.2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.2 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 0.8 \\ 0.2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 \\ 0.8 \\ 0.1 \end{pmatrix} \right).$$

Unabhängig davon, ob man in die Beobachtergleichung (7.13) exakte oder unsichere Ein- und Ausgabewahrscheinlichkeitsvektoren einsetzt, erhält man einen Zustandswahrscheinlichkeitsvektor dessen Verteilung im Folgenden mit

$$\mathbf{p}_z(k+1) = \begin{pmatrix} \Pr(Z(k+1) = 1 \mid \mathbf{P}_v(0...k), \mathbf{P}_w(0...k)) \\ \Pr(Z(k+1) = 2 \mid \mathbf{P}_v(0...k), \mathbf{P}_w(0...k)) \\ \vdots \\ \Pr(Z(k+1) = N \mid \mathbf{P}_v(0...k), \mathbf{P}_w(0...k)) \end{pmatrix} \in [0, 1]^N \quad (7.14)$$

bezeichnet wird. Dieser Vektor repräsentiert die Zustandswahrscheinlichkeitsverteilung zur Zeit $k+1$ unter der Bedingung, dass bis zur Zeit k die E/A Folgen $(\mathbf{P}_v(0...k), \mathbf{P}_w(0...k))$ durchlaufen wurden. In Erweiterung von (3.59) hängen die Zustandswahrscheinlichkeiten zum Zeitpunkt $k+1$ nicht zwangsläufig nur von den vergangenen und exakt bekannten Ein- und Ausgabefolgen $V(0...k), W(0...k)$ ab, sondern von den zurückliegenden Folgen von Eingabe- und Ausgabewahrscheinlichkeitsvektoren $\mathbf{P}_v(0...k)$ und $\mathbf{P}_w(0...k)$, die entweder exakte oder unsichere Ein- und Ausgaben repräsentieren. Zur Bezeichnung des Zustandswahrscheinlichkeitsvektors (7.14) siehe auch Anmerkung 7.2.

Basierend auf dem Zustandswahrscheinlichkeitsvektor (7.14) und Lemma 7.1 kann nun eine *erweiterte Beobachtergleichung* definiert werden, die sowohl für exakte als auch unsichere Ein- und Ausgaben verwendet werden kann.

Definition 7.3 (Erweiterte Tensor-Beobachtergleichung): *Sind die Folgen von Ein- und Ausgabewahrscheinlichkeitsvektoren $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ gegeben und konsistent mit dem stochastischen Automaten \mathcal{A} , dann können die Zustandswahrscheinlichkeitsverteilungen mit*

$$k \geq 0 : \quad \mathbf{p}_z(k+1) = \frac{\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle}{\left\| \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle \right\|_1} \in [0, 1]^N, \quad (7.15)$$

$$k = 0 : \quad \mathbf{p}_z(0) = \mathbf{p}_0$$

für alle Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden. Dabei gilt $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in [0, 1]^Q$ sowie $\mathbf{p}_v(k) \in [0, 1]^M$.

Alternativ kann die Beobachtergleichung (7.13), beziehungsweise (7.15) auch elementweise notiert werden:

$$\begin{aligned} k \geq 0 : \quad p_z(k+1, \bar{z}) &= \frac{\sum_{w=1}^Q \sum_{z=1}^N \sum_{v=1}^M l(\bar{z}, w, z, v) \cdot p_w(k, w) \cdot p_z(k, z) \cdot p_v(k, v)}{\sum_{\bar{z}=1}^N \sum_{w=1}^Q \sum_{z=1}^N \sum_{v=1}^M l(\bar{z}, w, z, v) \cdot p_w(k, w) \cdot p_z(k, z) \cdot p_v(k, v)}, \\ k = 0 : \quad p_z(0, z) &:= (p_0)_z. \end{aligned} \quad (7.16)$$

Die erweiterte Beobachtergleichung (7.15), beziehungsweise deren elementweise Notation (7.16), kann dabei als die tensorbasierte Variante einer bereits in [113, S. 163–165] beschriebenen und bewiesenen Gleichung zur Berücksichtigung unsicherer Ein- und Ausgaben betrachtet werden.

Für beliebige Folgen von Ein- und Ausgabewahrscheinlichkeitsvektoren $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ erhält man mit (7.15) eine Folge von Zustandswahrscheinlichkeitsvektoren

$$\mathbf{P}_z(0...T+1 \mid \mathbf{P}_v(0...T), \mathbf{P}_w(0...T)) = (\mathbf{p}_z(0), \mathbf{p}_z(1), \mathbf{p}_z(2), \dots, \mathbf{p}_z(T+1)) . \quad (7.17)$$

Dabei ist klar, dass Definition 7.3 für exakte E/A Folgen $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$, die ausschließlich Elemente $\mathbf{p}_w(k) \in \{0, 1\}^Q$ und $\mathbf{p}_v(k) \in \{0, 1\}^M$ für $k = 0, 1, \dots, T$ besitzen, in Lemma 7.1 übergeht und das in diesem Fall die Gleichheit

$$\mathbf{P}_z(0...T+1 \mid \mathbf{P}_v(0...T), \mathbf{P}_w(0...T)) = \mathbf{P}_z(0...T+1 \mid V(0...T), W(0...T)) \quad (7.18)$$

gilt. Die Elemente $\mathbf{p}_z(k)$ der Folge (7.17) besitzen in diesem Fall die Verteilung (3.59). Die erweiterte Beobachtergleichung (7.15) kann damit als eine allgemeine Darstellungsform betrachtet werden, da sie keine Restriktionen für die Ein- und Ausgaben vorsieht, aber als Spezialfall die Lösung für die Beobachtung mit exakt bekannten Ein- und Ausgaben beinhaltet. Für die theoretischen Betrachtungen im nächsten Abschnitt wird, solange nicht explizit auf die Berücksichtigung von Unsicherheiten hingewiesen wird, davon ausgegangen, dass die Eingaben und Ausgaben exakt bekannt sind.

Anmerkung 7.2 (Notation des Zustandsvektors): *Auch wenn der Vektor der Zustandswahrscheinlichkeitsverteilung (7.14) bei unsicheren Ein- und Ausgaben eine andere Verteilung repräsentiert als der Zustandsvektor des autonomen Automaten (3.18) und der Zustandsvektor bei exakten skalaren Ein- und Ausgabefolgen (3.59), werden zugunsten der Übersichtlichkeit, alle Vektoren mit dem gleichen Symbol \mathbf{p}_z bezeichnet. Welche Verteilung dem Zustandsvektor in den weiteren Betrachtungen zugrunde liegt, wird dabei jederzeit aus dem Kontext ersichtlich sein.*

Konsistenzprüfung der Ein- und Ausgabefolgen: Der erweiterte Beobachtergleichung aus Definition 7.3 kann ebenso zur Fehlererkennung verwendet werden. Die Prüfung, ob die E/A Folgen $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ konsistent mit dem stochastischen Automaten \mathcal{A} sind, erfolgt wieder anhand des Nenners der erweiterten Beobachtergleichung (7.15), der mit

$$b(k) := \left\| \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle \right\|_1 \in [0, 1] \quad (7.19)$$

abgekürzt wird. Mit dem Wissen, dass (7.19) nur für exakt bekannte Ein- und Ausgaben auf die gleichen Werte wie Gleichung (3.62) führt, wird auch hier das Symbol $b(k)$ zur Abkürzung des Nenners verwendet. Für mit dem stochastischen Automaten konsistente E/A Folgen erhält man damit die Folge

$$B(0...T \mid \mathbf{P}_v(0...T), \mathbf{P}_w(0...T)) = (b(0), b(0), \dots, b(T)) , \quad (7.20)$$

die ebenfalls als Verhaltenssignal bezeichnet wird. Für *exakte* E/A Folgen ergibt sich aus Lemma 7.1 der Zusammenhang, dass entsprechend

$$B(0...T \mid \mathbf{P}_v(0...T), \mathbf{P}_w(0...T)) = B(0...T \mid V(0...T), W(0...T)) \quad (7.21)$$

gilt (s. a. Gleichung (7.18)). Die erweiterte Beobachtergleichung (7.15) erzeugt somit für exakte vektorielle E/A Folgen das gleiche Verhaltenssignal (3.63) wie die einfache Beobachtergleichung (7.6). Die Konsistenzprüfung der vektoriellen Ein- und Ausgabefolgen erfolgt dabei analog zur Beschreibung im Absatz 3.5.3. Das bedeutet, eine E/A Folge $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ mit den E/A Paaren $(\mathbf{p}_v(k), \mathbf{p}_w(k))$ ist konsistent mit dem stochastischen Automaten \mathcal{A} , wenn für die Elemente $b(k)$ der Verhaltensfolge (7.20) $b(k) > 0$ für $k = 0, 1, 2, \dots, T$ gilt. Hingegen ist eine E/A Folge $(\mathbf{P}_v(0...T), \mathbf{P}_w(0...T))$ inkonsistent mit dem stochastischen Automaten \mathcal{A} , wenn ein Element $b(k)$ der Verhaltensfolge (7.20) den Wert Null annimmt.

7.2 Tensorrepräsentation qualitativer Modelle

Analog zum Verhaltenstensor des stochastischen Automaten in Definition 7.1 definiert man auch den Verhaltenstensor für das qualitative Modell aus Definition 4.1. Dieser tritt dabei an die Stelle der Verhaltensmatrix aus Definition 4.2.

7.2.1 Verhaltenstensor

Die approximierten bedingten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$, die man aus der qualitativen stochastischen Identifikation in Gleichung (4.27) erhält, werden durch einen entsprechenden Verhaltenstensor vierter Ordnung repräsentiert.

Definition 7.4 (Verhaltenstensor des qualitativen Modells): *Der Tensor*

$$\tilde{\mathbf{L}} \in [0, 1]^{N \times Q \times N \times M} \quad (7.22)$$

für dessen Elemente

$$\tilde{l}(\bar{z}, w, z, v) = \tilde{\mathcal{L}}(\bar{z}, w | z, v) \quad \forall \bar{z}, z \in \mathcal{X}, w \in \mathcal{Y}, v \in \mathcal{V} \quad (7.23)$$

gilt, heißt Verhaltenstensor des qualitativen Modells.

Der Verhaltenstensor (7.22) enthält für ein bestimmtes qualitatives Modell stets die gleichen Wahrscheinlichkeiten wie eine entsprechende Verhaltensmatrix. Das Beispiel 7.3 zeigt dies im Vergleich zur Verhaltensmatrix (5.16) aus dem Beispiel in Abschnitt 5.2, Seite 93.

Beispiel 7.3: Wie man im Vergleich mit der Verhaltensmatrix (5.25) aus den dem Beispiel in Abschnitt 5.2 feststellen kann, ändert sich durch die Verwendung eines Tensors zur Speicherung der bedingten Wahrscheinlichkeiten lediglich die Indexierung. Dabei wird der Index-Operator »:« aus Abschnitt 6.1.2 verwendet, um alle Indizes \bar{z} und z für eine gegebene E/A Kombination (w, v) auszuwählen. Anstelle der Verhaltensmatrix $\tilde{\mathbf{L}} \in [0, 1]^{18 \times 12}$ mit den Blöcken $\tilde{\mathbf{L}} \in [0, 1]^{6 \times 6}$ tritt der Verhaltenstensor $\tilde{\mathbf{L}} \in [0, 1]^{6 \times 3 \times 6 \times 2}$ mit den Tensorscheiben

$$\begin{aligned} \tilde{\mathbf{L}}(:, 1, :, 1) &= \begin{pmatrix} 0.921 & 0 & 0 & 0 & 0 & 0 \\ 0.079 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}}(:, 1, :, 2) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\ \tilde{\mathbf{L}}(:, 2, :, 1) &= \begin{pmatrix} 0 & 0.155 & 0 & 0 & 0 & 0 \\ 0 & 0.721 & 0 & 0 & 0.333 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.124 & 0 & 0 & 0.667 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}}(:, 2, :, 2) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0.128 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.756 & 0 \\ 0 & 0 & 0 & 0 & 0.112 & 0 \end{pmatrix}, \\ \tilde{\mathbf{L}}(:, 3, :, 1) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}}(:, 3, :, 2) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.055 \\ 0 & 0 & 0 & 0 & 0 & 0.944 \end{pmatrix}. \end{aligned} \quad (7.24)$$

Die bedingte Wahrscheinlichkeit $\mathcal{L}(6, 3, 6, 2) = 0.944$ dafür, dass das quantisierte System bei der qualitativen Eingabe $v = 2$ vom Zustand $z = 6$ in den Zustand $\bar{z} = 5$ wechselt und die Ausgabe $w = 3$ erzeugt, steht somit an der Position $\tilde{l}(6, 3, 6, 2)$.

Analog zu den Tensor-Beobachtergleichungen des stochastischen Automaten (7.6) und (7.15) erhält man die qualitativen Beobachtergleichungen für die Tensorrepräsentation qualitativer Modelle, was im nächsten Abschnitt erläutert wird.

7.2.2 Qualitative Beobachtergleichung

Zunächst können basierend auf Definition 7.1, die approximierten Wahrscheinlichkeiten $\tilde{\mathcal{L}}(\bar{z}, w | z, v)$ aus Gleichung (4.27) in einem Tensor $\tilde{\mathbf{L}} \in [0, 1]^{N \times Q \times N \times M}$ notiert werden. Damit kann die Beobachtergleichung für die Tensorrepräsentation qualitativer Modelle analog zu Definition 7.2 formuliert werden.

Definition 7.5 (Qualitative Tensor-Beobachtergleichung): *Sind die quantisierten Ein- und Ausgangsfolgen $([\mathbf{U}(0...T)], [\mathbf{Y}(0...T)])$ des quantisierten Systems konsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$, dann kann eine Approximation der Wahrscheinlichkeitsverteilungen der qualitativen Zustände mit*

$$\begin{aligned} k \geq 0 : \quad \tilde{\mathbf{p}}_z(k+1) &= \frac{\tilde{\mathbf{L}}(:, w(k), :, v(k)) \cdot \tilde{\mathbf{p}}_z(k)}{\|\tilde{\mathbf{L}}(:, w(k), :, v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1} \in [0, 1]^N, \\ k = 0 : \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0 \end{aligned} \quad (7.25)$$

für alle Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden. Der qualitative Eingang $v(k)$ und Ausgang $w(k)$ sind dabei in jedem Zeitschritt k durch die quantisierte Eingangsgröße $[\mathbf{u}(k)] = v(k)$ und Ausgangsgröße $[\mathbf{y}(k)] = w(k)$ gegeben.

Der Wahrscheinlichkeitsvektor der qualitativen Zustände zur Zeit $k+1$ besitzt dabei die Verteilung aus Gleichung (4.38). Wendet man die qualitative Beobachtergleichung (7.25) auf gegebene quantisierte E/A Folgen $([\mathbf{U}(0...T)], [\mathbf{Y}(0...T)])$ an, erhält man die mit (4.41) identische Folge von Wahrscheinlichkeitsvektoren

$$\tilde{\mathbf{P}}_z(0...T+1 | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]) = (\tilde{\mathbf{p}}_z(0), \tilde{\mathbf{p}}_z(1), \tilde{\mathbf{p}}_z(2), \dots, \tilde{\mathbf{p}}_z(T+1)) , \quad (7.26)$$

welche die approximierten Wahrscheinlichkeitsverteilungen der qualitativen Zustände bis zum Zeithorizont $T+1$ beschreiben. Außerdem wird der Nenner der qualitativen Beobachtergleichung (7.25) mit

$$\tilde{b}(k) := \|\tilde{\mathbf{L}}(:, w(k), :, v(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1 \in [0, 1] , \quad (7.27)$$

abgekürzt, womit man das für die Fehlererkennung wichtige Verhaltenssignal

$$\tilde{B}(0...T | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]) = (\tilde{b}(0), \tilde{b}(1), \tilde{b}(2), \dots, \tilde{b}(T)) \quad (7.28)$$

des qualitativen Modells bilden kann. Dieses ist wiederum identisch mit Gleichung (4.43).

Konsistenzprüfung der Ein- und Ausgangsfolgen: Die Bewertung, ob die quantisierten E/A Folgen $([\mathbf{U}(0...T)], [\mathbf{Y}(0...T)])$ konsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$ sind, erfolgt analog zu den Gleichungen (4.44) und (4.45) in Abschnitt 4.4.2. Damit kann ein einfacher Algorithmus zur qualitativen Fehlererkennung definiert werden.

7.2.3 Algorithmus zur qualitativen Fehlererkennung

Der Ablauf des Beobachteralgorithmus 7.1 ist identisch mit dem Beobachteralgorithmus 4.1, der auf der Matrixrepräsentation qualitativer Modelle beruht.

Algorithmus 7.1: Algorithmus zur qualitativen Fehlererkennung (Tensorrepräsentation)

Gegeben: Qualitatives Modell \tilde{A} , Gl. (4.15) mit Verhaltenstensor \tilde{L} , Gl. (7.22)
 Quantisierte Eingangsfolge $[U(0...T)]$ mit Elementen $[u(k)] = v(k)$, Gl. (4.4)
 Quantisierte Ausgangsfolge $[Y(0...T)]$ mit Elementen $[y(k)] = w(k)$, Gl. (4.6)

Initialisiere: $k = 0$, $\tilde{p}_z(0) := \tilde{p}_0$

Iteriere:

- 1 **solange** $0 \leq k \leq T$
- 2 **berechne** $\tilde{L}(:, w(k), :, v(k)) \cdot \tilde{p}_z(k)$ Gl. (7.25)
- 3 **berechne** $\|\tilde{L}(:, w(k), :, v(k)) \cdot \tilde{p}_z(k)\|_1 =: \tilde{b}(k)$, Gl. (7.25)
- 4 **berechne** $\tilde{b}_m(k)$, Gl. (4.46) und $\tilde{b}_b(k)$, Gl. (4.48)
- 5 **wenn** $\tilde{b}(k) = 0$ **dann**
- 6 **setze** $\tilde{p}_z(k+1) = \tilde{p}_0$
- 7 **setze** $k := k+1$ und fahre fort mit Schritt 1
- 8 **sonst**
- 9 **berechne** $\tilde{p}_z(k+1)$, Gl. (7.25)
- 10 **setze** $k := k+1$ und fahre fort mit Schritt 1

Ergebnis: Folge $\tilde{P}_z(0...T+1 | [U(0...T)], [Y(0...T)])$, Gl. (7.26)
 Folge $\tilde{B}(0...T | [U(0...T)], [Y(0...T)])$, Gl. (7.28)
 Folge $\tilde{B}_m(0...T)$, Gl. (4.47)
 Folge $\tilde{B}_b(0...T)$, Gl. (4.49)

7.2.4 Erweiterte qualitative Beobachtergleichung

Im Folgenden wird eine erweiterte qualitative Beobachtergleichung eingeführt, mit der sich Unsicherheiten in den quantisierten Ein- und Ausgangsgrößen berücksichtigen lassen. Dies stellt eine interessante Möglichkeit für die praktische Anwendung qualitativer Modelle dar, denn reale Messdaten können durch unterschiedliche Gründe, wie korrodierte Stecker, schlechte elektrische Abschirmungen oder äußere Störeinflüsse, mit Unsicherheiten behaftet sein. Beispiel 7.4 zeigt hierbei anschaulich, wie sich Unsicherheiten in den Ein- und Ausgängen auf die Zustandswahrscheinlichkeiten auswirken.

Qualitative Ein- und Ausgangswahrscheinlichkeitsvektoren

Zunächst können die Wahrscheinlichkeiten der qualitativen Ausgänge $\Pr([y(k)] = w)$ und der qualitativen Eingänge $\Pr([u(k)] = v)$ analog zu Abschnitt 7.1.3 für alle $w \in \mathcal{Y}$ und $v \in \mathcal{U}$ vektoriell notiert werden:

$$\tilde{p}_v(k) = \begin{pmatrix} \Pr([u(k)] = 1) \\ \Pr([u(k)] = 2) \\ \vdots \\ \Pr([u(k)] = M) \end{pmatrix} \in [0, 1]^M, \quad (7.29)$$

$$\tilde{p}_w(k) = \begin{pmatrix} \Pr([y(k)] = 1) \\ \Pr([y(k)] = 2) \\ \vdots \\ \Pr([y(k)] = Q) \end{pmatrix} \in [0, 1]^Q. \quad (7.30)$$

Der Vektor $\tilde{\mathbf{p}}_v(k)$ wird in diesem Kontext als *Wahrscheinlichkeitsvektor der qualitativen Ausgänge* und der Vektor $\tilde{\mathbf{p}}_w(k)$ als *Wahrscheinlichkeitsvektor der qualitativen Eingänge* bezeichnet. Die quantisierten Ein- und Ausgangsfolgen $[U(0...T)]$ und $[Y(0...T)]$ werden damit wieder durch eine Folge von Wahrscheinlichkeitsvektoren

$$\tilde{\mathbf{P}}_v(0...T) = (\tilde{\mathbf{p}}_v(0), \tilde{\mathbf{p}}_v(1), \tilde{\mathbf{p}}_v(2), \dots, \tilde{\mathbf{p}}_v(T)) \quad (7.31)$$

und

$$\tilde{\mathbf{P}}_w(0...T) = (\tilde{\mathbf{p}}_w(0), \tilde{\mathbf{p}}_w(1), \tilde{\mathbf{p}}_w(2), \dots, \tilde{\mathbf{p}}_w(T)) \quad (7.32)$$

ersetzt.

Exakte qualitative Ein- und Ausgangswahrscheinlichkeitsvektoren: Sind die qualitativen Eingänge $v \in \mathcal{U}$ und Ausgänge $w \in \mathcal{W}$ zu jeder Zeit $k = 0, 1, \dots, T$ exakt bekannt und durch die eindeutigen quantisierten Eingangs- und Ausgangsfolgen $[U(0...T)]$ und $[Y(0...T)]$ gegeben, dann besitzen die qualitativen Ein- und Ausgangswahrscheinlichkeitsvektoren wieder nur an der $v(k)$ -ten und $w(k)$ -ten Stelle den Wert Eins, während alle anderen Komponenten den Wert Null haben. Für die Wahrscheinlichkeitsvektoren der qualitativen Ein- und Ausgänge gilt dann entsprechend $\tilde{\mathbf{p}}_v(k) \in \{0, 1\}^M$ und $\tilde{\mathbf{p}}_w(k) \in \{0, 1\}^Q$ für $k = 0, 1, 2, \dots, T$. Die Folgen der Wahrscheinlichkeitsvektoren der qualitativen Eingänge (7.31) und Ausgänge (7.32) werden in diesem Fall wieder als *exakt* bezeichnet. Siehe hierzu auch Abschnitt 7.1.3 und Beispiel 7.1. Im Folgenden wird kürzer wieder einfach von E/A Folgen $(\tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T))$ und E/A Paaren $(\tilde{\mathbf{p}}_v(k), \tilde{\mathbf{p}}_w(k))$ gesprochen. Die erweiterte qualitative Beobachtergleichung folgt somit unmittelbar aus Gleichung (7.15) und wird hier wieder in einer Allgemeinen Form definiert.

Definition 7.6 (Erweiterte qualitative Tensor-Beobachtergleichung): *Sind die Folgen von Wahrscheinlichkeitsvektoren qualitativer Ein- und Ausgänge $(\tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T))$ gegeben und konsistent mit dem qualitativen Modell \tilde{A} , dann kann eine Approximation der Wahrscheinlichkeitsverteilungen der qualitativen Zustände mit*

$$\begin{aligned} k \geq 0 : \quad \tilde{\mathbf{p}}_z(k+1) &= \frac{\langle \tilde{\mathbf{L}} \mid \llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket \rangle}{\left\| \langle \tilde{\mathbf{L}} \mid \llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket \rangle \right\|_1} \in [0, 1]^N, \\ k = 0 : \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0 \end{aligned} \quad (7.33)$$

für $k = 0, 1, 2, \dots, T$ berechnet werden. Dabei ist $\tilde{\mathbf{p}}_w(k) \in [0, 1]^Q$, $\tilde{\mathbf{p}}_v(k) \in [0, 1]^M$ und $\tilde{\mathbf{p}}_z(k) \in [0, 1]^N$.

In Gleichung (7.33) ist

$$\llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket = \tilde{\mathbf{p}}_w(k) \circ \tilde{\mathbf{p}}_z(k) \circ \tilde{\mathbf{p}}_v(k) \in [0, 1]^{Q \times N \times M} \quad (7.34)$$

wiederum ein Kruskal Rang-1 Tensor wie in Gleichung (7.12). Analog zum Zustandsvektor (7.14) wird der Wahrscheinlichkeitsvektor der qualitativen Zustände zur Zeit $k+1$ durch

$$\tilde{\mathbf{p}}_z(k+1) = \begin{pmatrix} \Pr([\mathbf{x}(k+1)] = 1 \mid \tilde{\mathbf{P}}_v(0...k), \tilde{\mathbf{P}}_w(0...k)) \\ \Pr([\mathbf{x}(k+1)] = 2 \mid \tilde{\mathbf{P}}_v(0...k), \tilde{\mathbf{P}}_w(0...k)) \\ \vdots \\ \Pr([\mathbf{x}(k+1)] = N \mid \tilde{\mathbf{P}}_v(0...k), \tilde{\mathbf{P}}_w(0...k)) \end{pmatrix} \in [0, 1]^N \quad (7.35)$$

in einer allgemeinen Form definiert. Für mit dem qualitativen Modell $\tilde{\mathcal{A}}$ konsistente Ein- und Ausgangsfolgen $(\tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T))$ generiert die Beobachtergleichung (7.33) eine Folge von Zustandswahrscheinlichkeitsvektoren, die mit

$$\tilde{\mathbf{P}}_z(0...T+1 | \tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T)) = (\tilde{\mathbf{p}}_z(0), \tilde{\mathbf{p}}_z(1), \tilde{\mathbf{p}}_z(2), \dots, \tilde{\mathbf{p}}_z(T+1)) \quad (7.36)$$

bezeichnet wird. Aus dem Nenner der Beobachtergleichung folgt wiederum die Abkürzung

$$\tilde{b}(k) := \left\| \left\langle \tilde{\mathbf{L}} \mid \llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket \right\rangle \right\|_1, \quad (7.37)$$

woraus sich dann das Verhaltenssignal

$$\tilde{B}(0...T | \tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T)) = (\tilde{b}(0), \tilde{b}(1), \tilde{b}(2), \dots, \tilde{b}(T)) \quad (7.38)$$

des qualitativen Modells ergibt. Auch hier werden somit zugunsten einer einheitlichen Beschreibung der Algorithmen zur Fehlerdetektion die gleichen Symbole $\tilde{\mathbf{p}}_z$ und \tilde{b} verwendet wie bei der einfachen Beobachtergleichung in (7.25) und (7.27). Dabei gilt analog zu den Beziehungen in den Gleichungen (7.18) und (7.21), dass wenn die E/A Folgen exakt sind und ausschließlich Elemente $\tilde{\mathbf{p}}_v(k) \in \{0, 1\}^M$ und $\tilde{\mathbf{p}}_w(k) \in \{0, 1\}^Q$ enthalten, wiederum folgende Äquivalenz gilt:

$$\tilde{\mathbf{P}}_z(0...T+1 | \tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T)) = \tilde{\mathbf{P}}_z(0...T+1 | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]) , \quad (7.39)$$

$$\tilde{B}(0...T | \tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T)) = \tilde{B}(0...T | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]) . \quad (7.40)$$

Die Gleichung (7.33) erzeugt damit für exakte E/A Folgen von Wahrscheinlichkeitsvektoren die gleiche approximierte vektorielle Zustandsfolge (7.26) und Verhaltensfolge (7.28) wie die qualitative Beobachtergleichung (7.25). Dies ist wiederum eine direkte Schlussfolgerung aus dem Lemma 7.1. Wichtig zu wissen ist, dass selbst für sicher bekannte qualitative E/A Folgen, beide Beobachtergleichungen grundsätzlich nur dann die tatsächlichen Zustandswahrscheinlichkeiten (4.40) des quantisierten Systems beschreiben können, wenn dieses die Markow-Eigenschaft besitzt und das qualitative Modell vollständig ist. Beide Bedingungen sind, wie bereits erwähnt, im Allgemeinen nicht erfüllt.

Konsistenzprüfung der Ein- und Ausgangsfolgen: Die Konsistenzprüfung der E/A Folgen erfolgt analog zu Abschnitt 7.1.3. Das heißt, eine E/A Folge $(\tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T))$ mit den E/A Paaren $(\tilde{\mathbf{p}}_v(k), \tilde{\mathbf{p}}_w(k))$ ist konsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$, wenn für die Elemente $\tilde{b}(k)$ der Verhaltensfolge (7.38) $\tilde{b}(k) > 0$ für $k = 0, 1, 2, \dots, T$ gilt. Dementsprechend ist eine E/A Folge $(\tilde{\mathbf{P}}_v(0...T), \tilde{\mathbf{P}}_w(0...T))$ inkonsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$, wenn $\tilde{b}(k) = 0$ ist.

Beispiel 7.4 (Unsichere Ein- und Ausgangsvektoren): Die Abbildung 7.1 zeigt am Beispiel eines Erhitzers einer RLT-Anlage, welchen Einfluss eine 20%-ige Unsicherheit in den qualitativen Ein- und Ausgangswahrscheinlichkeitsvektoren auf die qualitativen Zustandswahrscheinlichkeitsverteilungen hat. Es handelt sich hierbei um ein SISO-System mit zwei Zuständen, die durch die Wasseraustrittstemperatur T_{WA} und die Wassereintrittstemperatur T_{WE} gegeben sind. Die Größe T_{WA} besitzt zwei und die Größe T_{WE} drei qualitative Zustände. Die obere Reihe in Abbildung 7.1 zeigt die qualitativen Zustandswahrscheinlichkeiten für exakte Ein- und Ausgänge. In den unteren beiden Abbildungen sind die Zustandswahrscheinlichkeitsverteilungen unter Berücksichtigung einer Unsicherheit von 20% dargestellt. Wie zu erkennen ist, führt diese Unsicherheit insbesondere bei der Wasseraustrittstemperatur T_{WA} zu einer Änderung der Wahrscheinlichkeitsverteilungen der qualitativen Zustände. Bei der Wassereintrittstemperatur T_{WE} ist die Auswirkung geringer und in der Abbildung kaum zu erkennen, wobei jedoch auch hier ein Unterschied besteht. So beträgt beispielsweise zum Zeitpunkt $k = 1$ und für exakte Ein- und Ausgänge, die Wahrscheinlichkeit dafür, dass sich die Wassereintrittstemperatur im qualitativen Zustand $z = 1$ (untere Partition) befindet 4.8%. Für den qualitativen Zustand $z = 2$ (mittlere Partition) liegt diese bei 66.8% und für den Zustand $z = 3$ (obere Partition) bei 28.4%. Für die unsicheren Ein- und Ausgänge betragen die qualitativen Zustandswahrscheinlichkeiten hingegen 8.3% ($z = 1$), 62.1% ($z = 2$) und 29.6% ($z = 3$).

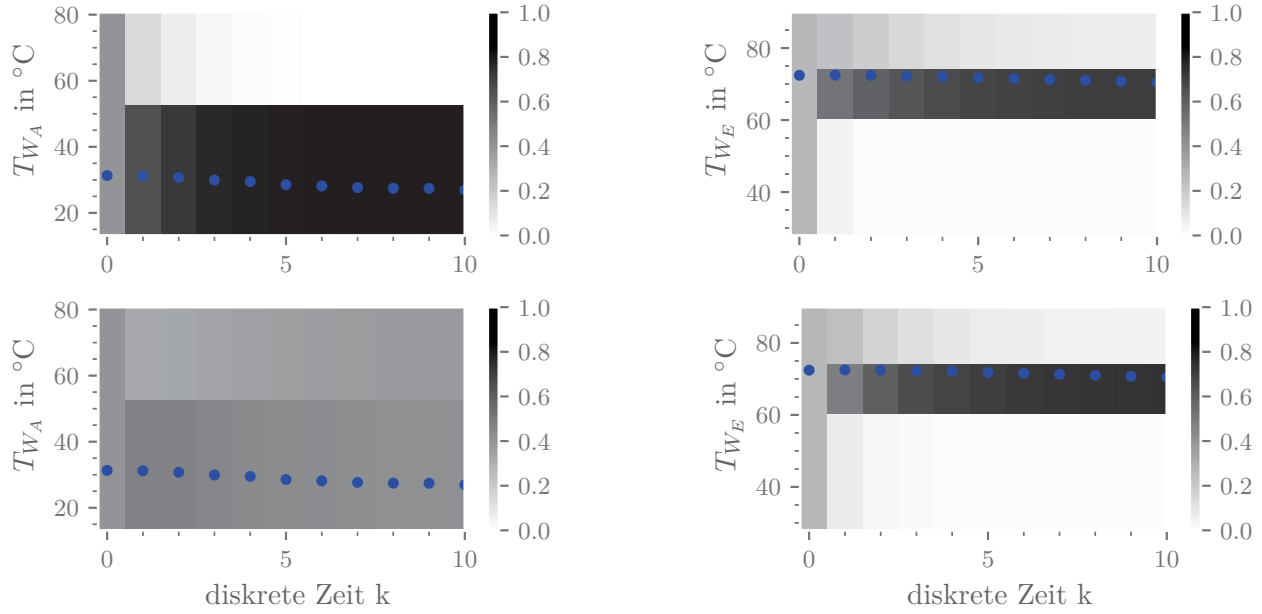


Abbildung 7.1: Qualitative Zustandswahrscheinlichkeiten für exakte qualitative Ein- und Ausgänge (oben) und mit 20% Unsicherheit (unten)

7.2.5 Erweiterter Algorithmus zur qualitativen Fehlererkennung

Der Ablauf des erweiterten Algorithmus 7.2 zur qualitativen Fehlerdetektion ist identisch zu dem der Algorithmen 4.1 und 7.1.

Algorithmus 7.2: Erweiterter Algorithmus zur qualitativen Fehlererkennung

Gegeben: Qualitatives Modell \tilde{A} , Gl. (4.15) mit Verhaltenstensor \tilde{L} , Gl. (7.22)
 Folge von Wahrscheinlichkeitsvektoren qualitativer Eingänge $\tilde{\mathbf{P}}_v(0..T)$, Gl. (7.31)
 Folge von Wahrscheinlichkeitsvektoren qualitativer Ausgänge $\tilde{\mathbf{P}}_w(0..T)$, Gl. (7.32)

Initialisiere: $k = 0$, $\tilde{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$

Iteriere:

- 1 solange $0 \leq k \leq T$
- 2 berechne $\left\langle \tilde{L} \mid \llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket \right\rangle$, Gl. (7.33)
- 3 berechne $\left\| \left\langle \tilde{L} \mid \llbracket \tilde{\mathbf{p}}_w(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v(k) \rrbracket \right\rangle \right\|_1 =: \tilde{b}(k)$, Gl. (7.33)
- 4 berechne $\tilde{b}_m(k)$, Gl. (4.46) und $\tilde{b}_b(k)$, Gl. (4.48)
- 5 wenn $\tilde{b}(k) = 0$ dann
- 6 setze $\tilde{\mathbf{p}}_z(k+1) = \tilde{\mathbf{p}}_0$
- 7 setze $k := k+1$ und fahre fort mit Schritt 1
- 8 sonst
- 9 berechne $\tilde{\mathbf{p}}_z(k+1)$, Gl. (7.33)
- 10 setze $k := k+1$ und fahre fort mit Schritt 1

Ergebnis: Folge $\tilde{\mathbf{P}}_z(0..T+1 \mid \tilde{\mathbf{P}}_v(0..T), \tilde{\mathbf{P}}_w(0..T))$, Gl. (7.36)
 Folge $\tilde{B}(0..T \mid \tilde{\mathbf{P}}_v(0..T), \tilde{\mathbf{P}}_w(0..T))$, Gl. (7.38)
 Folge $\tilde{B}_m(0..T)$, Gl. (4.47)
 Folge $\tilde{B}_b(0..T)$, Gl. (4.49)

Verwendet man den Algorithmus mit exakten E/A Folgen, erhält man entsprechend den Gleichungen (7.39) und (7.40), die Folge $\tilde{\mathbf{P}}_z(0...T + 1 | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)])$ sowie das Verhaltenssignal $\tilde{B}(0...T | [\mathbf{U}(0...T)], [\mathbf{Y}(0...T)])$. Die Anmerkung 4.2 zur Speicherung der Zustandsfolge gilt in Bezug auf den Algorithmus 7.2 im gleichen Sinne.

8

Tensordekomposition

In diesem Kapitel werden zwei verschiedene Methoden der Tensordekomposition beschrieben sowie deren Grundlagen erläutert und die entsprechenden Notationen und Begriffe eingeführt. Einleitend wird in Abschnitt 8.1 ein kurzer Überblick bezüglich der historischen Entwicklung und des Ursprungs der Dekompositionsmethoden gegeben. In Abschnitt 8.2 wird die CP-Dekomposition vorgestellt und gezeigt, wie diese zur Reduktion des Speicherbedarfs des Verhaltenstensors eingesetzt werden kann und welche Werkzeuge hierfür zur Verfügung stehen. Abschnitt 8.3 behandelt die TT-Dekomposition und stellt die Vor- und Nachteile gegenüber der CP-Dekomposition sowie die Unterschiede im Hinblick auf den Speicherbedarf heraus.

8.1 Einleitung

Die *Tensordekomposition* ist ein Faktorisierungsverfahren, das im Allgemeinen eine Generalisierung einer Matrix Faktorisierung darstellt und dessen Grundidee bereits 1927 von Hitchcock [42] beschrieben wurde [56, S. 456–457]. Die wohl bekannteste Methode zur Faktorisierung oder Dekomposition von Matrizen ist die aus der linearen Algebra bekannte *Singulärwertzerlegung* (engl.: Singular Value Decomposition (SVD)). Eine der ersten Erweiterungen der SVD für das Gebiet der multilinearen Algebra stellt die Tucker Dekomposition dar, die 1966 durch Tucker [119] für dreidimensionale Tensoren entwickelt wurde und von der mittlerweile mehrere Verallgemeinerungen für N -dimensionale Tensoren existieren, wie beispielsweise die von De Lathauwer et al. [28] publizierte *Singulärwertzerlegung höherer Ordnung* (engl.: Higher Order Singular Value Decomposition (HOSVD)) [56, S. 474–477].

Eine weitere Tensordekompositionsmethode ist die *CP-Dekomposition* (engl.: CP-Decomposition), welche ebenfalls als eine Generalisierung der SVD angesehen werden kann [25, S. 43]. Die Abkürzung **CP** hat ihren Ursprung in der Arbeit von Kiers [48] und steht für die 1970 entwickelten Faktorisierungsmodelle **CANDECOMP** (CANonical DECOMPosition [20]) und **PARAFAC** (PARAllel Factors [37]). Neben der Tucker und der CP-Dekomposition existieren auch aktuellere und modernere Verfahren, wie zum Beispiel die im Jahr 2011 von Oseledets [99] vorgestellte *Tensor-Train (TT) Dekomposition*. Das Verfahren der TT-Dekomposition stammt aus dem Bereich der numerischen Analysis und der Quantenphysik [35, S. 4], während die CP-Dekomposition ihren Ursprung in der Psychometrie hat [56, S. 456]. Die aktuellen Anwendungsfelder beider Dekompositionsmethoden erstrecken sich über viele Gebiete der Mathematik, Physik und Ingenieurwissenschaften. So finden diese Ansätze zum Beispiel Anwendung in der Signalverarbeitung (Signaltrennung und Rauschunterdrückung), in der Bildverarbeitung und -kompression, bei bildgebenden medizinischen Verfahren wie dem EEG sowie bei Methoden des maschinellen Lernens und der *Modellreduktion* [25, S. xvii–xxi], [56, S. 456].

Von besonderem Interesse in dieser Arbeit ist der Aspekt der Modellreduktion, unter der in diesem Fall die Repräsentation des Verhaltenstensors durch dessen approximierter und in der Dimensionalität reduzierter Faktorisierung verstanden wird [25, S. xvii]. In den folgenden Abschnitten wird erläutert, wie sich die CP-Dekomposition (Abschnitt 8.2) und die TT-Dekomposition (Abschnitt 8.3) zur Faktorisierung des in Definition 7.1 eingeführten Verhaltenstensors nutzen lassen. Die Erläuterungen beziehen sich dabei stets auf den Verhaltenstensor des stochastischen Automaten L , sind aber analog auf den Verhaltenstensor des qualitativen Modells \tilde{L} in Definition 7.4 anwendbar.

8.2 CP-Dekomposition

In den folgenden Abschnitten werden die algebraischen Grundlagen der *CP-Dekomposition* und deren Anwendung beschrieben. Grundlagen- und Übersichtsliteratur bieten dabei insbesondere die Arbeiten von Cichocki et al. [25], Comon et al. [26], Grasedyck et al. [35], Hackbusch [36], Kiers [48] und Kolda und Bader [56].

8.2.1 Grundlagen der CP-Dekomposition

Definition 8.1 (CP-Dekomposition [56, S. 463–464]): Die *CP-Dekomposition* eines Tensors N -ter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ ist definiert als dessen Zerlegung in eine Summe von R Rang-1 Tensoren

$$\mathbf{Y} \approx \llbracket \boldsymbol{\lambda}; \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)} \rrbracket \equiv \sum_{r=1}^R \lambda_r \bigcirc_{n=1}^N \mathbf{x}_{:r}^{(n)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}, \quad (8.1)$$

die für $n = 1, \dots, N$ durch die dyadischen Produkte der Spaltenvektoren $\mathbf{x}_{:r}^{(n)} \in \mathbb{R}^{J_n}$ der Faktormatrizen $\mathbf{X}^{(n)} \in \mathbb{R}^{J_n \times R}$ gegeben und mit den Faktoren λ_r des Vektors $\boldsymbol{\lambda} \in \mathbb{R}^R$ gewichtet sind. Die elementweise Notation ist dabei durch

$$y(j_1, j_2, \dots, j_N) \approx \sum_{r=1}^R \lambda_r \cdot x_{j_1 r}^{(1)} \cdot x_{j_2 r}^{(2)} \cdot \dots \cdot x_{j_N r}^{(N)} \quad \forall j_n \in \{1, \dots, J_n\}, n = 1, \dots, N \quad (8.2)$$

gegeben.

Das Ergebnis der CP-Dekomposition eines Tensor $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ ist also ein Kruskal-Tensor $\llbracket \cdot \rrbracket$ nach Definition 6.18, der eine Anzahl von N Faktormatrizen $\mathbf{X}^{(n)} \in \mathbb{R}^{J_n \times R}$ mit J_n Zeilen und je R Spalten sowie den Gewichtungsvektor $\boldsymbol{\lambda}$ der Länge R enthält. Der Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}^R$ mit den Komponenten λ_r enthält dabei sogenannte Skalierungswerte, falls die Spaltenvektoren der Faktormatrizen normalisiert wurden, was für verschiedene Anwendungen hilfreich ist [56, S. 464]. Erfolgt keine Normalisierung gilt allgemein $\boldsymbol{\lambda} = \mathbf{1}$ (vgl. Gleichung (6.42)). Die Gleichung (8.1) wird auch als *CP-Repräsentation* des Tensors \mathbf{Y} bezeichnet.

Die folgende Abbildung zeigt eine grafische Interpretation der CP-Dekomposition eines Tensors dritter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ als Summe der dyadischen Produkte dreier Spaltenvektoren $\mathbf{x}_{:r}^{(1)} \in \mathbb{R}^{J_1}$, $\mathbf{x}_{:r}^{(2)} \in \mathbb{R}^{J_2}$ und $\mathbf{x}_{:r}^{(3)} \in \mathbb{R}^{J_3}$ für $r = 1, \dots, R$.

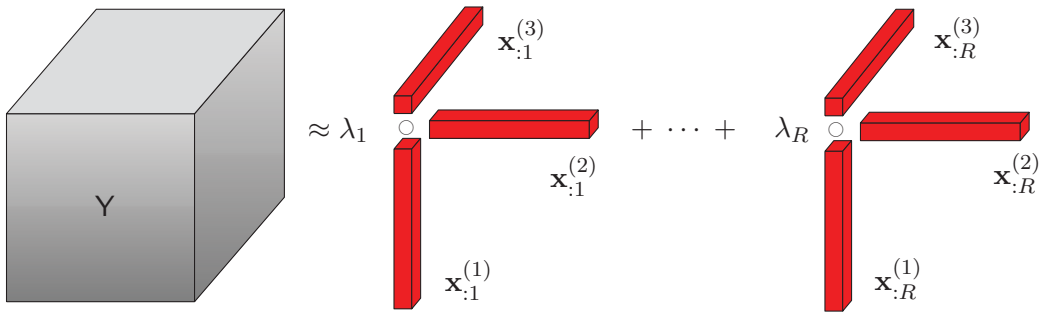


Abbildung 8.1: CP-Dekomposition eines Tensors dritter Ordnung, nach [56, S. 463]

Das Konzept der CP-Dekomposition besteht somit darin, dass nicht der gesamte Tensor \mathbf{Y} , sondern nur dessen Zerlegung, in Form des Kruskal-Tensors mit den Faktormatrizen $\mathbf{X}^{(n)} \in \mathbb{R}^{J_n \times R}$ und dem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}^R$ gespeichert werden muss. Die CP-Dekomposition muss dabei

im Allgemeinen nicht zwangsläufig eine Approximation darstellen, denn generell kann jeder beliebige Tensor $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ durch einen entsprechenden Kruskal-Tensor exakt repräsentiert werden [22, S. 284]. Die Frage, die sich in der Praxis dabei stellt, ist, wie groß die Anzahl an Spalten R in den Faktormatrizen $\mathbf{X}^{(n)} \in \mathbb{R}^{J_n \times R}$ des Kruskal-Tensors, beziehungsweise die Anzahl zu summierender Rang-1 Tensoren sein muss, damit im Idealfall Gleichheit in (8.1) besteht oder aber der ursprüngliche Tensor \mathbf{Y} mit zumindest hinreichender Genauigkeit approximiert wird. Wie präzise die CP-Dekomposition den Ursprungstensor wiedergibt, hängt dabei im Wesentlichen vom Wert R ab, den man als *Dekompositionsrang* oder *Rang der Dekomposition* bezeichnet [58, S. 8].

Setzt man einen aus der CP-Dekomposition resultierenden Kruskal-Tensor wieder zusammen, das heißt man summiert die dyadischen Produkte der Spaltenvektoren $\mathbf{x}_{:r}^{(n)} \in \mathbb{R}^{J_n}$ der Faktormatrizen $\mathbf{X}^{(n)} \in \mathbb{R}^{J_n \times R}$ unter Einbeziehung der Gewichte λ_r auf, dann bezeichnet man dies als *Komposition*. Stellt die CP-Dekomposition dabei wie in Definition 8.1 nur eine Approximation des Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ dar, dann führt die Komposition des Kruskal-Tensors entsprechend auf einen Tensor $\hat{\mathbf{Y}} \approx \mathbf{Y}$, der lediglich eine Näherung des Ursprungstensors repräsentiert. Im folgenden Abschnitt werden die Zusammenhänge zwischen dem Rang der Dekomposition R , einer exakten CP-Dekomposition und einer Approximation näher erläutert.

Tensorrang-Dekomposition und Rang-R Approximation

Zwischen dem in Definition 6.17 festgelegten Begriff des Tensorrangs und der CP-Dekomposition besteht der Zusammenhang, dass der Rang eines Tensors der minimalen Anzahl von Rang-1 Termen R in einer *exakten* CP-Dekomposition entspricht [56, S. 464]. Eine CP-Dekomposition mit dieser Eigenschaft wird auch als *Tensorrang-Dekomposition* (engl.: tensor rank decomposition [56], [123]) oder *kanonische polyadische Dekomposition* (engl.: canonical polyadic decomposition (CPD) [31]) bezeichnet.

Definition 8.2 (Tensorrang-Dekomposition [56, S. 464]): *Eine exakte CP-Dekomposition eines Tensors \mathbf{Y} mit $R = \text{rang}(\mathbf{Y})$ Rang-1 Termen, ist eine Tensorrang-Dekomposition.*

Es soll angemerkt werden, dass eine exakte CP-Dekomposition nicht unbedingt eine Tensorrang-Dekomposition darstellt, da auch eine Faktorisierung mit $R > \text{rang}(\mathbf{Y})$ exakt sein kann und somit die Anzahl der Rang-1 Terme nicht minimal ist.

Von besonderem Interesse sind häufig die sogenannten besten *Niedrig-Rang Approximationen* (engl.: low-rank approximations [35], [56])¹. Im Kontext der CP-Dekomposition spricht man dabei auch von der besten *Rang-R Approximation* (engl.: rank-r approximation [27]). Damit sind CP-Dekompositionen mit einem Dekompositionsrang $R < \text{rang}(\mathbf{Y})$ gemeint, für die es folglich keine exakte Lösung geben kann und die den ursprünglichen Tensor näherungsweise repräsentieren. Die Existenz der besten Rang-R Approximation ist dabei nicht zwangsläufig gewährleistet, da für manche Tensoren auch mit $R < \text{rang}(\mathbf{Y})$ beliebig präzise Approximationen berechnet werden können [56, S. 469]. Dieses Problem wird in der Literatur auch als Degenerierung der CP-Dekomposition bezeichnet [56, S. 469], [59, S. 117].

Sowohl die Tensorrang Dekomposition als auch die beste Rang-R Approximation sind eher theoretischer Natur, da beide letztlich die Kenntnis des Tensorrangs voraussetzen. Die Berechnung des Tensorrangs ist jedoch, wie bereits in Abschnitt 6.5.1 erläutert, nur für Tensoren niedriger Ordnung und geringer Dimensionsgrößen möglich. Die in dieser Arbeit auftretenden Verhaltenstensoren sind in ihrer Dimensionalität deutlich größer als die in der Literatur behandelten Beispiele, für die eine

¹Es sei angemerkt, dass der Begriff der Niedrig-Rang Approximation nicht nur für die CP-Dekomposition, sondern auch für andere Faktorisierungsverfahren verwendet wird. Siehe hierzu [35].

Berechnung des Tensorrangs theoretisch möglich wäre. Eine simple Lösung dieses Problems scheint daher zunächst die Verwendung eines *Brute-Force* Algorithmus zu sein, der numerisch einfach so viele CP-Dekompositionen für $R = 1, 2, \dots$ ausführt, bis eine exakte Faktorisierung gefunden ist. Dies stellt sich in der Praxis jedoch schwierig dar, da die Algorithmen zur Berechnung der CP-Dekomposition aus verschiedenen Gründen die Berechnung einer exakten Dekomposition nicht sicherstellen können [56, S. 471] (s. a. Abschnitt 8.2.2). Allerdings existieren Algorithmen, die ebensolche Brute-Force Methoden nutzen, um innerhalb vordefinierter Approximationskriterien eine Dekomposition zu finden, die ein Optimum im Verhältnis zwischen Speicherreduktion und Genauigkeit bietet [127, S. 29–30].

Aufgrund der Unkenntnis bezüglich des Rangs des Verhaltenstensors und der bereits erwähnten algorithmischen Limitierung kann hier mit großer Sicherheit angenommen werden, dass es in der Regel nicht möglich sein wird, eine exakte CP-Dekomposition zu berechnen. Gleiches gilt für die beste Rang- R Approximation, weshalb wir im Folgenden einfach von einer *Rang- R Approximation* sprechen.

Definition 8.3 (Rang- R Approximation): *Die CP-Dekomposition eines Tensors \mathbf{Y} mit einem festen Dekompositionsrang R , wird als Rang- R Approximation bezeichnet.*

Das Ziel der CP-Dekomposition des Verhaltenstensors ist im Weiteren somit nicht die Berechnung einer exakten Faktorisierung, sondern vielmehr die Generierung einer Dekomposition, die zum einen präzise genug ist um die ursprünglichen Eigenschaften genau genug abzubilden und zum anderen eine ausreichend hohe Reduktion des Speicherbedarfs ermöglicht. Die Berechnung der CP-Dekomposition und welche Werkzeuge dafür zur Verfügung stehen wird im nächsten Abschnitt beschrieben.

8.2.2 Berechnung der CP-Dekomposition

Die Berechnung der CP-Dekomposition erfolgt durch die Lösung eines Optimierungsproblems mit dem Ziel, für einen gegebenen Dekompositionsrang R , die bestmögliche Approximation, beziehungsweise die minimale Differenz zwischen dem ursprünglichen Tensor und dessen Zerlegung zu finden. Da der Verhaltenstensor mit den bedingten Wahrscheinlichkeiten nur nichtnegative Werte enthält, werden im Folgenden nur solche Optimierungsalgorithmen verwendet, welche die Elemente der Faktormatrizen mit der Beschränkung auf die Menge der nichtnegativen reellen Zahlen $\mathbb{R}_{\geq 0}$ berechnen. CP-Dekompositionen dieser Art werden als *nichtnegative CP-Dekomposition* (engl.: non-negative CP (NNCP) [56]) oder *nichtnegative Tensor Faktorisierung* (engl.: non-negative tensor factorization (NTF) [25]) bezeichnet. Das Optimierungsproblem lässt sich für einen Tensor $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{J_1 \times J_2 \times \dots \times J_N}$ durch

$$\min_{\boldsymbol{\lambda}, \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)}} \frac{1}{2} \left\| \mathbf{Y} - \llbracket \boldsymbol{\lambda}; \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)} \rrbracket \right\|_{\text{F}}^2 \quad (8.3)$$

beschreiben, wobei für die Elemente der Faktormatrizen und des Gewichtungsvektors

$$\lambda_r > 0, \quad x_{jnr}^{(n)} \geq 0 \quad \text{für } j_n = 1, \dots, J_n, \quad n = 1, \dots, N \quad \text{und } r = 1, \dots, R$$

gilt [56, S. 471], [24].

Die Notation (8.3) ist eine der üblichen Darstellungen des Optimierungsproblems der NNCP-Dekomposition und basiert auf der Minimierung der quadratischen Frobenius Norm $\|\cdot\|_{\text{F}}^2$, bzw. der quadratischen Summe der Tensorelemente [114]. Lösungsalgorithmen für diese Art von Optimierungsproblem basieren häufig auf der allgemeinen Methode der *alternierenden kleinsten Quadrate* (engl.: alternating least squares (ALS)) und werden entsprechend als ALS Algorithmen bezeichnet [24]. Da das

Optimierungsproblem (8.3) als Nebenbedingung die Nichtnegativität der Elemente der Faktormatrizen impliziert, spricht man im Speziellen auch von ANLS (*engl.*: alternating non-negative least squares [53]) oder NNLS (*engl.*: non-negative least squares [24]) Algorithmen.

Grundsätzlich existieren zahlreiche Algorithmen und unterschiedliche Kostenfunktionen zur Berechnung der NNCP-Dekomposition, wovon jedoch nur ein Bruchteil in den zur Verfügung stehenden Computer Toolboxen implementiert sind. Die aktuell (Stand: 2017) in MATLAB[®] und PYTHON[™] verfügbaren Algorithmen für die NNCP-Dekomposition gehören fast ausschließlich zur Gruppe der ALS Algorithmen, wobei insbesondere der sogenannte HALS (*engl.*: hierarchical alternating least squares [24]) Algorithmus erwähnenswert ist, da er speziell für die Dekomposition großer Tensoren entwickelt wurde. Weitere moderne Ansätze aus der Familie der ALS Algorithmen sind der ANLS-AS (*engl.*: alternating non-negative least squares with active set [49], [50]) und der ANLS-BPP (*engl.*: alternating non-negative least squares with block principal pivoting [52]) Algorithmus. Der sogenannte MU (*engl.*: multiplicative update [61], [130]) Algorithmus ist einer der ältesten Ansätze zur Berechnung der NNCP-Dekomposition und der einzige in MATLAB[®] und PYTHON[™] verfügbare Ansatz, der nicht zur ALS Gruppe gehört. Auf eine detaillierte Beschreibung der Funktionsweise oben aufgeführter Algorithmen soll an dieser Stelle verzichtet werden; weitere Informationen können den jeweils angegebenen Quellen entnommen werden.

Werkzeuge zur Berechnung der NNCP-Dekomposition

Tabelle 8.1 gibt einen Überblick bezüglich der verfügbaren MATLAB[®] und PYTHON[™] Toolboxen sowie der darin enthaltenen Algorithmen für die Kalkulation der NNCP-Dekomposition.

Tabelle 8.1: Toolboxen und Algorithmen für die NNCP-Dekomposition

Symbol	Name	Algorithmen
(A)	MATLAB [®] Tensor Toolbox [6]	MU
(B)	MATLAB [®] Tensorlab [126]	NLS ^a , MINF ^a
(C)	MATLAB [®] Nonnegative Matrix and Tensor Factorization Algorithms Toolbox [51]	ANLS-BPP, ANLS-AS, HALS, MU
(D)	PYTHON [™] Scikit-Tensor Library [97]	Unterstützt keine NNCP-Dekomposition, dient jedoch als Basisbibliothek für (E)
(E)	PYTHON [™] NCP.py ^b [103], [111]	ANLS-BPP, ANLS-AS, HALS, MU

^a Beide Algorithmen gehören zur ALS Gruppe, wobei NLS hier abweichend für *Non-linear Least Square* steht, während mit MINF ein nicht näher spezifizierter nichtlinearer unbeschränkter Optimierungsalgorithmus (*engl.*: *nonlinear unconstrained optimization algorithm*) bezeichnet wird [127, S. 24].

^b Stellt die PYTHON[™] Übersetzung von (C) dar.

Die Verwendung der in Tabelle 8.1 aufgeführten Toolboxen (A), (C), (D) und (E) ist prinzipiell sehr einfach, da die Dekomposition eines Tensors mit nur einer einzigen Funktion initiiert wird. Neben dem Festlegen des Dekompositions-rangs R bieten diese Toolboxen die Möglichkeit, weitere Funktionsargumente zu übergeben, wie beispielsweise die Anzahl der Iterationsschritte, das Initialisierungsverfahren oder verschiedene Konvergenzkriterien. Die mit Abstand umfangreichste Toolbox ist das MATLAB[®] Tensorlab (B), dessen Funktionsweise sich zumindest dann von der der anderen Toolboxen unterscheidet, wenn Nebenbedingungen für die Berechnung der Dekomposition definiert werden sollen. Für diesen Fall wird ein sogenanntes SDF (*engl.*: structured data fusion [127]) Framework bereitgestellt, welches eine eigne Syntax besitzt. Mithilfe dieses Frameworks können verschiedene Restriktionen definiert werden, wie zum Beispiel die Forderung der Nichtnegativität der Elemente

der Faktormatrizen. Erwähnenswert ist auch, dass alle Toolboxen eine Vielzahl von mathematischen Produkten der linearen und multilinearen Algebra bereitstellen, wodurch sich viele Berechnungen einfach und sehr kompakt implementieren lassen.

Zu allen Tensortoolboxen ist anzumerken, dass diese – bis auf (C) und (E) – deutlich mehr Algorithmen für die normale CP-Dekomposition bereitstellen, als für die NNCP-Dekomposition. Beispiele zur Verwendung der Toolboxen und deren Funktionen können für (C), (D) und (E) innerhalb der Programmumgebung über die entsprechenden Funktionsaufrufe gefunden werden oder auch im Quellcode, der direkt unter den in Tabelle 8.1 angegebenen Quellen einsehbar ist. Gleiches gilt auch für (A), wobei hier zusätzlich die Publikationen [4], [5] hilfreich sind. Für (B) wird ein komplettes Handbuch [127] zum Download angeboten, das eine Vielzahl an Beispielen und Anwendungsmöglichkeiten enthält.

Konvergenz des Optimierungsproblems

Zur Berechnung der CP-Dekomposition soll abschließend kurz auf die Thematik der Konvergenz des Optimierungsproblems eingegangen werden. Wie bereits erwähnt wurde, ist die Existenz der besten Rang- R Approximation nicht sichergestellt (Degeneration der CP-Dekomposition), was in direktem Zusammenhang damit steht, dass generell keine globale Lösung für eine Approximation mit $R < \text{rank}(\mathbf{Y})$ existiert [75, S. 436]. Hier besteht ein Unterschied zwischen der CP und NNCP-Dekomposition, denn die NNCP ist nicht degenerativ und für einen gegebenen Dekompositionsrang R existiert stets ein globales Minimum, womit auch eine beste Rang- R Approximation existiert [75, S. 433], [105]. Die theoretische Existenz einer globalen besten Lösung hilft in der Praxis jedoch nicht zwangsläufig weiter, da die Qualität des Ergebnisses wesentlich von den Werten der Elemente des ursprünglichen Tensors und des verwendeten Lösungsalgorithmus des Optimierungsproblems abhängt. Wie die praktische Anwendung der NNCP-Dekomposition zeigt, können die zur Verfügung stehenden ALS und MU Algorithmen in Tabelle 8.1 das Konvergieren in einem globalen Minimum, insbesondere für große Tensoren, in der Regel nicht sicherstellen [88]. Grund hierfür ist vor allem die Nicht-Konvexität des Optimierungsproblems, weshalb die Startwerte der Faktormatrizen, mit denen die Iteration initialisiert wird, einen entscheidenden Einfluss auf das Ergebnis haben, da jede Dekomposition, die mit verschiedenen Startwerten begonnen wird, zu einem anderen Ergebnis und in der Regel nur *lokalen* Minimum führt [88, S. 2].

Die Toolboxen in Tabelle 8.1 initialisieren die Startwerte der Faktormatrizen zufällig, wodurch im Allgemeinen jede Dekomposition auch zu unterschiedlichen Ergebnissen führt. Es sind also in der Regel stets mehrere Dekompositionen nötig, um eine gute Approximation für einen vorgegebenen Dekompositionsrang zu finden. Allerdings ist es auch möglich, die Startwerte manuell vorzugeben, was jedoch Kenntnis über die innere Struktur des zu faktorisierenden Tensors erfordert.

8.2.3 Speicherbedarf der CP-Dekomposition

Wie aus Definition 8.1 ersichtlich ist, führt die Faktorisierung eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ mit Dekompositionsrang R zu einer Anzahl von N Faktormatrizen der Größe $J_n \times R$. Damit erhält man für alle $n = 1, 2, \dots, N$ eine Anzahl von $J_n \cdot R$ Werten sowie den Vektor $\boldsymbol{\lambda}$, der die Länge R besitzt. Der Speicherbedarf² der CP-Dekomposition sowie ein Reduktionsfaktor, der die Kompressionsrate wiedergibt, kann damit wie folgt definiert werden.

²Der Begriff des Speicherbedarfs bezieht sich auf die Anzahl der zu speichernden Werte.

Definition 8.4 (Speicherbedarf der CP-Dekomposition): Der Speicherbedarf der CP-Dekomposition eines Tensors N -ter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ mit dem Dekompositionsrang R beträgt

$$\gamma_{\text{CP}} = R \cdot \left(\sum_{n=1}^N J_n \right) + R \quad (8.4)$$

Werte, woraus sich ein Reduktionsfaktor

$$\xi_{\text{CP}} = \frac{\prod_{n=1}^N J_n}{R \cdot \left(\sum_{n=1}^N J_n \right) + R} \quad (8.5)$$

ergibt.

Das Verhältnis zwischen Speicherreduktion und Approximationsgenauigkeit bei der CP-Dekomposition lässt sich anhand einer Bildkompression gut veranschaulichen.

Beispiel 8.1 (Bildkompression durch NNCP-Dekomposition): Die Daten des Fotos des Fraunhofer-ISE Hauptgebäudes in Abbildung 8.2(a) können in einem Tensor dritter Ordnung $\mathbf{Y} \in \mathbb{N}^{480 \times 725 \times 3}$ mit Elementen $y(i, j, k)$ gespeichert werden. Dabei stehen die ersten beiden Modi des Tensors für die Auflösung und der letzte Modus für den RGB Farbraum. Die frontalen Scheiben $\mathbf{Y}(:, :, k) \in \mathbb{N}^{480 \times 725}$ enthalten also für $k = 1, 2, 3$ die Farbwerte für rot, grün und blau als 8-Bit Integer mit Werten in $\{0, 1, \dots, 255\}$.



(a) Original, 1 044 000 Werte



(b) $R = 200$, $\gamma_{\text{CP}} = 241\,800$, $\xi_{\text{CP}} = 4.32$



(c) $R = 100$, $\gamma_{\text{CP}} = 120\,900$, $\xi_{\text{CP}} = 8.64$



(d) $R = 25$, $\gamma_{\text{CP}} = 30\,225$, $\xi_{\text{CP}} = 34.54$

Abbildung 8.2: Beispiele für die Bildkompression anhand verschiedener Rang- R Approximationen des Tensors $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{480 \times 725 \times 3}$, (Quelle Original: ©Fraunhofer ISE/Foto: Guido Kirsch)

Für den Dekompositionsrang $R = 25$ ergeben sich damit aus der NNCP-Dekomposition des Originalbildes die Faktormatrizen $\mathbf{X}^{(1)} \in \mathbb{R}_{\geq 0}^{480 \times 25}$, $\mathbf{X}^{(2)} \in \mathbb{R}_{\geq 0}^{725 \times 25}$ und $\mathbf{X}^{(3)} \in \mathbb{R}_{\geq 0}^{3 \times 25}$ sowie der Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{25}$ (vgl. Definition 8.1). Während für das Original in Abbildung 8.2(a) der Speicherbedarf $480 \cdot 725 \cdot 3 = 1\,044\,000$ Werte beträgt, benötigt die CP-Repräsentation $\llbracket \boldsymbol{\lambda}; \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)} \rrbracket$ bei einem Dekompositionsrang von $R = 25$ nur $25 \cdot (480 + 725 + 3) + 25 = 30\,225$ Werte, woraus sich ein Reduktionsfaktor $\xi_{\text{CP}} = 34.54$ ergibt (vgl. Gleichungen (8.4), (8.5)). Das Bild in Abbildung 8.2(d) wurde dabei anhand des wieder komponierten Tensors $\hat{\mathbf{Y}} \in \mathbb{R}_{\geq 0}^{480 \times 725 \times 3}$ erzeugt, der damit nur eine Approximation des ursprünglichen Tensor \mathbf{Y} darstellt. Die Abbildungen 8.2(b) und 8.2(c) zeigen die Resultate für die gleiche Vorgehensweise, jedoch mit Dekompositions-rängen von $R = 100$ und $R = 200$. Dabei wird offensichtlich, dass umso höher der Rang R gewählt wird, desto besser ist die Approximationsgüte und umso größer der Speicherbedarf. Die NNCP-Dekomposition des Tensors \mathbf{Y} erfolgte dabei mithilfe des HALS Algorithmus der MATLAB[®] Toolbox (C) aus Tabelle 8.1.

Die in den Abbildungen 8.2(b)-8.2(d) angegebene Anzahl der zu speichernden Werte und der Reduktionsfaktor beziehen sich auf den dekomponierten Tensor, also auf die Elemente aller Faktormatrizen zuzüglich des Gewichtungsvektors. Das Beispiel dient in erster Linie der Veranschaulichung der Approximationsgenauigkeit in Abhängigkeit des Dekompositions-rangs. Um die Abbildungen darstellen zu können, müssen die jeweiligen Faktorisierungen jedoch wieder komponiert werden, womit der praktische Nutzen gering ist, da der komponierte Tensor $\hat{\mathbf{Y}}$ die gleiche Größe besitzt, wie der ursprüngliche Tensor \mathbf{Y} (vgl. Definition 8.1). Um die Speicherreduktion durch die CP-Dekomposition nutzen zu können, müssen also entsprechende Algorithmen definiert werden, die auf Basis der CP-Repräsentation, also des Kruskal-Tensors arbeiten, ohne dass dieser wieder komponiert werden muss. Für den Verhaltenstensor bedeutet dies die Anpassung der Beobachteralgorithmen an die Struktur des aus der CP-Dekomposition resultierenden Kruskal-Tensors. Die CP-Dekomposition des Verhaltenstensors wird im folgenden Abschnitt beschrieben. Wie die Beobachteralgorithmen für den stochastischen Automaten und die qualitativen Modelle an die dekomponierte Struktur des Verhaltenstensors angepasst werden können, ist Gegenstand des Kapitels 9.

8.2.4 CP-Repräsentation des Verhaltenstensors

In diesem Abschnitt wird die CP-Dekomposition des Verhaltenstensors \mathbf{L} aus Definition 7.1 erläutert. Für die Berechnung der Faktorisierung kommen ausschließlich Algorithmen für die NNCP-Dekomposition aus Tabelle 8.1 zum Einsatz, da der Verhaltenstensor mit den bedingten Wahrscheinlichkeiten nur nichtnegative Werte enthalten kann. Für die praktische Anwendung ist insbesondere die Rang- R Approximation des Verhaltenstensors aus Definition 7.1 von Bedeutung, während für die formale Notation der im nächsten Kapitel eingeführten Beobachteralgorithmen, eine exakte Definition erforderlich ist. Im Folgenden werden beide Definition eingeführt und erläutert.

Definition 8.5 (Rang- R Approximation des Verhaltenstensors): Die Approximation des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ durch einen Tensor

$$\hat{\mathbf{L}} \approx \mathbf{L} \quad (8.6)$$

mit der CP-Repräsentation

$$\hat{\mathbf{L}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \rrbracket \equiv \sum_{r=1}^R \lambda_r \mathbf{a}_{:r} \circ \mathbf{b}_{:r} \circ \mathbf{c}_{:r} \circ \mathbf{d}_{:r} \in \mathbb{R}_{\geq 0}^{N \times Q \times N \times M} \quad (8.7)$$

und Faktormatrizen

$$\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}, \quad \mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}, \quad \mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}, \quad \mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}$$

sowie dem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$, wird als Rang- R Approximation des Verhaltenstensors bezeichnet.

In Gleichung (8.7) steht $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}$ somit für die Faktormatrix der Folgezustände des Verhaltenstensors und $\mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}$ sowie $\mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}$ repräsentieren die Faktormatrizen der Ausgaben, Zustände und Eingaben. Die Vektoren $\mathbf{a}_{:,r} \in \mathbb{R}_{\geq 0}^N$, $\mathbf{b}_{:,r} \in \mathbb{R}_{\geq 0}^Q$, $\mathbf{c}_{:,r} \in \mathbb{R}_{\geq 0}^N$ und $\mathbf{d}_{:,r} \in \mathbb{R}_{\geq 0}^M$ stellen den jeweils r -ten Spaltenvektor der entsprechenden Faktormatrix dar. Der Dekompositionsrang R kann dabei als ein justierbarer Parameter zur Einstellung des Verhältnisses zwischen Speicherbedarf und Approximationsgüte interpretiert werden.

Dass die Rang- R Approximation des Verhaltenstensor $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ nur eine Näherung darstellt, hat direkte Konsequenzen in Bezug auf die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{z}, w | z, v)$, die durch die Elemente $l(\bar{z}, w, z, v) \in [0, 1]$ des Verhaltenstensors repräsentiert werden. Während die Werte der Elemente $l(\bar{z}, w, z, v)$ des ursprünglichen Verhaltenstensors \mathbf{L} innerhalb des Einheitsintervalls $[0, 1]$ liegen, gilt für den approximierten Verhaltenstensor in Definition 8.5 allgemein nur

$$\hat{\mathbf{L}} \in \mathbb{R}_{\geq 0}^{N \times Q \times N \times M} . \quad (8.8)$$

Das bedeutet, dessen Elemente $\hat{l}(\bar{z}, w, z, v)$ können somit auch Werte größer Eins annehmen. Es handelt sich dabei also nicht mehr um bedingte Wahrscheinlichkeiten im definitionsgemäßen Sinn, sondern nur noch um entsprechende Näherungen, für die lediglich $\hat{l}(\bar{z}, w, z, v) \approx l(\bar{z}, w, z, v)$ gilt. Ein Element des approximierten Verhaltenstensor kann mit Gleichung (8.9) bestimmt werden:

$$\hat{l}(\bar{z}, w, z, v) = \sum_{r=1}^R \lambda_r \cdot a_{\bar{z}r} \cdot b_{wr} \cdot c_{zr} \cdot d_{vr} . \quad (8.9)$$

Da die Eigenschaft, dass die Rang- R Approximation nur eine Näherung der bedingten Wahrscheinlichkeiten des Verhaltenstensors generieren kann, für formale mathematische Beschreibungen ungünstig ist, muss auch der Fall einer exakten Dekomposition des Verhaltenstensors eingeführt werden, der wie folgt definiert ist.

Definition 8.6 (Exakte CP-Repräsentation des Verhaltenstensors): *Wenn Gleichung (8.6) mit Gleichheit gilt, sprich die Differenz in Gleichung (8.3) ist Null, dann ist*

$$\hat{\mathbf{L}} = \mathbf{L} \quad (8.10)$$

und die CP-Dekomposition des Verhaltenstensors ist exakt.

Es sei an dieser Stelle angemerkt, dass eine exakte CP-Dekomposition des Verhaltenstensors nach Definition 8.6 nicht bedeutet, dass es sich um eine Tensorrang-Dekomposition handelt (vgl. Abschnitt 8.2.1). Das heißt, im Allgemeinen wird der Rang R einer exakten Dekomposition, nicht dem Rang des Verhaltenstensors \mathbf{L} entsprechen ($R > \text{rank}(\mathbf{L})$).

Wie die CP-Repräsentation (8.7) des Verhaltenstensor zur Beobachtung stochastischer Automaten und zur qualitativen genutzt werden kann, wird in Kapitel 9 beschrieben.

8.3 TT-Dekomposition*

Dieser Abschnitt beschreibt die *Tensor-Train (TT) Dekomposition* und ihre Anwendung in Bezug auf den Verhaltenstensor. Die Entwicklung dieser Tensor-Dekompositionsmethode geht im Wesentlichen zurück auf Oseledets [99] und Oseledets und Tyrtshnikov [102]. Gute Überblicke und Erläuterungen der algebraischen Grundlagen bieten aber auch Cichocki et al. [22], Holtz et al. [43], N. Lee und Cichocki [62], [63], [64] und N. Lee et al. [65].

8.3.1 Grundlagen der TT-Dekomposition*

Die TT-Dekomposition unterscheidet sich in mehreren Aspekten von der CP-Dekomposition. Während bei der CP-Dekomposition ein Tensor $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ in N zweidimensionale Faktormatrizen zerlegt wird, erfolgt die Faktorisierung bei der TT-Dekomposition in eine Anzahl von N dreidimensionalen *TT-Kernen* (engl.: TT-Cores [99]). Im Gegensatz zur CP-Dekomposition, bei der die Approximationsgüte zum einen vom gewählten Dekompositionsrang R und zum anderen von den Startwerten der Faktormatrizen abhängt, erlaubt die TT-Dekomposition die Vorgabe einer festen Approximationsgenauigkeit, die mit ε bezeichnet wird [99, S. 2297].

Da in dieser Arbeit die TT-Dekomposition genau wie die CP-Dekomposition, als eine Approximation verwendet wird, werden auch die entsprechenden Gleichungen in Form von Näherungen angegeben. Für die folgende Definition der TT-Dekomposition wird das kontrahierte Modus- $(P|1)$ Produkt aus Definition 6.13 benötigt.

Definition 8.7 (TT-Dekomposition [22, S. 368], [64, S. 931]): *Die TT-Dekomposition ist definiert als Approximation eines Tensors N -ter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ durch eine Folge kontrahierter Modus- $(P|1)$ Produkte*

$$\mathbf{Y} \approx \langle\langle \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)} \rangle\rangle \equiv \mathbf{X}^{(1)} \odot \mathbf{X}^{(2)} \odot \dots \odot \mathbf{X}^{(N)}, \quad (8.11)$$

die für $n = 1, \dots, N$ durch die TT-Kerne $\mathbf{X}^{(n)} \in \mathbb{R}^{R_{n-1} \times J_n \times R_n}$ gegeben sind.

Die Anzahl der Elemente entlang des Modus-2 der TT-Kerne $\mathbf{X}^{(n)} \in \mathbb{R}^{R_{n-1} \times J_n \times R_n}$ entspricht dabei für $n = 1, \dots, N$ stets der Anzahl der Elemente des korrespondierenden Modus- n des Ursprungstensors. Die Anzahl der Elemente entlang des Modus-1 und Modus-3 der TT-Kerne wird durch die sogenannten TT-Ränge (engl.: TT-ranks) R_1, \dots, R_N bestimmt, wobei $R_0 = R_N = 1$ gilt [99, S. 2296–2297]. Da demzufolge der erste TT-Kern $\mathbf{X}^{(1)} \in \mathbb{R}^{1 \times J_1 \times R_1}$ und der letzte TT-Kern $\mathbf{X}^{(N)} \in \mathbb{R}^{R_{N-1} \times J_N \times 1}$ jeweils eine Dimension der Länge Eins besitzen, können diese ebenso als Tensoren zweiter Ordnung, beziehungsweise als Matrizen $\mathbf{X}^{(1)} \in \mathbb{R}^{J_1 \times R_1}$ und $\mathbf{X}^{(N)} \in \mathbb{R}^{R_{N-1} \times J_N}$ interpretiert werden [63, S. 603]. Im Folgenden wird, solange nicht explizit erwähnt, auf diese Möglichkeit verzichtet und alle TT-Kerne werden wie in Definition 8.7, als Tensoren dritter Ordnung notiert, da sich daraus eine einheitliche Darstellung sowie Vorteile in der späteren Notation des Beobachteralgorithmus ergeben. Die in Gleichung (8.11) auftretenden Klammern $\langle\langle \cdot \rangle\rangle$ können dabei, vergleichbar mit der Notation des Kruskal-Tensors $\llbracket \cdot \rrbracket$ bei der CP-Dekomposition, als eine Kurznotation für eine Folge kontrahierter Modus- $(P|1)$ Produkte interpretiert werden.³ Für weitere Notationen der TT-Dekomposition siehe [22, S. 363–368].

Die Abbildung 8.3 zeigt schematisch die TT-Dekomposition anhand eines Tensors fünfter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3 \times J_4 \times J_5}$. Die Bestimmung des ersten kontrahierten Modus-1 Produkts entspricht dabei der in Beispiel 6.4 veranschaulichten Berechnung.

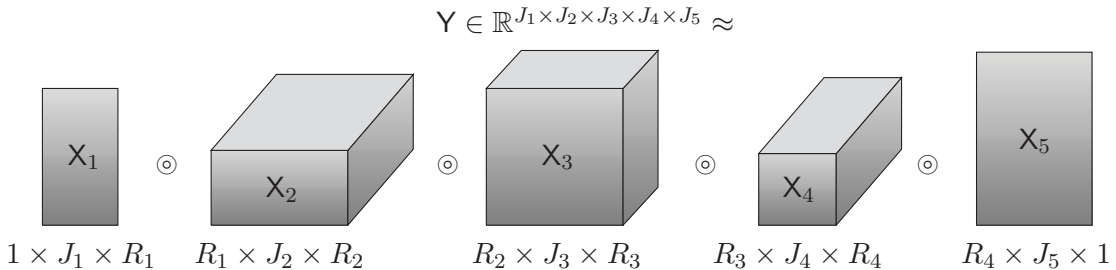


Abbildung 8.3: TT-Dekomposition eines Tensors fünfter Ordnung, nach [23, S. 155]

³Die Klammern $\langle\langle \cdot \rangle\rangle$ dürfen nicht mit den einfachen Klammern $\langle \cdot \rangle$ des kontrahierten Tensorprodukts in Definition 6.14 und 6.15 verwechselt werden.

Die Berechnung eines Elements des Tensors \mathbf{Y} in Definition 8.7 erfolgt durch eine Folge von Matrix Produkten

$$y(j_1, j_2, \dots, j_{N-1}, j_N) \approx \mathbf{X}_{:j_1:}^{(1)} \cdot \mathbf{X}_{:j_2:}^{(2)} \cdots \mathbf{X}_{:j_{N-1}:}^{(N-1)} \cdot \mathbf{X}_{:j_N:}^{(N)}, \quad (8.12)$$

wobei $\mathbf{X}_{:j_n:}^{(n)} \in \mathbb{R}^{R_{n-1} \times R_n}$ für $n = 1, \dots, N$ gilt [64, S. 931]. Die Terme $\mathbf{X}_{:j_2:}^{(2)}, \dots, \mathbf{X}_{:j_{N-1}:}^{(N-1)}$ stellen somit laterale Scheiben der TT-Kerne $\mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N-1)}$ dar (s. a. Abbildung 6.3). Aufgrund der Beziehung $R_0 = R_N = 1$ kann der erste Faktor $\mathbf{X}_{:j_1:}^{(1)} \in \mathbb{R}^{1 \times R_1}$ als Zeilenvektor $\mathbf{x}_{j_1:}^{(1)} \in \mathbb{R}^{1 \times R_1}$ und der letzte Faktor $\mathbf{X}_{:j_N:}^{(N)} \in \mathbb{R}^{R_{N-1} \times 1}$ als Spaltenvektor $\mathbf{x}_{:j_N}^{(N)} \in \mathbb{R}^{R_{N-1} \times 1}$ aufgefasst werden. Alternativ zu Gleichung (8.12) lässt sich die elementweise Kalkulation auch als Summe elementweiser Produkte darstellen [64, S. 933]:

$$y(j_1, j_2, \dots, j_{N-1}, j_N) \approx \sum_{r_1=1}^{R_1} \cdots \sum_{r_{N-1}=1}^{R_{N-1}} x_{1j_1r_1}^{(1)} \cdot x_{r_1j_2r_2}^{(2)} \cdots x_{r_{N-2}j_{N-1}r_{N-1}}^{(N-1)} \cdot x_{r_{N-1}j_N1}^{(N)}. \quad (8.13)$$

Ähnlich wie bei der CP-Dekomposition, bei der die Tensorrang Dekomposition (s. Definition 8.2) eine exakte Faktorisierung mit minimaler Anzahl an Rang-1 Termen R beschreibt, existiert bei der TT-Dekomposition der Begriff der *minimalen* TT-Dekomposition. Ob eine TT-Dekomposition minimal ist, hängt dabei vom Rang der *matrizisierten* TT-Kerne ab.

Definition 8.8 (Minimale TT-Dekomposition [64, S. 932]): *Die TT-Dekomposition (8.7) eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ ist minimal, wenn für die Modus-1 und Modus-3 Matrizisierung der TT-Kerne und deren TT-Ränge*

$$R_n = \text{rang}\left(\mathbf{X}_{(3)}^{(n)}\right) \quad \text{und} \quad R_{n-1} = \text{rang}\left(\mathbf{X}_{(1)}^{(n)}\right) \quad (8.14)$$

für alle $n = 1, 2, \dots, N$ gilt.

Der Begriff der *minimalen* TT-Dekomposition bedeutet dabei, dass die Dekomposition exakt ist [10]. Die *minimale* TT-Dekomposition ist außerdem *eindeutig* [43, S. 709], das heißt es existiert nur eine einzige Kombination an TT-Rängen, die den ursprünglichen Tensor exakt wiedergibt. Das bedeutet, die TT-Ränge R_n der minimalen TT-Dekomposition entsprechen für alle $n = 1, \dots, N-1$ dem Rang der n -ten kanonischen Matrizisierung des Tensors \mathbf{Y} , sodass $\text{rang}(\mathbf{Y}_{<n>}) = R_n$ ist [64, S. 932]. Im Gegensatz zur CP-Dekomposition existiert außerdem immer eine minimale TT-Dekomposition als auch eine *beste* TT Approximation [102, S. 72], [99, S. 2300]. Zum Begriff der Matrizisierung siehe Abschnitt 6.3.2. Im folgenden Abschnitt wird die Berechnung der TT-Dekomposition für den in dieser Arbeit wichtigen Fall der näherungsweisen Faktorisierung eines Tensors erläutert.

8.3.2 Berechnung der TT-Dekomposition*

Das Ziel der TT-Dekomposition ist die Approximation eines Tensors $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ durch TT-Kerne $\mathbf{X}^{(n)} \in \mathbb{R}^{R_{n-1} \times J_n \times R_n}$, $n = 1, \dots, N$ und kleinstmöglichen TT-Rängen R_1, \dots, R_{N-1} sowie vordefinierter Approximationsgüte ε . Diese Forderung lässt sich formal wie folgt definieren [99, S. 2301]:

$$\left\| \mathbf{Y} - \left\langle \left\langle \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)} \right\rangle \right\rangle \right\|_{\text{F}} \leq \varepsilon \|\mathbf{Y}\|_{\text{F}}. \quad (8.15)$$

Der Lösungsalgorithmus, der die TT-Dekomposition als eine Folge von Singulärwertzerlegungen (SVD) berechnet [102, S. 73], wird als TT-SVD [99, S. 2301] bezeichnet und ermöglicht auch die Berechnung einer minimalen TT-Dekomposition [64, S. 932]. Für die TT-Dekomposition des Verhaltenstensors ist diese Tatsache jedoch zweitrangig, da auch hier wie bei der CP-Dekomposition, ein Kompromiss zwischen Approximationsgenauigkeit und Speicherbedarf angestrebt wird und daher keine minimale TT-Dekomposition, sondern eine *TT-Approximation* berechnet werden soll. Eine

interessante Eigenschaft des TT-SVD Algorithmus ist, dass dieser auch die Berechnung der *besten* TT-Approximation sicherstellt [99, S. 2300]. Dabei wird der zu dekomponierende Tensor $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ für alle $n = 1, \dots, N-1$ kanonisch matrizisiert und die generierten Matrizen $\mathbf{Y}_{<n>} \in \mathbb{R}^{J_1 \dots J_n \times J_{n+1} \dots J_N}$ durch Singulärwertzerlegungen

$$\mathbf{Y}_{<n>} \approx \mathbf{U}^{(n)} \cdot \mathbf{S}^{(n)} \cdot \mathbf{V}^{(n)} = \hat{\mathbf{Y}}^{(n)} \in \mathbb{R}^{J_1 \dots J_n \times J_{n+1} \dots J_N}$$

approximiert [102, S. 73–77]. Die Approximation erfolgt dabei als die *beste* Singulärwertzerlegung für die vorgegebene Genauigkeit ε . Diese Näherung wird auch als *beste* Rang- R_n Approximation bezeichnet [99, S. 2299], [102, S. 73], da die Ränge der Matrizen $\hat{\mathbf{Y}}^{(1)}, \dots, \hat{\mathbf{Y}}^{(N-1)}$ mit $\text{rang}(\hat{\mathbf{Y}}^{(n)}) = R_n$, die TT-Ränge R_1, \dots, R_{N-1} der zu generierenden TT-Kerne $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N-1)}$ für $n = 1, 2, \dots, N$ vorgeben. Die TT Approximation ist somit *quasi optimal* [99, S. 2300], denn sie stellt die beste Approximation für eine gegebene Approximationsgenauigkeit ε dar. Die TT-Kerne $\mathbf{X}^{(n)}$ werden dann für $n = 1, \dots, N-1$ durch Umordnen der Elemente der Matrizen $\mathbf{U}^{(n)}$ in ihre dreidimensionale Struktur $R_{n-1} \times J_n \times R_n$ überführt, wobei der letzte TT-Kern $\mathbf{X}^{(N)}$ sich aus dem Matrixprodukt $\mathbf{S}^{(N-1)} \cdot \mathbf{V}^{(N-1)}$ ergibt [99, S. 2301].

Werkzeuge zur Berechnung der TT-Dekomposition*

Zur Berechnung der TT-Dekomposition steht momentan (Stand 2017) die *Tensor-Train Toolbox* [100] für MATLAB[®] sowie die *TTpy Toolbox* [101] für PYTHONTM zur Verfügung. Leider existieren bisher jedoch keine Implementierungen, die die Berechnung der TT-Kerne mit nichtnegativen Elementen erlauben. Allerdings sind bereits Algorithmen für die nichtnegative TT-Dekomposition theoretisch beschrieben worden [65], was Hoffnung gibt, dass diese in naher Zukunft auch als entsprechende Werkzeuge verfügbar sein werden. Auch wenn die Eigenschaft, dass die TT-Kerne negative Elemente enthalten können prinzipiell ungünstig für die Dekomposition des Verhaltenstensors ist, kann die TT-Dekomposition aufgrund der einstellbaren Approximationsgüte, als eine Alternative zur CP-Dekomposition dienen. Wie das Problem der möglicherweise negativen Elemente in den TT-Kernen mit einem heuristischen Ansatz gelöst werden kann, wird in Kapitel 9 beschrieben.

8.3.3 Speicherbedarf der TT-Dekomposition*

Zum Speicherbedarf der TT-Dekomposition sei angemerkt, dass dieser nicht wie bei der CP-Dekomposition durch die Vorgabe eines festen Dekompositions-rangs R bereits vor der Faktorisierung feststeht. Dadurch, dass bei der TT-Dekomposition lediglich die Approximationsgüte ε vorgegeben wird, hängt der Speicherbedarf von der Struktur des zugrundeliegenden Tensors ab. Einfach ausgedrückt: Für dimensionsgleiche Tensoren mit unterschiedlich komplexen Datenstrukturen wird für die gleiche vorgegebene Approximationsgüte in der Regel unterschiedlich viel Speicher benötigt.

Definition 8.9 (Speicherbedarf der TT-Dekomposition): *Der Speicherbedarf eines Tensors N -ter Ordnung $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$, der durch die TT-Repräsentation $\langle\langle \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)} \rangle\rangle$ mit TT-Kernen $\mathbf{X}^{(n)} \in \mathbb{R}^{R_{n-1} \times J_n \times R_n}$ für $n = 1, \dots, N$ gegeben ist, beträgt*

$$\gamma_{\text{TT}} = \sum_{n=1}^N R_{n-1} \cdot J_n \cdot R_n \quad (8.16)$$

Werte, woraus sich ein Reduktionsfaktor von

$$\xi_{\text{TT}} = \frac{\prod_{n=1}^N J_n}{\sum_{n=1}^N R_{n-1} \cdot J_n \cdot R_n} \quad (8.17)$$

ergibt.

8.3.4 TT-Repräsentation des Verhaltenstensors*

In diesem Abschnitt wird gezeigt, wie die TT-Dekomposition zur Faktorisierung des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ aus Definition 7.1 genutzt werden kann. Aufgrund der Tatsache, dass die TT-Kerne über ihre rechten und linken Ränge miteinander verknüpft sind (vgl. Abbildung 8.3), spielt im Hinblick auf die Entwicklung von Algorithmen, die auf Basis der TT-Repräsentation (8.11) arbeiten, die Reihenfolge der Modi des zu dekomponierenden Tensors eine Rolle. Im Folgenden werden die Modi des Verhaltenstensors für die TT-Repräsentation permutiert, was wie in der Literatur üblich, durch die Notation

$$\underline{\mathbf{L}} = \text{permute}(\mathbf{L}, [Q, M, N, N]) \quad (8.18)$$

erfolgt.⁴ Die Permutation der Modi des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ mit den Elementen $l(\bar{z}, w, z, v)$ führt dementsprechend zu einer alternativen Repräsentation $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ mit Elementen

$$\underline{l}(w, v, \bar{z}, z) = l(\bar{z}, w, z, v) = \mathcal{L}(\bar{z}, w | z, v) \quad \forall \bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V}. \quad (8.19)$$

Der Unterstrich kennzeichnet somit im Folgenden den permutierten Verhaltenstensor, beziehungsweise dessen Elemente. Anstatt die Elemente eines bereits vorhandenen Verhaltenstensors umzuordnen, kann auch direkt ein Tensor des Formats $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ angelegt werden, dem dann mit

$$\underline{l}(w, v, \bar{z}, z) = \mathcal{L}(\bar{z}, w | z, v) \quad \forall \bar{z}, z \in \mathcal{Z}, w \in \mathcal{W}, v \in \mathcal{V}, \quad (8.20)$$

die bedingten Wahrscheinlichkeiten der Verhaltensrelation direkt zugewiesen werden (s. a. Gleichung (7.2)). Die Permutation der Tensormodi führt somit zur einer alternativen Darstellung des Verhaltenstensors, die jedoch exakt den gleichen Informationsgehalt besitzt. Damit kann die *TT-Approximation des Verhaltenstensors*, wie folgt definiert werden.

Definition 8.10 (TT-Approximation des Verhaltenstensors): *Die Approximation des Verhaltenstensors $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ durch einen Tensor*

$$\hat{\underline{\mathbf{L}}} \approx \underline{\mathbf{L}} \quad (8.21)$$

mit der TT-Repräsentation

$$\hat{\underline{\mathbf{L}}} = \langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle \equiv \mathbf{E} \odot \mathbf{F} \odot \mathbf{G} \odot \mathbf{H} \quad (8.22)$$

und TT-Kernen

$$\mathbf{E} \in \mathbb{R}^{1 \times Q \times R_1}, \quad \mathbf{F} \in \mathbb{R}^{R_1 \times M \times R_2}, \quad \mathbf{G} \in \mathbb{R}^{R_2 \times N \times R_3}, \quad \mathbf{H} \in \mathbb{R}^{R_3 \times N \times 1},$$

wird als *TT-Approximation des Verhaltenstensors* bezeichnet.

In Definition 8.10 wird der TT-Kern $\mathbf{E} \in \mathbb{R}^{1 \times Q \times R_1}$ als TT-Kern der Ausgaben bezeichnet und die TT-Kerne $\mathbf{F} \in \mathbb{R}^{R_1 \times M \times R_2}$, $\mathbf{G} \in \mathbb{R}^{R_2 \times N \times R_3}$ und $\mathbf{H} \in \mathbb{R}^{R_3 \times N \times 1}$ als TT-Kerne der Eingaben, Folgezustände und Zustände.

Aufgrund der Tatsache, dass die TT-Dekomposition nach Definition 8.10 eine Approximation darstellt und die Lösung des Optimierungsproblems (8.15) ohne die Nebenbedingung der Nichtnegativität der TT-Kerne erfolgt, gilt für den approximierten Verhaltenstensor

$$\hat{\underline{\mathbf{L}}} \in \mathbb{R}^{Q \times M \times N \times N}. \quad (8.23)$$

⁴Die Permutation des Verhaltenstensors ist aus mathematischer Sicht nicht notwendig. Sie bietet allerdings Vorteile bei der Implementierung der Gleichungen in der Programmiersprache PYTHONTM.

Das bedeutet, dass die Elemente $\widehat{\underline{l}}(w, v, \bar{z}, z)$ sowohl negative als auch Werte größer Eins annehmen können. Auch hier handelt es sich somit nicht mehr um bedingte Wahrscheinlichkeiten, sondern um Näherungen, für die lediglich $\widehat{\underline{l}}(w, v, \bar{z}, z) \approx \underline{l}(w, v, \bar{z}, z)$ gilt. Ein Element des approximierten Verhaltenstensors $\widehat{\underline{l}}$ kann dann mit Gleichung (8.24) berechnet werden:

$$\widehat{\underline{l}}(w, v, \bar{z}, z) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e(1, w, r_1) \cdot f(r_1, v, r_2) \cdot g(r_2, \bar{z}, r_3) \cdot h(r_3, z, 1) . \quad (8.24)$$

Auch für die TT-Dekomposition muss für die weiteren Betrachtungen der exakte Fall eingeführt werden.

Definition 8.11 (Exakte TT-Dekomposition des Verhaltenstensors): *Wenn die Gleichung (8.21) mit Gleichheit gilt, sprich die Differenz in Gleichung (8.15) ist Null, dann ist*

$$\widehat{\underline{l}} = \underline{l} \quad (8.25)$$

und die TT-Dekomposition des Verhaltenstensors ist exakt.

9

Beobachteralgorithmen für die CP- und TT-Repräsentation

Dieses Kapitel stellt die Beobachteralgorithmen zur qualitativen Fehlererkennung, basierend auf der CP- und TT-Repräsentation des Verhaltenstensors vor. Einleitend wird in Abschnitt 9.1 eine Beobachtergleichung auf Grundlage der Komposition von Submatrizen des Verhaltenstensors stochastischer Automaten aufgezeigt. In Erweiterung hierzu erfolgt die Einführung einer effizienteren Beobachtergleichung, mit der die Zustandswahrscheinlichkeitsvektoren des stochastischen Automaten unmittelbar auf Grundlage der CP-Repräsentation des Verhaltenstensor bestimmt werden können. Abschließend wird der Beobachteralgorithmus für die qualitative Fehlererkennung definiert. In Abschnitt 9.2 wird eine Beobachtergleichung für die TT-Repräsentation stochastischer Automaten definiert und ein entsprechender qualitativer Beobachteralgorithmus für die Fehlererkennung beschrieben.

9.1 Beobachtung mit der CP-Repräsentation des Verhaltenstensors

9.1.1 Beobachtung durch Komposition von Submatrizen

Die Beobachtergleichung in Definition 7.2 basiert auf der Matrix-Vektor Multiplikation der in Abhängigkeit von der aktuellen Ein- und Ausgabe selektierten Submatrix $\mathbf{L}(:, w(k), :, v(k)) \in [0, 1]^{N \times N}$ des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ des stochastischen Automaten \mathcal{A} , mit dem Zustandswahrscheinlichkeitsvektor $\mathbf{p}_z(k) \in [0, 1]^N$. Um die Beobachtergleichung des stochastischen Automaten (7.6) an die CP-Repräsentation des Verhaltenstensors in Definition 8.5 anzupassen, erscheint es daher naheliegend, ebendiese Submatrix auf Basis der CP-Repräsentation $[\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!]$ des Verhaltenstensors zu berechnen. Die Vorgehensweise hierbei wird im Folgenden erläutert.

Proposition 9.1: *Unter der Voraussetzung einer exakten CP-Dekomposition des Verhaltenstensors $\mathbf{L} = [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!]$ nach Definition 8.6 gilt für die in der allgemeinen Beobachtergleichung (7.6) im Zähler und Nenner auftretende Submatrix*

$$\mathbf{L}(:, w(k), :, v(k)) = [\![\boldsymbol{\alpha}(k); \mathbf{A}, \mathbf{C}]\!] \in [0, 1]^N, \quad (9.1)$$

mit dem Vektor

$$\boldsymbol{\alpha}(k) = \boldsymbol{\lambda} \otimes (\mathbf{b}(w(k), :) \otimes \mathbf{d}(v(k), :))^T \in \mathbb{R}_{\geq 0}^R. \quad (9.2)$$

Dabei sind

$$\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}, \mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}, \mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}, \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R,$$

die Faktormatrizen und der Gewichtungsvektor. In (9.2) ist $\mathbf{b}(w(k), :) \in \mathbb{R}_{\geq 0}^{1 \times R}$ demzufolge die w -te Zeile der Faktormatrix \mathbf{B} und $\mathbf{d}(v(k), :) \in \mathbb{R}_{\geq 0}^{1 \times R}$ die v -te Zeile der Faktormatrix \mathbf{D} .

Beweis: Der Beweis von Proposition 9.1 findet sich im Anhang B.2.

Wie Proposition 9.1 und der Beweis zeigen, kann mit Gleichung (9.1) für alle Eingänge $v(k)$ und Ausgänge $w(k)$ die entsprechende Submatrix aus der CP-Repräsentation des Verhaltenstensors berechnet werden. Somit lässt sich eine Beobachtergleichung für die CP-Repräsentation wie folgt notieren:

Lemma 9.1 (CP-Beobachtergleichung für komponierte Submatrizen): *Durch Einsetzen von Gleichung (9.1) in Gleichung (7.6) erhält man die Beobachtergleichung für die CP-Repräsentation des Verhaltenstensors:*

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\llbracket \boldsymbol{\alpha}(k); \mathbf{A}, \mathbf{C} \rrbracket \cdot \mathbf{p}_z(k)}{\|\llbracket \boldsymbol{\alpha}(k); \mathbf{A}, \mathbf{C} \rrbracket \cdot \mathbf{p}_z(k)\|_1} \in [0, 1]^{N \times N}, \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0. \end{aligned} \quad (9.3)$$

Für den in Gleichung (9.1) und (9.3) auftretenden Kruskal-Tensor gilt nach Definition 6.18 die Beziehung

$$\llbracket \boldsymbol{\alpha}(k); \mathbf{A}, \mathbf{C} \rrbracket = \sum_{r=1}^R \alpha_r(k) \mathbf{a}_{:r} \circ \mathbf{c}_{:r}. \quad (9.4)$$

Aufgrund der Tatsache, dass das äußere Vektorprodukt $\mathbf{a}_{:r} \circ \mathbf{c}_{:r}$ zeitinvariant ist, muss dieses nicht für jeden Zeitschritt k berechnet werden, sondern kann mit

$$\mathbf{K}_r = \mathbf{a}_{:r} \circ \mathbf{c}_{:r} \in \mathbb{R}_{\geq 0}^{N \times N} \quad (9.5)$$

für alle $r = 1, \dots, R$ einmalig berechnet werden. Somit erhält man mit $\mathbf{K}_1, \dots, \mathbf{K}_R$ eine Anzahl von R Matrizen der Größe $N \times N$. Da die Skalarmultiplikation mit dem äußeren Produkt verträglich ist, gilt für den skalaren Multiplikator $\alpha_r(k)$ allgemein

$$\alpha_r(k) \mathbf{a}_{:r} \circ \mathbf{c}_{:r} = \alpha_r(k) (\mathbf{a}_{:r} \circ \mathbf{c}_{:r}) = \alpha_r(k) \cdot \mathbf{K}_r, \quad (9.6)$$

womit sich Gleichung (9.1) wie folgt darstellen lässt

$$\mathbf{L}(:, w(k), :, v(k)) = \sum_{r=1}^R \alpha_r(k) \cdot \mathbf{K}_r \in [0, 1]^{N \times N}. \quad (9.7)$$

Die Gleichungen (9.1) und (9.7) sind somit äquivalent und durch die Beziehung in (9.6) gilt weiterhin Beweis B.2. Anstelle von (9.3) kann dann die Gleichung

$$\mathbf{p}_z(k+1) = \frac{\left(\sum_{r=1}^R \alpha_r(k) \cdot \mathbf{K}_r \right) \cdot \mathbf{p}_z(k)}{\left\| \left(\sum_{r=1}^R \alpha_r(k) \cdot \mathbf{K}_r \right) \cdot \mathbf{p}_z(k) \right\|_1} \in [0, 1]^{N \times N}, \quad (9.8)$$

verwendet werden, bei der der Berechnungsaufwand durch die einmalige Vorprozessierung des Produkts (9.5) entsprechend geringer ist. Für eine exakte CP-Dekomposition des Verhaltenstensors führen die Gleichungen (9.8) und (9.3) entsprechend auf die gleichen Resultate wie die Beobachtergleichung (7.6).

Die Methode der Beobachtung durch Komposition der entsprechenden Submatrizen des Verhaltenstensors kann insbesondere bei vielen Zuständen N und großen Dekompositionsrängen R sehr rechen- und speicherintensiv sein. Generell empfiehlt sich daher die Vorabberechnung der äußeren Produkte, also die Verwendung von Gleichung (9.8), da damit die Anzahl der Rechenoperationen pro Zeitschritt k erheblich reduziert wird. Unabhängig davon, ob die Beobachtergleichung (9.3) oder (9.8) verwendet wird, besteht bei der Komposition der Submatrizen jedoch das grundlegende Problem, dass der Speicherbedarf gegenüber dem der CP-Repräsentation des Verhaltenstensors wieder anwächst. Dies wird im Folgenden anhand der Beobachtergleichung (9.8) erläutert.

Speicherbedarf bei der Komposition von Submatrizen

Der Speicherbedarf bei der Komposition von Submatrizen beträgt

$$\gamma_{\text{CP}_K} = (Q + M) \cdot R + N^2 \cdot R + R \quad (9.9)$$

Werte, da neben den Faktormatrizen $\mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}$, $\mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}$ und dem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$ zusätzlich für alle Indizes $r = 1, \dots, R$ die Matrizen $\mathbf{K}^{(r)} \in \mathbb{R}_{\geq 0}^{N \times N}$ gespeichert werden müssen. Abbildung 9.1 zeigt für verschiedenen Dekompositionsrange R und eine äquivalente Anzahl von Eingaben M , Ausgaben Q und Zuständen N , den Speicherbedarf der CP-Repräsentation des Verhaltenstensors γ_{CP} aus Gleichung (8.4), im Vergleich zum Speicherbedarf γ_{CP_K} aus Gleichung (9.9).

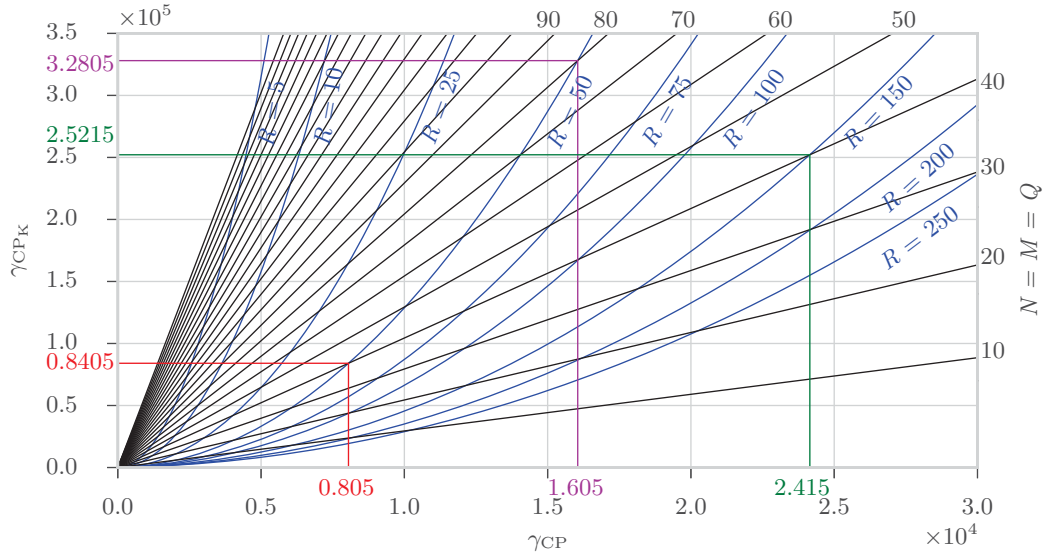


Abbildung 9.1: Speicherbedarf (8.4) eines CP-dekomponierten Verhaltenstensors (x -Achse), im Vergleich zum Speicherbedarf (9.9) mit vorprozessierten Matrizen $\mathbf{K}^{(r)}$ (y -Achse). Der Randmaßstab auf der rechten und oberen Seite steht für die Anzahl der Eingaben M , Ausgaben Q und Zustände N , wobei $N = M = Q$ gilt.

Beispielsweise führt die CP-Dekomposition des Verhaltenstensors eines Systems mit $N = M = Q = 40$ und einem Dekompositionsrang von $R = 50$ zu einer Anzahl von $(40 + 40 + 40 + 40) \cdot 50 + 50 = 8050$ Werten, während bei der Vorberechnung der Matrizen $\mathbf{K}^{(r)}$, der Speicherbedarf um über eine Größenordnung höher ist $((40 + 40) \cdot 50 + 40 \cdot 40 \cdot 50 + 50 = 84\,050)$. Dies ist beispielhaft in Abbildung 9.1 in Rot dargestellt. Allgemein gilt, dass der Speicherbedarf γ_{CP_K} und γ_{CP} für beliebige aber gleiche Anzahlen von Ein-, Ausgaben und Zuständen $N = M = Q$ mit zunehmendem Rang R in der Art linear wächst, dass eine Verdopplung des Rangs zu einer Verdreifachung der zu speichernden Werte führt. Für das in Abbildung 9.1 in Rot dargestellte Beispiel ergibt sich beispielsweise für eine Verdreifachung des Rangs ($R = 150$) und $N = M = Q = 40$ sowohl für den Speicherbedarf γ_{CP_K} als auch für γ_{CP} jeweils exakt der dreifache Wert, was in Abbildung 9.1 in Grün dargestellt ist. Anders verhält es sich, wenn man für einen beliebigen Dekompositionsrang R die Anzahl der Zustände N erhöht. Während der Wert N in Gleichung (8.4) nur einen linearen Einfluss auf den Speicherbedarf γ_{CP} hat, besteht in (9.9) ein quadratischer Zusammenhang zwischen dem Speicherbedarf γ_{CP_K} und der Anzahl der Zustände. Wieder ausgehend von dem in Rot eingezeichneten Beispiel erhält man nun für $R = 50$ und eine Verdopplung der Anzahlen von Eingängen, Ausgängen und Zuständen ($N = M = Q = 80$), den in Magenta eingezeichneten Speicherbedarf $\gamma_{\text{CP}} = 16\,050$. Damit ergibt sich in etwa eine Verdopplung der zu speichernden Werte, während sich der Speicherbedarf $\gamma_{\text{CP}_K} = 328\,050$ circa vervierfacht. Für dieses Beispiel beträgt der Speicherbedarf γ_{CP_K} aufgrund der Vorprozessierung der Matrizen $\mathbf{K}^{(r)}$

bereits über das 20-fache von dem der reinen CP-Dekomposition. Dies ist wohlgermerkt aber immer noch deutlich weniger, als wenn man den Verhaltenstensor $\mathbf{L} \in [0, 1]^{80 \times 80 \times 80 \times 80}$ mit 40 960 000 Werten in der vollen Darstellung speichern würde.

Wie aus den obigen Ausführungen klar wird, scheint es insbesondere für Systeme mit einer hohen Anzahl von Zuständen nicht zielführend zu sein, die Submatrizen des Verhaltenstensor zu komponieren. Es ist somit naheliegend, dass um das gesamte Potenzial der CP-Dekomposition auszuschöpfen, die Berechnung des Zustandswahrscheinlichkeitsvektors $\mathbf{p}_z(k+1)$, direkt auf Basis der Faktormatrizen erfolgen muss – und zwar ohne die Vorprozessierung der Submatrizen. Eine Beobachtergleichung die dieser Anforderung gerecht wird, wird im Folgenden beschrieben.

9.1.2 CP-Beobachtergleichung

Nachstehend wird eine Beobachtergleichung für die CP-Repräsentation des Verhaltenstensors vorgestellt, welche die direkte Berechnung des Zustandswahrscheinlichkeitsvektors ermöglicht.

Proposition 9.2: *Unter der Voraussetzung einer exakten CP-Dekomposition des Verhaltenstensors $\mathbf{L} = [\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$ nach Definition 8.6 gilt für das in der erweiterten Beobachtergleichung (7.15) im Zähler und Nenner auftretende kontrahierte Tensorprodukt*

$$\begin{aligned} & \left\langle \mathbf{L} \mid [\mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k)] \right\rangle \\ &= \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\mathbf{B}^\top \cdot \mathbf{p}_w(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\mathbf{D}^\top \cdot \mathbf{p}_v(k) \right) \right) \in [0, 1]^N. \end{aligned} \quad (9.10)$$

Dabei sind die Faktormatrizen, der Gewichtungsvektor und die Wahrscheinlichkeitsvektoren der Eingaben, Ausgaben und Zustände wie folgt gegeben:

$$\begin{aligned} \mathbf{A} &\in \mathbb{R}_{\geq 0}^{N \times R}, \quad \mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}, \quad \mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}, \quad \mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}, \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R, \\ \mathbf{p}_z(k) &\in [0, 1]^N, \quad \mathbf{p}_w(k) \in [0, 1]^Q, \quad \mathbf{p}_v(k) \in [0, 1]^M. \end{aligned}$$

Beweis: Der Beweis von Proposition 9.2 findet sich im Anhang B.3.

Lemma 9.2 (CP-Beobachtergleichung): *Durch Einsetzen von Gleichung (9.10) in den Zähler und Nenner von Gleichung (7.15) folgt mit*

$$\begin{aligned} k \geq 0: \quad \mathbf{p}_z(k+1) &= \frac{\mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\mathbf{B}^\top \cdot \mathbf{p}_w(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\mathbf{D}^\top \cdot \mathbf{p}_v(k) \right) \right)}{\left\| \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\mathbf{B}^\top \cdot \mathbf{p}_w(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\mathbf{D}^\top \cdot \mathbf{p}_v(k) \right) \right\|_1} \in [0, 1]^N, \\ k = 0: \quad \mathbf{p}_z(0) &= \mathbf{p}_0 \end{aligned} \quad (9.11)$$

die allgemeine CP-Beobachtergleichung für exakt CP-dekomponierte Verhaltenstensoren. Dabei gilt $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in [0, 1]^Q$ sowie $\mathbf{p}_v(k) \in [0, 1]^M$.

Korollar 9.1: *Aus dem Beweis der Proposition 9.2 und dem Beweis der Proposition 7.1 folgt unmittelbar, dass*

$$\begin{aligned} & \mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k) \\ &= \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\mathbf{B}^\top \cdot \mathbf{p}_w(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\mathbf{D}^\top \cdot \mathbf{p}_v(k) \right) \right) \in [0, 1]^N \end{aligned} \quad (9.12)$$

gilt, wenn die Ein- und Ausgabewahrscheinlichkeitsvektoren exakt und durch $\mathbf{p}_w(k) \in \{0, 1\}^Q$ und $\mathbf{p}_v(k) \in \{0, 1\}^M$ gegeben sind.

Ein direkter Beweis des Korollars 9.1 findet sich auch in [91]. Zusammenfassend können nun zwei grundlegende Aussagen bezüglich der Beobachtung mit CP-dekomponierten Verhaltenstensen formuliert werden:

1. Lemma 9.2 führt für eine exakte CP-Dekomposition des Verhaltenstensors und unsichere Ein- und Ausgaben auf die gleichen Zustandswahrscheinlichkeiten wie die erweiterte Beobachtergleichung (7.15), wobei die Zustandswahrscheinlichkeitsverteilungen durch (7.14) gegeben sind.
2. Aus dem Korollar 9.1 folgt, dass Lemma 9.2 für eine exakte CP-Dekomposition und exakt bekannte Ein- und Ausgaben die gleichen Zustandswahrscheinlichkeiten wie die Beobachtergleichung (7.6) generiert. Die Zustandswahrscheinlichkeitsverteilungen sind dann entsprechend durch (3.59) gegeben.

Ist die CP-Dekomposition nicht exakt, das heißt der Verhaltenstensor stellt nur eine Approximation $\hat{\mathbf{L}} \in \mathbb{R}_{\geq 0}^{N \times Q \times N \times M}$ mit Elementen $\hat{l}(\bar{z}, w, z, v)$ dar, dann ist offensichtlich, dass auch der mit der Beobachtergleichung (9.11) berechnete Vektor der Zustandswahrscheinlichkeiten nur eine Approximation $\hat{\mathbf{p}}_z(k+1) \in \mathbb{R}_{\geq 0}^N$ repräsentiert.

Die Struktur der CP-Beobachtergleichung kann zum besseren Verständnis auch grafisch interpretiert werden. Abbildung 9.2 zeigt die Visualisierung des Terms (9.10) aus Proposition 9.2, der sowohl im Nenner als auch im Zähler des Lemmas 9.2 auftritt.

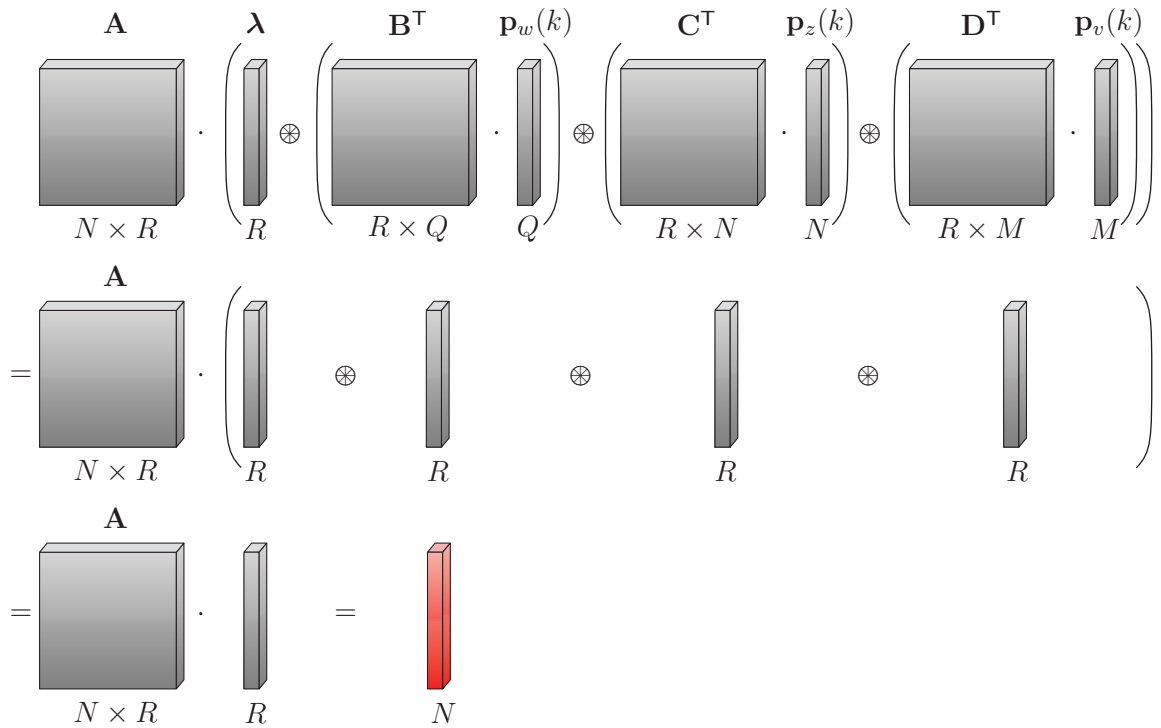


Abbildung 9.2: Grafische Interpretation der CP-Beobachtergleichung

9.1.3 CP-Algorithmus zur qualitativen Fehlererkennung

Der Algorithmus 9.1 zur qualitativen Fehlerdetektion ist hier für den in der Praxis relevanten Fall eines approximierten Verhaltenstensors notiert. Die approximierten Größen des qualitativen Modells werden dabei durch das Symbol $\hat{\cdot}$ gekennzeichnet. Der Verhaltenstensor $\hat{\mathbf{L}} \in [0, 1]^{N \times Q \times N \times M}$ des qualitativen Modells ist somit durch die CP-Repräsentation $\hat{\mathbf{L}} = [\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}] \in \mathbb{R}_{\geq 0}^{N \times Q \times N \times M}$ gegeben, welche durch die Rang- R Approximation in Definition 8.5 berechnet wird. Dabei sei darauf hingewiesen, dass im approximativen Fall die Summennorm im Nenner der Beobachtergleichung

mitunter Werte $\hat{b}(k) > 1$ annimmt (vgl. Abschnitt 8.2.4). Außerdem führt die Approximation des Verhaltenstensors dazu, dass dessen Nullelemente in der Näherung $\hat{\mathbf{L}}$ durch sehr kleine Werte ungleich null repräsentiert werden. Da die Nullelemente jedoch eine entscheidende Rolle bei der Fehlerdetektion spielen, ist das Festlegen des Schwellwerts τ aus Gleichung (4.48) in der Regel zwingend erforderlich (s. Abschnitt 4.4.3). Welchen Wert genau man für den Schwellwert τ wählen muss, lässt sich nicht pauschal beantworten, da dies zum einen von der Approximationsgüte der Dekomposition abhängt und zum anderen eine Frage der gewünschten Sensitivität ist.

Der Ablauf des nachstehend angegebenen Algorithmus 9.1 zur qualitativen Fehlererkennung erfolgt analog zur Beschreibung in Abschnitt 4.4.4.

Algorithmus 9.1: CP-Algorithmus zur qualitativen Fehlererkennung

Gegeben: Qualitatives Modell $\tilde{\mathbf{A}}$, Gl. (4.15) mit Verhaltenstensor
 in CP-Repräsentation $\hat{\mathbf{L}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \rrbracket$, Gl. (8.7)
 Folge von Wahrscheinlichkeitsvektoren qualitativer Eingänge $\tilde{\mathbf{P}}_v(0 \dots T)$, Gl. (7.31)
 Folge von Wahrscheinlichkeitsvektoren qualitativer Ausgänge $\tilde{\mathbf{P}}_w(0 \dots T)$, Gl. (7.32)

Initialisiere: $k = 0$, $\hat{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$

Iteriere:

- 1 **solange** $0 \leq k \leq T$
- 2 **berechne** $\mathbf{A} \cdot (\boldsymbol{\lambda} \otimes (\mathbf{B}^\top \cdot \tilde{\mathbf{p}}_w(k)) \otimes (\mathbf{C}^\top \cdot \hat{\mathbf{p}}_z(k)) \otimes (\mathbf{D}^\top \cdot \tilde{\mathbf{p}}_v(k)))$, Gl. (9.11)
- 3 **berechne** $\|\mathbf{A} \cdot (\boldsymbol{\lambda} \otimes (\mathbf{B}^\top \cdot \tilde{\mathbf{p}}_w(k)) \otimes (\mathbf{C}^\top \cdot \hat{\mathbf{p}}_z(k)) \otimes (\mathbf{D}^\top \cdot \tilde{\mathbf{p}}_v(k)))\|_1 =: \hat{b}(k)$,
- 4 **berechne** $\hat{b}_m(k)$, Gl. (4.46) und $\hat{b}_b(k)$, Gl. (4.48)
- 5 **wenn** $\hat{b}(k) = 0$ **dann**
- 6 **setze** $\hat{\mathbf{p}}_z(k+1) = \tilde{\mathbf{p}}_0$
- 7 **setze** $k := k + 1$ und fahre fort mit Schritt 1
- 8 **sonst**
- 9 **berechne** $\hat{\mathbf{p}}_z(k+1)$, Gl. (9.11)
- 10 **setze** $k := k + 1$ und fahre fort mit Schritt 1

Ergebnis: Folge $\hat{\mathbf{P}}_z 0 \dots T + 1 \mid \tilde{\mathbf{P}}_v(0 \dots T), \tilde{\mathbf{P}}_w(0 \dots T)$, Gl. (7.36)
 Folge $\hat{\hat{B}}(0 \dots T \mid \tilde{\mathbf{P}}_v(0 \dots T), \tilde{\mathbf{P}}_w(0 \dots T))$, Gl. (7.38)
 Folge $\hat{\hat{B}}_m(0 \dots T)$, Gl. (4.47)
 Folge $\hat{\hat{B}}_b(0 \dots T)$, Gl. (4.49)

Eine Demonstration des CP-Algorithmus 9.1 zur qualitativen Fehlererkennung erfolgt in Beispiel 9.1 für den simulierten Erhitzer aus Abschnitt 5.2.6 und in Beispiel 9.2 für den der realen RLT-Anlage aus Abschnitt 5.3.

Beispiel 9.1 (CP-Beobachtergleichung): Im Folgenden wird das Prinzip der CP-Beobachtergleichung (9.11) an einer Rang-4 Approximation des Verhaltenstensors $\tilde{\mathbf{L}} \in [0, 1]^{6 \times 3 \times 6 \times 2}$ aus (7.24) demonstriert. Dies ist der Verhaltenstensor eines qualitativen Erhitzermodells mit den Modellparametern aus Tabelle 5.5 und der Beobachterzeitschrittdauer $T_b = 6$. Die Rang-4 Approximation des Tensor führt dabei zu den Faktormatrizen

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0.186 & 0.996 \\ 0.075 & 0.001 & 0.983 & 0.085 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.996 & 0.058 & 0 & 0 \\ 0.053 & 0.998 & 0 & 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^{6 \times 4}, \quad \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^{3 \times 4},$$

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 0.924 \\ 0.959 & 0 & 0.709 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.9745 & 0 & 0.298 & 0 \\ 0 & 0.945 & 0 & 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^{6 \times 4}, \quad \mathbf{D} = \begin{pmatrix} 0.412 & 0 & 1 & 1 \\ 0.911 & 1 & 0 & 0 \end{pmatrix} \in \mathbb{R}_{\geq 0}^{2 \times 4}$$

und dem Gewichtungsvektor $\boldsymbol{\lambda} = \mathbf{1}^4$, der ausschließlich die Werte Eins enthält. Der qualitative Eingang zur Zeit $k = 0$ ist $v(0) = 2$ und für den Ausgang gilt $w(0) = 3$. Damit ist der Wahrscheinlichkeitsvektor des qualitativen Eingangs durch

$$\tilde{\mathbf{p}}_v(0) = (0 \ 1)^T$$

und der qualitative Ausgangswahrscheinlichkeitsvektor mit

$$\tilde{\mathbf{p}}_w(0) = (0 \ 0 \ 1)^T$$

gegeben. Der initiale Wahrscheinlichkeitsvektor der qualitativen Zustände sei

$$\hat{\mathbf{p}}_z(0) = \tilde{\mathbf{p}}_0 = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T.$$

Nun können die Matrix-Vektor-Produkte $\mathbf{B}^T \cdot \tilde{\mathbf{p}}_w(0)$, $\mathbf{C}^T \cdot \hat{\mathbf{p}}_z(0)$ und $\mathbf{D}^T \cdot \tilde{\mathbf{p}}_v(0)$ in der CP-Beobachtergleichung (9.11) berechnet werden. Da jeder dieser Vektoren nur ein Einselement besitzt, wird durch das Matrix-Vektorprodukt jeweils nur diejenige Spalte der transponierten Faktormatrix selektiert, die dem Index des Einselements des entsprechenden Vektors entspricht. Es ist somit

$$\begin{aligned} \mathbf{B}^T \cdot \tilde{\mathbf{p}}_w(0) &= (0 \ 1 \ 0 \ 0)^T, \\ \mathbf{C}^T \cdot \hat{\mathbf{p}}_z(0) &= (0 \ 0.945 \ 0 \ 0)^T, \\ \mathbf{D}^T \cdot \tilde{\mathbf{p}}_v(0) &= (0.911 \ 1 \ 0 \ 0)^T. \end{aligned}$$

Damit kann der Nenner der CP-Beobachtergleichung notiert und berechnet werden:

$$\begin{aligned} & \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes (\mathbf{B}^T \cdot \tilde{\mathbf{p}}_w(0)) \otimes (\mathbf{C}^T \cdot \hat{\mathbf{p}}_z(0)) \otimes (\mathbf{D}^T \cdot \tilde{\mathbf{p}}_v(0)) \right) \\ &= \begin{pmatrix} 0 & 0 & 0.186 & 0.996 \\ 0.075 & 0.001 & 0.983 & 0.085 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.996 & 0.058 & 0 & 0 \\ 0.053 & 0.998 & 0 & 0 \end{pmatrix} \cdot \left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0.945 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0.911 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right) \\ &= \begin{pmatrix} 0 & 0 & 0.186 & 0.996 \\ 0.075 & 0.001 & 0.983 & 0.085 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.996 & 0.058 & 0 & 0 \\ 0.053 & 0.998 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0.945 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0.001 \\ 0 \\ 0 \\ 0.055 \\ 0.943 \end{pmatrix} \tag{9.13} \end{aligned}$$

Im Nenner der CP-Beobachtergleichung steht wiederum die Summennorm, was bedeutet, dass jedes Element des Vektors (9.13) durch die Summe seiner Elemente dividiert werden muss. Diese Summe

beträgt hier 0.999, womit man unmittelbar den Vektor der qualitativen Zustandswahrscheinlichkeiten zur Zeit $k = 1$ erhält

$$\hat{\mathbf{p}}_z(1) = \begin{pmatrix} 0 \\ 0.001 \\ 0 \\ 0 \\ 0.055 \\ 0.944 \end{pmatrix}. \quad (9.14)$$

Der Wert des Vektors (9.14) stellt dabei eine sehr gute Näherung der Wahrscheinlichkeiten der sechsten Spalte des Subtensors $\tilde{\mathbf{L}}(:, 3, :, 2)$ aus Gleichung (7.24) dar. Ebendiese Zustandswahrscheinlichkeiten erhält man für das gegebene E/A Paar $w(0) = 3, v(0) = 2$ auch mit der Tensor-Beobachtergleichung (7.25) des nicht dekomponierten Verhaltenstensors aus Gleichung (7.24):

$$\tilde{\mathbf{p}}_z(1) = \frac{\tilde{\mathbf{L}}(:, 3, :, 2) \cdot \tilde{\mathbf{p}}_z(0)}{\|\tilde{\mathbf{L}}(:, 3, :, 2) \cdot \tilde{\mathbf{p}}_z(0)\|_1} = \begin{pmatrix} 0 \\ 0.001 \\ 0 \\ 0 \\ 0.055 \\ 0.944 \end{pmatrix}.$$

Der Speicherbedarf des qualitativen Modells beträgt nach Gleichung (8.4) bei der Rang-4 Approximation $\gamma_{CP} = 4 \cdot (6 + 3 + 6 + 2) + 4 = 72$ Werte, was nur noch 28.1 % des ursprünglichen Speicherbedarfs des Verhaltenstensors von $\gamma = 256$ Werten entspricht. Abbildung 9.3 zeigt dabei einen Vergleich der Verhaltenssignale für den gesamten Zeitbereich der Anwendungsdaten aus Abbildung 5.8. Wie man erkennen kann, reichen eine Rang-2 und Rang-3 Dekomposition nicht aus, um das Verhalten des Erhitzers hinsichtlich seines Nominal- und Fehlerverhaltens ordnungsgemäß zu bewerten. Die Rang-4 Approximation liefert hingegen ein sehr gutes Ergebnis, welches nahezu dem ohne Dekomposition entspricht.

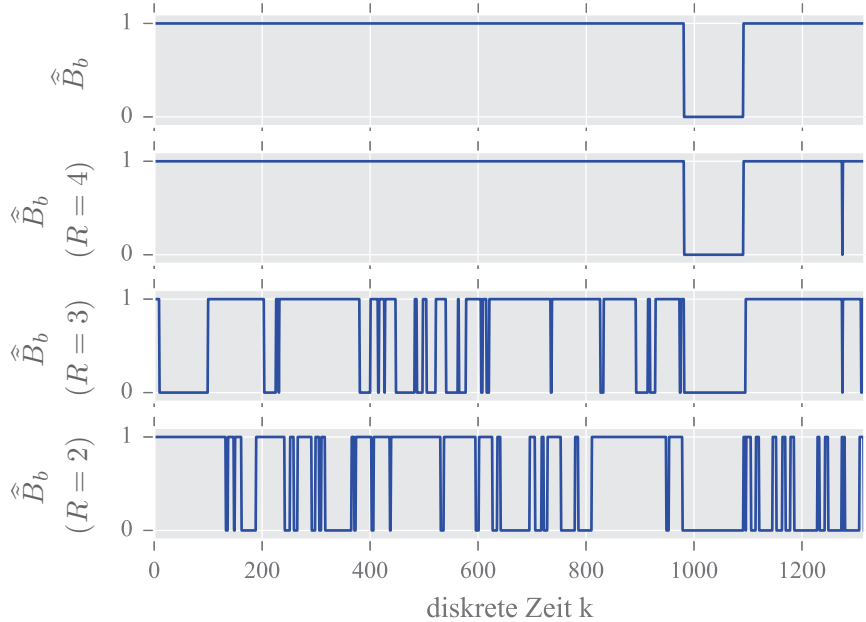


Abbildung 9.3: Verhaltenssignale \hat{B}_b von oben nach unten: ohne Dekomposition, Rang-4 (RMSD=0.039), Rang-3 (RMSD=0.46), Rang-2 Approximation (RMSD=0.55). Alle Signale wurden mit den Parametern $\phi = 2$, $\tau = 1\%$ und $T_b = 6$ bestimmt.

Die in der Beschriftung von Abbildung 9.3 angegebene Wurzel der mittleren quadratischen Abweichung (engl.: root-mean-square deviation (RMSD)) wurde jeweils anhand des Verhaltenssignals ohne Dekomposition \tilde{B} und denjenigen mit Dekomposition $\hat{\tilde{B}}$ berechnet. Dabei wurden die jeweils unveränderten Verhaltenssignale verwendet, also ohne Mittelwertbildung und Anwendung des Schwellwerts τ .

Beispiel 9.2 (Fehlererkennung mit der CP-Repräsentation): Die qualitative Beobachtung und Fehlererkennung mit der CP-Repräsentation wird im Folgenden anhand einer Rang-50 Approximation des Verhaltenstensors

$$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 36 \times 36 \times 36} \quad (9.15)$$

eines qualitativen Modells des Erhitzers aus Abschnitt 5.3 gezeigt. Die Dimensionen des Verhaltenstensors (9.15) ergeben sich aus der in Tabelle 5.11 gegebenen Anzahl der qualitativen Zustände $N = 36$, Ausgänge $Q = 36$ und Eingänge $M = 36$. Die Rang-50 Approximation dieses Tensors führt entsprechend Definition (8.5) auf die CP-Repräsentation

$$\hat{\tilde{\mathbf{L}}} = [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!] \in \mathbb{R}_{\geq 0}^{36 \times 36 \times 36 \times 36} \quad (9.16)$$

mit den Faktormatrizen

$$\mathbf{A} \in \mathbb{R}_{\geq 0}^{36 \times 50}, \mathbf{B} \in \mathbb{R}_{\geq 0}^{36 \times 50}, \mathbf{C} \in \mathbb{R}_{\geq 0}^{36 \times 50}, \mathbf{D} \in \mathbb{R}_{\geq 0}^{36 \times 50}$$

und dem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{50}$. Für die Dekomposition wurde der HALS Algorithmus (C) aus Tabelle 8.1 mit 200 Iterationsschritten verwendet. Der Wahl des Dekompositionsrangs $R = 50$ stellt dabei den kleinstmöglichen Wert dar, mit dem sich ein vergleichbares Ergebnis wie bei der Betrachtung ohne Dekomposition realisieren ließ. Da das Optimierungsproblem der CP-Dekomposition mit zufälligen Startwerten iterativ gelöst wird, ist es jedoch nicht auszuschließen, dass man auch präzisere Zerlegungen mit womöglich niedrigerem Dekompositionsrang R finden kann, wenn man eine Vielzahl weiterer Dekompositionen berechnet.

Basierend auf der Zerlegung (9.16) kann nun der CP-Algorithmus 9.1 zur qualitativen Fehlerdetektion verwendet werden. Die Abbildung 9.4 zeigt dabei den Vergleich der Zustandswahrscheinlichkeiten der Wasseraustrittstemperatur T_{WA} des Erhitzers für den fehlerhaften Zeitbereich (vgl. Abbildung 5.19). Auf der linken Seite ist das Ergebnis des CP-Beobachteralgorithmus 9.1 dargestellt und auf der rechten Seite das Resultat ohne Dekomposition, berechnet mit Gleichung (7.25).

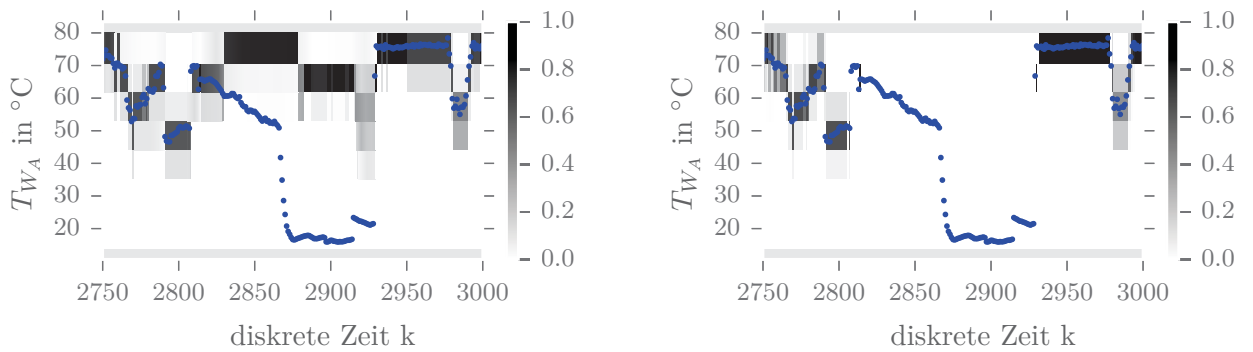


Abbildung 9.4: Vergleich der qualitativen Zustandswahrscheinlichkeiten mit Dekomposition (links) und ohne Dekomposition (rechts)

Das Ergebnis des CP-Algorithmus erweckt dabei zunächst den Eindruck eines völlig anderen Ergebnisses, welches den fehlerhaften Bereich mit positiven Zustandswahrscheinlichkeiten bewertet. Ein anderes Ergebnis zeigt sich jedoch, wenn man die Zustandswahrscheinlichkeiten jeweils vor der Normierung durch den Nenner der jeweiligen Beobachtergleichung betrachtet, was in Abbildung 9.5 dargestellt ist.

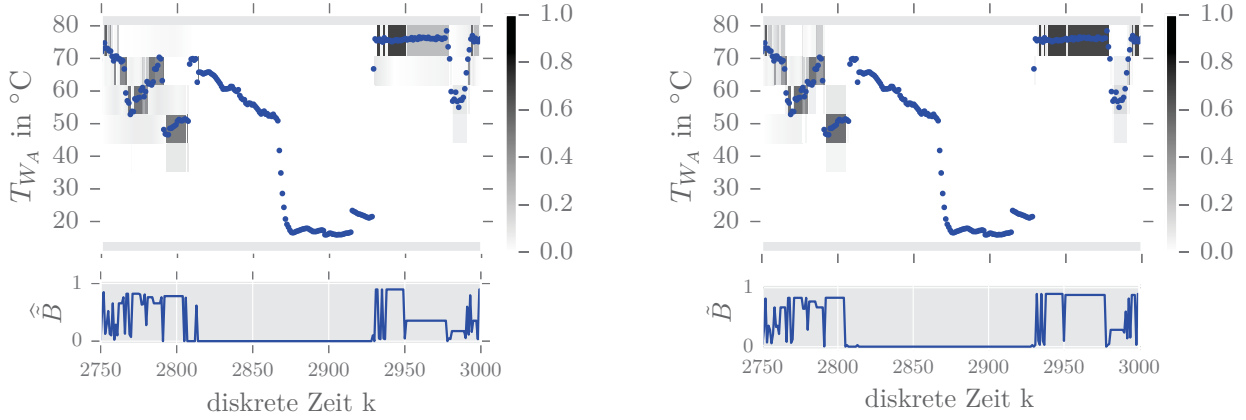


Abbildung 9.5: Vergleich der qualitativen Zustandswahrscheinlichkeiten vor der Normierung und der Verhaltenssignale mit Dekomposition \hat{B} (links) und ohne Dekomposition \tilde{B} (rechts)

Ein eindeutigeres Resultat zeigen grundsätzlich die Verhaltenssignale \hat{B} (links) und \tilde{B} (rechts) in Abbildung 9.5, welche die Normierungssummen der Zustandswahrscheinlichkeitsvektoren für jeden Zeitschritt beschreiben. Während diese innerhalb des Fehlerbereichs auf der rechten Seite tatsächlich Null sind, besitzen diese im dekomponierten Fall größtenteils sehr kleine Werte. Aus diesem Grund ist es bei den Algorithmen, die auf der dekomponierten Struktur des Verhaltenstensors basieren in der Regel notwendig, den Schwellwert für die Fehlererkennung τ auf einen kleinen Wert zu setzen, damit solche niedrigen Signalwerte entsprechend als Fehler detektiert werden können. Die beiden nachstehenden Abbildungen zeigen die Fehlersignale mit Dekomposition (links) und ohne Dekomposition (rechts) für einen Schwellwert $\tau = 1\%$ sowie einen gleitenden Mittelwert mit einer Fensterbreite von $\phi = 2$.

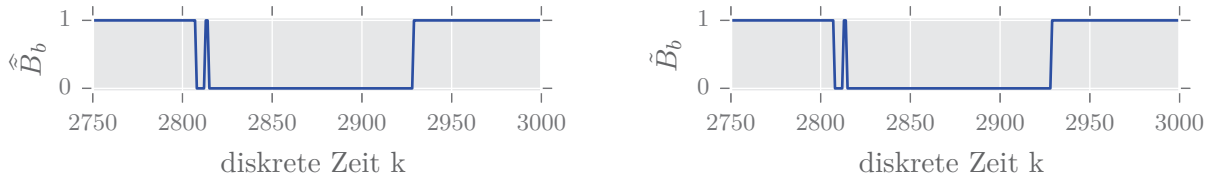


Abbildung 9.6: Vergleich der Verhaltenssignale \hat{B}_b (links) und \tilde{B}_b (rechts)

Wie dieses Beispiel zeigt, lassen sich mit der CP-Dekomposition entsprechend gleichwertige Fehlererkennungsergebnisse erzielen, und das, obwohl der Speicherbedarf in diesem Beispiel um den Faktor 231 mal geringer ist als bei der vollen Repräsentation des Verhaltenstensors. Anstelle von 1 679 616 Werte müssen dabei lediglich 7250 Werte gespeichert werden. Die Tabelle 9.1 fasst dabei die Ergebnisse dieses Beispiels zusammen.

Tabelle 9.1: Speicherbedarf des Verhaltenstensors des qualitativen Erhitzermodells ohne Dekomposition und für die Rang-50 Approximation der CP-Repräsentation

Repräsentation des Verhaltenstensors	Fehlererk. Algorithmus	Speicher- bedarf	Reduktions- faktor	RMSD \tilde{B} zu \hat{B}
$\tilde{L} \in [0, 1]^{36 \times 36 \times 36 \times 36}$, Gl. (9.15)	7.1	$\gamma = 1\,679\,616$	0	0
$\hat{L} = [\mathbf{A}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$, Gl. (9.16)	9.1	$\gamma_{CP} = 7250$	$\xi_{CP} = 231$	0.179

In der letzten Spalte der Tabelle ist die Wurzel der mittleren quadratischen Abweichung (RMSD) der Verhaltenssignale \tilde{B} und \hat{B} für den Trainingszeitraum des qualitativen Modells angegeben, die

hier als Indikator der Approximationsgüte der Dekomposition angesehen werden kann und im Folgenden zum Vergleich verschiedener Dekompositionen verwendet wird. Zwar geben alle Toolboxes zur Tensordekomposition in Tabelle 8.1 einen Approximationsfehler für die Faktorisierung an; jedoch basiert dessen Berechnung nicht bei allen auf der gleichen Vorgehensweise. Aus diesem Grund wird hier die RMSD zwischen den Verhaltenssignalen mit und ohne Dekomposition, basierend auf dem Trainingszeitraum des qualitativen Modells, verwendet.

9.1.4 Vergleich der Rechenzeiten

Die Tabellen 9.2 und 9.3 zeigen einen Vergleich der Rechenzeiten des einfachen Beobachteralgorithmus 7.1, des erweiterten Beobachteralgorithmus 7.2 sowie des CP-Beobachteralgorithmus 9.1. Die Berechnungen wurden dabei als »worst-case« Szenario durchgeführt, das heißt alle Vektoren, Matrizen und Tensoren besaßen keine Nullelemente.

Tabelle 9.2: $\mathbf{L} \in [0, 1]^{10 \times 10 \times 10 \times 10}$, beste Zeit aus 10.000 Wiederholungen

Algorithmus	Rechenzeit für einen Zeitschritt k in μs								
7.1	2.0								
7.2	62								
9.1	Rang	5	10	25	50	100	250	500	1000
		8.4	8,7	9,08	9.18	10,6	14,3	18,3	33,4

Tabelle 9.3: $\mathbf{L} \in [0, 1]^{100 \times 100 \times 100 \times 100}$, beste Zeit aus 1.000 Wiederholungen (*beste aus 100 Wdh.)

Algorithmus	Rechenzeit für einen Zeitschritt k in μs								
7.1	10								
7.2	555 000*								
9.1	Rang	5	10	25	50	100	250	500	1000
		17.6	12.5	13.1	20.3	24.1	39.3	58.8	190

Alle oben gezeigten Ergebnisse wurden mit einem Intel® Core™ i5 CPU @2.50GHz mit 8 GB RAM berechnet. Wie man sieht, bietet die einfache Beobachtermatrixgleichung mit Abstand die kürzesten Rechenzeiten. Allerdings ist ein Vergleich mit den anderen beiden Algorithmen nur schwer möglich, da diese mit den vektoriellen Ein- und Ausgaben auch Unsicherheiten berücksichtigen können. Jedoch zeigen die Tabellen deutlich, dass der CP-Beobachteralgorithmus selbst für große Dekompositionsränge wesentlich kürzere Rechenzeiten bietet, als der erweiterte Beobachteralgorithmus.

9.2 Beobachtung mit der TT-Repräsentation des Verhaltenstensors*

9.2.1 TT-Beobachtermatrixgleichung*

Auch für den TT-dekomponierten und permutierten Verhaltenstensor in Definition 8.10 lässt sich eine Beobachtermatrixgleichung ableiten, die auf Basis der TT-Kerne, den Wahrscheinlichkeitsvektor $\mathbf{p}_z(k+1)$ berechnet. Bevor die im Folgenden als *TT-Beobachtermatrixgleichung* bezeichnete Formel aufgestellt werden kann, müssen jedoch noch die einfachen Beobachtermatrixgleichungen an den permutierten Verhaltenstensor $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ mit den Elementen $\underline{\mathbf{L}}(w, v, \bar{z}, z)$ aus Gleichung (8.18) angepasst werden. Die einfache Matrix-Vektor Beobachtermatrixgleichung (7.6) nimmt dabei folgende Form an:

$$\begin{aligned}
 k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\underline{\mathbf{L}}(w(k), v(k), :, :) \cdot \mathbf{p}_z(k)}{\|\underline{\mathbf{L}}(w(k), v(k), :, :) \cdot \mathbf{p}_z(k)\|_1} \in [0, 1]^N, \\
 k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0.
 \end{aligned} \tag{9.17}$$

Da für die Elemente $\underline{l}(w, v, \bar{z}, z)$ des Verhaltenstensors $\underline{L} \in [0, 1]^{Q \times M \times N \times N}$ die Beziehung

$$\underline{l}(w, v, \bar{z}, z) = l(\bar{z}, w, z, v) = \mathcal{L}(\bar{z}, w | z, v) \quad (9.18)$$

gilt, führt auch Gleichung (9.17) auf die gleichen Ergebnisse wie Gleichung (7.6). Um die erweiterte Beobachtergleichung (7.15) auf Basis des kontrahierten Tensorprodukts für den permutierten Verhaltenstensor \underline{L} aufzustellen, lässt sich analog zu Proposition 7.1, die folgende Behauptung aufstellen:

Proposition 9.3: *Für exakte Ein- und Ausgangswahrscheinlichkeitsvektoren gilt auf Basis des permutierten Verhaltenstensors*

$$\left\langle \underline{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)} = \underline{L}(w(k), v(k), :, :) \cdot \mathbf{p}_z(k) \in [0, 1]^N \quad (9.19)$$

mit $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in \{0, 1\}^Q$ sowie $\mathbf{p}_v(k) \in \{0, 1\}^M$.

Beweis: Der Beweis von Proposition 9.3 findet sich im Anhang B.4.

Dabei ist zu beachten, dass sich die Reihenfolge der Eingangs-, Ausgangs- und Zustandsvektoren innerhalb des Kruskal-Rang-1 Tensors $\llbracket \cdot \rrbracket$ geändert hat.¹ Da die Modi entlang derer die Tensoren in (9.19) kontrahiert werden, nicht mehr der Vorgabe aus Definition 6.15 entsprechen, müssen diese hier explizit angegeben werden. Analog zur erweiterten Beobachtergleichung (7.15), kann damit die erweiterte Beobachtergleichung für den permutierten Verhaltenstensor notiert werden:

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\left\langle \underline{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)}}{\left\| \left\langle \underline{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)} \right\|_1} \in [0, 1]^N, \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0. \end{aligned} \quad (9.20)$$

Mit Gleichung (9.20) erhält man entsprechend für exakte Ein- und Ausgänge, die Zustandswahrscheinlichkeitsverteilungen (3.54) und für unsichere Ein- und Ausgänge, diejenigen aus der Gleichung (7.14). Damit kann die TT-Beobachtergleichung wie folgt aufgestellt werden:

Proposition 9.4: *Unter der Voraussetzung einer exakten TT Dekomposition des Verhaltenstensors $\underline{L} = \langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle$ nach Definition 8.11 gilt für das in der erweiterten Beobachtergleichung (9.20) im Zähler und Nenner auftretende kontrahierte Tensorprodukt*

$$\begin{aligned} &\left\langle \underline{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)} \\ &= (\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k)) \in [0, 1]^N. \end{aligned} \quad (9.21)$$

Dabei sind die TT-Kerne sowie die Wahrscheinlichkeitsvektoren der Eingänge, Ausgänge und Zustände durch

$$\begin{aligned} \mathbf{E} &\in \mathbb{R}^{1 \times Q \times R_1}, \quad \mathbf{F} \in \mathbb{R}^{R_1 \times M \times R_2}, \quad \mathbf{G} \in \mathbb{R}^{R_2 \times N \times R_3}, \quad \mathbf{H} \in \mathbb{R}^{R_3 \times N \times 1}, \\ \mathbf{p}_z(k) &\in [0, 1]^N, \quad \mathbf{p}_w(k) \in \{0, 1\}^Q, \quad \mathbf{p}_v(k) \in \{0, 1\}^M, \end{aligned}$$

gegeben.

Beweis: Der Beweis von Proposition 9.4 findet sich im Anhang B.5.

¹Die Änderung der Reihenfolge der Vektoren im Kruskal-Tensor ist nicht zwingend erforderlich. Man kann anstelle von (9.19) auch $\left\langle \underline{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle_{(1,4,2|1,2,3)}$ schreiben.

Lemma 9.3 (TT-Beobachtergleichung): *Durch Einsetzen von Gleichung (9.21) in den Zähler und Nenner von Gleichung (9.20) folgt mit*

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{(\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k))}{\|(\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k))\|_1} \in [0, 1]^N, \quad (9.22) \\ k = 0 : \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0 \end{aligned}$$

die allgemeine Beobachtergleichung für exakt TT-dekomponierte Verhaltenstensenoren. Dabei gilt $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w(k) \in [0, 1]^Q$ sowie $\mathbf{p}_v(k) \in [0, 1]^M$.

Korollar 9.2: *Aus dem Beweis der Proposition 9.4 sowie dem Beweis der Proposition 9.3 folgt unmittelbar, dass*

$$\underline{\mathbf{L}}(w(k), v(k), :, :) \cdot \mathbf{p}_z(k) = (\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k)) \in [0, 1]^N \quad (9.23)$$

gilt, wenn die Ein- und Ausgabewahrscheinlichkeitsvektoren exakt und durch $\mathbf{p}_w(k) \in \{0, 1\}^Q$ und $\mathbf{p}_v(k) \in \{0, 1\}^M$ gegeben sind.

Ein direkter Beweis des Korollars 9.2 findet sich auch in [91]. Aus Lemma 9.3 und Korollar 9.2 lassen sich analog zur CP-Beobachtergleichung zwei grundlegende Aussagen formulieren:

1. Lemma 9.3 führt für eine exakte TT-Dekomposition des permutierten Verhaltenstensors und unsichere Ein- und Ausgaben auf die gleichen Zustandswahrscheinlichkeiten wie die erweiterte Beobachtergleichung (9.20), wobei die Zustandswahrscheinlichkeitsverteilungen durch (7.14) gegeben sind.
2. Aus dem Korollar 9.2 folgt, dass Lemma 9.3 für eine exakte TT-Dekomposition und exakt bekannte Ein- und Ausgaben die gleichen Zustandswahrscheinlichkeiten wie die Beobachtergleichung (9.17) generiert. Die Zustandswahrscheinlichkeitsverteilungen sind dann wiederum durch (3.54) gegeben.

Dabei gilt auch für die TT-Dekomposition, dass wenn diese nur eine Näherung des permutierten Verhaltenstensors $\hat{\underline{\mathbf{L}}} \in \mathbb{R}^{Q \times M \times N \times N}$ mit den Elementen $\hat{\underline{\mathbf{L}}}(w, v, \bar{z}, z)$ darstellt, dann wird auch der Vektor der Zustandswahrscheinlichkeitsverteilungen nur eine Approximation $\hat{\mathbf{p}}_z(k+1) \in \mathbb{R}^N$ repräsentieren. Die Struktur der TT-Beobachtergleichung lässt sich ebenfalls visualisieren, was in Abbildung 9.7 auf Seite 169 für die Gleichung (9.21) gezeigt ist.

9.2.2 TT-Algorithmus zur qualitativen Fehlererkennung*

Der Algorithmus 9.2 zur qualitativen Fehlerdetektion ist hier für den Fall eines approximierten Verhaltenstensors notiert. Es wird also angenommen, dass der Verhaltenstensor des qualitativen Modells $\tilde{\underline{\mathbf{L}}} \in [0, 1]^{Q \times M \times N \times N}$ durch eine approximierte TT-Repräsentation $\langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle = \hat{\underline{\mathbf{L}}} \in \mathbb{R}^{Q \times M \times N \times N}$ nach Definition 8.10 gegeben ist. Bei der TT-Dekomposition muss im approximativen Fall jedoch berücksichtigt werden, dass die Elemente der TT-Kerne auch negative Elemente enthalten können, was in (9.22) zu einem Vektor $\hat{\mathbf{p}}_z(k+1) \in \mathbb{R}^N$ mit möglicherweise negativen Komponenten führt. Auch darf in diesem Fall die Normierung des Vektors nicht direkt durch die Summennorm $\|\cdot\|_1$ erfolgen, da diese die *Beträge* der Vektorkomponenten aufsummiert, was zu einer falschen Normalisierung führt. Es wird somit ein Zwischenschritt

$$\boldsymbol{\Omega}(k) = (\mathbf{E} \bar{\times}_{(2)} \tilde{\mathbf{p}}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \tilde{\mathbf{p}}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \hat{\mathbf{p}}_z(k)) \in \mathbb{R}^N \quad (9.24)$$

eingeführt, bei dem der Vektor $\mathbf{\Omega}(k)$ mit den Komponenten $\Omega_i(k)$ für $i = 1, \dots, N$, auf einen Vektor $\mathbf{\Psi}(k) \in \mathbb{R}_{\geq 0}^N$ mit Komponenten

$$\Psi_i(k) = \begin{cases} 0 & \text{wenn } \Omega_i(k) < 0, \\ \Omega_i(k) & \text{sonst,} \end{cases} \quad (9.25)$$

abgebildet wird. Der Vektor der approximierten Zustandswahrscheinlichkeiten ist dann definiert durch

$$\hat{\mathbf{p}}_z(k+1) = \frac{\mathbf{\Psi}(k)}{\|\mathbf{\Psi}(k)\|_1} \in [0, 1]^N. \quad (9.26)$$

Die Funktionalität des TT-Algorithmus 9.2 zur qualitativen Fehlererkennung wird in Beispiel 9.3 wieder anhand des Erhitzers der realen RLT-Anlage aus Abschnitt 5.3 demonstriert.

Algorithmus 9.2: TT-Algorithmus zur qualitativen Fehlererkennung

Gegeben: Qualitatives Modell $\tilde{\mathcal{A}}$, Gl. (4.15) mit Verhaltenstensor in

TT-Repräsentation $\hat{\underline{\mathbf{L}}} = \langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle$, Gl. (8.22)

Folge von Wahrscheinlichkeitsvektoren qualitativer Eingänge $\tilde{\mathbf{P}}_v(0 \dots T)$, Gl. (7.31)

Folge von Wahrscheinlichkeitsvektoren qualitativer Ausgänge $\tilde{\mathbf{P}}_w(0 \dots T)$, Gl. (7.32)

Initialisiere: $k = 0$, $\hat{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$

Iteriere:

- 1 **solange** $0 \leq k \leq T$
- 2 **berechne** $\mathbf{\Omega}(k)$, Gl. (9.24)
- 3 **berechne** $\mathbf{\Psi}(k)$, Gl. (9.25)
- 4 **berechne** $\|\mathbf{\Psi}(k)\|_1 =: \hat{b}(k)$, Gl. (9.26)
- 5 **berechne** $\hat{b}_m(k)$, Gl. (4.46) und $\hat{b}_b(k)$, Gl. (4.48)
- 6 **wenn** $\hat{b}(k) = 0$ **dann**
- 7 **setze** $\hat{\mathbf{p}}_z(k+1) = \tilde{\mathbf{p}}_0$
- 8 **setze** $k := k+1$ und fahre fort mit Schritt 1
- 9 **sonst**
- 10 **berechne** $\hat{\mathbf{p}}_z(k+1) = \mathbf{\Psi}(k)/\|\mathbf{\Psi}(k)\|_1$, Gl. (9.26)
- 11 **setze** $k := k+1$ und fahre fort mit Schritt 1

Ergebnis: Folge $\hat{\mathbf{P}}_z(0 \dots T+1) \mid \tilde{\mathbf{P}}_v(0 \dots T), \tilde{\mathbf{P}}_w(0 \dots T)$, Gl. (7.36)

Folge $\hat{\hat{B}}(0 \dots T \mid \tilde{\mathbf{P}}_v(0 \dots T), \tilde{\mathbf{P}}_w(0 \dots T))$, Gl. (7.38)

Folge $\hat{\hat{B}}_m(0 \dots T)$, Gl. (4.47)

Folge $\hat{\hat{B}}_b(0 \dots T)$, Gl. (4.49)

Beispiel 9.3 (Fehlererkennung mit der TT-Repräsentation): Dieses Beispiel zeigt die qualitative Beobachtung und Fehlererkennung anhand der TT-Repräsentation des Verhaltenstensors eines qualitativen Modells des Erhitzers aus Abschnitt 5.3 mit den Modellparametern aus Tabelle 5.11. Die Elemente des entsprechenden Verhaltenstensors wurden dabei nach Gleichung (8.18) umgeordnet, sodass hier anstelle des Verhaltenstensors (9.15) der permutierte Verhaltenstensors $\tilde{\underline{\mathbf{L}}} \in [0, 1]^{36 \times 36 \times 36 \times 36}$ verwendet wird. Die TT-Approximation

$$\hat{\underline{\mathbf{L}}} = \langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle \in \mathbb{R}^{36 \times 36 \times 36 \times 36} \quad (9.27)$$

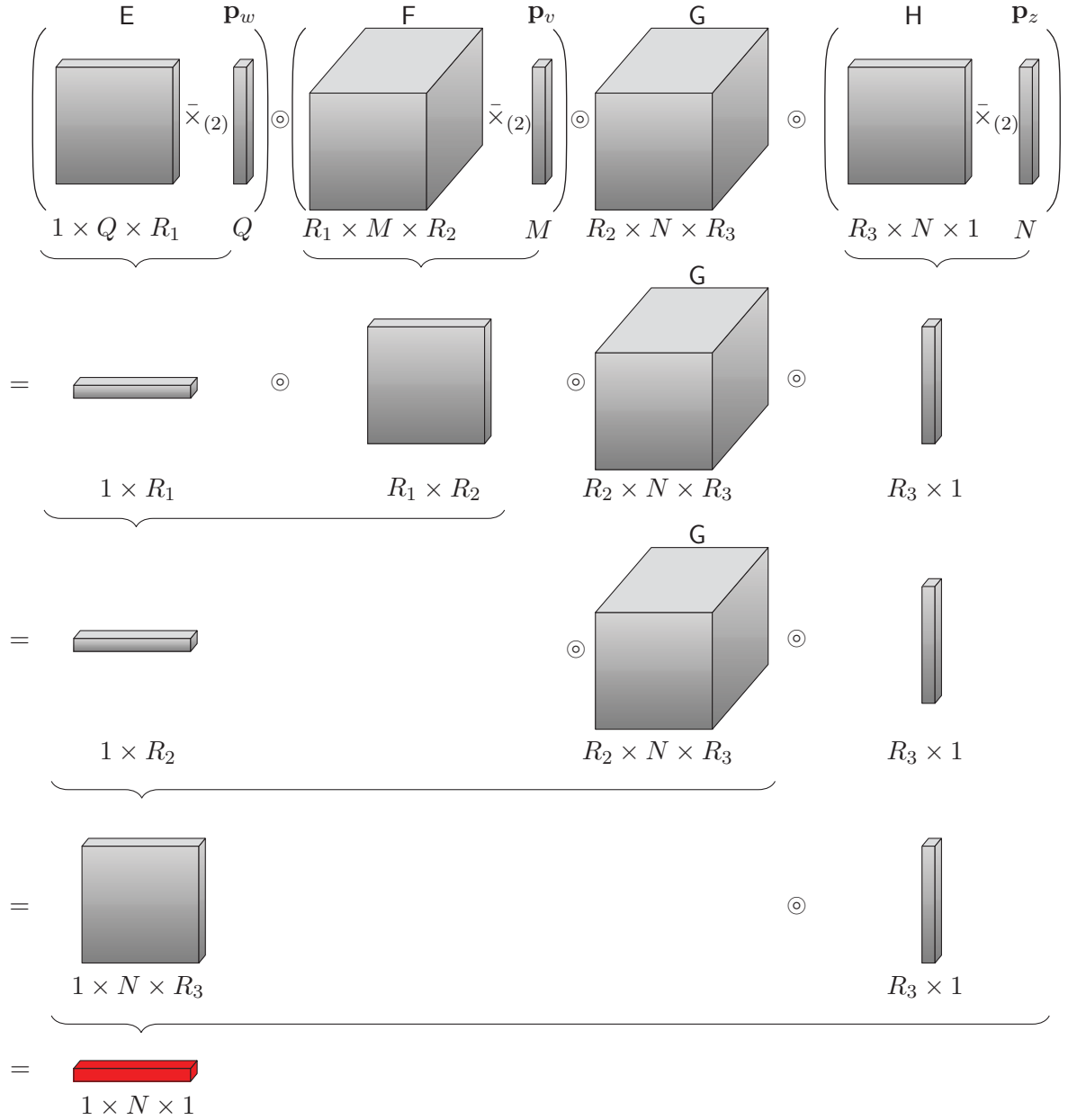


Abbildung 9.7: Grafische Interpretation der TT-Beobachtergleichung

des Verhaltenstensors nach Definition 8.10 wurde dabei mit einer Approximationsgüte von $\varepsilon = 0.27$ berechnet und besitzt die TT-Kerne

$$\mathbf{E} \in \mathbb{R}^{1 \times 36 \times 27}, \mathbf{F} \in \mathbb{R}^{27 \times 36 \times 67}, \mathbf{G} \in \mathbb{R}^{67 \times 36 \times 25}, \mathbf{H} \in \mathbb{R}^{25 \times 36 \times 1}.$$

Es gilt also für die TT-Ränge $R_1 = 27, R_2 = 67$ und $R_3 = 25$, wobei die Berechnung der TT-Dekomposition mit der MATLAB[®] Toolbox [100] und einer Approximationsgüte $\varepsilon = 0.27$ realisiert wurde. Die Wahl der Approximationsgüte erfolgte auch hier als die kleinstmögliche, mit der sich näherungsweise das gleiche Fehlererkennungsverhalten wie ohne Dekomposition abbilden ließ. Basierend auf der TT-Repräsentation des Verhaltenstensors kann nun der Algorithmus 9.2 zur qualitativen Fehlerdetektion auf die Messdaten des Erhitzers angewendet werden. Die Abbildung 9.8 zeigt auf der rechten Seite das Ergebnis ohne Dekomposition und auf der linken Seite das des TT-Algorithmus für die Wasseraustrittstemperatur T_{WA} .

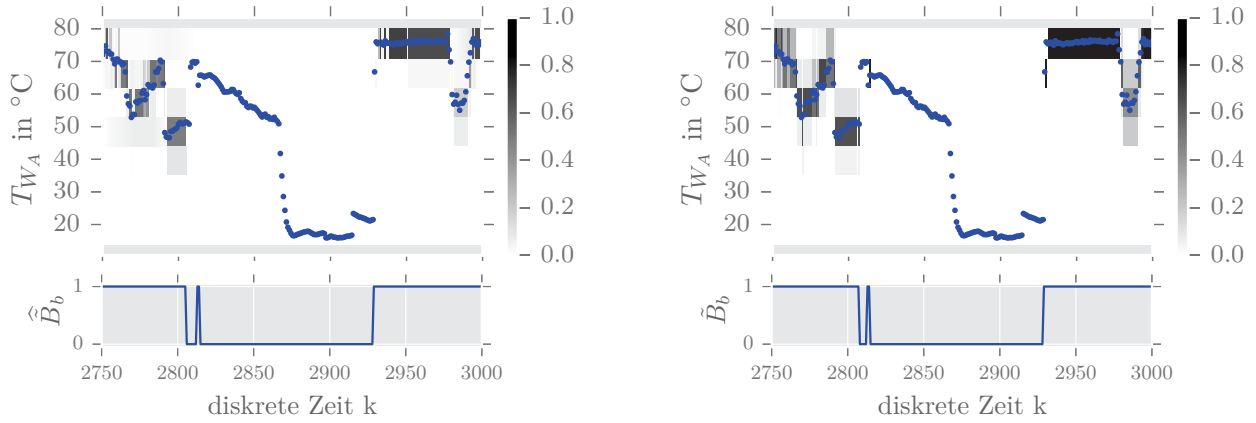


Abbildung 9.8: Vergleich der qualitativen Zustandswahrscheinlichkeiten und der Verhaltenssignale mit Dekomposition \hat{B}_b (links) und ohne Dekomposition \tilde{B}_b (rechts) ($\tau = 1\%$, $\phi = 2$)

Wie die Abbildung 9.8 zeigt, approximiert die TT-Dekomposition den Verhaltenstensor mit hoher Genauigkeit. Dies ist daran zu erkennen, dass innerhalb des nicht nominalen Zeitbereichs keine Zustandswahrscheinlichkeiten existieren. Die Werte des Verhaltenssignals sind hier tatsächlich Null, was einen Vorteil gegenüber der CP-Dekomposition im Anwendungsbeispiel 9.2 darstellt. Wie die Tabelle 9.4 zeigt, benötigt die TT-Dekomposition mit 127 296 Werten jedoch deutlich mehr Speicher als die CP-Dekomposition.

Tabelle 9.4: Speicherbedarf des Verhaltenstensors des qualitativen Erhitzermodells ohne Dekomposition, für die Rang-50 Approximation der CP-Repräsentation und die TT-Approximation mit $\varepsilon = 0.27$

Repräsentation des Verhaltenstensors	Fehlererk. Algorithmus	Speicher- bedarf	Reduktions- faktor	RMSD \tilde{B} zu \hat{B}
$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 36 \times 36 \times 36}$, Gl. (9.15)	7.1	$\gamma = 1\,679\,616$	0	0
$\hat{\mathbf{L}} = [\mathbf{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]$, Gl. (9.16)	9.1	$\gamma_{\text{CP}} = 7250$	$\xi_{\text{CP}} = 231$	0.179
$\hat{\underline{\mathbf{L}}} = \langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle$, Gl. (9.27)	9.2	$\gamma_{\text{TT}} = 127\,296$	$\xi_{\text{TT}} = 13$	0.096

10

Beobchateralgorithmen für die erweiterte CP-Repräsentation

Dieses Kapitel beschreibt die Beobachtung und Fehlererkennung mit qualitativen Modellen, deren Ein-, Ausgänge und Zustände komponentenweise quantisiert sind. Abschnitt 10.1 bildet mit der formalen Beschreibung eines stochastischen Automaten mit vektoriellen Ein- und Ausgaben die Grundlage für ein entsprechendes qualitatives Modell mit komponentenweise quantisierten Ein- und Ausgängen, welches in Abschnitt 10.2 eingeführt wird. Abschnitt 10.3 beschreibt einen autonomen stochastischen Automaten mit vektoriellen Zuständen. Ein stochastischer Automat mit vektoriellen Eingaben, Ausgaben und Zuständen wird in Abschnitt 10.4 eingeführt, der dann wiederum die Basis für ein qualitatives Modell bildet, bei dem alle Systemgrößen komponentenweise quantisiert sind, was in Abschnitt 10.5 beschrieben wird. Das allgemeine Ziel dieses Kapitels ist dabei die weitere Reduktion des Speicherbedarfs stochastischer Automaten und qualitativer Modelle.

10.1 Stochastischer Automat mit vektoriellen Ein- und Ausgaben

In diesem Abschnitt wird beschrieben, wie ein stochastischer Automat \mathcal{A} zur Beobachtung eines stochastischen Prozesses mit vektoriellen Ein- und Ausgaben verwendet werden kann. Die folgenden Erläuterungen bilden damit die automatentheoretische Grundlage für ein qualitatives Modell, welches ein quantisiertes System mit *komponentenweise* quantisierten Ein- und Ausgängen beschreiben soll.

10.1.1 Stochastischer Prozess

Ein stochastischer Prozess mit vektoriellen Ein- und Ausgabefolgen ist dadurch definiert, dass er nicht nur eine einzige Folge von Ein und Ausgaben besitzt, sondern eine Anzahl von m Eingabefolgen und q Ausgabefolgen (s. Abbildung 10.1).

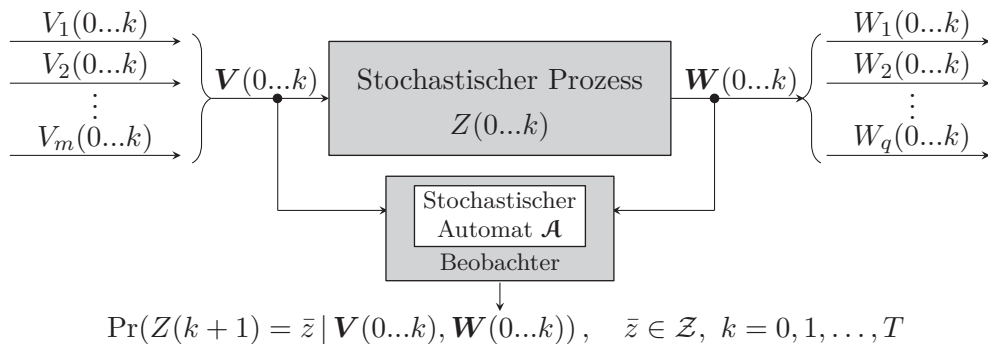


Abbildung 10.1: Stochastischer Prozess mit vektoriellen Ein- und Ausgabefolgen

In Abbildung 10.1 besitzt jede Eingabefolge $V_t(0...T)$ die Elemente $v_t(k) \in \mathcal{V}_t := \{1, 2, \dots, M_t\}$ für Indizes $t = 1, 2, \dots, m$ und jede Ausgabefolge $W_j(0...T)$ die Elemente $w_j(k) \in \mathcal{W}_j := \{1, 2, \dots, Q_j\}$

für $j = 1, 2, \dots, q$. Die Eingaben und Ausgaben können damit zu *Ein- und Ausgabevektoren*

$$\mathbf{v}(k) = \begin{pmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_m(k) \end{pmatrix} \in \mathcal{V}, \quad \mathbf{w}(k) = \begin{pmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_q(k) \end{pmatrix} \in \mathcal{W} \quad (10.1)$$

zusammengefasst werden, wobei

$$\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2 \times \dots \times \mathcal{V}_m \subseteq \mathbb{N}^m \quad (10.2)$$

die Menge aller möglichen Eingabetupel und

$$\mathcal{W} = \mathcal{W}_1 \times \mathcal{W}_2 \times \dots \times \mathcal{W}_q \subseteq \mathbb{N}^q \quad (10.3)$$

die Menge aller Ausgabetupel repräsentiert. Wie in der Abbildung 10.1 dargestellt, können die Vektoren der Ein- und Ausgaben (10.1) für $k = 0, 1, 2, \dots, T$ durch die Vektorfolgen

$$\mathbf{V}(0..T) = (\mathbf{v}(0), \mathbf{v}(1), \mathbf{v}(2), \dots, \mathbf{v}(T)) \quad (10.4)$$

und

$$\mathbf{W}(0..T) = (\mathbf{w}(0), \mathbf{w}(1), \mathbf{w}(2), \dots, \mathbf{w}(T)) \quad (10.5)$$

repräsentiert werden. Damit kann nun ein stochastischer Automat definiert werden, der den stochastischen Prozess in Abbildung 10.1 beschreibt. Da der stochastische Prozess über vektorielle E/A Folgen verfügt, wird der entsprechende Automat im Folgenden als *initialer stochastischer Automat mit vektoriellen Ein- und Ausgaben bezeichnet*.

10.1.2 Stochastischer Automat

Definition 10.1 (Initialer stochastischer Automat mit vektoriellen Ein- und Ausgaben): *Das 5-Tupel*

$$\mathcal{A} = (\mathcal{Z}, \mathcal{V}, \mathcal{W}, \mathcal{L}, \mathbf{p}_0) , \quad (10.6)$$

mit der Eingabemenge \mathcal{V} , der Ausgabemenge \mathcal{W} , der Zustandsmenge \mathcal{Z} sowie der Verhaltensrelation \mathcal{L} und dem initialen Wahrscheinlichkeitsvektor der Zustände \mathbf{p}_0 , heißt *initialer stochastischer Automat mit vektoriellen Ein- und Ausgaben*.

In (10.6) ist \mathcal{L} wieder eine Funktion, welche die bedingte Wahrscheinlichkeitsverteilung

$$\mathcal{L} : \mathcal{Z} \times \mathcal{W} \times \mathcal{Z} \times \mathcal{V} \longrightarrow [0, 1]$$

repräsentiert und analog zu (3.34), als Verhaltensrelation des stochastischen Automaten \mathcal{A} bezeichnet wird. Diese Verhaltensrelation ist durch

$$\mathcal{L}(\bar{z}, \mathbf{w}, z, \mathbf{v}) = \Pr \left(\begin{array}{c} Z(k+1) = \bar{z}, \\ \mathbf{W}(k) = \mathbf{w} \end{array} \middle| \begin{array}{c} Z(k) = z, \\ \mathbf{V}(k) = \mathbf{v} \end{array} \right) \quad (10.7)$$

gegeben und beschreibt die bedingte Wahrscheinlichkeit dafür, dass der Automat bei der vektoriellen Eingabe \mathbf{v} vom Zustand z in den Folgezustand \bar{z} wechselt und die vektorielle Ausgabe \mathbf{w} erzeugt. Da die Verhaltensrelation (10.7) eine bedingte Wahrscheinlichkeit repräsentiert, wird zur Verdeutlichung dieser Tatsache die Notation $\mathcal{L}(\bar{z}, \mathbf{w} | z, \mathbf{v})$ anstelle von $\mathcal{L}(\bar{z}, \mathbf{w}, z, \mathbf{v})$ verwendet. Dabei ist zu beachten, dass (10.7) eine Kurznotation der expliziten Schreibweise

$$\begin{aligned} & \mathcal{L}(\bar{z}, w_1, w_2, \dots, w_q | z, v_1, v_2, \dots, v_m) \\ &= \Pr \left(\begin{array}{c} Z(k+1) = \bar{z}, \\ W_1(k) = w_1, \dots, W_q(k) = w_q \end{array} \middle| \begin{array}{c} Z(k) = z, \\ V_1(k) = v_1, \dots, V_m(k) = v_m \end{array} \right) \end{aligned} \quad (10.8)$$

darstellt.

10.1.3 Tensorrepräsentation

Die bedingten Übergangswahrscheinlichkeiten der Verhaltensrelation $\mathcal{L}(\bar{z}, \mathbf{w} \mid z, \mathbf{v})$ des stochastischen Automaten (10.6) können nun wiederum in einem Verhaltenstensor angeordnet werden. Dabei wird das gleiche Symbol \mathbf{L} verwendet wie in Abschnitt 7.1.1, da im Folgenden stets aus dem Kontext ersichtlich sein wird, um welche Repräsentation es sich handelt. Der Verhaltenstensor ist damit durch den Tensor

$$\mathbf{L} \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m} \quad (10.9)$$

mit den Elementen

$$l(\bar{z}, w_1, w_2, \dots, w_q, z, v_1, v_2, \dots, v_m) = \mathcal{L}(\bar{z}, w_1, w_2, \dots, w_q \mid z, v_1, v_2, \dots, v_m) \quad (10.10)$$

definiert. Im Folgenden wird, wenn keine explizite Indexierung der Tensorelemente nötig ist, auch hier die Kurznotation $l(\bar{z}, \mathbf{w}, z, \mathbf{v})$ verwendet. Die CP-Repräsentation des Verhaltenstensors (10.9) ist dabei durch

$$\mathbf{L} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_q, \mathbf{C}, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_m \rrbracket \quad (10.11)$$

gegeben, wobei angenommen wird, dass es sich um eine exakte Faktorisierung handelt. Dabei bezeichnet $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}$ die Faktormatrix der Folgezustände und $\mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}$ die Faktormatrix der Zustände. Die Faktormatrizen der Ausgaben sind durch $\mathbf{B}_j \in \mathbb{R}_{\geq 0}^{Q_j \times R}$ für Indizes $j = 1, \dots, q$ und die Faktormatrizen der Eingaben durch $\mathbf{D}_i \in \mathbb{R}_{\geq 0}^{M_i \times R}$ für $i = 1, \dots, m$ gegeben. Auf Grundlage des Verhaltenstensors (10.9) lassen sich nun zunächst die Gleichungen zur Zustandsbeobachtung aufstellen.

10.1.4 Allgemeine Beobachtung

Eine einfache Beobachtergleichung auf Basis des Verhaltenstensors (10.9) kann analog zu Definition 7.2 aufgestellt werden:

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\mathbf{L}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \mathbf{p}_z(k)}{\|\mathbf{L}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \mathbf{p}_z(k)\|_1} \in [0, 1]^N, \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0. \end{aligned} \quad (10.12)$$

Der einzige Unterschied zur Beobachtergleichung (7.6) besteht somit darin, dass die Eingaben und Ausgaben nun durch die Vektoren $\mathbf{v}(k)$ und $\mathbf{w}(k)$ gegeben sind. Die in der Abbildung 10.1 eingezeichneten Zustandswahrscheinlichkeiten zur Zeit $k+1$ sind dann für alle $\bar{z} \in \mathcal{Z}$ durch den Vektor

$$\mathbf{p}_z(k+1) = \begin{pmatrix} p_z(k+1, 1) \\ p_z(k+1, 2) \\ \vdots \\ p_z(k+1, N) \end{pmatrix} := \begin{pmatrix} \Pr(Z(k+1) = 1 \mid \mathbf{V}(0...k), \mathbf{W}(0...k)) \\ \Pr(Z(k+1) = 2 \mid \mathbf{V}(0...k), \mathbf{W}(0...k)) \\ \vdots \\ \Pr(Z(k+1) = N \mid \mathbf{V}(0...k), \mathbf{W}(0...k)) \end{pmatrix} \in [0, 1]^N \quad (10.13)$$

gegeben. Wendet man die Beobachtergleichung entsprechend auf gegebene vektorielle E/A Folgen $(\mathbf{V}(0...T), \mathbf{W}(0...T))$ an, erhält man die Folge

$$\mathbf{p}_z(0...T+1 \mid \mathbf{V}(0...T), \mathbf{W}(0...T)) = (\mathbf{p}_z(0), \mathbf{p}_z(1), \mathbf{p}_z(2), \dots, \mathbf{p}_z(T+1)) \quad (10.14)$$

von Zustandswahrscheinlichkeitsvektoren.

Zur Berücksichtigung unsicherer Ein- und Ausgaben kann wie in Abschnitt 7.1.3 eine erweiterte Beobachtergleichung auf Grundlage des kontrahierten Tensorprodukts (6.36) definiert werden. Dabei treten für $t = 1, 2, \dots, m$ und $j = 1, 2, \dots, q$ anstelle der skalaren Eingabefolgen $V_t(0 \dots T)$ und Ausgabefolgen $W_j(0 \dots T)$ die Folgen von Wahrscheinlichkeitsvektoren

$$\begin{aligned} \mathbf{P}_v^{(t)}(0 \dots T) &= (\mathbf{p}_v^{(t)}(0), \mathbf{p}_v^{(t)}(1), \mathbf{p}_v^{(t)}(2), \dots, \mathbf{p}_v^{(t)}(T)) , \\ \mathbf{P}_w^{(j)}(0 \dots T) &= (\mathbf{p}_w^{(j)}(0), \mathbf{p}_w^{(j)}(1), \mathbf{p}_w^{(j)}(2), \dots, \mathbf{p}_w^{(j)}(T)) \end{aligned}$$

deren Elemente für $k = 0, 1, 2, \dots, T$ durch

$$\mathbf{p}_v^{(t)}(k) = \begin{pmatrix} \Pr(V_t(k) = 1) \\ \Pr(V_t(k) = 2) \\ \vdots \\ \Pr(V_t(k) = M_t) \end{pmatrix} \in [0, 1]^{M_t}, \quad \mathbf{p}_w^{(j)}(k) = \begin{pmatrix} \Pr(W_j(k) = 1) \\ \Pr(W_j(k) = 2) \\ \vdots \\ \Pr(W_j(k) = Q_j) \end{pmatrix} \in [0, 1]^{Q_j} \quad (10.15)$$

gegen sind. Bei der folgenden Verwendung der Wahrscheinlichkeitsvektoren der Eingaben und Ausgaben (10.15) sei dabei die Annahme impliziert, dass die Zufallsvariablen der Eingaben $V_1(k), \dots, V_m(k)$ und der Ausgaben $W_1(k), \dots, W_q(k)$ jeweils voneinander *stochastisch unabhängig* sind (s. hierzu Definition 10.2).

Definition 10.2 (Stochastische Unabhängigkeit): Zufallsvariablen $Z_1(k), \dots, Z_n(k)$ mit Werten in $\mathcal{Z}_1 = \{1, \dots, N_1\}, \dots, \mathcal{Z}_n = \{1, \dots, N_n\}$ heißen *stochastisch unabhängig*, wenn für alle $z_1 \in \mathcal{Z}_1$ bis $z_n \in \mathcal{Z}_n$ zum Zeitpunkt k folgendes gilt:

$$\Pr(Z_1(k) = z_1, \dots, Z_n(k) = z_n) = \Pr(Z_1(k) = z_1) \cdots \Pr(Z_n(k) = z_n) . \quad (10.16)$$

Die Annahme der stochastischen Unabhängigkeit ist beispielsweise dann erfüllt, wenn die Ein- und Ausgaben exakt bekannt und die Wahrscheinlichkeitsvektoren (10.15) auf der Menge $\{0, 1\}$ definiert sind und somit nur ein Einselement besitzen, während alle anderen Komponenten Null sind (s. Abschnitt 7.1.3, S. 130). Für solche wieder als *exakt* bezeichneten Ein- und Ausgabewahrscheinlichkeitsvektoren kann nun gezeigt werden, dass folgendes gilt:

Proposition 10.1: Für exakte Ein- und Ausgabewahrscheinlichkeitsvektoren gilt

$$\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w^{(1)}(k), \dots, \mathbf{p}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle = \mathbf{L}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \mathbf{p}_z(k) \quad (10.17)$$

mit $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w^{(j)}(k) \in \{0, 1\}^{Q_j}$ für $j = 1, 2, \dots, q$ sowie $\mathbf{p}_v^{(t)}(k) \in \{0, 1\}^{M_t}$ für $t = 1, 2, \dots, m$.

Beweis: Der Beweis der Proposition 10.1 findet sich im Anhang B.6.

Durch Einsetzen von (10.17) in (10.12) erhält man damit die erweiterte Beobachtergleichung

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z(k+1) &= \frac{\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w^{(1)}(k), \dots, \mathbf{p}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle}{\left\| \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w^{(1)}(k), \dots, \mathbf{p}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle \right\|_1} \in [0, 1]^N , \\ k = 0 : \quad \mathbf{p}_z(0) &= \mathbf{p}_0 \end{aligned} \quad (10.18)$$

die für gegebene Folgen von Wahrscheinlichkeitsvektoren der Eingaben $\mathbf{P}_v^{(t)}(0 \dots T)$, $t = 1, \dots, m$ und Ausgaben $\mathbf{P}_w^{(j)}(0 \dots T)$, $j = 1, \dots, q$ auf eine Folge von Zustandswahrscheinlichkeitsvektoren

$$\begin{aligned} &\mathbf{P}_z(0 \dots T+1 \mid \mathbf{P}_v^{(1)}(0 \dots T), \dots, \mathbf{P}_v^{(m)}(0 \dots T), \mathbf{P}_w^{(1)}(0 \dots T), \dots, \mathbf{P}_w^{(q)}(0 \dots T)) \\ &= (\mathbf{p}_z(0), \mathbf{p}_z(1), \mathbf{p}_z(2), \dots, \mathbf{p}_z(T+1)) \end{aligned} \quad (10.19)$$

führt. Unsicherheiten in den Eingaben und Ausgaben können dann wie bereits in Abschnitt 7.1.3 auf Seite 132 beschrieben berücksichtigt werden, indem man die Ein- und Ausgabewahrscheinlichkeitsvektoren entsprechend auf dem Einheitsintervall $[0, 1]$ definiert. Für exakte Ein- und Ausgabefolgen erhält man mit Gleichung (10.18) aufgrund von Proposition 10.1 die gleichen Ergebnisse wie mit Gleichung (10.12), das heißt, für die Folge (10.19) der Zustandswahrscheinlichkeitsvektoren gilt

$$\begin{aligned} & \mathbf{P}_z(0 \dots T + 1 \mid \mathbf{P}_v^{(1)}(0 \dots T), \dots, \mathbf{P}_v^{(m)}(0 \dots T), \mathbf{P}_w^{(1)}(0 \dots T), \dots, \mathbf{P}_w^{(q)}(0 \dots T)) \\ &= \mathbf{P}_z(0 \dots T + 1 \mid \mathbf{V}(0 \dots T), \mathbf{W}(0 \dots T)) . \end{aligned} \quad (10.20)$$

10.1.5 Beobachtung mit der CP-Repräsentation

Für die Beobachtung mit der CP-Repräsentation des Verhaltenstensors (10.11) gilt die folgende Proposition.

Proposition 10.2: *Für eine exakte CP-Dekomposition des Verhaltenstensors (10.11) gilt für das in der erweiterten Beobachtergleichung (10.18) auftretende kontrahierte Tensorprodukt*

$$\begin{aligned} & \left\langle \mathbf{L} \mid \llbracket \mathbf{P}_w^{(1)}(k), \dots, \mathbf{P}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle \\ &= \mathbf{A} \cdot \left(\lambda \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right) \end{aligned} \quad (10.21)$$

mit $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w^{(j)}(k) \in [0, 1]^{Q_j}$ für $j = 1, 2, \dots, q$ sowie $\mathbf{p}_v^{(t)}(k) \in [0, 1]^{M_t}$ für $t = 1, 2, \dots, m$.

Beweis: Der Beweis von Proposition 10.2 befindet sich im Anhang B.7.

Lemma 10.1 (CP-Beobachtergleichung für stochastische Automaten mit vektoriellen Ein- und Ausgaben): *Durch Einsetzen von Gleichung (10.21) in den Zähler und Nenner von Gleichung (7.6) folgt mit*

$$\begin{aligned} k \geq 0: \quad \mathbf{p}_z(k+1) &= \frac{\mathbf{A} \cdot \left(\lambda \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right)}{\left\| \mathbf{A} \cdot \left(\lambda \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right\|_1} \in [0, 1]^N, \\ k = 0: \quad \mathbf{p}_z(0) &= \mathbf{p}_0 \end{aligned} \quad (10.22)$$

die CP-Beobachtergleichung für stochastische Automaten mit vektoriellen Ein- und Ausgaben nach Definition 10.1. Dabei ist $\mathbf{p}_z(k) \in [0, 1]^N$, $\mathbf{p}_w^{(j)}(k) \in [0, 1]^{Q_j}$ für $j = 1, 2, \dots, q$ sowie $\mathbf{p}_v^{(t)}(k) \in [0, 1]^{M_t}$ für $t = 1, 2, \dots, m$.

Korollar 10.1: *Aus den Beweisen der Proposition 10.1 und dem Beweis der Proposition 10.2 folgt unmittelbar, dass*

$$\mathbf{A} \cdot \left(\lambda \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right) = \mathbf{L}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \mathbf{p}_z(k) \quad (10.23)$$

gilt, wenn die Ein- und Ausgabewahrscheinlichkeitsvektoren exakt und durch $\mathbf{p}_w^{(j)}(k) \in \{0, 1\}^{Q_j}$ und $\mathbf{p}_v^{(t)}(k) \in \{0, 1\}^{M_t}$ für $t = 1, 2, \dots, m$ und $j = 1, 2, \dots, q$ gegeben sind.

Die Erkenntnisse aus Lemma 10.1 und dem Korollar 10.1 lassen sich in den folgenden beiden Aussagen zusammenfassen:

1. Mit Lemma 10.1 erhält man für eine exakte CP-Dekomposition des Verhaltenstensors (10.11) und unsichere Folgen von Ein- und Ausgaben die gleiche Folge von Zustandswahrscheinlichkeitsvektoren (10.19) wie mit der erweiterten Beobachtergleichung (10.18).
2. Mit dem Korollar 10.1 ist gezeigt, dass Lemma 10.1 für eine exakte CP-Dekomposition sowie exakt bekannte Folgen von Ein- und Ausgaben zur gleichen Folge von Zustandswahrscheinlichkeitsvektoren (10.14) führt wie die einfache Beobachtergleichung (10.12).

10.2 Qualitatives Modell für komponentenweise quantisierte Ein- und Ausgänge

In diesem Abschnitt wird das qualitative Modell für komponentenweise quantisierte Ein- und Ausgänge eingeführt, welches auf einem stochastischen Automaten nach Definition 10.1 basiert. Die folgende Tabelle 10.1 zeigt dabei den Speicherbedarf γ_{CP} für eine Rang-50 Approximation des Verhaltenstensors $\tilde{\mathbf{L}} \in [0, 1]^{36 \times 36 \times 36 \times 36}$ des qualitativen Erhitzermodells aus Beispiel 9.2 und dem eines qualitativen Modells mit komponentenweise quantisierten Ein- und Ausgängen $\tilde{\mathbf{L}} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$.

Tabelle 10.1: Speicherbedarf γ und γ_{CP} für verschiedendimensionale Verhaltenstensenoren qualitativer Modelle des Erhitzers mit den Modellparametern aus Tabelle 5.11

Anzahl qualitativer Größen	Dimensionalität des Verhaltenstensors	Speicherbedarf γ	Speicherbedarf γ_{CP} ($R = 50$)
$N = Q = M = 36$	$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 36 \times 36 \times 36}$	1 679 616	7250
$N = 36, M_1 = M_2 = 6, Q_1 = Q_2 = 6$	$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$	1 679 616	4850

Wie man in Tabelle 10.1 erkennt, ist der Speicherbedarf für einen gleichen Dekompositionsrang von $R = 50$ für die beiden Verhaltenstensenoren unterschiedlich, obwohl diese in der vollen Repräsentation eine identische Anzahl von Werten als auch den gleichem Informationsgehalt besitzen. Welcher Zusammenhang zwischen den beiden Darstellungen besteht, wird im Folgenden erläutert.

10.2.1 Komponentenweise Quantisierung

Die komponentenweise Quantisierung wird im Folgenden wieder anhand des quantitativen Zustandsvektors $\mathbf{x} \in \mathbb{R}^n$ eines zeitdiskreten Prozesses wie in Abbildung 1.3 erläutert und gilt gleichermaßen für den Eingangsvektor $\mathbf{u} \in \mathbb{R}^m$ und den Ausgangsvektor $\mathbf{y} \in \mathbb{R}^q$.

Als *komponentenweise Signalquantisierung* wird eine Methode bezeichnet, welche alle Komponenten $x_i \in \mathbb{R}$ eines reellwertigen Zustandsvektors $\mathbf{x} \in \mathbb{R}^n$ auf einzelne qualitative Werte $[x_i]$ abbildet. Die komponentenweise Quantisierung des Zustands kann durch die Funktionen

$$E_{x_i} : \mathbb{R} \rightarrow \mathcal{X}_i \quad (10.24)$$

beschrieben werden. Dabei wird für alle $i = 1, 2, \dots, n$ der eindimensionale Zustandsraum \mathbb{R} auf den qualitativen Zustandsraum $\mathcal{X}_i = \{1, 2, \dots, N_i\}$ und jeder reellwertige Zustand $x_i \in \mathbb{R}$ auf seinen quantisierten Wert $[x_i] \in \mathcal{X}_i$ abgebildet:

$$x_i \in \mathbb{R} \xrightarrow{\text{Quantisierung}} [x_i] \in \mathcal{X}_i \quad (10.25)$$

Die Quantisierung erfolgt durch die Festlegung von Partitionierungen $\mathcal{E}_{x_i}(z_i)$, $z_i \in \mathcal{X}_i$, die ebenfalls wieder als beschränkt und disjunkt für alle $i = 1, 2, \dots, n$ angenommen werden. Der qualitative Wert des quantisierten Zustands $[x_i]$ wird von der Nummer $z_i \in \mathcal{X}_i$ bestimmt, die angibt, zu welcher Menge $\mathcal{E}_{x_i}(z_i)$ der quantitative Zustand x_i gehört:

$$[x_i] = z_i \iff x_i \in \mathcal{E}_{x_i}(z_i) \quad \forall i = 1, 2, \dots, n. \quad (10.26)$$

Die qualitativen Werte der quantisierten Zustände $[x_i]$ sind damit für $i = 1, \dots, n$ durch die *qualitativen Zustände* $z_i \in \mathcal{X}_i$ gegeben, was sich in einer vektoriellen Notation

$$[\mathbf{x}] = \begin{pmatrix} [x_1] \\ [x_2] \\ \vdots \\ [x_n] \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} = \mathbf{z} \in \mathcal{X} \quad (10.27)$$

zusammenfassen lässt. Der Vektor $\mathbf{z} \in \mathcal{X}$, dessen Komponenten durch die qualitativen Zustände $z_i \in \mathcal{X}_i$ gegeben sind, wird als *qualitativer Zustandsvektor* bezeichnet (s. Anmerkung 10.1). Die Menge \mathcal{X} stellt dabei das kartesische Produkt

$$\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_n \subseteq \mathbb{N}^n \quad (10.28)$$

der einzelnen qualitativen Zustandsräume \mathcal{X}_i dar. Der Umstand, dass hier der Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$ nicht auf einen einzigen qualitativen Zustand $z \in \mathcal{X}$, sondern auf einen qualitativen Zustandsvektor $\mathbf{z} \in \mathcal{X}$ abgebildet wird, wird durch die Verwendung fetter Klammern $[\cdot]$ gekennzeichnet. Der Ausdruck $[\mathbf{x}] = \mathbf{z} \in \mathcal{X}$ kennzeichnet somit einen komponentenweise quantisierten Zustandsvektor $\mathbf{x} \in \mathbb{R}^n$, während die Notation $[\mathbf{x}] = z \in \mathcal{X}$ die allgemeine Quantisierung ausdrückt (vgl. Gleichung (4.11)).

Anmerkung 10.1 (Qualitativer Zustandsvektor): *Genau genommen stellt die vektorielle Darstellung des qualitativen Zustandsvektors (10.27) einen leichten Missbrauch der Vektornotation dar. Der Grund dafür ist, dass die mit \mathcal{X} bezeichnete Menge aller möglichen qualitativen Zustandstupel keinen Vektorraum bildet, sondern nur eine Teilmenge $\mathbb{N}^n \subseteq \mathbb{R}^n$.*

Auch der quantitative Eingangsvektor $\mathbf{u} \in \mathbb{R}^m$ mit den reellwertigen Eingänge $u_j \in \mathbb{R}$, $j = 1, 2, \dots, m$ und der quantitative Ausgangsvektor $\mathbf{y} \in \mathbb{R}^q$ mit den Ausgängen $y_t \in \mathbb{R}$, $t = 1, 2, \dots, q$ werden komponentenweise durch die Partitionierung der qualitativen Eingangsräume $\mathcal{U}_j = \{1, 2, \dots, M_j\}$ und Ausgangsräume $\mathcal{Y}_t = \{1, 2, \dots, Q_t\}$ quantisiert. Dabei wird jeder reellwertige Eingang $u_j \in \mathbb{R}$ auf einen qualitativen Eingang $[u_j] = v_j \in \mathcal{U}_j$ und jeder Ausgang $y_t \in \mathbb{R}$ auf einen qualitativen Ausgang $[y_t] = w_t \in \mathcal{Y}_t$ abgebildet. Die Partitionierungen der einzelnen Ein- und Ausgänge werden mit $\mathcal{E}_{v_j}(v_j)$, $v_j \in \mathcal{U}_j$ und $\mathcal{E}_{w_t}(w_t)$, $w_t \in \mathcal{Y}_t$ bezeichnet, womit wieder folgender Zusammenhang gilt:

$$[u_j] = v_j \iff u_j \in \mathcal{E}_{u_j}(v_j) \quad \forall j = 1, 2, \dots, m, \quad (10.29)$$

$$[y_t] = w_t \iff y_t \in \mathcal{E}_{y_t}(w_t) \quad \forall t = 1, 2, \dots, q. \quad (10.30)$$

Die Beziehungen (10.29) und (10.30) können dann wieder in einer vektoriellen Repräsentation ausgedrückt werden:

$$[\mathbf{u}] = \begin{pmatrix} [u_1] \\ [u_2] \\ \vdots \\ [u_m] \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = \mathbf{v} \in \mathcal{U}, \quad [\mathbf{y}] = \begin{pmatrix} [y_1] \\ [y_2] \\ \vdots \\ [y_q] \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_q \end{pmatrix} = \mathbf{w} \in \mathcal{Y}. \quad (10.31)$$

Der Vektor $\mathbf{v} \in \mathcal{U}$ wird als *qualitativer Eingangsvektor* und der Vektor $\mathbf{w} \in \mathcal{Y}$ als *qualitativer Ausgangsvektor* bezeichnet, wobei entsprechend Anmerkung 10.1 gilt. Die Mengen \mathcal{U} und \mathcal{Y} sind wieder durch die kartesischen Produkte

$$\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \cdots \times \mathcal{U}_m \subseteq \mathbb{N}^m \quad (10.32)$$

und

$$\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_q \subseteq \mathbb{N}^q \quad (10.33)$$

gegeben. Werden die reellwertigen Eingangs-, Ausgangs- und Zustandsfolgen (4.1)–(4.3) nun mithilfe der Beziehungen (10.26), (10.29) und (10.30) komponentenweise quantisiert, dann ergibt sich eine vektorielle quantisierte Eingangsfolge

$$[\mathbf{U}(0 \dots T)] = ([\mathbf{u}(0)], [\mathbf{u}(1)], [\mathbf{u}(2)], \dots, [\mathbf{u}(T)]) , \quad (10.34)$$

eine vektorielle quantisierte Zustandsfolge

$$[\mathbf{X}(0 \dots T+1)] = ([\mathbf{x}(0)], [\mathbf{x}(1)], [\mathbf{x}(2)], \dots, [\mathbf{x}(T+1)]) \quad (10.35)$$

sowie eine vektorielle quantisierte Ausgangsfolge

$$[\mathbf{Y}(0 \dots T)] = ([\mathbf{y}(0)], [\mathbf{y}(1)], [\mathbf{y}(2)], \dots, [\mathbf{y}(T)]) . \quad (10.36)$$

Die komponentenweise Quantisierung wird abschließend anhand des folgenden Beispiels erläutert.

Beispiel 10.1 (Komponentenweise Quantisierung): Gegeben ist wieder ein Prozess mit den quantitativen Zuständen $x_1 \in \mathbb{R}$ und $x_2 \in \mathbb{R}$, die durch den Zustandsvektor $\mathbf{x} \in \mathbb{R}^2$ gegeben sind und deren Verlauf links in Abbildung 10.2 dargestellt ist (vgl. Abbildung 4.2). Das Signal $x_1 \in \mathbb{R}$ wird dabei auf

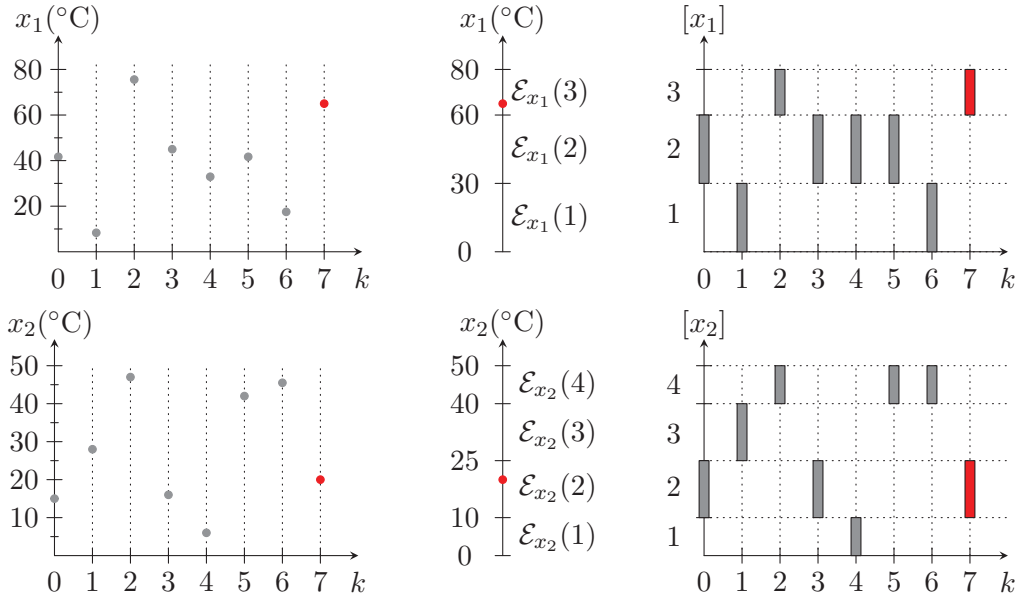


Abbildung 10.2: Komponentenweise Quantisierung eines Zustandssignals $\mathbf{x} \in \mathbb{R}^2$

einen qualitativen Zustandsraum $\mathcal{X}_1 = \{1, 2, 3\}$, also einen Zustandsraum mit $N_1 = 3$ qualitativen Zuständen $z_1 \in \mathcal{X}_1$ abgebildet. Oben in der Mitte der Abbildung ist eine mögliche Partitionierung des quantitativen Zustandsraumes \mathbb{R} durch die drei Bereiche $\mathcal{E}_{x_1}(z_1)$ für $z_1 = 1, 2, 3$ dargestellt. Für

den zur Zeit $k = 7$ in Rot eingezeichneten Zustand $x_1(7) = 65^\circ\text{C}$ ist offensichtlich, dass dieser zur Menge $\mathcal{E}_{x_1}(3)$ gehört und sein qualitativer Wert damit durch $[x_1(7)] = 3$ gegeben ist.

Ebenso erfolgt die Quantisierung des Zustandes $x_2 \in \mathbb{R}$, der durch die Partitionen $\mathcal{E}_{x_2}(1), \dots, \mathcal{E}_{x_2}(4)$ auf den qualitativen Zustandsraum $\mathcal{X}_2 = \{1, 2, 3, 4\}$ abgebildet wird und für dessen qualitative Zustände entsprechend $z_2 \in \mathcal{X}_2$ gilt. Der in Rot eingezeichnete Zustand $x_2(7) = 20^\circ\text{C}$ gehört hier zur Menge $\mathcal{E}_{x_2}(2)$ und besitzt den qualitativen Wert $[x_2(7)] = 2$. Die komponentenweise Quantisierung des Zustandsvektors $\mathbf{x}(7) \in \mathbb{R}^2$ ist damit wie folgt gegeben:

$$\mathbf{x}(7) = \begin{pmatrix} x_1(7) \\ x_2(7) \end{pmatrix} = \begin{pmatrix} 65 \\ 20 \end{pmatrix} \in \mathbb{R}^2 \xrightarrow[\text{Quantisierung}]{\text{komponentenweise}} [\mathbf{x}(7)] = \begin{pmatrix} [x_1(7)] \\ [x_2(7)] \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \in \mathcal{X}.$$

Die Menge \mathcal{X} ist nach (10.28) durch das kartesische Produkt

$$\mathcal{X}_1 \times \mathcal{X}_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4)\}$$

gegeben und enthält für $z_1 \in \mathcal{X}_1$ und $z_2 \in \mathcal{X}_2$ alle 12 möglichen qualitativen Zustandstupel (z_1, z_2) . Die komponentenweise Quantisierung führt damit für alle in der Abbildung eingezeichneten Zeitschritte auf eine quantisierte vektorielle Zustandsfolge

$$[\mathbf{X}(0\dots 7)] = \left(\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right).$$

Die komponentenweise Quantisierung der Zustände, Eingänge und Ausgänge kann dabei einfach in die allgemeine Quantisierung aus Abschnitt 4.2.2 überführt werden. Wie sich dies algorithmisch für beliebig dimensionale Eingangs-, Ausgangs-, und Zustandsräume verwirklichen lässt, wird im nächsten Abschnitt beschrieben.

Enumerationsgleichung und -algorithmus

Die folgende Erläuterung bezüglich der Enumerationsgleichung erfolgt wieder anhand der Zustände und gilt entsprechend für die Ein- und Ausgangsgrößen.

Der qualitative Zustand $z \in \mathcal{X} = \{1, 2, \dots, N\}$ bei der allgemeinen Quantisierung in Abschnitt 4.2.2 kann direkt aus den qualitativen Zuständen $z_i \in \mathcal{X}_i = \{1, 2, \dots, N_i\}$, $i = 1, 2, \dots, n$ der komponentenweisen Quantisierung bestimmt werden:

Definition 10.3 (Enumerationsgleichung ($z_i \in \mathcal{X}_i \longrightarrow z \in \mathcal{X}$)): Sind die qualitativen Zustände der komponentenweisen Quantisierung $z_i \in \mathcal{X}_i = \{1, 2, \dots, N_i\}$ für $i = 1, 2, \dots, n$ gegeben, dann ist

$$z = \overline{z_1 z_2 \dots z_n} = z_n + (z_{n-1} - 1)N_n + (z_{n-2} - 1)N_{n-1}N_n + \dots + (z_2 - 1)N_3 N_4 \dots N_n + (z_1 - 1)N_2 N_3 N_4 \dots N_n \in \mathcal{X} \quad (10.37)$$

der qualitative Zustand bei der allgemeinen Quantisierung. Dabei gilt entsprechend $\mathcal{X} = \{1, 2, \dots, N\}$ mit $N = N_1 N_2 \dots N_n$.

In Gleichung (10.37) ist $\overline{z_1 z_2 \dots z_n}$ der Multiindex aus Definition 6.4. Dabei werden hier jedoch nicht die Indizes eines mehrdimensionalen Tensors, sondern die qualitativen Zustände z_1, z_2, \dots, z_n auf den skalaren Wert $z \in \mathcal{X}$ abgebildet. Die Funktion des Multiindex ist jedoch in beiden Fällen identisch. Durch die Enumerationsgleichung ist der Zusammenhang zwischen den qualitativen Zuständen $z_i \in \mathcal{X}_i$ der komponentenweisen Quantisierung und dem einzelnen qualitativen Zustand $z \in \mathcal{X}$ der allgemeinen Quantisierung eindeutig definiert. Bei der Enumerationsgleichung (10.37) ist jedoch zu beachten, dass sich diese nicht ohne Weiteres umkehren lässt. Sie kann also nicht dazu verwendet

werden, um aus einem qualitativen Zustand $z \in \mathcal{X}$, die qualitativen Zustände $z_i \in \mathcal{X}_i, i = 1, \dots, n$ bei der komponentenweisen Quantisierung zu berechnen. Dies kann mit dem Algorithmus 10.1 realisiert werden.

Algorithmus 10.1: Enumerationsalgorithmus ($z \in \mathcal{X} \longrightarrow z_i \in \mathcal{X}_i$)

Gegeben: Qualitativer Zustand $z \in \mathcal{X} = \{1, 2, \dots, N\}$,
Anzahlen qualitativer Zustände N_1, N_2, \dots, N_n aus der
komponentenweisen Quantisierung

Initialisiere: $j_n = z - 1$ für ein beliebiges $z \in \mathcal{X}$

Iteriere: 1 für $i = n, n-1, \dots, 1$
 2 $z_i = (j_i \bmod N_i) + 1$ (a)
 3 $j_{i-1} = \left\lfloor \frac{j_i}{N_i} \right\rfloor$ (b)

Ergebnis: Qualitative Zustände $z_i \in \mathcal{X}_i = \{1, \dots, N_i\}$ für $i = n, n-1, \dots, 1$

^aFür natürliche Zahlen $x \in \mathbb{N}$ und $y \in \mathbb{N}_{\geq 0}$ bestimmt die *modulo* Funktion $(x \bmod y) \in \mathbb{N}$ den ganzzahligen Rest der Division x/y . Es ist beispielsweise $(5 \bmod 3) = 2$.

^bDer Ausdruck $\lfloor x \rfloor$ bezeichnet wiederum eine Abrundungsfunktion, mit der eine reelle Zahl $x \in \mathbb{R}$ auf die größte ganze Zahl abgerundet wird, die kleiner oder gleich x ist. Es ist beispielsweise $\lfloor 5.97 \rfloor = 5$ (s. a. Abschnitt 5.2.2, S. 78).

In der Praxis enthalten Programmiersprachen wie MATLAB[®] oder PYTHON[™] Standardfunktionen, die ebensolche Zuordnungen wie in Definition in 10.3 und Algorithmus 10.1 einfach ermöglichen.

Beispiel 10.2 (Enumerationsgleichung und -algorithmus): Für ein System mit $n = 2$ quantitativen Zuständen geht die Gleichung (10.37) in folgende Form über:

$$z = \overline{z_1 z_2} = z_2 + (z_1 - 1)N_2 \in \mathcal{X}.$$

Für die in Abbildung 10.2 gegebenen Partitionierungen $\mathcal{E}_x(z_1)$ mit $z_1 \in \mathcal{X}_1 := \{1, 2, 3\}$ und $\mathcal{E}_x(z_2)$ mit $z_2 \in \mathcal{X}_2 := \{1, 2, 3, 4\}$ folgt mit $N = N_1 N_2 = 3 \cdot 4 = 12$, dass für die Nummerierung des mehrdimensionalen qualitativen Zustandsraums $z \in \mathcal{X} = \{1, 2, \dots, 12\}$ gilt. Damit können für alle Kombinationen von z_1 und z_2 die Werte z berechnet werden:

$$\begin{aligned} \overline{11} &= 1 + (1 - 1) \cdot 4 = 1 \\ \overline{21} &= 1 + (2 - 1) \cdot 4 = 5 \\ \overline{31} &= 1 + (3 - 1) \cdot 4 = 9 \\ \overline{12} &= 2 + (1 - 1) \cdot 4 = 2 \\ \overline{22} &= 2 + (2 - 1) \cdot 4 = 6 \\ \overline{32} &= 2 + (3 - 1) \cdot 4 = 10 \\ &\vdots \\ \overline{34} &= 4 + (3 - 1) \cdot 4 = 12 \end{aligned}$$

Damit ist festgelegt wie die Partitionierungen $\mathcal{E}_x(z), z \in \mathcal{X}$ bei der allgemeinen Quantisierung nummeriert sind und welche Kombinationen aus qualitativen Zuständen z_1 und z_2 zu einem qualitativen Zustand z führen (vgl. Abbildung 4.2).

Möchte man nun beispielsweise vom qualitativen Zustand $z = 10$ zurückschließen auf die qualitativen Zustände z_1, z_2 der komponentenweisen Quantisierung, dann kann der Algorithmus 10.1 verwendet werden. Für $z = 10$ und $n = 2$ wird der Algorithmus mit $j_2 = 10 - 1 = 9$ initialisiert und die Iterationsschleife beginnt in Schritt 1 mit $i = 2$. Damit erhält man in Schritt 2 den qualitativen Zustand

$$z_2 = (j_2 \bmod N_2) + 1 = (9 \bmod 4) + 1 = 2.$$

In Schritt 3 ergibt sich

$$j_1 = \left\lfloor \frac{j_2}{N_2} \right\rfloor = \left\lfloor \frac{9}{4} \right\rfloor = 2 .$$

Nun beginnt die Schleife in Schritt 1 mit $i = 1$ von neuem und man erhält in Schritt 2 den qualitativen Zustand

$$z_1 = (j_1 \bmod N_1) + 1 = (2 \bmod 3) + 1 = 3 .$$

Formal folgt in Schritt 3 noch die Berechnung des Werts j_0 , der jedoch nicht mehr zur weiteren Berechnung benötigt wird, da die Schleife nicht erneut durchlaufen wird.

10.2.2 Quantisiertes System

Abbildung 10.3 zeigt nun ein quantisiertes System, bei dem die qualitativen Werte der reellwertigen Ein- und Ausgangsvektoren durch die komponentenweise Quantisierung $[\mathbf{u}(k)] = \mathbf{v} \in \mathcal{U}$ sowie $[\mathbf{y}(k)] = \mathbf{w} \in \mathcal{Y}$ gegeben sind (vgl. Abschnitt 10.2.1). Dabei ist \mathbf{v} der qualitative Eingangsvektor und \mathbf{w} der qualitative Ausgangsvektor aus Gleichung (10.31). Das heißt, für die Elemente der qualitativen Eingangs- und Ausgangsvektoren gilt $v_t \in \mathcal{U}_t := \{1, \dots, M_t\}$, $t = 1, \dots, m$ sowie $w_j \in \mathcal{Y}_j := \{1, \dots, Q_j\}$, $j = 1, \dots, q$. Die qualitativen Zustände sind jedoch durch die allgemeine Quantisierung $[\mathbf{x}(k)] = z \in \mathcal{X} := \{1, \dots, N\}$ gegeben (vgl. Abschnitt 4.2.2).

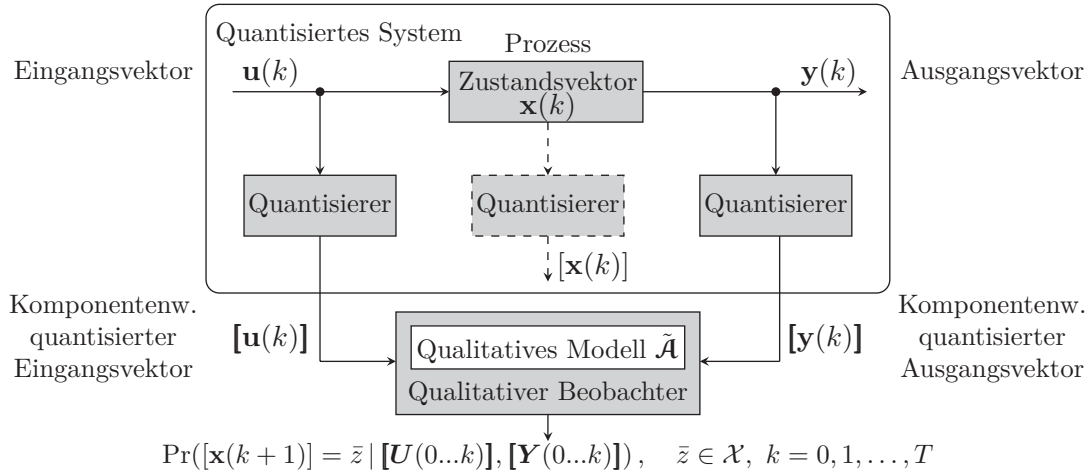


Abbildung 10.3: Quantisiertes System mit komponentenweise quantisierten Ein- und Ausgängen

Ein solches quantisiertes System ist durch einen stochastischen Automaten

$$\tilde{\mathcal{A}} = (\mathcal{X}, \mathcal{U}, \mathcal{Y}, \tilde{\mathcal{L}}, \mathbf{p}_0) , \quad (10.38)$$

mit der Eingabemenge \mathcal{U} , der Ausgabemenge \mathcal{Y} , der Zustandsmenge \mathcal{X} sowie der Verhaltensrelation $\tilde{\mathcal{L}}$ beschreibbar (vgl. Definition 10.6). Der stochastische Automat $\tilde{\mathcal{A}}$ wird dabei wieder als qualitatives Modell des quantisierten Systems bezeichnet. Die Verhaltensrelation des qualitativen Modells ist nun durch die Beziehung

$$\tilde{\mathcal{L}}(\bar{z}, \mathbf{w} | z, \mathbf{v}) = \Pr \left(\begin{array}{c} [\mathbf{x}(k+1)] = \bar{z} , \\ [\mathbf{y}(k)] = \mathbf{w} \end{array} \middle| \begin{array}{c} [\mathbf{x}(k)] = z , \\ [\mathbf{u}(k)] = \mathbf{v} \end{array} \right) \quad (10.39)$$

gegeben und beschreibt die bedingte Wahrscheinlichkeit dafür, dass das quantisierte System vom qualitativen Zustand $[\mathbf{x}(k)] = z$ bei qualitativem Eingang $[\mathbf{u}(k)] = \mathbf{v}$, in den qualitativen Folgezustand $[\mathbf{x}(k+1)] = \bar{z}$ wechselt und dabei den qualitativen Ausgang $[\mathbf{y}(k)] = \mathbf{w}$ erzeugt. Dabei ist die explizite Notation der Verhaltensrelation durch die Beziehung

$$\begin{aligned} \tilde{\mathcal{L}}(\bar{z}, w_1, \dots, w_q | z, v_1, \dots, v_m) \\ = \Pr \left(\begin{array}{c} [\mathbf{x}(k+1)] = \bar{z}, \\ [y_1(k)] = w_1, \dots, [y_q(k)] = w_q \end{array} \middle| \begin{array}{c} [\mathbf{x}(k)] = z, \\ [u_1(k)] = v_1, \dots, [u_m(k)] = v_m \end{array} \right) \end{aligned} \quad (10.40)$$

gegeben. Die Verhaltensrelation $\tilde{\mathcal{L}}$ stellt dabei erneut nur eine Approximation dar, da wie bereits erwähnt nicht angenommen werden kann, dass das quantisierte System die Markow-Eigenschaft besitzt. Zudem kann die hier verwendete Methode der stochastischen qualitativen Identifikation nicht die Vollständigkeit des qualitativen Modells garantieren. Die Prozedur zur Generierung des qualitativen Modells (10.38), beziehungsweise die Ermittlung der bedingten Wahrscheinlichkeiten der Verhaltensrelation $\tilde{\mathcal{L}}$ erfolgt analog zur Beschreibung in Abschnitt 4.3.2. Dabei treten anstelle der Gleichung (4.22) die Tupel

$$\begin{aligned} [\Theta(k)] &= ([\mathbf{x}(k+1)], [\mathbf{y}(k)], [\mathbf{x}(k)], [\mathbf{u}(k)]) \\ &= ([\mathbf{x}(k+1)], [y_1(k)], \dots, [y_q(k)], [\mathbf{x}(k)], [u_1(k)], \dots, [u_m(k)]) , \end{aligned}$$

womit alle weiteren Schritte (4.23)–(4.27) entsprechend folgen und in (4.27) zu einer Approximation $\tilde{\mathcal{L}}(\bar{z}, w_1, \dots, w_q | z, v_1, \dots, v_m)$ der Verhaltensrelation führen. Die somit berechneten bedingten Wahrscheinlichkeiten werden dann in einem Verhaltenstensor

$$\tilde{\mathbf{L}} \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m} \quad (10.41)$$

mit Elementen

$$\tilde{l}(\bar{z}, w_1, w_2, \dots, w_q, z, v_1, v_2, \dots, v_m) = \tilde{\mathcal{L}}(\bar{z}, w_1, w_2, \dots, w_q | z, v_1, v_2, \dots, v_m)$$

gespeichert. Im Folgenden wird zur Indexierung der Tensorelemente wenn möglich die Kurznotation $\tilde{l}(\bar{z}, \mathbf{w}, z, \mathbf{v})$ verwendet.

10.2.3 Qualitative Beobachtung

Die Definition einer einfachen Beobachtergleichung für komponentenweise quantisierte Ein- und Ausgänge, mit der die bedingte Wahrscheinlichkeit $\Pr([\mathbf{x}(k+1)] = \bar{z} | [\mathbf{U}(0 \dots k)], [\mathbf{Y}(0 \dots k)])$ aus Abbildung 10.3 näherungsweise berechnet werden kann, erfolgt analog zu Gleichung (10.12):

Definition 10.4 (Qualitative Beobachtergleichung für komponentenweise quantisierte Ein- und Ausgänge): *Sind die komponentenweise quantisierten E/A Folgen $[\mathbf{U}(0 \dots T)]$, $[\mathbf{Y}(0 \dots T)]$ des quantisierten Systems konsistent mit dem qualitativen Modell $\tilde{\mathcal{A}}$, dann kann eine Approximation der Wahrscheinlichkeitsverteilungen der qualitativen Zustände mit*

$$\begin{aligned} k \geq 0 : \quad \tilde{\mathbf{p}}_z(k+1) &= \frac{\tilde{\mathbf{L}}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \tilde{\mathbf{p}}_z(k)}{\|\tilde{\mathbf{L}}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \tilde{\mathbf{p}}_z(k)\|_1} \in [0, 1]^N, \\ k = 0 : \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0 \end{aligned} \quad (10.42)$$

für alle Zeiten $k = 0, 1, 2, \dots, T$ berechnet werden. Der qualitative Eingangsvektor $\mathbf{v}(k)$ und Ausgangsvektor $\mathbf{w}(k)$ sind dabei in jedem Zeitschritt k durch die komponentenweise quantisierte Eingangsgröße $[\mathbf{u}(k)] = \mathbf{v}(k)$ und Ausgangsgröße $[\mathbf{y}(k)] = \mathbf{w}(k)$ gegeben.

Die Zustandswahrscheinlichkeiten $\Pr([\mathbf{x}(k+1)] = \bar{z} \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)])$, $z \in \mathcal{X}$ sind dann entsprechend durch den Vektor

$$\tilde{\mathbf{p}}_z(k+1) = \begin{pmatrix} [\mathbf{x}(k+1)] = 1 \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)] \\ [\mathbf{x}(k+1)] = 2 \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)] \\ \vdots \\ [\mathbf{x}(k+1)] = N \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)] \end{pmatrix} \in [0, 1]^N \quad (10.43)$$

gegeben. Wie bereits erwähnt, repräsentiert der Vektor $\tilde{\mathbf{p}}_z$ nur eine Approximation der auf der rechten Seite der Gleichung gegebenen Wahrscheinlichkeiten. Wendet man die Beobachtergleichung (10.42) auf komponentenweise quantisierte Folgen von Ein- und Ausgängen $[\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]$ an, erhält man die Folge

$$\tilde{\mathbf{P}}_z(0...T+1 \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)]) = (\tilde{\mathbf{p}}_z(0), \tilde{\mathbf{p}}_z(1), \tilde{\mathbf{p}}_z(2), \dots, \tilde{\mathbf{p}}_z(T+1)) \quad (10.44)$$

von Zustandswahrscheinlichkeitsvektoren.

Erweiterte Beobachtergleichung: Eine Beobachtergleichung zur Berücksichtigung von Unsicherheiten in den komponentenweise quantisierten Ein- und Ausgängen kann analog zu Gleichung (10.18) definiert werden. Die Grundlage hierbei bilden die analog zu Gleichung (10.15) und für $t = 1, \dots, m$ und $j = 1, \dots, q$ definierten Wahrscheinlichkeitsvektoren der qualitativen Ein- und Ausgänge

$$\tilde{\mathbf{p}}_v^{(t)}(k) = \begin{pmatrix} \Pr([u_t(k)] = 1) \\ \Pr([u_t(k)] = 2) \\ \vdots \\ \Pr([u_t(k)] = M_t) \end{pmatrix} \in [0, 1]^{M_t}, \quad \tilde{\mathbf{p}}_w^{(j)}(k) = \begin{pmatrix} \Pr([y_j(k)] = 1) \\ \Pr([y_j(k)] = 2) \\ \vdots \\ \Pr([y_j(k)] = Q_j) \end{pmatrix} \in [0, 1]^{Q_j}. \quad (10.45)$$

Wie in Gleichung (10.15) sei auch hier angenommen, dass die Eingänge für $t = 1, \dots, m$ zur Zeit k stochastisch unabhängig voneinander sind, wobei entsprechendes für die Ausgänge gilt (s. a. Abschnitt 10.2.5). Für reale quantisierte Systeme ist diese Annahme kritisch, da die Ausgangsgrößen von den Zustandsgrößen abhängig sind. Für exakt bekannte quantisierte Ein- und Ausgangsfolgen $[\mathbf{U}(0...T)], [\mathbf{Y}(0...T)]$ ist diese Annahme jedoch gerechtfertigt, denn die Wahrscheinlichkeitsvektoren (10.45) besitzen dann für alle Zeiten $k = 0, 1, \dots, T$ nur ein Einselement und sind auf der Menge $\{0, 1\}$ definiert (vgl. Abschnitt 7.2.4, S. 137). Allgemein können die Wahrscheinlichkeitsvektoren der qualitativen Ein- und Ausgänge (10.45) wiederum zu Vektorfolgen $\tilde{\mathbf{P}}_v^{(t)}(0...T)$ und $\tilde{\mathbf{P}}_w^{(j)}(0...T)$ zusammengefasst werden. Mit den Vektoren (10.45), dem Verhaltenstensor (10.41) sowie dem Zustandsvektor (10.43) kann dann die erweiterte Beobachtergleichung (10.18) verwendet werden:

$$\begin{aligned} k \geq 0: \quad \tilde{\mathbf{p}}_z(k+1) &= \frac{\left\langle \mathbf{L} \mid \llbracket \tilde{\mathbf{p}}_w^{(1)}(k), \dots, \tilde{\mathbf{p}}_w^{(q)}(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v^{(1)}(k), \dots, \tilde{\mathbf{p}}_v^{(m)}(k) \rrbracket \right\rangle}{\left\| \left\langle \mathbf{L} \mid \llbracket \tilde{\mathbf{p}}_w^{(1)}(k), \dots, \tilde{\mathbf{p}}_w^{(q)}(k), \tilde{\mathbf{p}}_z(k), \tilde{\mathbf{p}}_v^{(1)}(k), \dots, \tilde{\mathbf{p}}_v^{(m)}(k) \rrbracket \right\rangle \right\|_1} \in [0, 1]^N, \\ k = 0: \quad \tilde{\mathbf{p}}_z(0) &= \tilde{\mathbf{p}}_0. \end{aligned} \quad (10.46)$$

Diese Gleichung führt auf die Folge von Zustandswahrscheinlichkeitsvektoren

$$\begin{aligned} &\tilde{\mathbf{P}}_z(0...T+1 \mid \tilde{\mathbf{P}}_v^{(1)}(0...T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0...T), \tilde{\mathbf{P}}_w^{(1)}(0...T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0...T)) \\ &= (\tilde{\mathbf{p}}_z(0), \tilde{\mathbf{p}}_z(1), \tilde{\mathbf{p}}_z(2), \dots, \tilde{\mathbf{p}}_z(T+1)) \end{aligned} \quad (10.47)$$

für die im Falle exakt bekannter quantisierter Ein- und Ausgänge die Gleichheit

$$\begin{aligned} &\tilde{\mathbf{P}}_z(0...T+1 \mid \tilde{\mathbf{P}}_v^{(1)}(0...T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0...T), \tilde{\mathbf{P}}_w^{(1)}(0...T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0...T)) \\ &= \tilde{\mathbf{P}}_z(0...T+1 \mid [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)]) \end{aligned}$$

gilt. Im folgenden Absatz wird der Algorithmus zur Fehlererkennung für qualitative Modelle mit komponentenweise quantisierten Ein- und Ausgängen auf Basis der CP-Repräsentation des Verhaltenstensors (10.41) angegeben. Auf die ausführliche Angabe der Algorithmen zur qualitativen Fehlerdetektion mit den Beobachtergleichungen (10.42) und (10.46) soll an dieser Stelle verzichtet werden, da sich deren Struktur nicht von der Algorithmen 7.1 und 7.2 unterscheidet.

10.2.4 CP-Algorithmus zur qualitativen Fehlererkennung

Die CP-Dekomposition des Verhaltenstensors (10.41) ist in der Regel wieder durch eine Approximation gegeben, die mit

$$\hat{\mathbf{L}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_q, \mathbf{C}, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_m \rrbracket \in \mathbb{R}_{\geq 0}^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m} \quad (10.48)$$

bezeichnet wird. Dabei wird der Zähler der Beobachtergleichung aus Lemma 10.1 mit der Variablen

$$\boldsymbol{\rho}(k) := \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^T \cdot \tilde{\mathbf{p}}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^T \cdot \hat{\mathbf{p}}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^T \cdot \tilde{\mathbf{p}}_v^{(t)}(k) \right) \right) \in \mathbb{R}_{\geq 0}^N \quad (10.49)$$

und der Nenner mit der bekannten Variablen

$$\hat{b}(k) := \left\| \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^T \cdot \tilde{\mathbf{p}}_w^{(j)}(k) \right) \otimes \left(\mathbf{C}^T \cdot \hat{\mathbf{p}}_z(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^T \cdot \tilde{\mathbf{p}}_v^{(t)}(k) \right) \right) \right\|_1 \in \mathbb{R}_{\geq 0} \quad (10.50)$$

abgekürzt. Die Gleichung (10.50) führt für konsistente vektorielle E/A Folgen und für diskrete Zeiten $k = 0, 1, 2, \dots, T$ entsprechend auf die Folge

$$\begin{aligned} & \hat{\tilde{B}}(0 \dots T \mid \tilde{\mathbf{P}}_v^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0 \dots T), \tilde{\mathbf{P}}_w^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0 \dots T)) \\ &= (\hat{b}(0), \hat{b}(1), \hat{b}(2), \dots, \hat{b}(T)) \end{aligned} \quad (10.51)$$

Der Algorithmus 10.2 zur qualitativen Fehlererkennung ist auf Seite 185 dargestellt und wird am Anwendungsbeispiel 10.3 auf Seite 186 demonstriert.

10.2.5 Äquivalenz qualitativer Modelle

Zwischen einem qualitativen Modell, welches durch die stochastische qualitative Identifikation mit allgemeiner Quantisierung generiert wurde und einem qualitativen Modell bei die die Ein- und Ausgänge komponentenweise quantisiert wurden, besteht unter gewissen Voraussetzungen eine grundlegende Äquivalenz. Zur Erläuterung sei beispielsweise ein qualitatives Modell $\tilde{\mathcal{A}}$ gemäß Definition (4.15) gegeben, welches eine Anzahl von $M = M_1 M_2 \dots M_m$ qualitativen Eingängen, $Q = Q_1 Q_2 \dots Q_q$ qualitativen Ausgängen und $N = N_1 N_2 \dots N_n$ qualitativen Zuständen besitzt und entsprechend durch einen vierdimensionalen Verhaltenstensor $\tilde{\mathbf{L}} \in [0, 1]^{N \times Q \times N \times M}$ mit Elementen $\tilde{l}(\bar{z}, w, z, v)$ beschrieben wird. Außerdem sei nun ein qualitatives Modell $\tilde{\mathcal{A}}$ entsprechend Gleichung (10.38) gegeben, welches hier durch einen Verhaltenstensor $\tilde{\mathbf{L}}^* \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m}$ mit den Elementen $\tilde{l}^*(\bar{z}, w_1, w_2, \dots, w_q, z, v_1, v_2, \dots, v_m)$ definiert wird. Der Tensor $\tilde{\mathbf{L}}^*$ kann nun mithilfe des Multiindex in einen Tensor

$$\tilde{\mathbf{L}}^\dagger \in [0, 1]^{N \times Q \times N \times M}$$

mit Elementen

$$\tilde{l}^\dagger(\bar{z}, \overline{w_1 w_2 \dots w_q}, z, \overline{v_1 v_2 \dots v_m}) = \tilde{l}^*(\bar{z}, w_1, w_2, \dots, w_q, z, v_1, v_2, \dots, v_m) \quad (10.52)$$

Algorithmus 10.2: CP-Algorithmus zur qualitativen Fehlererkennung mit komponentenweise quantisierten Ein- und Ausgängen

Gegeben: *Qualitatives Modell* $\tilde{\mathcal{A}}$, Gl. (10.38) mit *Verhaltenstensor* in
 CP-Repräsentation $\hat{\mathbf{L}} = [\mathbf{A}; \mathbf{B}_1, \dots, \mathbf{B}_q, \mathbf{C}, \mathbf{D}_1, \dots, \mathbf{D}_m]$, Gl. (10.48)
 Folgen von Wahrscheinlichkeitsvektoren qualitativer Eingänge
 $\tilde{\mathbf{P}}_v^{(t)}(0 \dots T)$ für $t = 1, \dots, m$
 Folgen von Wahrscheinlichkeitsvektoren qualitativer Ausgänge
 $\tilde{\mathbf{P}}_w^{(j)}(0 \dots T)$ für $j = 1, \dots, q$

Initialisiere: $k = 0$, $\hat{\mathbf{p}}_z(0) := \tilde{\mathbf{p}}_0$

Iteriere: 1 **solange** $0 \leq k \leq T$
 2 *berechne* $\boldsymbol{\rho}(k)$, Gl. (10.49)
 3 *berechne* $\hat{b}(k)$, Gl. (10.50)
 4 *berechne* $\hat{b}_m(k)$, Gl. (4.46) und $\hat{b}_b(k)$, Gl. (4.48)
 5 **wenn** $\hat{b}(k) = 0$ **dann**
 6 *setze* $\hat{\mathbf{p}}_z(k+1) = \tilde{\mathbf{p}}_0$
 7 *setze* $k := k+1$ und *fahre fort mit Schritt 1*
 8 **sonst**
 9 *berechne* $\hat{\mathbf{p}}_z(k+1) = \boldsymbol{\rho}(k)/\hat{b}(k)$
 10 *setze* $k := k+1$ und *fahre fort mit Schritt 1*

Ergebnis: $\hat{\mathbf{P}}_z(0 \dots T+1 | \tilde{\mathbf{P}}_v^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0 \dots T), \tilde{\mathbf{P}}_w^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0 \dots T))$ Gl. (10.47),
 $\hat{\mathbf{B}}(0 \dots T | \tilde{\mathbf{P}}_v^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0 \dots T), \tilde{\mathbf{P}}_w^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0 \dots T))$, Gl. (10.51),
 $\hat{\mathbf{B}}_m(0 \dots T)$, Gl. (4.47),
 $\hat{\mathbf{B}}_b(0 \dots T)$, Gl. (4.49).

überführt werden. Unter der Voraussetzung, dass die beiden qualitativen Modelle $\tilde{\mathcal{A}}$ und $\tilde{\mathcal{A}}$ für das gleiche quantisierte System, mit identischen Partitionierungen und Anzahlen an qualitativen Ein-, Ausgängen und Zuständen stochastisch identifiziert wurden, gilt dann die Gleichheit

$$\tilde{\mathbf{L}}^\dagger = \tilde{\mathbf{L}}.$$

Die Repräsentation des Verhaltenstensors $\tilde{\mathbf{L}}^\star \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m}$ sowie die vierdimensionale Darstellung $\tilde{\mathbf{L}} \in [0, 1]^{N \times Q \times N \times M}$ sind somit ineinander überführbar, was sich in der Praxis als hilfreich erweist, da die Modelle nicht getrennt voneinander gebildet werden müssen, sondern durch Umordnen der Tensorelemente in die jeweils andere Struktur überführt werden können. Für jede Programmiersprache, welche die *big-endian* Konvention verwendet ist eine zu (10.52) identische Operation mithilfe der sogenannten *reshape* Funktion möglich. Dann ist beispielsweise

$$\tilde{\mathbf{L}} = \text{reshape}(\tilde{\mathbf{L}}^\star, [N, Q, M, N]) \in [0, 1]^{N \times Q \times N \times M}$$

und umgekehrt

$$\tilde{\mathbf{L}}^\star = \text{reshape}(\tilde{\mathbf{L}}, [N, Q_1, \dots, Q_q, M_1, \dots, M_m, N]) \in [0, 1]^{N \times Q_1 \times \dots \times Q_q \times N \times M_1 \times M_m}.$$

Dies Gleichheit führt auch dazu, dass die mit den jeweiligen einfachen Beobachtergleichungen berechneten Zustandswahrscheinlichkeiten identisch sind. Das heißt, die auf Basis des vierdimensionalen

Verhaltenstensors mit der Beobachtergleichung (7.25) bestimmten bedingten Zustandswahrscheinlichkeiten $\Pr(Z(k) = \bar{z} | [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)])$, $\bar{z} \in \mathcal{X}$ sind identisch mit den Zustandswahrscheinlichkeiten $\Pr(Z(k) = \bar{z} | [\mathbf{U}(0...k)], [\mathbf{Y}(0...k)])$, $\bar{z} \in \mathcal{X}$ die man aus der Beobachtergleichung (10.42) erhält. Dies ist eine logische Konsequenz aus der Enumerationsgleichung (10.37) und des Enumerationsalgorithmus 10.1, beziehungsweise der Partitionierung der Signalmräume mit welcher der Zusammenhang zwischen den komponentenweise quantisierten Ein- und Ausgangsgrößen und denen der allgemeinen Quantisierung hergestellt wird (vgl. Abschnitt 10.2.1, Seite 179). So ist beispielsweise bei einem System mit $m = 2$ numerischen Eingängen und $M_1 = 3$ sowie $M_2 = 4$ qualitativen Eingängen festgelegt, dass bei der komponentenweisen Quantisierung die qualitativen Eingänge $[\mathbf{u}] = ([u_1] = 2 \ [u_2] = 4)^\top$ dem qualitativen Eingang $[\mathbf{u}] = 8$ bei der allgemeinen Quantisierung entsprechen (vgl. Abbildung 4.2).

Die Äquivalenz der Zustandswahrscheinlichkeiten gilt für exakte quantisierte Ein- und Ausgänge entsprechend auch für die erweiterten Beobachtergleichungen (10.46) und (7.33). Sind die Ein- und Ausgänge nicht sicher besteht jedoch eine Einschränkung, die der Struktur der Beobachtergleichung (10.46) geschuldet ist. Während die erweiterte Beobachtergleichung (7.33) für den vierdimensionalen Verhaltenstensor die Zustandswahrscheinlichkeiten für Eingangswahrscheinlichkeitsvektoren der Länge M und Ausgangswahrscheinlichkeitsvektoren der Länge Q bestimmt, werden bei der Beobachtergleichung (10.46) jeweils m Vektoren der Länge M_1, M_2, \dots, M_m und q Vektoren der Länge Q_1, Q_2, \dots, Q_q verwendet. Für zum Beispiel wieder $M_1 = 3$, $M_2 = 4$ und $M = 12$ qualitative Eingänge besitzt der Eingangswahrscheinlichkeitsvektor $\mathbf{p}_v \in [0, 1]^{12}$ beim vierdimensionalen Verhaltenstensor zwölf Komponenten, während bei der höherdimensionalen Tensorrepräsentation die Eingangswahrscheinlichkeitsvektoren $\mathbf{p}_v^{(1)} \in [0, 1]^3$, $\mathbf{p}_v^{(2)} \in [0, 1]^4$ verwendet werden, die zusammen nur sieben Elemente besitzen. Damit ist klar, dass mit den einzelnen Eingangswahrscheinlichkeitsvektoren nicht die gleiche Menge an Informationen darstellbar ist, wie mit einem einzelnen Eingangswahrscheinlichkeitsvektor. Mit den Eingangsvektoren $\mathbf{p}_v^{(1)} \in [0, 1]^3$ und $\mathbf{p}_v^{(2)} \in [0, 1]^4$ ist eine zum Vektor $\mathbf{p}_v \in [0, 1]^{12}$ äquivalente Repräsentation somit nur für all diejenigen Wahrscheinlichkeitsverteilungen möglich, die sich beispielsweise als Kronecker-Produkt der einzelnen Vektoren darstellen lassen. Anders formuliert, die einzelnen Ein- und Ausgänge müssen voneinander *stochastisch unabhängig* sein.

Beispiel 10.3 (Fehlererkennung mit der CP-Repräsentation und komponentenweise quantisierten Ein- und Ausgängen): *Die Grundlage dieses Beispiels bildet der stochastische Automat (10.38), der für die Betrachtung komponentenweise quantisierter Ein- und Ausgänge definiert ist und ein qualitatives Modell des Erhitzers aus Abschnitt 5.3 mit den Modellparametern aus Tabelle 5.11. Aufgrund der Tatsache, dass die Eingangsgrößen u_1, u_2 des Erhitzers je sechs ($M_1 = M_2 = 6$) qualitative Werte $[u_1] = v_1 \in \{1, \dots, M_1\}$ und $[u_2] = v_2 \in \{1, \dots, M_2\}$ annehmen können und gleiches entsprechend für die Ausgänge gilt, ist der Verhaltenstensor nach Gleichung (10.41) durch*

$$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6} \quad (10.53)$$

gegeben. Die qualitative Beobachtung wird im folgenden wieder anhand einer Rang-50 Approximation des Verhaltenstensor gezeigt, die mit dem MU-Algorithmus (E) aus Tabelle 8.1 in PYTHONTM berechnet wurde. Dabei werden die Ergebnisse der einfachen Beobachtergleichung (10.42) mit denen des CP-Algorithmus 10.2 anhand der visualisierten Zustandswahrscheinlichkeiten der Wasseraustrittstemperatur T_{W_A} sowie der Verhaltenssignale verglichen.

Die Rang-50 Approximation des Verhaltenstensors (10.53) führt nun entsprechend Gleichung (10.48) auf die CP-Repräsentation

$$\hat{\mathbf{L}} = [\mathbf{\lambda}; \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}, \mathbf{D}_1, \mathbf{D}_1] \in \mathbb{R}_{\geq 0}^{36 \times 6 \times 6 \times 36 \times 6 \times 6} \quad (10.54)$$

mit den Faktormatrizen

$$\mathbf{A} \in \mathbb{R}_{\geq 0}^{36 \times 50}, \mathbf{B}_1 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \mathbf{B}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \mathbf{C} \in \mathbb{R}_{\geq 0}^{36 \times 50}, \mathbf{D}_1 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \mathbf{D}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50}$$

und dem Gewichtungsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{50}$. In der Abbildung 10.4 sind die Ergebnisse der qualitativen Beobachtung basierend auf der CP-Repräsentation (10.54) (links) denen ohne Dekomposition gegenübergestellt (rechts).

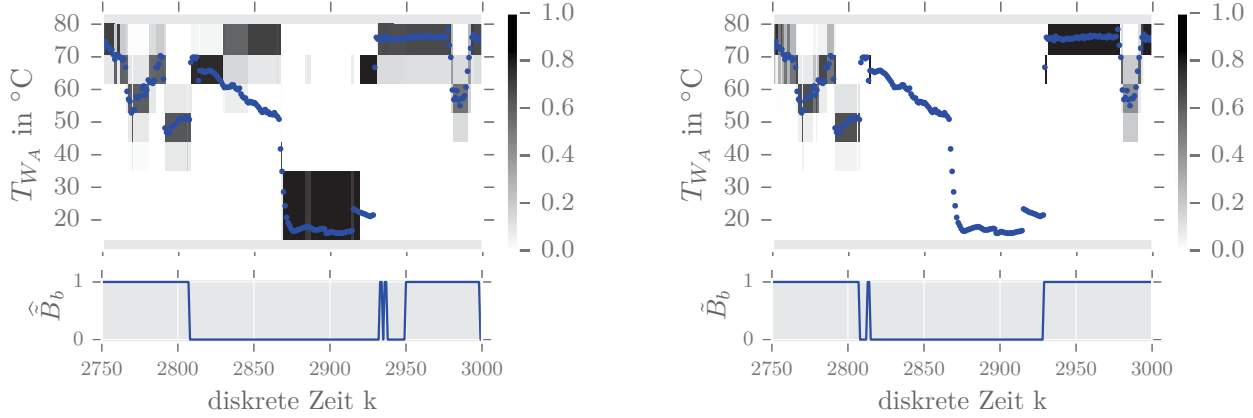


Abbildung 10.4: Vergleich der qualitativen Zustandswahrscheinlichkeiten und der Verhaltenssignale mit Dekomposition \hat{B}_b (links) und ohne Dekomposition \tilde{B}_b (rechts)

Die rechte Darstellung basiert dabei auf der Tensorrepräsentation $\tilde{\mathbf{L}} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$, wobei zur Berechnung der Zustandswahrscheinlichkeiten und des Verhaltenssignals die Gleichung (10.42) verwendet wurde. Die Abbildung für die CP Repräsentation zeigt dabei wieder positive Zustandswahrscheinlichkeiten für den fehlerhaften Bereich, was wie bereits im Anwendungsbeispiel 9.2 beschrieben daran liegt, dass die Nullen des Verhaltenstensors durch sehr kleine Werte approximiert werden. Die Verhaltenssignale der Approximation \hat{B} und der vollen Tensorrepräsentation \tilde{B} zeigen dabei für gleiche Werte von $\tau = 1\%$ und $\phi = 2$ ein leicht unterschiedliches Ergebnis. Der nicht nominale Zeitbereich wird zwar von beiden Algorithmen zur gleichen Zeit erkannt, jedoch zeigt die linke Darstellung den Fehler länger an als die Rechte. Durch das ändern des Schwellwerts τ und der Fensterbreite ϕ des gleitenden Mittelwertes ließen sich hier bei der CP-Darstellung noch weitere Optimierungen durchführen, womit jedoch die Vergleichbarkeit der beiden Darstellungen nicht mehr gegeben wäre. Dies zeigt auch die Schwierigkeit einen sinnvollen Schwellwert für die Fehlererkennung festzulegen, was durch den approximativen Charakter der CP-Repräsentation zusätzlich erschwert wird.

Tabelle 10.2: Speicherbedarf für verschiedene Repräsentationen des Verhaltenstensors qualitativer Modelle des Erhitzers aus Abschnitt 5.3 mit den Modellparametern aus Tabelle 5.11

Repräsentation des Verhaltenstensors	Fehlererk. Algorithmus	Speicher- bedarf	Reduktions- faktor	MSD \tilde{B} zu \hat{B}
$\tilde{\mathbf{L}} \in [0, 1]^{36 \times 36 \times 36 \times 36}$, Gl. (9.15)	7.1	$\gamma = 1\,679\,616$	0	0
$\hat{\underline{\mathbf{L}}} = \langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle$, Gl. (9.27)	9.2	$\gamma_{\text{TT}} = 127\,296$	$\xi_{\text{TT}} = 13$	0.096
$\hat{\underline{\mathbf{L}}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \rrbracket$, Gl. (9.16)	9.1	$\gamma_{\text{CP}} = 7250$	$\xi_{\text{CP}} = 231$	0.179
$\hat{\underline{\mathbf{L}}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}, \mathbf{D}_1, \mathbf{D}_2 \rrbracket$, Gl. (10.54)	10.2	$\gamma_{\text{CP}} = 4850$	$\xi_{\text{CP}} = 346$	0.445

Wie die Tabelle 10.2 zeigt, ist der Speicherbedarf gegenüber dem der CP-Dekomposition des vierdimensionalen Verhaltenstensor aus dem Anwendungsbeispiel 9.2 noch einmal geringer, wobei beide Verhaltenstensoren in der vollen Tensorrepräsentation den gleichen Informationsgehalt besitzen und beide Dekompositionen mit einem Rang von $R = 50$ berechnet wurden. Die mittlere quadratische Abweichung der Verhaltenssignale ist in diesem Beispiel jedoch größer als bei den vorangegangenen. Nichtsdestotrotz zeigt dieses Beispiel, dass ungeachtet der weiteren Reduktion des Speicherbedarfs, die Fehlerdetektion mit der CP-Repräsentation des Verhaltenstensors immer noch möglich ist.

Abschließend soll kurz auf die bereits in Abschnitt 10.2.5 beschriebene Äquivalenz qualitativer Modelle eingegangen werden. Demnach bilden das (nicht dekomponierte) qualitative Modell des Anwendungsbeispiels 5.3 mit dem Verhaltenstensor $\tilde{L} \in [0, 1]^{36 \times 36 \times 36 \times 36}$ und das in diesem Beispiel verwendete Modell mit dem Verhaltenstensor $\tilde{L} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$ äquivalente Repräsentationen. Die Abbildung 10.5 zeigt einen Vergleich beider Tensorrepräsentationen für den fehlerhaften Zeitbereich, die erwartungsgemäß identisch ausfallen. Dabei wurde in beiden Darstellungen für die Verhaltenssignale ein gleitender Mittelwert $\phi = 2$ und ein Schwellwert von $\tau = 1\%$ verwendet.

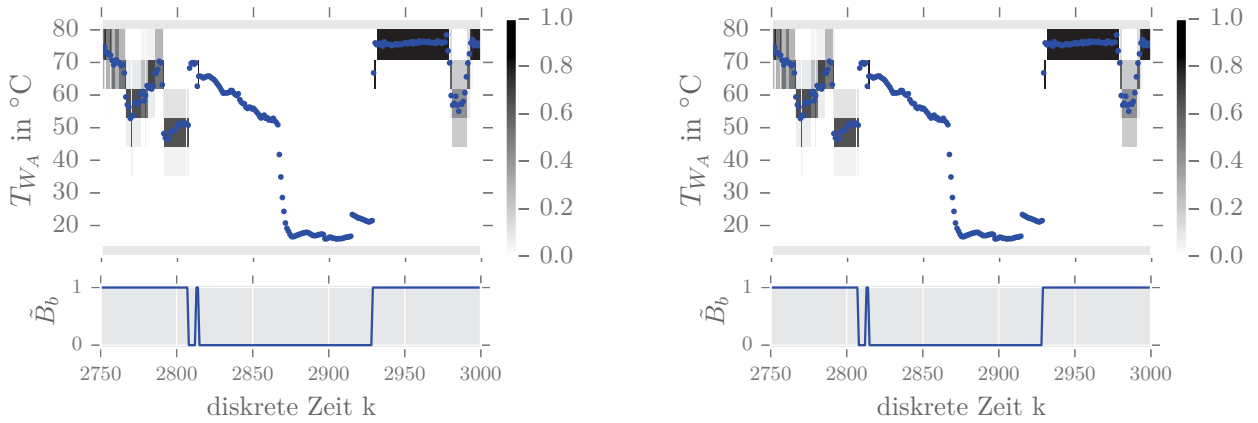


Abbildung 10.5: Vergleich der qualitativen Zustandswahrscheinlichkeiten und der Verhaltenssignale \tilde{B}_b für den Verhaltenstensor $\tilde{L} \in [0, 1]^{36 \times 36 \times 36 \times 36}$ (links), berechnet mit Gleichung (7.25), mit denen der Tensorrepräsentation $\tilde{L} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$ (rechts), berechnet mit Gleichung (10.42)

10.3 Autonomer stochastischer Automat mit vektoriellen Zustandsfolgen*

Hintergrund dieses Abschnitts ist die Einführung der automatentheoretischen Grundlage für die qualitative Modellierung quantisierter Systeme mit *komponentenweise* quantisierten Zuständen. Das allgemeine Konzept wird hierbei zunächst anhand eines autonomen stochastischen Automaten erläutert, während in Abschnitt 10.4 die Ergebnisse dieses und des Abschnitts 10.1 zusammengeführt werden.

10.3.1 Betrachtung stochastischer Teilprozesse*

In Abschnitt 3.2 wurde der autonome stochastische Automat \mathcal{A}_a als Modell eines einzelnen stochastischen Prozesses, der durch eine Folge von Zufallsvariablen $(Z(k))_{k \in \mathbb{N}}$ gegeben ist, beschrieben. Im Folgenden wird ein stochastischer Automat \mathcal{A}_a eingeführt, der das Verhalten mehrerer stochastischer

Prozesse $(Z_1(k))_{k \in \mathbb{N}}, (Z_2(k))_{k \in \mathbb{N}}, \dots, (Z_n(k))_{k \in \mathbb{N}}$ beschreiben soll. Dabei wird jeder stochastische Prozess $(Z_i(k))_{k \in \mathbb{N}}$ für $i = 1, \dots, n$ durch einen autonomen stochastischen Automaten $\mathcal{A}_a^{(i)}$ beschrieben. Dieses Konzept basiert somit auf der Betrachtung stochastischer Teilprozesse, die zusammengenommen den stochastischen Gesamtprozess repräsentieren (s. Abbildung 10.6).

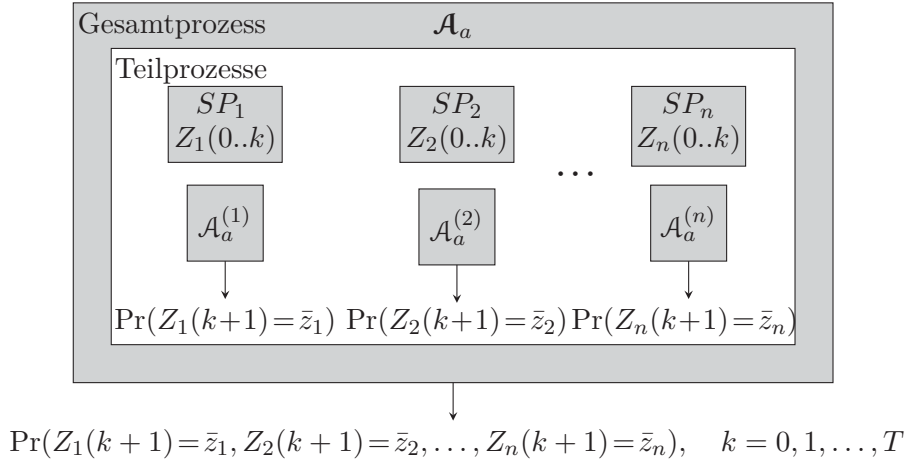


Abbildung 10.6: Betrachtung stochastischer Teilprozesse

In Abbildung 10.6 stellt jeder Automat $\mathcal{A}_a^{(i)}$, für $i = 1, 2, \dots, n$ einen autonomen stochastischen Automaten nach Definition 3.2 dar. Das bedeutet, jeder dieser Automaten ist durch ein 3-Tupel $\mathcal{A}_a^{(i)} = (\mathcal{Z}_i, \mathcal{F}_i, \mathbf{p}_0^{(i)})$ definiert und besitzt einen Zustandsraum $\mathcal{Z}_i := \{1, 2, \dots, N_i\}$ mit N_i Zuständen, die durch $z_i \in \mathcal{Z}_i$ gegeben sind. Die Übergangswahrscheinlichkeiten der stochastischen Automaten sind durch die Zustandsübergangsrelationen $\mathcal{F}_i(\bar{z}_i | z_i)$ gegeben und die initialen Zustandswahrscheinlichkeiten $(p_0)_{z_i} = \Pr(Z_i(0) = z_i)$, $z_i \in \mathcal{Z}$ durch die Vektoren $\mathbf{p}_0^{(i)} \in [0, 1]^{N_i}$. Die Berechnung der Zustandswahrscheinlichkeiten der Teilprozesse $\Pr(Z_i(k) = z_i)$, $i = 1, \dots, n$, erfolgt dabei analog zur Beschreibung in Abschnitt 3.2. Möchte man nun auf Basis dieser Wahrscheinlichkeiten, die *gemeinsame Zustandswahrscheinlichkeit* $\Pr(Z_1(k) = z_1, Z_2(k) = z_2, \dots, Z_n(k) = z_n)$ der Teilprozesse bestimmen, dann ist dies nur dann exakt möglich, wenn die Zufallsvariablen der stochastischen Teilprozesse $Z_1(k), Z_2(k), \dots, Z_n(k)$ *stochastisch unabhängig* sind.

Die gemeinsamen Zustandswahrscheinlichkeiten sind somit aus dem Produkt der Einzelwahrscheinlichkeiten berechenbar. Die Bestimmung dieser Wahrscheinlichkeiten kann wiederum vektoriell erfolgen, indem man die Zustandsübergangswahrscheinlichkeiten $\mathcal{F}(\bar{z}_i | z_i)$ der stochastischen Teilprozesse entsprechend Gleichung (3.19) in Übergangsmatrizen $\mathbf{F}_i \in [0, 1]^{N_i \times N_i}$ für $i = 1, \dots, n$ notiert. Die diskreten Zustandswahrscheinlichkeiten $\Pr(Z_i(k) = z_i)$, $z_i \in \mathcal{Z}_i$ jedes Automaten $\mathcal{A}_a^{(i)}$ werden dabei analog zu Gleichung (3.18) für alle $z_i \in \mathcal{Z}_i$ in Form von Wahrscheinlichkeitsvektoren

$$\mathbf{p}_z^{(i)}(k) = \begin{pmatrix} p_z^{(i)}(k, 1) \\ p_z^{(i)}(k, 2) \\ \vdots \\ p_z^{(i)}(k, N_i) \end{pmatrix} := \begin{pmatrix} \Pr(Z_i(k) = 1) \\ \Pr(Z_i(k) = 2) \\ \vdots \\ \Pr(Z_i(k) = N_i) \end{pmatrix} \in [0, 1]^{N_i} \quad (10.55)$$

notiert. Die Zustandswahrscheinlichkeitsverteilungen der Teilprozesse können dementsprechend für beliebige Zeiten $k = 0, \dots, T$ mit

$$\begin{aligned} \mathbf{p}_z^{(i)}(k+1) &= \mathbf{F}_i \cdot \mathbf{p}_z^{(i)}(k) \\ \mathbf{p}_z^{(i)}(0) &:= \mathbf{p}_0^{(i)} \end{aligned} \quad (10.56)$$

berechnet werden. Für stochastisch unabhängige Zufallsvariablen sind die gemeinsamen Wahrscheinlichkeiten zur Zeit $k + 1$ durch den Produktvektor

$$\mathbf{p}_z(k+1) = \bigotimes_{i=1}^n \mathbf{p}_z^{(i)}(k+1) \in [0, 1]^N \quad (10.57)$$

der Länge $N = N_1 N_2 \cdots N_n$ mit Komponenten

$$\begin{aligned} p_z(k+1, \bar{z}) &= \Pr(Z_1(k+1) = \bar{z}_1) \cdot \Pr(Z_2(k+1) = \bar{z}_2) \cdots \Pr(Z_n(k+1) = \bar{z}_n) \\ &= \Pr(Z_1(k+1) = \bar{z}_1, Z_2(k+1) = \bar{z}_2, \dots, Z_n(k+1) = \bar{z}_n) \end{aligned} \quad (10.58)$$

für $\bar{z} = \bar{z}_1 \bar{z}_2 \dots \bar{z}_n \in \{1, 2, \dots, N\}$ gegeben.

Die ausschließliche Betrachtung der stochastischen Teilprozesse, um auf das Verhalten des stochastischen Gesamtprozesses zu schließen, führt im Allgemeinen zu einer nur näherungsweisen Beschreibung der gemeinsamen Zustandswahrscheinlichkeiten, da in der Regel *nicht* angenommen werden kann, dass die Zufallsvariablen der einzelnen Prozesse stochastisch unabhängig sind. In der Literatur ist diese Vorgehensweise dennoch üblich, da es aus komplexitätstechnischen Gründen häufig nicht möglich oder sinnvoll ist, das vollständige Verhalten des Gesamtprozesses abzubilden. Ansätze die auf dieser komponentenorientierten Betrachtungsweise beruhen sind beispielsweise sogenannte Automatenetze, bei denen ein stochastisches Gesamtsystem durch gekoppelte oder ungekoppelte Teilprozesse beschrieben wird (s. hierzu [95], [113]). Das Komplexitätsproblem bei der Betrachtung des Gesamtverhaltens wird im folgenden Absatz deutlich werden. Dabei wird ein Ansatz zur Reduktion des Speicherbedarfs vorgestellt, der auf der CP-Dekomposition beruht. Dabei wird zur formalen Erläuterung der theoretischen Grundlagen zunächst ebenfalls angenommen, dass die Zufallsvariablen der Teilprozesse stochastisch unabhängig sind.

10.3.2 Betrachtung des stochastischen Gesamtprozesses*

Zunächst werden für die folgenden Beschreibungen die Zustände $z_i \in \mathcal{Z}_i$ der stochastischen Teilprozesse für alle $i = 1, \dots, n$ in einem *Zustandsvektor*

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \in \mathcal{Z} \quad (10.59)$$

zusammengefasst. Die Menge \mathcal{Z} ist durch das kartesische Produkt

$$\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \times \cdots \times \mathcal{Z}_n \subseteq \mathbb{N}^n$$

gegeben und repräsentiert somit die Menge aller möglichen Zustandstupel. Dass diese Zustandstupel hier als Vektoren interpretiert werden, stellt wiederum einen leichten Missbrauch der Vektornotation dar, der bereits in Anmerkung 10.1 erläutert wurde. Mit der vektoriellen Notation der Zustände $z_i \in \mathcal{Z}_i$ können nun auch die Zustandsfolgen $Z_i(0..T) = (z_i(0), z_i(1), \dots, z_i(T))$ der stochastischen Teilprozesse für $k = 0, 1, \dots, T$ in einer vektoriellen Zustandsfolge

$$\mathbf{Z}(0..T) = (\mathbf{z}(0), \mathbf{z}(1), \mathbf{z}(2), \dots, \mathbf{z}(T)) \quad (10.60)$$

zusammengefasst werden. Die gemeinsamen Zustandswahrscheinlichkeiten des Gesamtprozesses zur Zeit k können damit durch die Kurznotation

$$\Pr(\mathbf{Z}(k) = \mathbf{z}) := \Pr(Z_1(k) = z_1, Z_2(k) = z_2, \dots, Z_n(k) = z_n)$$

ausgedrückt werden. Der stochastische Automat zur Beschreibung des Gesamtprozesses \mathcal{A}_a kann somit wie folgt definiert werden:

Definition 10.5 (Autonomer stochastischer Automat mit vektoriellen Zuständen): *Das 3-Tupel*

$$\mathcal{A}_a = (\mathcal{Z}, \mathcal{F}, \mathbf{p}_0^{(i)}) \quad (10.61)$$

mit der bedingten Wahrscheinlichkeitsverteilung

$$\mathcal{F} : \mathcal{Z} \times \mathcal{Z} \rightarrow [0, 1] , \quad (10.62)$$

welche die bedingte Übergangswahrscheinlichkeit

$$\mathcal{F}(\bar{\mathbf{z}} | \mathbf{z}) = \Pr(\mathbf{Z}(k+1) = \bar{\mathbf{z}} | \mathbf{Z}(k) = \mathbf{z}) \quad (10.63)$$

für alle $\bar{\mathbf{z}}, \mathbf{z} \in \mathcal{Z}$ beschreibt, heißt autonomer stochastischer Automat mit vektoriellen Zuständen.

Die Übergangsrelation (10.63) beschreibt mit $\mathcal{F}(\bar{\mathbf{z}} | \mathbf{z})$ die bedingte Wahrscheinlichkeit dafür, dass der autonome Automat \mathcal{A}_a in den vektoriellen Zustand $\bar{\mathbf{z}} \in \mathcal{Z}$ übergeht, wenn er sich im vektoriellen Zustand $\mathbf{z} \in \mathcal{Z}$ befindet. Die Notation der Übergangsrelation (10.63) stellt dabei eine abkürzende Schreibweise der Verteilung

$$\mathcal{F}(\bar{z}_1, \dots, \bar{z}_n | z_1, \dots, z_n) = \Pr(Z_1(k+1)=z_1, \dots, Z_n(k+1)=z_n | Z_1(k)=z_1, \dots, Z_n(k)=z_n) \quad (10.64)$$

dar. Die Übergangswahrscheinlichkeiten sind somit in einem *Übergangstensor*

$$\mathbf{F} \in [0, 1]^{N_1 \times N_2 \times \dots \times N_n \times N_1 \times N_2 \times \dots \times N_n} \quad (10.65)$$

mit Elementen

$$f(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, z_1, z_2, \dots, z_n) = \mathcal{F}(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n | z_1, z_2, \dots, z_n)$$

darstellbar. Sind wie hier angenommen die Zufallsvariablen der stochastischen Teilprozesse stochastisch unabhängig, gilt dabei die Gleichheit

$$f(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, z_1, z_2, \dots, z_n) = f_{\bar{z}_1 z_1}^{(1)} \cdot f_{\bar{z}_2 z_2}^{(2)} \cdot \dots \cdot f_{\bar{z}_n z_n}^{(n)} , \quad (10.66)$$

wobei $f_{\bar{z}_i z_i}^{(i)}$ das (\bar{z}_i, z_i) -te Element der Übergangsmatrix \mathbf{F}_i des i -ten Teilprozesses ist. Ausgehend von einem »worst-case« Szenario, bei dem der Übergangstensor \mathbf{F} ausschließlich Werte ungleich null enthält, beträgt die Anzahl zu speichernder Werte bei der Betrachtung des stochastischen Gesamtprozesses

$$\prod_{i=1}^n N_i^2 ,$$

während bei der ausschließlichen Betrachtung der Teilprozesse nur die Übergangsmatrizen $\mathbf{F}_1, \dots, \mathbf{F}_n$ mit einer Gesamtanzahl von

$$\sum_{i=1}^n N_i^2$$

Werten gespeichert werden müssen. Für ein kleines Gesamtsystem, bestehend aus $n = 4$ Teilprozessen mit je $N_i = 6$ Zuständen beträgt der Speicherbedarf des Übergangstensors $6^2 \cdot 6^2 \cdot 6^2 \cdot 6^2 = 1\,679\,616$ Werte, während man zur Speicherung der Übergangsmatrizen der stochastischen Teilprozesse nur $6^2 + 6^2 + 6^2 + 6^2 = 144$ Werte benötigt. Dieser gravierende Unterschied macht deutlich, weshalb man häufig nur die stochastischen Teilprozesse betrachtet und damit bei stochastisch *abhängigen* Zufallsvariablen in Kauf nimmt, die gemeinsamen Zustandswahrscheinlichkeiten nur approximativ

bestimmen zu können. Dieses kleine Beispiel zeigt aber auch, welche Menge an Information bezüglich des Gesamtverhaltens des Prozesses bei dieser Vorgehensweise verloren geht. Dementsprechend erstrebenswert ist es, eine Möglichkeit zur Reduktion der Modellkomplexität zu finden, die nicht von vornherein schon grundlegende Information bezüglich des Gesamtverhaltens des Prozesses unbeachtet lässt. Durch die Methodik der Tensordekomposition bieten sich hierbei neue Wege zur Reduktion der Modellkomplexität, die es ermöglichen, selbst große Übergangstensoren durch eine approximierte CP-Repräsentation, auf einen überschaubaren Speicherbedarf zu reduzieren. Für die im Folgenden vorgestellte Gleichung, die auf der CP-Dekomposition des Übergangstensors (10.65) basiert, wird zwar für die Beweisführung ebenfalls die stochastische Unabhängigkeit der Zufallsvariablen gefordert, jedoch kann im Folgenden die Vermutung aufgestellt werden, dass diese für stochastisch *abhängige* Zufallsvariablen bessere Ergebnisse generiert, als wenn nur die stochastischen Teilprozesse betrachtet werden.

CP-Repräsentation des Übergangstensors: Für die CP-Dekomposition des Übergangstensors $F \in [0, 1]^{N_1 \times N_2 \times \dots \times N_n \times N_1 \times N_2 \times \dots \times N_n}$ wird angenommen, dass diese exakt und durch die Zerlegung

$$F = \llbracket \lambda; \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n \rrbracket \quad (10.67)$$

gegeben ist. Dabei stellen die Matrizen $\mathbf{A}_i \in \mathbb{R}_{\geq 0}^{N_i \times R}$ die Faktormatrizen der Folgezustände und die Matrizen $\mathbf{C}_i \in \mathbb{R}_{\geq 0}^{N_i \times R}$ die Faktormatrizen der Zustände dar. Ein Algorithmus, der die Zustandswahrscheinlichkeiten $\mathbf{p}_z^{(i)}(k+1)$ aus Gleichung (10.56) für $i = 1, \dots, n$ berechnet, kann damit wie folgt definiert werden:

Proposition 10.3: *Für eine exakte CP-Dekomposition und stochastisch unabhängige Zustände können die Zustandswahrscheinlichkeiten der stochastischen Teilprozesse durch*

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z^{(i)}(k+1) &= \mathbf{A}_i \cdot \left(\lambda \otimes \left(\bigotimes_{j \neq i}^n \sum_{s_j=1}^{N_j} (\mathbf{a}_{s_j}^{(j)})^\top \right) \otimes \left(\bigotimes_{t=1}^n \mathbf{C}_t^\top \cdot \mathbf{p}_z^{(t)}(k) \right) \right) \in [0, 1]^{N_i}, \\ k = 0 : \quad \mathbf{p}_z^{(t)}(0) &= \mathbf{p}_0^{(t)} \end{aligned} \quad (10.68)$$

für alle $i = 1, 2, \dots, n$ und Zeiten $k = 0, 1, 2, \dots, T$ bestimmt werden.

Beweis: Der Beweis der Proposition 10.3 ist in Anhang B.8 aufgeführt.

Ist die CP Dekomposition exakt, dann generiert die Gleichung (10.68) die gleichen Ergebnisse wie Gleichung (10.56). Der Vektor der gemeinsamen Zustandswahrscheinlichkeiten zur Zeit $k+1$ kann dann wiederum durch das Kronecker-Produkt (10.57) berechnet werden. Sind die Zufallsvariablen der Zustände stochastisch abhängig, kann die Gleichung (10.68) die gemeinsamen Zustandswahrscheinlichkeiten nur approximativ bestimmen. Es besteht jedoch Grund zur Annahme, dass Gleichung (10.68) bessere Ergebnisse liefert, als wenn von vornherein nur die Teilprozesse betrachtet werden, denn die Gleichung basiert auf der Verwendung aller Faktormatrizen der CP-Dekomposition des Gesamtprozesses (10.67). Der limitierende Faktor der Gleichung besteht somit nicht darin, dass Informationen bezüglich des Gesamtverhaltens grundsätzlich vernachlässigt werden, sondern in der Verwendung einzelner Zustandswahrscheinlichkeitsvektoren $\mathbf{p}_z^{(t)}(k)$, $t = 1, \dots, n$ der Längen N_t , anstelle eines gemeinsamen Zustandsvektors der Länge $N = N_1 N_2 \dots N_n$, was eine effiziente Berechnung sicherstellt.

10.4 Stochastischer Automat mit vektoriellen Ein-, Ausgabe-, und Zustandsfolgen*

Dieser Abschnitt führt die Erkenntnisse der Abschnitte 10.1 und 10.3 zusammen und beschreibt damit einen stochastischen Automaten mit vektoriellen Ein-, Ausgabe-, und Zustandsfolgen, der die Basis der qualitativen Modellierung quantisierter Systeme mit komponentenweise quantisierten Eingängen, Ausgängen und Zuständen bildet.

10.4.1 Stochastischer Prozess*

Die folgende Abbildung zeigt die Interpretation eines stochastischer Prozesses mit vektoriellen Ein-, Ausgabe- und Zustandsfolgen, der durch das Modell \mathcal{A} beschrieben wird.

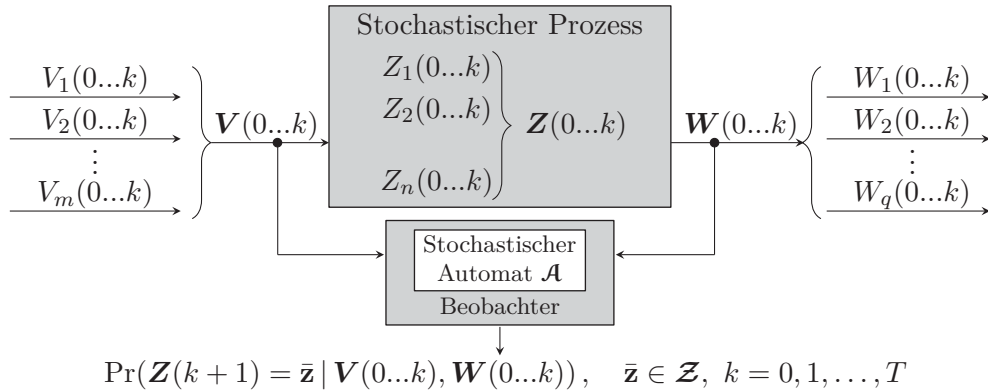


Abbildung 10.7: Stochastischer Prozess mit vektoriellen Ein-, Ausgabe- und Zustandsfolgen

Die Eingabefolgen $\mathbf{V}(0...T)$, Ausgabefolgen $\mathbf{W}(0...T)$ und Zustandsfolgen $\mathbf{Z}(0...T)$ sind dabei durch die Gleichungen (10.4), (10.5) sowie (10.60) definiert. Der stochastische Automat \mathcal{A} stellt somit eine Erweiterung des Automaten mit vektoriellen E/A Folgen (10.6) um vektorielle Zustandsfolgen dar. Die in Abbildung 10.7 eingezeichnete und gesuchte gemeinsame Wahrscheinlichkeit stellt dabei wiederum eine abkürzende Notation dar:

$$\begin{aligned} \Pr(\mathbf{Z}(k+1) = \bar{\mathbf{z}} \mid \mathbf{V}(0...k), \mathbf{W}(0...k)) \\ := \Pr(Z_1(k+1) = \bar{z}_1, \dots, Z_n(k+1) = \bar{z}_n \mid V_1(0..k), \dots, V_m(0..k), W_1(0..k), \dots, W_q(0..k)). \end{aligned} \quad (10.69)$$

10.4.2 Stochastischer Automat*

Definition 10.6 (Stochastischer Automat mit vektoriellen Ein-, Ausgabe- und Zustandsfolgen): *Das 5-Tupel*

$$\mathcal{A} = \left(\mathcal{Z}, \mathcal{V}, \mathcal{W}, \mathcal{L}, \mathbf{p}_0^{(i)} \right), \quad (10.70)$$

mit der Eingabemenge \mathcal{V} , der Ausgabemenge \mathcal{W} , der Zustandsmenge \mathcal{Z} sowie der Verhaltensrelation \mathcal{L} und den initialen Wahrscheinlichkeitsvektoren der Zustände $\mathbf{p}_0^{(i)}$ für $i = 1, \dots, n$, heißt initialer stochastischer Automat mit vektoriellen Ein- und Ausgabe- und Zustandsfolgen.

Dabei ist \mathcal{L} eine Funktion, die die bedingte Wahrscheinlichkeitsverteilung

$$\mathcal{L} : \mathcal{Z} \times \mathcal{W} \times \mathcal{Z} \times \mathcal{V} \longrightarrow [0, 1]$$

repräsentiert und mit

$$\mathcal{L}(\bar{\mathbf{z}}, \mathbf{w}, \mathbf{z}, \mathbf{v}) = \Pr \left(\begin{array}{c} \mathbf{Z}(k+1) = \bar{\mathbf{z}}, \\ \mathbf{W}(k) = \mathbf{w} \end{array} \middle| \begin{array}{c} \mathbf{Z}(k) = \mathbf{z}, \\ \mathbf{V}(k) = \mathbf{v} \end{array} \right) \quad (10.71)$$

die bedingte Wahrscheinlichkeit dafür beschreibt, dass der Automat vom vektoriellen Zustand \mathbf{z} bei der vektoriellen Eingabe \mathbf{v} , in den vektoriellen Folgezustand $\bar{\mathbf{z}}$ übergeht und die vektorielle Ausgabe \mathbf{w} erzeugt. Dabei wird im Folgenden wieder die Notation $\mathcal{L}(\bar{\mathbf{z}}, \mathbf{w} | \mathbf{z}, \mathbf{v})$ verwendet, um zu verdeutlichen, dass es sich hierbei um eine bedingte Wahrscheinlichkeit handelt. Dabei stellt (10.71) wiederum eine Kurznotation der bedingten Wahrscheinlichkeit

$$\begin{aligned} & \mathcal{L}(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q | z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m) \\ &= \Pr \left(\begin{array}{c} Z_1(k+1) = \bar{z}_1, \dots, Z_n(k+1) = \bar{z}_n, \\ W_1(k) = w_1, \dots, W_q(k) = w_q \end{array} \middle| \begin{array}{c} Z_1(k) = z_1, \dots, Z_n(k) = z_n, \\ V_1(k) = v_1, \dots, V_m(k) = v_m \end{array} \right) \end{aligned} \quad (10.72)$$

dar.

10.4.3 Tensorrepräsentation*

Die bedingten Wahrscheinlichkeiten $\mathcal{L}(\bar{\mathbf{z}}, \mathbf{w} | \mathbf{z}, \mathbf{v})$ können nun erneut in einem Verhaltenstensor

$$\mathbf{L} \in [0, 1]^{N_1 \times N_2 \times \dots \times N_n \times Q_1 \times Q_2 \times \dots \times Q_q \times N_1 \times N_2 \times \dots \times N_n \times M_1 \times M_2 \times \dots \times M_m} \quad (10.73)$$

mit Elementen

$$\begin{aligned} & l(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q, z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m) \\ &= \mathcal{L}(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q | z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m) \end{aligned}$$

notiert werden. Ist keine explizite Indexierung der Tensorelemente notwendig, wird auch hier die Kurznotation $l(\bar{\mathbf{z}}, \mathbf{w} | \mathbf{z}, \mathbf{v})$ verwendet. Auf die Angabe einer Beobachtergleichung, mit der die Zustandswahrscheinlichkeiten ohne Dekomposition berechnet werden können, wird an dieser Stelle verzichtet. Grundsätzlich ist diese Berechnung jedoch wieder mit dem kontrahierten Tensorprodukt möglich.

Für die CP-Dekomposition des Verhaltenstensors (10.73) wird im Weiteren wieder angenommen, dass diese exakt und durch die CP-Repräsentation

$$\mathbf{L} = \llbracket \boldsymbol{\lambda}; \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_q, \mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_m \rrbracket \quad (10.74)$$

gegeben ist. Darin sind $\mathbf{A}_i \in \mathbb{R}_{\geq 0}^{N_i \times R}$, $i = 1, \dots, n$ die Faktormatrizen der Folgezustände, $\mathbf{B}_j \in \mathbb{R}_{\geq 0}^{Q_j \times R}$ für $j = 1, \dots, q$ die Faktormatrizen der Ausgaben, $\mathbf{C}_s \in \mathbb{R}_{\geq 0}^{N_s \times R}$ für $s = 1, \dots, n$ die Faktormatrizen der Zustände und $\mathbf{D}_t \in \mathbb{R}_{\geq 0}^{N_t \times R}$ für $t = 1, \dots, m$ die Faktormatrizen der Eingaben. Außerdem ist R der Rang der Dekomposition und $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$ der Gewichtungsvektor.

10.4.4 Betrachtung stochastischer Teilprozesse*

Für die im folgenden vorgestellte Beobachtergleichung für den stochastischen Automaten (10.70), muss erneut die stochastische Unabhängigkeit der Zufallsvariablen der Zustände vorausgesetzt werden. Für den Beweis dieser Gleichung wird dabei vereinfachend angenommen, dass das Verhalten des Gesamtsystems in Abbildung (10.7) durch eine Anzahl von n stochastischen Teilprozessen, die durch die stochastischen Automaten $\mathcal{A}_i = (\mathcal{Z}_i, \mathcal{V}_i, \mathcal{W}_i, \mathcal{L}_i, \mathbf{p}_0^{(i)})$, $i = 1, \dots, n$ beschrieben werden, dargestellt

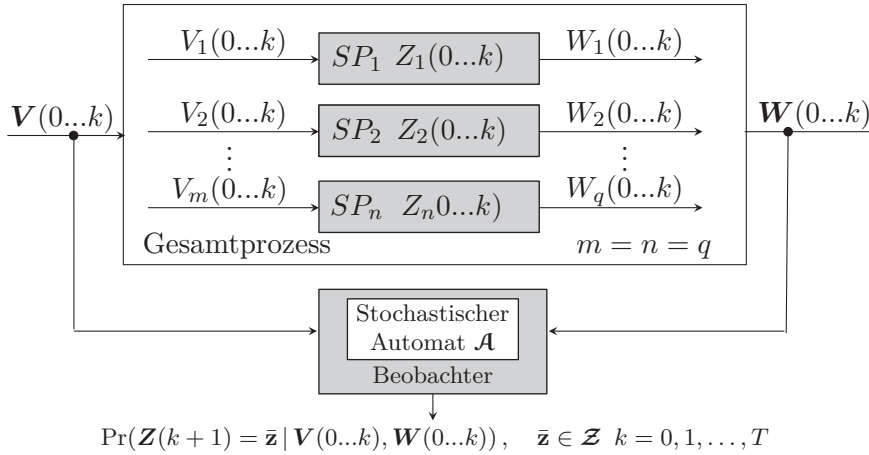


Abbildung 10.8: Stochastische Teilprozesse mit Ein-, Ausgaben und Zuständen

werden kann (s. Abbildung 10.8). Im Folgenden gilt somit die Annahme, dass für die Anzahlen von Eingabefolgen m , Zustandsfolgen n und Ausgabefolgen q der Zusammenhang $m = n = q$ gilt.¹

Die Zustandswahrscheinlichkeiten der stochastischen Teilprozesse SP_1, \dots, SP_n sind entsprechend für $i = 1, \dots, n$ durch

$$\begin{aligned} k \geq 0 : \quad \mathbf{p}_z^{(i)}(k+1) &= \frac{\mathbf{L}_i(:, w_i(k), :, v_i(k)) \cdot \mathbf{p}_z^{(i)}(k)}{\|\mathbf{L}_i(:, w_i(k), :, v_i(k)) \cdot \mathbf{p}_z^{(i)}(k)\|_1} \in [0, 1]^{N_i}, \\ k = 0 : \quad \mathbf{p}_z^{(i)}(0) &= \mathbf{p}_0^{(i)} \end{aligned} \quad (10.75)$$

gegeben. In Gleichung (10.75) ist $\mathbf{L}_i(:, w_i(k), :, v_i(k)) \in [0, 1]^{N_i \times N_i}$ eine Submatrix des i -ten Verhaltenstensors $\mathbf{L}_i \in [0, 1]^{N_i \times Q_i \times N_i \times M_i}$ und des i -ten stochastischen Teilprozesses. Jeder Zustandswahrscheinlichkeitsvektor $\mathbf{p}_z^{(i)}(k+1)$ repräsentiert damit die bedingte Zustandswahrscheinlichkeitsverteilung $\Pr(Z_i(k+1) = \bar{z}_i | V_i(0..k), W_i(0..k)), \bar{z}_i \in \mathcal{Z}_i$.

Die Berechnung der gemeinsamen Wahrscheinlichkeit (10.69) folgt aus dem Kronecker-Produkt

$$\mathbf{p}_z(k+1) = \bigotimes_{i=1}^n \mathbf{p}_z^{(i)}(k+1) \in [0, 1]^N, \quad (10.76)$$

welches in einem Vektor der Länge $N = N_1 N_2 \dots N_n$ mit Komponenten

$$\begin{aligned} p_z(k+1, \bar{z}) &= \Pr(Z_1(k+1) = \bar{z}_1 | V_1(0..k), W_1(0..k)) \dots \Pr(Z_n(k+1) = \bar{z}_n | V_n(0..k), W_n(0..k)) \\ &= \Pr(Z_1(k+1) = \bar{z}_1, \dots, Z_n(k+1) = \bar{z}_n | V_1(0..k), \dots, V_n(0..k), W_1(0..k), \dots, W_n(0..k)) \end{aligned} \quad (10.77)$$

für $\bar{z} = \bar{z}_1 \dots \bar{z}_n \in \{1, 2, \dots, N\}$ resultiert. Für die Übergangswahrscheinlichkeiten des Gesamtsystems (10.72), die durch die Elemente des Verhaltenstensors (10.73) beschrieben werden, gilt nun aufgrund der vorausgesetzten stochastischen Unabhängigkeit die Gleichheit [95]

$$\begin{aligned} l(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q, z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m) \\ = l_1(\bar{z}_1, w_1, z_1, v_1) \cdot l_2(\bar{z}_2, w_2, z_2, v_2) \dots l_n(\bar{z}_n, w_q, z_n, v_m). \end{aligned} \quad (10.78)$$

¹ Für die praktische Anwendung stellt die Annahme $n = m = q$ keine Einschränkung dar, da in der Anzahl »fehlende« Folgen von Automateingaben-, -ausgaben oder -zuständen stets durch eine entsprechende Folge, deren Elemente auf einer einelementigen Ein-, Ausgabe oder Zustandsmenge definiert sind, kompensiert werden können. Im Verhaltenstensor können die entsprechenden Modi dann durch Dimensionen der Länge Eins ergänzt werden, die keinen Informationsgehalt besitzen.

Dabei sind $l_i(\bar{z}_i, w_i, z_i, v_i)$ die Elemente des i -ten Verhaltenstensors $L_i \in [0, 1]^{N_i \times N_i}$ des i -ten stochastischen Teilprozesses. Die Berechnung der gemeinsamen Wahrscheinlichkeiten nach (10.78) kann dabei als eine Komposition des Verhaltenstensors L interpretiert werden, die entsprechend die stochastische Unabhängigkeit impliziert. Ein solches Zusammenführen der stochastischen Automaten \mathcal{A}_i zu einer Gesamtrepräsentation \mathcal{A} , ist hierbei vergleichbar mit einer einfachen parallelen Komposition stochastischer Automaten, die bei der Verwendung von Automatenetzen Anwendung findet (s. hierzu [95]).

Abschließend kann nun eine Beobachtergleichung basierend auf der CP-Repräsentation (10.74) aufgestellt werden.

10.4.5 Beobachtung mit der CP-Repräsentation*

Proposition 10.4: *Für eine exakte CP-Dekomposition, exakte Ein- und Ausgabewahrscheinlichkeitsvektoren und stochastisch unabhängige Zustände gilt unter der Annahme $m = n = q$ die Gleichung*

$$\begin{aligned} & L_i(:, w_i(k), :, v_i(k)) \cdot \mathbf{p}_z^{(i)}(k) \\ &= \mathbf{A}_i \cdot \left(\lambda \otimes \left(\bigotimes_{e \neq i}^n \sum_{h_e=1}^{N_e} (\mathbf{a}_{h_e \cdot}^{(e)})^\top \right) \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\bigotimes_{s=1}^n \mathbf{C}_s^\top \cdot \mathbf{p}_z^{(s)}(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right) \end{aligned} \quad (10.79)$$

für $i = 1, \dots, n$. Dabei ist $\mathbf{p}_z^{(s)}(k) \in [0, 1]^{N_s}$ für $s = 1, \dots, n$, $\mathbf{p}_w^{(j)}(k) \in \{0, 1\}^{Q_j}$ für $j = 1, 2, \dots, q$ sowie $\mathbf{p}_v^{(t)}(k) \in \{0, 1\}^{M_t}$ für $t = 1, 2, \dots, m$.

Beweis: Der Beweis der Proposition 10.4 befindet sich im Anhang B.9.

Lemma 10.2 (CP-Beobachtergleichung für stochastische Automaten mit vektoriellen Ein- und Ausgaben und Zuständen): *Durch Einsetzen von Gleichung (10.4) in den Zähler und Nenner von Gleichung (10.75) folgt mit*

$$\begin{aligned} & k \geq 0 : \mathbf{p}_z^{(i)}(k+1) \\ &= \frac{\mathbf{A}_i \cdot \left(\lambda \otimes \left(\bigotimes_{e \neq i}^n \sum_{h_e=1}^{N_e} (\mathbf{a}_{h_e \cdot}^{(e)})^\top \right) \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\bigotimes_{s=1}^n \mathbf{C}_s^\top \cdot \mathbf{p}_z^{(s)}(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right)}{\left\| \mathbf{A}_i \cdot \left(\lambda \otimes \left(\bigotimes_{e \neq i}^n \sum_{h_e=1}^{N_e} (\mathbf{a}_{h_e \cdot}^{(e)})^\top \right) \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \mathbf{p}_w^{(j)}(k) \right) \otimes \left(\bigotimes_{s=1}^n \mathbf{C}_s^\top \cdot \mathbf{p}_z^{(s)}(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \mathbf{p}_v^{(t)}(k) \right) \right\|_1} \end{aligned} \quad (10.80)$$

$$k = 0 : \mathbf{p}_z^{(i)}(0) = \mathbf{p}_0^{(i)}$$

die CP-Beobachtergleichung für stochastische Automaten nach Definition 10.6. Damit können für eine exakte CP-Dekomposition, exakte Ein- und Ausgaben, stochastisch unabhängige Zustände und unter der Annahme $m = n = q$, die Zustandswahrscheinlichkeiten der stochastischen Teilprozesse für $i = 1, 2, \dots, n$ bestimmt werden. Dabei ist $\mathbf{p}_z^{(s)}(k) \in [0, 1]^{N_s}$ für $s = 1, 2, \dots, n$, $\mathbf{p}_w^{(j)}(k) \in \{0, 1\}^{Q_j}$ für $j = 1, 2, \dots, q$ sowie $\mathbf{p}_v^{(t)}(k) \in \{0, 1\}^{M_t}$ für $t = 1, 2, \dots, m$.

Die gemeinsamen Wahrscheinlichkeiten $\Pr(\mathbf{Z}(k+1) = \bar{\mathbf{z}} \mid \mathbf{V}(0 \dots k), \mathbf{W}(0 \dots k))$ sind dann durch das Kronecker-Produkt der Vektoren $\mathbf{p}_z^{(i)}(k+1)$ nach Gleichung (10.76), beziehungsweise (10.77) gegeben.

Vermutung 10.1: *Zu Lemma 10.2 soll angemerkt werden, dass die Gleichung auch auf Systeme wie in Abbildung 10.7, die über beliebige und nicht identische Anzahlen von Zuständen n , Eingaben m und*

Ausgaben q verfügen, angewandt werden kann (s. Fußnote 1, Seite 195). Die Gleichung führt dabei für stochastisch unabhängige Zustände, exakte Ein- und Ausgaben sowie eine exakte CP-Dekomposition, auf die korrekten gemeinsamen Zustandswahrscheinlichkeiten (10.69). Diese Vermutung liegt darin begründet, dass die Gleichung (10.80) offenbar die bedingten Wahrscheinlichkeiten

$$\begin{aligned} & \Pr(Z_i(k+1) = \bar{z}_i \mid V_1(0..k), \dots, V_m(0..k), W_1(0..k), \dots, W_q(0..k)) \\ &= \Pr(Z_i(k+1) = \bar{z}_i \mid \mathbf{V}(0..k), \mathbf{W}(0..k)), \quad \bar{z}_i \in \mathcal{Z}_i \end{aligned} \quad (10.81)$$

für $i = 1, \dots, n$ bestimmt. Dies ist die Wahrscheinlichkeit dafür, dass die i -te Zufallsvariable Z_i den Zustand $\bar{z}_i \in \mathcal{Z}_i$ zur Zeit $k+1$ annimmt, unter der Bedingung, dass die Eingabe- und Ausgabe-folgen $\mathbf{V}(0..k), \mathbf{W}(0..k)$ bis zur Zeit k durchlaufen wurden. Für stochastisch unabhängige Zustände erhält man dann aus dem Produkt der bedingten Zustandswahrscheinlichkeiten (10.81) die gesuchte gemeinsame Wahrscheinlichkeit

$$\begin{aligned} & \prod_{i=1}^n \Pr(Z_i(k+1) = \bar{z}_i \mid V_1(0..k), \dots, V_m(0..k), W_1(0..k), \dots, W_q(0..k)) \\ &= \Pr(Z_1(k+1) = \bar{z}_1, \dots, Z_n(k+1) = \bar{z}_n \mid V_1(0..k), \dots, V_m(0..k), W_1(0..k), \dots, W_q(0..k)) \\ &= \Pr(\mathbf{Z}(k+1) = \bar{\mathbf{z}} \mid \mathbf{V}(0..k), \mathbf{W}(0..k)) . \end{aligned}$$

Die Vermutung lässt dabei die Annahme zu, dass sich im Falle stochastisch abhängiger Zustände, mit der Beobachtergleichung (10.80) bessere Ergebnisse erzielen lassen, als durch die ausschließliche Betrachtung der stochastischen Teilprozesse. Diese Annahme ist darin begründet, dass Gleichung (10.80) die Zustandswahrscheinlichkeiten (10.81) für jedes $i \in \{1, \dots, n\}$ anhand aller Faktormatrizen der CP-Repräsentation des Gesamtsystems (10.74) und unter Berücksichtigung aller Ein- und Ausgabe-folgen bestimmt (vgl. Abbildung 10.7). Bei der Betrachtung der stochastischen Teilprozesse werden die Zustandswahrscheinlichkeiten des i -ten Teilprozesses hingegen nur unter Berücksichtigung der diesem Prozess zugehörigen Ein- und Ausgabefolgen berechnet (vgl. Abbildung 10.8).

Die mit (10.80) berechneten Zustandswahrscheinlichkeitsvektoren repräsentieren damit für exakte Ein- und Ausgaben, entsprechend Gleichung (10.81), die Verteilungen

$$\mathbf{p}_z^{(i)}(k+1) = \begin{pmatrix} \Pr(Z_i(k+1) = 1 \mid \mathbf{V}(0..k), \mathbf{W}(0..k)) \\ \Pr(Z_i(k+1) = 2 \mid \mathbf{V}(0..k), \mathbf{W}(0..k)) \\ \vdots \\ \Pr(Z_i(k+1) = N \mid \mathbf{V}(0..k), \mathbf{W}(0..k)) \end{pmatrix} \in [0, 1]^{N_i} ,$$

Aus der Beobachtergleichung (10.80) ergeben sich dann für $k = 0, 1, \dots, T$ die Vektorfolgen

$$\mathbf{P}_z^{(i)}(0..T+1 \mid \mathbf{V}(0..T), \mathbf{W}(0..T)) = (\mathbf{p}_z^{(i)}(0), \mathbf{p}_z^{(i)}(1), \mathbf{p}_z^{(i)}(2), \dots, \mathbf{p}_z^{(i)}(T+1)) . \quad (10.82)$$

Da in (10.80) die Ein- und Ausgaben wieder durch Wahrscheinlichkeitsvektoren dargestellt werden, wird im Folgenden wieder eine allgemeine Notation des Zustandsvektors bevorzugt, die entsprechend durch

$$\mathbf{p}_z^{(i)}(k+1) = \begin{pmatrix} \Pr(Z_i(k+1) = 1 \mid \mathbf{P}_v^{(1)}(0..k), \dots, \mathbf{P}_v^{(m)}(0..k), \mathbf{P}_w^{(1)}(0..k), \dots, \mathbf{P}_w^{(q)}(0..k)) \\ \Pr(Z_i(k+1) = 2 \mid \mathbf{P}_v^{(1)}(0..k), \dots, \mathbf{P}_v^{(m)}(0..k), \mathbf{P}_w^{(1)}(0..k), \dots, \mathbf{P}_w^{(q)}(0..k)) \\ \vdots \\ \Pr(Z_i(k+1) = N \mid \mathbf{P}_v^{(1)}(0..k), \dots, \mathbf{P}_v^{(m)}(0..k), \mathbf{P}_w^{(1)}(0..k), \dots, \mathbf{P}_w^{(q)}(0..k)) \end{pmatrix} , \quad (10.83)$$

und für $k = 0, \dots, T$ durch die Folge

$$\begin{aligned} & \mathbf{P}_z^{(i)}(0..T+1 \mid \mathbf{P}_v^{(1)}(0..T), \dots, \mathbf{P}_v^{(m)}(0..T), \mathbf{P}_w^{(1)}(0..T), \dots, \mathbf{P}_w^{(q)}(0..T)) \\ &= (\mathbf{p}_z^{(i)}(0), \mathbf{p}_z^{(i)}(1), \mathbf{p}_z^{(i)}(2), \dots, \mathbf{p}_z^{(i)}(T+1)) \end{aligned} \quad (10.84)$$

gegeben ist. Dies lässt offen, ob die Ein- und Ausgaben exakt oder mit Unsicherheiten behaftet sind.

10.5 Qualitatives Modell für komponentenweise quantisierte Ein-, Ausgänge und Zustände*

In Erweiterung zu Tabelle 10.1, zeigt die Tabelle 10.3 zusätzlich den Speicherbedarf γ_{CP} des Verhaltenstensors $\tilde{L} \in [0, 1]^{6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6}$ eines qualitativen Modells des Erhitzers aus Abschnitt 5.3 für eine Rang-50 Approximation, wobei nicht nur die Ein- und Ausgänge, sondern auch die Zustände komponentenweise quantisiert sind, wodurch sich der Speicherbedarf weiter reduzieren lässt. Wie diese Darstellung des Verhaltenstensors zur qualitativen Fehlerdetektion verwendet werden kann, wird in diesem Abschnitt beschrieben.

Tabelle 10.3: Speicherbedarf γ und γ_{CP} für verschiedendimensionale Verhaltenstensen qualitativer Modelle des Erhitzers mit den Modellparametern aus Tabelle 5.11

Anzahl qualitativer Größen	Dimensionalität des Verhaltenstensors	Speicherbedarf γ	Speicherbedarf γ_{CP} ($R = 50$)
$N = Q = M = 36$	$\tilde{L} \in [0, 1]^{36 \times 36 \times 36 \times 36}$	1 679 616	7250
$N = 36, M_1 = M_2 = 6, Q_1 = Q_2 = 6$	$\tilde{L} \in [0, 1]^{36 \times 6 \times 6 \times 36 \times 6 \times 6}$	1 679 616	4850
$N_1 = N_2 = 6, M_1 = M_2 = 6, Q_1 = Q_2 = 6$	$\tilde{L} \in [0, 1]^{6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6}$	1 679 616	2450

10.5.1 Quantisiertes System*

Die Abbildung 10.9 zeigt die Interpretation eines quantisierten Systems bei dem in Erweiterung zu Abbildung 10.3 zusätzlich die Zustände komponentenweise quantisiert sind. Die qualitativen Zustände sind somit durch den komponentenweise quantisierten Zustand $[\mathbf{x}(k)] = \mathbf{z} \in \mathcal{X}$ gegeben, wobei der qualitative Zustandsvektor \mathbf{z} die Elemente $z_s \in \mathcal{X}_s := \{1, \dots, N_s\}$ für $s = 1, \dots, n$ besitzt (vgl. Abschnitt 10.2.1). Für die qualitativen Eingangsvektoren $[\mathbf{u}(k)] = \mathbf{v}$ und Ausgangsvektoren $[\mathbf{y}(k)] = \mathbf{w}$ gelten analog die Beziehungen aus Abschnitt 10.2.2.

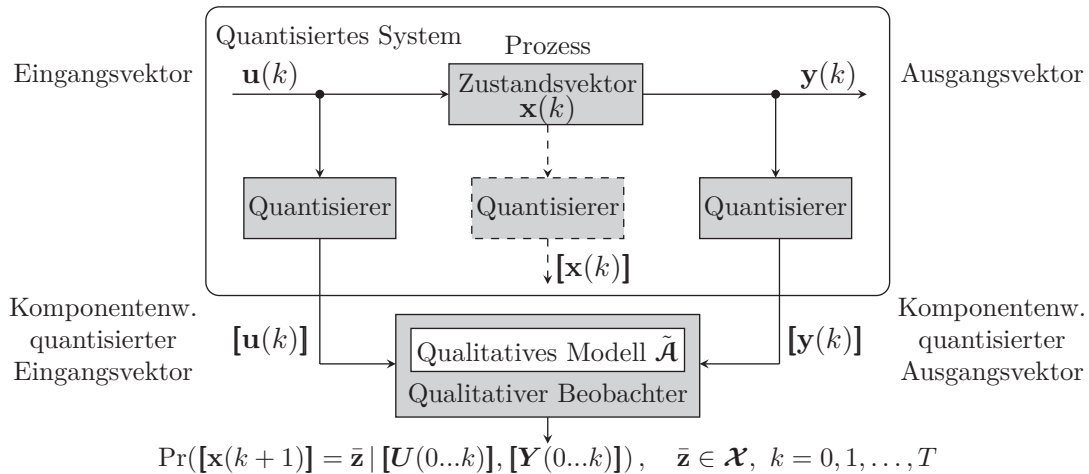


Abbildung 10.9: Quantisiertes System mit komponentenweise quantisierten Ein-, Ausgängen und Zuständen

Die Beschreibung des quantisierten Systems nach auf Definition 10.6, durch einen initialen stochastischen Automaten

$$\tilde{\mathcal{A}} = (\mathcal{X}, \mathcal{U}, \mathcal{Y}, \tilde{\mathcal{L}}, \mathbf{p}_0) \quad (10.85)$$

mit der Verhaltensrelation

$$\tilde{\mathcal{L}}(\bar{\mathbf{z}}, \mathbf{w} | \mathbf{z}, \mathbf{v}) = \Pr \left(\begin{array}{l} [\mathbf{x}(k+1)] = \bar{\mathbf{z}}, \\ [\mathbf{y}(k)] = \mathbf{w} \end{array} \middle| \begin{array}{l} [\mathbf{x}(k)] = \mathbf{z}, \\ [\mathbf{u}(k)] = \mathbf{v} \end{array} \right) \quad (10.86)$$

welche die bedingte Wahrscheinlichkeit dafür angibt, dass das quantisierte System beim qualitativen Eingang $[\mathbf{u}(k)] = \mathbf{v}$ vom qualitativen Zustand $[\mathbf{x}(k)] = \mathbf{z}$ in den qualitativen Folgezustand $[\mathbf{x}(k+1)] = \bar{\mathbf{z}}$ wechselt und den qualitativen Ausgang $[\mathbf{y}(k)] = \mathbf{w}$ erzeugt. Dies ist wiederum eine abkürzende Schreibweise für die Notation

$$\begin{aligned} \tilde{\mathcal{L}}(\bar{z}_1, \dots, \bar{z}_n, w_1, \dots, w_q | z_1, \dots, z_n, v_1, \dots, v_m) \\ = \Pr \left(\begin{array}{l} [x_1(k+1)] = \bar{z}_1, \dots, [x_n(k+1)] = \bar{z}_n, \\ [y_1(k)] = w_1, \dots, [y_q(k)] = w_q \end{array} \middle| \begin{array}{l} [x_1(k)] = z_1, \dots, [x_n(k)] = z_n, \\ [u_1(k)] = v_1, \dots, [u_m(k)] = v_m \end{array} \right). \end{aligned} \quad (10.87)$$

Die bedingten Wahrscheinlichkeiten der Verhaltensrelation werden dabei wieder durch die in Abschnitt 4.3.2 beschriebene Methode der stochastischen qualitativen Identifikation generiert. Anstelle der Tupel in Gleichung (4.22) verwendet man die Tupel komponentenweise quantisierter Werte

$$\begin{aligned} [\Theta(k)] &= ([\mathbf{x}(k+1)], [\mathbf{y}(k)], [\mathbf{x}(k)], [\mathbf{u}(k)]) \\ &= ([x_1(k+1)], \dots, [x_n(k+1)], [y_1(k)], \dots, [y_q(k)], [x_1(k)], \dots, [x_n(k)], [u_1(k)], \dots, [u_m(k)]). \end{aligned} \quad (10.88)$$

Die weitere Vorgehensweise folgt entsprechend den Gleichungen (4.23)–(4.27), womit man in (4.27) die Approximation $\tilde{\mathcal{L}}(\bar{z}_1, \dots, \bar{z}_n, w_1, \dots, w_q | z_1, \dots, z_n, v_1, \dots, v_m)$ der bedingten Wahrscheinlichkeiten der Verhaltensrelation erhält. Diese werden erneut in einem Verhaltenstensor

$$\tilde{\mathbf{L}} \in [0, 1]^{N_1 \times \dots \times N_n \times Q_1 \times \dots \times Q_q \times N_1 \times \dots \times N_n \times M_1 \times \dots \times M_m} \quad (10.89)$$

gespeichert, mit Elementen

$$\begin{aligned} \tilde{l}(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q, z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m) \\ = \tilde{\mathcal{L}}(\bar{z}_1, \bar{z}_2, \dots, \bar{z}_n, w_1, w_2, \dots, w_q | z_1, z_2, \dots, z_n, v_1, v_2, \dots, v_m). \end{aligned} \quad (10.90)$$

10.5.2 CP-Algorithmus zur qualitativen Fehlererkennung*

Auch hier wird wieder angenommen, dass die CP-Dekomposition des Verhaltenstensors (10.89) nur eine Näherung darstellt, die durch die CP-Repräsentation

$$\hat{\tilde{\mathbf{L}}} = [\![\boldsymbol{\lambda}; \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_q, \mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_m]\!] , \quad (10.91)$$

gegeben ist, wobei $\hat{\tilde{\mathbf{L}}} \in \mathbb{R}_{\geq 0}^{N_1 \times N_2 \times \dots \times N_n \times Q_1 \times Q_2 \times \dots \times Q_q \times N_1 \times N_2 \times \dots \times N_n \times M_1 \times M_2 \times \dots \times M_m}$ gilt. Die Beobachtergleichung für das quantisierte System in Abbildung 10.9 basiert entsprechend auf der Beobachtergleichung des Lemmas 10.2. Für den Beobachteralgorithmus zur qualitativen Fehlerdetektion, wird der Zähler der Beobachtergleichung mit

$$\rho^{(i)}(k) := \mathbf{A}_i \cdot \left(\boldsymbol{\lambda} \otimes \left(\bigotimes_{e \neq i}^n \sum_{h_e=1}^{N_e} (\mathbf{a}_{h_e}^{(e)})^\top \right) \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \tilde{\mathbf{p}}_w^{(j)}(k) \right) \otimes \left(\bigotimes_{s=1}^n \mathbf{C}_s^\top \cdot \hat{\tilde{\mathbf{p}}}_z^{(s)}(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \tilde{\mathbf{p}}_v^{(t)}(k) \right) \right) \quad (10.92)$$

und der Nenner mit

$$\hat{b}^{(i)}(k) := \left\| \mathbf{A}_i \cdot \left(\boldsymbol{\lambda} \otimes \left(\bigotimes_{e \neq i}^n \sum_{h_e=1}^{N_e} (\mathbf{a}_{h_e}^{(e)})^\top \right) \otimes \left(\bigotimes_{j=1}^q \mathbf{B}_j^\top \cdot \tilde{\mathbf{p}}_w^{(j)}(k) \right) \otimes \left(\bigotimes_{s=1}^n \mathbf{C}_s^\top \cdot \hat{\tilde{\mathbf{p}}}_z^{(s)}(k) \right) \otimes \left(\bigotimes_{t=1}^m \mathbf{D}_t^\top \cdot \tilde{\mathbf{p}}_v^{(t)}(k) \right) \right\|_1 \quad (10.93)$$

abgekürzt, wobei $\rho^{(i)}(k) \in \mathbb{R}_{\geq 0}^N$ und $\hat{b}^{(i)}(k) \in \mathbb{R}_{\geq 0}$ gilt. Der Beobachteralgorithmus 10.3 für qualitative Modelle mit komponentenweise quantisierten Ein-, Ausgängen und Zuständen kann somit, wie auf Seite 200 gezeigt, definiert werden.

Algorithmus 10.3: CP-Algorithmus zur qualitativen Fehlererkennung mit komponentenweise quantisierten Ein-, Ausgängen und Zuständen

Gegeben: Qualitatives Modell $\tilde{\mathcal{A}}$, Gl. (10.85) mit Verhaltenstensor in CP-Repräsentation $\hat{\mathbf{L}} = \llbracket \boldsymbol{\lambda}; \mathbf{A}_1, \dots, \mathbf{A}_n, \mathbf{B}_1, \dots, \mathbf{B}_q, \mathbf{C}_1, \dots, \mathbf{C}_n, \mathbf{D}_1, \dots, \mathbf{D}_m \rrbracket$, Gl. (10.91)
 Folgen von Wahrscheinlichkeitsvektoren qualitativer Eingänge $\tilde{\mathbf{P}}_v^{(t)}(0 \dots T)$ für $t = 1, \dots, m$
 Folgen von Wahrscheinlichkeitsvektoren qualitativer Ausgänge $\tilde{\mathbf{P}}_w^{(j)}(0 \dots T)$ für $j = 1, \dots, q$

Initialisiere: $k = 0$, $\hat{\mathbf{p}}_z^{(i)}(0) := \tilde{\mathbf{p}}_0^{(i)} \forall i$

Iteriere:

- 1 **solange** $0 \leq k \leq T$
- 2 **für** $i = 1, 2, \dots, n$
- 3 berechne $\boldsymbol{\rho}^{(i)}(k)$, Gl. (10.92)
- 4 berechne $\hat{b}^{(i)}(k)$, Gl. (10.93)
- 5 berechne $\hat{b}_m^{(i)}(k)$, Gl. (4.46) und $\hat{b}_b^{(i)}(k)$, Gl. (4.48)
- 6 **wenn** $\hat{b}^{(i)}(k) = 0$ **dann**
- 7 setze $\hat{\mathbf{p}}_z^{(i)}(k+1) = \tilde{\mathbf{p}}_0^{(i)}$
- 8 wenn $i < n$ fahre fort mit Schritt 2, sonst setze $k := k+1$
 und fahre fort mit Schritt 1
- 9 **sonst**
- 10 berechne $\hat{\mathbf{p}}_z^{(i)}(k+1) = \frac{\boldsymbol{\rho}^{(i)}(k)}{\hat{b}^{(i)}(k)}$
- 11 wenn $i < n$ fahre fort mit Schritt 2, sonst setze $k := k+1$
 und fahre fort mit Schritt 1

Ergebnis: $\hat{\mathbf{P}}_z^{(i)}(0 \dots T+1 | \tilde{\mathbf{P}}_v^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0 \dots T), \tilde{\mathbf{P}}_w^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0 \dots T))$,
 $\hat{B}^{(i)}(0 \dots T | \tilde{\mathbf{P}}_v^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_v^{(m)}(0 \dots T), \tilde{\mathbf{P}}_w^{(1)}(0 \dots T), \dots, \tilde{\mathbf{P}}_w^{(q)}(0 \dots T))$,
 $\hat{B}_m^{(i)}(0 \dots T)$, Gl. (4.47),
 $\hat{B}_b^{(i)}(0 \dots T)$, Gl. (4.49),
 für $i = 1, 2, \dots, n$.

Zu diesem Algorithmus sei nochmals erwähnt, dass er aus letztlich drei Gründen die realen qualitativen Zustandswahrscheinlichkeiten des zugrundeliegenden Systems nicht exakt beschreiben kann:

1. Das quantisierte System besitzt die Markow-Eigenschaft nicht
2. Die CP-Dekomposition ist nicht exakt
3. Die Zustände sind nicht stochastisch unabhängig

Dennoch besitzt dieser Algorithmus einen Vorteil gegenüber den bisher vorgestellten Algorithmen, was am folgenden Anwendungsbeispiel gezeigt wird.

Beispiel 10.4 (Fehlererkennung mit der CP-Repräsentation und komponentenweise quantisierten Ein-, Ausgängen und Zuständen): Die Basis dieses Beispiels bildet der stochastische Automat (10.85) sowie der Erhitzer aus dem Anwendungsbeispiel in Abschnitt 5.3 mit den Modellparametern aus

Tabelle 5.11. Der Erhitzer wird dabei ausschließlich über komponentenweise quantisierte Signale beschrieben. Das heißt, das qualitative Modell besitzt die qualitativen Eingänge $v_1 \in \mathcal{U}_1 := \{1, \dots, M_1\}$ und $v_2 \in \mathcal{U}_2 := \{1, \dots, M_2\}$, die qualitativen Zustände $z_1 \in \mathcal{X}_1 := \{1, \dots, N_1\}$, $z_2 \in \mathcal{X}_2 := \{1, \dots, N_2\}$ und die qualitativen Ausgänge $w_1 \in \mathcal{Y}_1 := \{1, \dots, Q_1\}$ und $w_2 \in \mathcal{Y}_2 := \{1, \dots, Q_2\}$, wobei entsprechend Tabelle 5.11 $M_1 = M_2 = Q_1 = Q_2 = N_1 = N_2 = 6$ gilt. Der Verhaltenstensor nach Gleichung (10.89) ist damit durch

$$\tilde{\mathbf{L}} \in [0, 1]^{6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6} \quad (10.94)$$

gegeben und besitzt die CP-Repräsentation

$$\hat{\mathbf{L}} = [\boldsymbol{\lambda}; \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2, \mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2] \in \mathbb{R}_{\geq 0}^{6 \times 6 \times 6 \times 6 \times 6 \times 6 \times 6} \quad (10.95)$$

mit den Faktormatrizen

$$\begin{aligned} \mathbf{A}_1 &\in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{A}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{B}_1 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{B}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \\ \mathbf{C}_1 &\in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{C}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{D}_1 \in \mathbb{R}_{\geq 0}^{6 \times 50}, \quad \mathbf{D}_2 \in \mathbb{R}_{\geq 0}^{6 \times 50} \end{aligned}$$

und dem Gewichtsvektor $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{50}$. Die CP-Dekomposition wurde also wieder mit einem Dekompositionsrank von $R = 50$ berechnet, wobei wie im Anwendungsbeispiel 10.3, der MU-Algorithmus (E) aus Tabelle 8.1 in PYTHONTM verwendet wurde.

Die Beobachtung und Fehlererkennung erfolgt dabei mit dem Algorithmus 10.3. Im Unterschied zu den vorherigen Anwendungsbeispielen bestimmt dieser Algorithmus die Zustandswahrscheinlichkeiten für jeden Zustand einzeln, sodass man für dieses Beispiel zwei Zustandsvektoren $\hat{\mathbf{p}}_z^{(1)}(k) \in \mathbb{R}_{\geq 0}^6$ und $\hat{\mathbf{p}}_z^{(2)}(k) \in \mathbb{R}_{\geq 0}^6$ erhält. Es sei an dieser Stelle nochmals darauf hingewiesen, dass man selbst für exakte Dekompositionen aus dem Produkt dieser Vektoren $\hat{\mathbf{p}}_z^{(1)}(k) \otimes \hat{\mathbf{p}}_z^{(2)}(k) = \hat{\mathbf{p}}_z(k) \in \mathbb{R}_{\geq 0}^{36}$ nur dann die gleichen Zustandswahrscheinlichkeiten wie in den vorangegangenen Beispielen erhält, wenn die Zustände stochastisch unabhängig sind, was jedoch in der Praxis nicht der Fall ist.

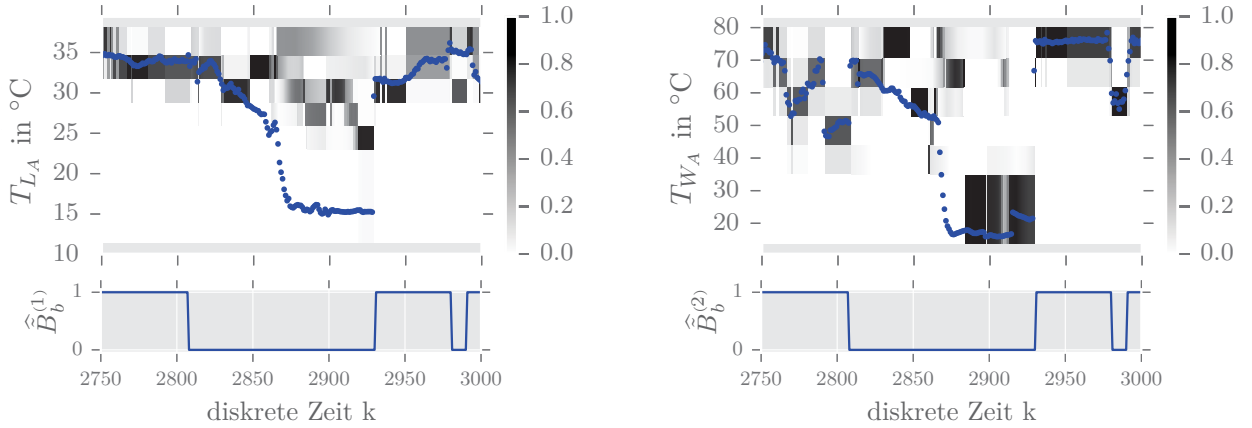


Abbildung 10.10: Zustandswahrscheinlichkeiten der Luft- und Wasseraustrittstemperaturen sowie deren Verhaltenssignale $\hat{B}_b^{(1)}$ und $\hat{B}_b^{(2)}$

Abbildung 10.10 zeigt die mit dem Algorithmus 10.3 berechneten Zustandswahrscheinlichkeiten der Luftaustritts- und Wasseraustrittstemperaturen. In diesem Fall handelt es sich um die unmittelbare grafische Darstellung der Vektoren $\hat{\mathbf{p}}_z^{(1)}(k)$ sowie $\hat{\mathbf{p}}_z^{(2)}(k)$. Die jeweils unterhalb der Zustandswahrscheinlichkeitsverteilungen dargestellten Verhaltenssignale $\hat{B}_b^{(1)}$ und $\hat{B}_b^{(2)}$ basieren entsprechend auf den Normierungssummen der einzelnen Zustandsvektoren und der Anwendung des gleitenden Mittelwertes sowie des Schwellwerts für die Fehlerdetektion, welche wieder die Werte $\tau = 1\%$ und $\phi = 2$ besitzen.

Wie die Abbildungen zeigen, ist auch mit diesem Beobachteralgorithmus die Erkennung des nicht nominalen Betriebs des Erhitzers anhand der Verhaltenssignale möglich. Wie die Tabelle 10.4 zeigt, ermöglicht die CP-Repräsentation (10.95) eine Reduktion des Speicherbedarfs um einen Faktor von über 685, was somit der größten Minimierung aller hier vorgestellten Darstellungen des Verhaltenstensors entspricht. Dabei bietet der Beobachteralgorithmus 10.3 aus praktischer Sicht einen weiteren Vorteil gegenüber allen zuvor beschriebenen Beobachteralgorithmen. Während bei allen anderen Beobachteralgorithmen in den Anwendungsbeispielen ein Zustandsvektor der Länge $N = 36$ berechnet wird, werden hier nur zwei Vektoren der Länge $N_1 = 6, N_2 = 6$ berechnet. Insbesondere bei Systemen mit einer deutlich größeren Anzahl an Zuständen, ergibt sich daraus ein erheblicher Vorteil in Bezug auf die Komplexität der Berechnungen, die in jedem Zeitschritt ausgeführt werden müssen.

Tabelle 10.4: Speicherbedarf für verschiedene Repräsentationen des Verhaltenstensors qualitativer Modelle des Erhitzers aus Abschnitt 5.3 und den Modellparametern aus Tabelle 5.11

Repräsentation des Verhaltenstensors	Fehlererk. Algorithmus	Speicher- bedarf	Reduktions- faktor	RMSD \tilde{B} zu \hat{B}
$\tilde{L} \in [0, 1]^{36 \times 36 \times 36 \times 36}$, Gl. (9.15)	7.1	$\gamma = 1\,679\,616$	0	0
$\hat{L} = \langle E, F, G, H \rangle$, Gl. (9.27)	9.2	$\gamma_{TT} = 127\,296$	$\xi_{TT} = 13$	0.096
$\hat{L} = [\lambda; A, B, C, D]$, Gl. (9.16)	9.1	$\gamma_{CP} = 7250$	$\xi_{CP} = 231$	0.179
$\hat{L} = [\lambda; A, B_1, B_2, C, D_1, D_1]$, Gl. (10.54)	10.2	$\gamma_{CP} = 4850$	$\xi_{CP} = 346$	0.445
$\hat{L} = [\lambda; A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2]$, Gl. (10.95)	10.3	$\gamma_{CP} = 2450$	$\xi_{CP} = 685.55$	—

Da die mit diesem Algorithmus berechneten Zustandswahrscheinlichkeiten aus den beschriebenen Gründen nicht direkt mit denen der anderen Algorithmen vergleichbar sind, wurde hier der RMSE nicht berechnet.

Wie die Ergebnisse in Tabelle 10.4 abschließend zeigen, bietet die Dekomposition des Verhaltenstensors ein erhebliches Potenzial, um den Speicherbedarf qualitativer Modelle zu reduzieren. Selbst bei sehr hohen Reduktionsfaktoren ξ blieb die Fähigkeit zur Fehlererkennung erhalten.

11

Zusammenfassung und Ausblick

Zusammenfassung

In Kapitel 2 wurde der Stand der Technik der Fehlererkennung in der Gebäudeautomation erläutert und die Vor- und Nachteile üblicher Verfahren zur Fehlererkennung wurden herausgestellt. Zudem wurde ein Überblick hinsichtlich moderner modellbasierter Ansätze zur Fehlererkennung, deren Vor- und Nachteilen sowie eine Abgrenzung zu den qualitativen Modellen erarbeitet. In Kapitel 3 wurden die aus der Literatur bekannten Grundlagen stochastischer Automaten und die Zustandsbeobachtung eingeführt, welche die Grundlage der konsistenzbasierten Fehlererkennung bildet. Die qualitativen Modelle, deren Fundament stochastische Automaten bilden und die auf der Beschreibung des qualitativen Verhaltens quantisierter Systeme beruhen, wurden in Kapitel 4 eingeführt. Für die Interpretation der aus der qualitativen Zustandsbeobachtung gewonnenen Information bezüglich des qualitativen Systemverhaltens (nominal/fehlerhaft), wurde ein einfach zu interpretierendes, in der Sensibilität einstellbares und nutzerfreundliches Verhaltenssignal eingeführt. Ein einfacher Algorithmus zur qualitativen Fehlererkennung wurde eingeführt und erläutert. Dass sich qualitative Modelle zur Fehlererkennung in der Gebäudeautomation erfolgreich anwenden lassen, konnte insbesondere in Kapitel 5 gezeigt werden. Dabei wurde in ausführlichen Schritten beschrieben, wie ausgehend von verfügbaren Trainingsdaten, über die Partitionierung der Signalmräume und die Quantisierung, ein qualitatives Modell generiert werden kann und was beachtet werden muss. Die hieraus gewonnenen Erkenntnisse wurden in einem Zwischenfazit in Abschnitt 5.6 zusammengefasst. Die Kapitel 6 bis 10 widmeten sich der Lösung des Komplexitätsproblems stochastischer Automaten und qualitativer Modelle, mit dem Ziel, deren Speicherbedarf zu reduzieren, sodass diese effizient auf Gebäudeautomationssystemen oder Kleinstrechnern anwendbar sind. In Kapitel 6 wurden zunächst die aus der Literatur bekannten Grundlagen der linearen und multilinearen Algebra eingeführt, welche wiederum die Basis der tensorbasierten qualitativen Modellierung und Fehlererkennung bilden. Kapitel 7 zeigte auf, wie sich stochastische Automaten und qualitative Modelle durch Speicherung der Übergangswahrscheinlichkeiten der Verhaltensrelation in einem Verhaltenstensor definieren lassen. Entsprechende Beobachtergleichungen – auch zur Berücksichtigung von Unsicherheiten in den Automaten ein- und ausgaben – wurden aufgestellt, bewiesen und in Algorithmen zur qualitativen Fehlererkennung zusammengefasst. In Kapitel 8 wurden die Grundlagen der Tensordekomposition in Form der CP- und TT-Dekomposition eingeführt, die Vorteile in Bezug auf die Reduktion des Speicherbedarfs herausgestellt und entsprechende CP- und TT-Repräsentationen des Verhaltenstensors eingeführt. Beobachtergleichungen und -algorithmen zur Fehlererkennung auf Grundlage der speicher-reduzierten und approximierten CP- und TT-Repräsentationen des Verhaltenstensors wurden in Kapitel 9 erarbeitet und die entsprechenden Gleichungen bewiesen und erläutert. In Kapitel 10 wurde ein Ansatz zur weiteren Speicherreduzierung qualitativer Modelle entwickelt, welcher auf der komponentenweisen Quantisierung der Systemgrößen beruht. Unter Inkaufnahme weiterer Approximationen durch die in der Praxis im allgemeinen nicht haltbare Annahme der stochastischen Unabhängigkeit der Wahrscheinlichkeitsverteilungen der komponentenweisen quantisierten Systemgrößen untereinander, konnten weitere Reduktionen des Speicherbedarfs aufgezeigt werden und die entsprechenden Beobachtergleichungen bewiesen und in Algorithmen formalisiert werden.

Auch wenn insbesondere in den Kapiteln 8 bis 10 durch die Vorstellung verschiedener Algorithmen zur qualitativen Fehlerdetektion für die CP- und TT-Repräsentationen des Verhaltenstensors gezeigt wurde, dass die Methoden der Tensordekomposition ein großes Potenzial besitzen um qualitative Modelle für die Praxis anwendbar zu machen, bleibt in mehreren Bereichen weiterer Forschungsbedarf.

Ausblick

- **CP-Dekomposition:** Die CP-Dekomposition bietet sehr große Speicherreduktionen und ermöglicht es, wie in den Anwendungsbeispielen gezeigt wurde, die grundlegende Systemdynamik widerzuspiegeln, wodurch die Fehlerdetektion selbst bei groben Approximationen und kleinen Dekompositionsrängen realisierbar ist. Als schwierig stellt sich dabei die Berechnung einer hinreichend präzisen Approximation für gegebene Dekompositionsränge R dar, was wie erwähnt, an der zufälligen Initialisierung der Startwerte der Faktormatrizen liegt. An dieser Stelle wäre es hilfreich, durch eine Strukturanalyse des Verhaltenstensors, die Startwerte manuell vorzugeben, sodass die Dekomposition zu deterministischen Ergebnissen führt. Ein weiterer ungünstiger Aspekt ist die Tatsache, dass die Faktormatrizen Werte größer eins enthalten können. Hierbei wäre eine Nebenbedingung des Optimierungsproblems wünschenswert, die es ermöglicht, die Elemente der Faktormatrizen mit einer Beschränkung auf das Einheitsintervall $[0, 1]$ zu berechnen. Damit wäre sichergestellt, dass die mit den CP-Algorithmen berechneten approximierten Zustandswahrscheinlichkeiten stets echte Wahrscheinlichkeitsverteilungen darstellen. Eine solche Realisierung ist beispielsweise mit dem Tensorlab [126] möglich, konnte aber im Rahmen dieser Arbeit leider nicht mehr umgesetzt werden.
- **TT-Dekomposition:** Die TT-Dekomposition bietet die sehr hilfreiche Möglichkeit, die Approximationsgüte für die Zerlegung des Verhaltenstensors vorzugeben. Allerdings ist dabei im Gegensatz zur CP-Dekomposition, der Speicherbedarf nicht bereits schon vor der Dekomposition ersichtlich. Im Allgemeinen benötigt die TT-Dekomposition aufgrund der dreidimensionalen Struktur der TT-Kerne mehr Speicherbedarf als die CP-Dekomposition. Auch lässt sich kein großer Gewinn in Bezug auf den Speicherbedarf erzielen, indem man qualitative Modelle mit komponentenweise quantisierten Systemgrößen verwendet. Ein weiterer Punkt ist, dass die TT-Dekomposition die Elemente der TT-Kerne ohne Beschränkung auf die nicht-negativen Zahlen berechnet. Dies führt wie erläutert dazu, dass Zwischenschritte im Algorithmus zur Fehlerdetektion nötig werden, welche die möglicherweise negativen Wahrscheinlichkeiten des Zustandsvektors in Nullen umwandeln, was zu einer weiteren Approximation der Zustandswahrscheinlichkeiten führt.
- **Unvollständigkeit der qualitativen Modelle:** Die Unvollständigkeit der qualitativen Modelle, die sich letztlich bei der Verwendung von Messdaten zur Modellidentifikation kaum vermeiden lässt, stellt ein grundsätzliches Problem dar, welches auf alle modellbasierten Methoden zutrifft, welche ihr »Wissen« aus realen Messdaten generieren. Das Problem bei der Unvollständigkeit der qualitativen Modelle ist dabei, dass womöglich grundlegende Informationen bezüglich der Dynamik des zugrundeliegenden Prozesses nicht vom Modell abgebildet werden können. Um dieses Problem zu minimieren ist es hilfreich, über lange Trainingszeiträume fehlerfreier Daten zu verfügen. Dies stellt in Praxis wiederum ein Problem dar, denn das Aufarbeiten von Messdaten, egal ob automatisiert oder händisch, ist ein sehr zeitintensiver Prozess, der letztlich auch nur schwer sicherstellen kann, dass tatsächlich nur nominale Daten vorliegen. Dennoch bieten modellbasierte Methoden, auch wenn Sie als Black-Box Modelle aus Messdaten generiert werden, Vorteile gegenüber den häufig anzutreffenden regelbasierten Systemen zur Fehlerdetektion, da sie nicht nur feste Grenzwerte überwachen, sondern – wenn auch nicht

immer vollständig – die Systemdynamik widerspiegeln. Da heutzutage aufgrund der steigenden Technisierung im Gebäudesektor immer größere Mengen an Daten verfügbar sind, erscheint es erstrebenswert, dieses Potenzial entsprechend zu nutzen.

Eine weiterer Punkt bezüglich der Vollständigkeit des qualitativen Modells ist dessen strukturelle Vollständigkeit. Diese bezieht sich nicht darauf, dass das Modell tatsächlich alle dem Prozess möglichen Verhaltensformen abbilden kann, sondern dass die Normierungsbedingungen (3.36) und (3.37) für die bedingten Wahrscheinlichkeiten der Verhaltensrelation, beziehungsweise für den Verhaltenstensor erfüllt sind. Dies ist in der Regel bei der Verwendung von Messdaten zur Modellbildung nicht der Fall und trifft auch auf die hier gezeigten Anwendungsbeispiele zu. Eine interessante Möglichkeit dieses Problem zumindest approximativ zu lösen, bietet dabei wiederum die Tensordekomposition. Für das TensorLab [126] sowie die Tensor-Toolbox [6] sind Algorithmen verfügbar, die basierend auf der multilinearern Tensorstruktur, als fehlende Daten gekennzeichnete Tensorelemente mithilfe eines Optimierungsalgorithmus interpolieren [125]. Leider konnte diese vielversprechende Methode im Rahmen dieser Arbeit nicht mehr umgesetzt werden.

- **Weitere Optimierungspotenziale:** In dieser Arbeit wurden ausschließlich rechtwinklige Partitionierungen des qualitativen Eingangs-, Ausgangs- und Zustandsraums verwendet. Prinzipiell können diese Partitionierungen jedoch beliebig komplexe Formen besitzen. Andere Partitionierungen können dabei helfen, bestimmte Systemeigenschaften präziser darzustellen. Eine Möglichkeit bieten hierbei die kanonischen Partitionierungen, die beispielsweise in [69] beschrieben sind. Eine weitere Möglichkeit, mehr Informationen bezüglich des zugrundeliegenden Prozesses abbilden zu können, ist die Berücksichtigung vergangener Zeitschritte bei der Modellbildung. Diese Methode fand bisher kaum Anwendung, da beispielsweise schon die Berücksichtigung eines einzigen vergangenen Zeitschritts für einen einzelnen Zustand zu einer weiteren Dimension des Verhaltenstensor und damit einem deutlich größeren Speicherbedarf führt. Dieser Möglichkeit kommt jedoch durch die Tensordekomposition und die damit zur Verfügung stehenden Speicherreduktionspotenziale eine neue Bedeutung zu und sollte entsprechend untersucht werden.

Literatur

- [1] Abadi, M. et al. (2016): *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. arXiv: 1603.04467v2. Software online verfügbar: <https://www.tensorflow.org>.
- [2] AG Energiebilanzen (AGEB) (2018): *Auswertungstabellen zur Energiebilanz Deutschland*. Stand: 07/2018. Berlin, Bergheim.
- [3] Bacher, J., Pöge, A. und Wenzig, K. (2011): *Clusteranalyse: anwendungsorientierte Einführung in Klassifikationsverfahren*. 3., erg., vollst. überarb. und neu gestaltete Aufl. München: Oldenbourg Verlag. ISBN: 978-3-486-58457-8.
- [4] Bader, B. W. und Kolda, T. G. (2006): »Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping«. In: *ACM Transactions on Mathematical Software* Bd. 32, Nr. 4, S. 635–653. DOI: 10.1145/1186785.1186794.
- [5] Bader, B. W. und Kolda, T. G. (2008): »Efficient MATLAB Computations with Sparse and Factored Tensors«. In: *SIAM Journal on Scientific Computing* Bd. 30, Nr. 1, S. 205–231. DOI: 10.1137/060676489.
- [6] Bader, B. W., Kolda, T. G. et al. (2015): *MATLAB Tensor Toolbox Version 2.6*. Verfügbar online. URL: <http://www.sandia.gov/~tgkolda/TensorToolbox/>.
- [7] Beichelt, F. (1997): *Stochastische Prozesse für Ingenieure*. Stuttgart: Teubner. ISBN: 978-3-519-02989-2.
- [8] Benndorf, G. A., Wystrcil, D. und Réhault, N. (2018): »Energy Performance Optimization in Buildings: A Review on Semantic Interoperability, Fault Detection, and Predictive Control«. In: *Applied Physics Reviews* Bd. 5, Nr. 4, S. 041501-1–041501-19. DOI: 10.1063/1.5053110.
- [9] Benndorf, G. A. und Réhault, N. (2016): »Density-Based Clustering Algorithm for Fault Detection and Identification in HVAC Systems«. In: *Conference Proceedings of Central European Symposium on Building Physics CESBP 2016 / BauSIM 2016*, Dresden, Deutschland, S. 243–250. ISBN: 978-3-8167-9798-2.
- [10] Bigoni, D., Engsig-Karup, A. P. und Marzouk, Y. M. (2016): »Spectral Tensor-Train Decomposition«. In: *SIAM Journal on Scientific Computing* Bd. 38, Nr. 4, A2405–A2439. DOI: 10.1137/15M1036919.
- [11] Blanke, M., Kinnaert, M., Lunze, J. und Staroswiecki, M. (2006): *Diagnosis and Fault-Tolerant Control*. Second Edition. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-540-35652-3.
- [12] Blanke, M., Kinnaert, M., Lunze, J. und Staroswiecki, M. (2016): *Diagnosis and Fault-Tolerant Control*. Third Edition. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-662-47942-1.
- [13] Bukharaev, R. G. (1995): *Theorie der stochastischen Automaten*. Leitfäden der Informatik. Wiesbaden: Vieweg + Teubner. ISBN: 978-3-519-02124-7.
- [14] Bundesanzeiger (2014): *Forschungsförderung im 6. Energieforschungsprogramm „Forschung für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung“*. BAnz AT 30.12.2014 B1. Veröffentlicht am 30.12.2014.
- [15] Bundesministerium für Wirtschaft und Energie (BMWi) (2011): *Forschung für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung - Das 6. Energieforschungsprogramm der Bundesregierung*. Stand: 07/2011. Berlin.
- [16] Bundesministerium für Wirtschaft und Energie (BMWi) (2014): *Mehr Aus Energie Machen - Nationaler Aktionsplan Energieeffizienz*. Stand: 12/2014. Berlin.

- [17] Bundesministerium für Wirtschaft und Energie (BMWi) (2015a): *Energieeffizienzstrategie Gebäude - Wege zu einem nahezu klimaneutralen Gebäudebestand*. Stand: 11/2015. Berlin.
- [18] Bundesministerium für Wirtschaft und Energie (BMWi) (2015b): *Forschung für Energieoptimiertes Bauen. Förderkonzept der Forschungsinitiative EnOB*. Stand: 01/2015. Bonn.
- [19] Bundesministerium für Wirtschaft und Energie (BMWi) (2018): *Sechster Monitoring-Bericht zur Energiewende - Die Energie der Zukunft. Berichtsjahr 2016*. Stand: 06/2018. Berlin.
- [20] Carroll, J. D. und Chang, J.-J. (1970): »Analysis of Individual Differences in Multidimensional Scaling via an N-Way Generalization of “Eckart-Young” Decomposition«. In: *Psychometrika* Bd. 35, Nr. 3, S. 283–319. DOI: 10.1007/BF02310791.
- [21] *CASCADE - ICT for Energy Efficient Airports* (2011–2014). Gefördert durch das 7. Forschungsrahmenprogramm der Europäischen Kommission mit dem Fördervertrag Nr. 284920. URL: <http://www.cascade-eu.org>.
- [22] Cichocki, A., Lee, N., Oseledets, I. V., Phan, A.-H., Zhao, Q. und Mandic, D. (2016): »Tensor Networks for Dimensionality Reduction and Large-Scale Optimization. Part 1 Low-Rank Tensor Decompositions«. In: *Foundations and Trends® in Machine Learning* Bd. 9, Nr. 4-5, S. 249–429. DOI: 10.1561/22000000059.
- [23] Cichocki, A., Mandic, D., Lathauwer, L. D., Zhou, G., Zhao, Q., Caiafa, C. und Phan, H. A. (2015): »Tensor Decompositions for Signal Processing Applications: From Two-Way to Multiway Component Analysis«. In: *IEEE Signal Processing Magazine* Bd. 32, Nr. 2, S. 145–163. DOI: 10.1109/MSP.2013.2297439.
- [24] Cichocki, A. und Phan, A.-H. (2009): »Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations«. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* Bd. E92-A, Nr. 3, S. 708–721. DOI: 10.1587/transfun.E92.A.708.
- [25] Cichocki, A., Zdunek, R., Phan, H. A. und Amari, S.-I. (2009): *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. John Wiley & Sons, Ltd. ISBN: 978-0-470-74666-0.
- [26] Comon, P., Luciani, X. und de Almeida, A. L. F. (2009): »Tensor Decompositions, Alternating Least Squares and Other Tales«. In: *Journal of Chemometrics* Bd. 23, Nr. 7-8, S. 393–405. DOI: 10.1002/cem.1236.
- [27] Comon, P. und Lim, L.-H. (2011): *Sparse Representation and Low-Rank Tensor Approximations*. Bericht ISRN I3S/RR-2011-02-FR. Sophia Antipolis, France: Informatique, Signaux et Systèmes De Sophia Antipolis. URL: <https://hal.archives-ouvertes.fr/hal-00703494>.
- [28] De Lathauwer, L., De Moor, B. und Vandewalle, J. (2000): »A Multilinear Singular Value Decomposition«. In: *SIAM Journal on Matrix Analysis and Applications* Bd. 21, Nr. 4, S. 1253–1278. DOI: 10.1137/S0895479896305696.
- [29] Deutsche Energie-Agentur (dena) (2018): *Dena-Gebäudereport Kompakt - Statistiken und Analysen zur Energieeffizienz im Gebäudebestand*. Berlin.
- [30] DIN EN 15251:2012-12 (2012): *Eingangsparameter für das Raumklima zur Auslegung und Bewertung der Energieeffizienz von Gebäuden – Raumluftqualität, Temperatur, Licht und Akustik*.
- [31] Domanov, I. und De Lathauwer, L. (2015): *Canonical Polyadic Decomposition of Third-Order Tensors: Relaxed Uniqueness Conditions and Algebraic Algorithm*. arXiv: 1501.07251v2 [math.SP].

- [32] Eisenbarth, M. (2014): »Erkennung von Anomalien durch Residuenanalyse multivariater Zeitreihen«. Masterarbeit. Universität Freiburg.
- [33] Fernandez, N., Katipamula, S., Wang, W., Xie, Y., Zhao, M. und Corbin, C. (2017): *Impacts of Commercial Building Controls on Energy Savings and Peak Load Reduction*. PNNL-25985. Richland, Washington, USA: Pacific Northwest National Laboratory.
- [34] Gertler, J. (1998): *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker Inc. ISBN: 978-0-8247-9427-9.
- [35] Grasedyck, L., Kressner, D. und Tobler, C. (2013): »A Literature Survey of Low-Rank Tensor Approximation Techniques«. In: *GAMM-Mitteilungen* Bd. 36, Nr. 1, S. 53–78. DOI: 10.1002/gamm.201310004.
- [36] Hackbusch, W. (2012): *Tensor Spaces and Numerical Tensor Calculus*. Bd. 42. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-642-28027-6.
- [37] Harshman, R. A. (1970): »Foundations of the PARAFAC Procedure: Models and Conditions for an “Explanatory” Multi-modal Factor Analysis«. In: *UCLA Working Papers in Phonetics* Bd. 16, S. 1–84. URL: <http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf>.
- [38] Håstad, J. (1990): »Tensor Rank Is NP-Complete«. In: *Journal of Algorithms* Bd. 11, Nr. 4, S. 644–654. DOI: 10.1016/0196-6774(90)90014-6.
- [39] Hastie, T., Tibshirani, R. und Friedman, J. (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2. Aufl. New York: Springer-Verlag. ISBN: 978-0-387-84857-0.
- [40] Henze, N. (2010): *Stochastik für Einsteiger: Eine Einführung in die faszinierende Welt des Zufalls*. 8., erw. Aufl. Wiesbaden: Vieweg + Teubner. ISBN: 978-3-8348-0815-8.
- [41] Hillar, C. und Lim, L.-H. (2009): *Most Tensor Problems Are NP-Hard*. arXiv: 0911.1393v5 [cs.CC].
- [42] Hitchcock, F. L. (1927): »The Expression of a Tensor or a Polyadic as a Sum of Products«. In: *Journal of Mathematics and Physics* Bd. 6, Nr. 1-4, S. 164–189. DOI: 10.1002/sapm192761164.
- [43] Holtz, S., Rohwedder, T. und Schneider, R. (2012): »On Manifolds of Tensors of Fixed TT-Rank«. In: *Numerische Mathematik* Bd. 120, Nr. 4, S. 701–731. DOI: 10.1007/s00211-011-0419-7.
- [44] Hyafil, L. und Rivest, R. L. (1976): »Constructing Optimal Binary Decision Trees Is NP-Complete«. In: *Information Processing Letters* Bd. 5, Nr. 1, S. 15–17. DOI: 10.1016/0020-0190(76)90095-8.
- [45] Isermann, R. (2006): *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-540-24112-6.
- [46] Jacob, D., Dietz, S., Komhard, S., Neumann, C. und Herkel, S. (2010): »Black-Box Models for Fault Detection and Performance Monitoring of Buildings«. In: *Journal of Building Performance Simulation* Bd. 3, Nr. 1, S. 53–62. DOI: 10.1080/19401490903414454.
- [47] Katipamula, S. und Brambley, M. (2005): »Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems – A Review, Part I«. In: *HVAC&R Research* Bd. 11, Nr. 1, S. 3–25. DOI: 10.1080/10789669.2005.10391123.
- [48] Kiers, H. A. L. (2000): »Towards a Standardized Notation and Terminology in Multiway Analysis«. In: *Journal of Chemometrics* Bd. 14, Nr. 3, S. 105–122. DOI: 10.1002/1099-128X(200005/06)14:3<105::AID-CEM582>3.0.CO;2-I.

- [49] Kim, H. und Park, H. (2008): »Nonnegative Matrix Factorization Based on Alternating Non-negativity Constrained Least Squares and Active Set Method«. In: *SIAM Journal on Matrix Analysis and Applications* Bd. 30, Nr. 2, S. 713–730. DOI: 10.1137/07069239X.
- [50] Kim, H., Park, H. und Elden, L. (2007): »Non-Negative Tensor Factorization Based on Alternating Large-Scale Non-Negativity-Constrained Least Squares«. In: *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, Boston, USA, S. 1147–1151. DOI: 10.1109/BIBE.2007.4375705.
- [51] Kim, J. (2014): *Nonnegative Matrix and Tensor Factorization Algorithms Toolbox*. Verfügbar online. URL: <https://github.com/kimjingu/nonnegfac-matlab>.
- [52] Kim, J. und Park, H. (2008): »Toward Faster Nonnegative Matrix Factorization: A New Algorithm and Comparisons«. In: *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italien, S. 353–362. DOI: 10.1109/ICDM.2008.149.
- [53] Kim, J., He, Y. und Park, H. (2014): »Algorithms for Nonnegative Matrix and Tensor Factorizations: A Unified View Based on Block Coordinate Descent Framework«. In: *Journal of Global Optimization* Bd. 58, Nr. 2, S. 285–319. DOI: 10.1007/s10898-013-0035-4.
- [54] Kim, W. und Katipamula, S. (2018): »A Review of Fault Detection and Diagnostics Methods for Building Systems«. In: *Science and Technology for the Built Environment* Bd. 24, Nr. 1, S. 3–21. DOI: 10.1080/23744731.2017.1318008.
- [55] Kolda, T. G. (2006): *Multilinear Operators for Higher-Order Decompositions*. Tech. Report SAND2006-2081. Sandia National Laboratories, Albuquerque, NM, Livermore, CA, S. 1–28. URL: <http://www.sandia.gov/~tgkolda/pubs/pubfiles/SAND2006-2081.pdf>.
- [56] Kolda, T. G. und Bader, B. W. (2009): »Tensor Decompositions and Applications«. In: *SIAM Review* Bd. 51, Nr. 3, S. 455–500. DOI: 10.1137/07070111X.
- [57] Kruskal, J. B. (1977): »Three-Way Arrays: Rank and Uniqueness of Trilinear Decompositions, with Application to Arithmetic Complexity and Statistics«. In: *Linear Algebra and its Applications* Bd. 18, Nr. 2, S. 95–138. DOI: 10.1016/0024-3795(77)90069-6.
- [58] Kruskal, J. B. (1989): »Rank, Decomposition, and Uniqueness for 3-Way and N-Way Arrays«. In: *Multiway Data Analysis*. Hrsg. von R. Coppi und S. Bolasco. Amsterdam, The Netherlands: North-Holland Publishing Co., S. 7–18. ISBN: 978-0-444-87410-8. URL: <http://psychology.uwo.ca/faculty/harshman/jbkrank.pdf>.
- [59] Kruskal, J. B., Harshman, R. A. und Lundy, M. E. (1989): »How 3-MFA Data Can Cause Degenerate Parafac Solutions, among Other Relationships«. In: *Multiway Data Analysis*. Hrsg. von R. Coppi und S. Bolasco. Amsterdam, The Netherlands: North-Holland Publishing Co., S. 115–122. ISBN: 978-0-444-87410-8. URL: www.psychology.uwo.ca/faculty/harshman/jbkdegen.pdf.
- [60] LeBlanc, D. C. (2004): *Statistics: Concepts and Applications for Science*. Boston, Toronto, London, Singapore: Jones & Bartlett Learning. ISBN: 978-0-7637-4699-5.
- [61] Lee, D. D. und Seung, H. S. (2001): »Algorithms for Non-Negative Matrix Factorization«. In: *Advances in Neural Information Processing Systems* Bd. 13, S. 556–562.
- [62] Lee, N. und Cichocki, A. (2015): »Estimating a Few Extreme Singular Values and Vectors for Large-Scale Matrices in Tensor Train Format«. In: *SIAM Journal on Matrix Analysis and Applications* Bd. 36, Nr. 3, S. 994–1014. DOI: 10.1137/140983410.

- [63] Lee, N. und Cichocki, A. (2016): »Regularized Computation of Approximate Pseudoinverse of Large Matrices Using Low-Rank Tensor Train Decompositions«. In: *SIAM Journal on Matrix Analysis and Applications* Bd. 37, Nr. 2, S. 598–623. DOI: 10.1137/15M1028479.
- [64] Lee, N. und Cichocki, A. (2018): »Fundamental Tensor Operations for Large-Scale Data Analysis Using Tensor Network Formats«. In: *Multidimensional Systems and Signal Processing* Bd. 29, Nr. 3, S. 921–960. DOI: 10.1007/s11045-017-0481-0.
- [65] Lee, N., Phan, A.-H., Cong, F. und Cichocki, A. (2016): »Nonnegative Tensor Train Decompositions for Multi-Domain Feature Extraction and Clustering«. In: *Neural Information Processing. Lecture Notes in Computer Science* 9949. Springer International Publishing, S. 87–95. ISBN: 978-3-319-46674-3. DOI: 10.1007/978-3-319-46675-0_10.
- [66] León, F. P. und Kiencke, U. (2013): *Ereignisdiskrete Systeme: Modellierung und Steuerung verteilter Systeme*. vollst. überarb. und erg. Auflage. München: Oldenburg Verlag. ISBN: 978-3-486-76971-5.
- [67] Lichtenberg, G. und Lunze, J. (1996): »Identification of Discrete Event Models for Continuous-Variable Systems«. In: *Control '96, UKACC International Conference*, Exeter, Vereinigtes Königreich, S. 711–715. DOI: 10.1049/cp:19960638.
- [68] Lichtenberg, G., Lunze, J. und Münchmeyer, W.-E. (1994): »On Qualitative Identification of Linear Dynamical Systems«. In: *Second International Conference on Intelligent Systems Engineering*, Hamburg-Harburg, Deutschland, S. 95–100. DOI: 10.1049/cp:19940608.
- [69] Lichtenberg, G. (1998): *Theorie und Anwendung der qualitativen Modellierung zeitdiskreter dynamischer Systeme durch nichtdeterministische Automaten*. Vortschrittberichte VDI, Reihe 8: Mess-, Steuerungs- und Regelungstechnik, Nr. 686. Düsseldorf: VDI Verlag GmbH. ISBN: 3-18-368608-2.
- [70] Lichtenberg, G. und Luetzenberg, J. (2004): »Using Discrete Invariants for Fault Detection of Hybrid Systems«. In: 15th International Workshop on Principles of Diagnosis. Hrsg. von L. Trave-Massuyes. Carcassonne, S. 149–154.
- [71] Lichtenberg, G. und Lunze, J. (1997): »Observation of Qualitative States by Means of a Qualitative Model«. In: *International Journal of Control* Bd. 66, Nr. 6, S. 885–903. DOI: 10.1080/002071797224441.
- [72] Lichtenberg, G., Lunze, J. und Raisch, J. (1999a): »Zwei Wege zur Modellierung des Qualitativen Verhaltens dynamischer Systeme«. In: *at - Automatisierungstechnik* Bd. 47, Nr. 5, S. 187–198. DOI: 10.1524/auto.1999.47.5.187.
- [73] Lichtenberg, G., Lunze, J., Scheuring, R. und Schröder, J. (1999b): »Prozeßdiagnose mittels Qualitativer Modelle am Beispiel eines Wasserstoffverdichters«. In: *at - Automatisierungstechnik* Bd. 47, Nr. 3, S. 101–109. DOI: 10.1524/auto.1999.47.3.101.
- [74] Lichtenberg, G. und Steele, A. (1996): »An Approach to Fault Diagnosis Using Parallel Qualitative Observers«. In: *International Workshop on Discrete Event Systems*, Edinburgh, Vereinigtes Königreich, S. 290–295.
- [75] Lim, L.-H. und Comon, P. (2009): »Nonnegative Approximations of Nonnegative Tensors«. In: *Journal of Chemometrics* Bd. 23, Nr. 7-8, S. 432–441. DOI: 10.1002/cem.1244.
- [76] Lunze, J. (1992a): »Bemerkungen zur Qualitativen Beschreibung dynamischer Systeme«. In: *at - Automatisierungstechnik* Bd. 40, Nr. 8. DOI: 10.1515/auto-1992-0803.

- [77] Lunze, J. (1992b): »Qualitative Modelling of Continuous-Variable Systems by Means of Non-deterministic Automata«. In: *Intelligent Systems Engineering* Bd. 1, Nr. 1, S. 22–30. DOI: 10.1049/ise.1992.0003.
- [78] Lunze, J. (1993): »Ein Ansatz Zur Qualitativen Modellierung Und Regelung Dynamischer Systeme / An Approach to Qualitative Modelling and Control of Dynamical Systems«. In: *at - Automatisierungstechnik* Bd. 41, Nr. 12. DOI: 10.1515/auto-1993-1204.
- [79] Lunze, J. (1994): »Qualitative Modelling of Linear Dynamical Systems with Quantized State Measurements«. In: *Automatica* Bd. 30, Nr. 3, S. 417–431. DOI: 10.1016/0005-1098(94)90119-8.
- [80] Lunze, J. (1998a): »On the Markov Property of Quantised State Measurement Sequences«. In: *Automatica* Bd. 34, Nr. 11, S. 1439–1444. DOI: 10.1016/S0005-1098(98)00099-5.
- [81] Lunze, J. (1998b): »Qualitative Modellierung Dynamischer Systeme durch Stochastische Automaten«. In: *at - Automatisierungstechnik* Bd. 46, Nr. 6, S. 271–283. DOI: 10.1524/auto.1998.46.6.271.
- [82] Lunze, J., Nixdorf, B. und Schröder, J. (1999): »Deterministic Discrete-Event Representations of Linear Continuous-Variable Systems«. In: *Automatica* Bd. 35, Nr. 3, S. 395–406. DOI: 10.1016/S0005-1098(98)00176-9.
- [83] Lunze, J. und Schröder, J. (2001): »State Observation and Diagnosis of Discrete-Event Systems Described by Stochastic Automata«. In: *Discrete Event Dynamic Systems-Theory and Applications* Bd. 11, Nr. 4, S. 319–369. DOI: 10.1023/A:1011273108731.
- [84] Lunze, J. (2000): »Diagnosis of Quantised Systems«. In: *IFAC Proceedings Volumes* Bd. 33, Nr. 11: *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2000)*, Budapest, Ungarn, S. 29–40. DOI: 10.1016/S1474-6670(17)37337-8.
- [85] Lunze, J. (2010): *Regelungstechnik. 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. 8. neu bearb. Aufl. Springer-Lehrbuch. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-642-13808-9.
- [86] Lunze, J. (2012): *Ereignisdiskrete Systeme: Modellierung und Analyse dynamischer Systeme mit Automaten, Markovketten und Petrinetzen*. 2. überarb. Aufl. München: Oldenbourg. ISBN: 978-3-486-71885-0.
- [87] Lunze, J. (2016): *Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. 4. überarb. Aufl. Berlin, Boston: De Gruyter GmbH & Co KG. ISBN: 978-3-11-046557-0.
- [88] Luo, D., Huang, H. und Ding, C. (2009): *Are Tensor Decomposition Solutions Unique? On the Global Convergence of HOSVD and ParaFac Algorithms*. arXiv: 0902.4521v1 [cs.CV].
- [89] Mehrotra, K., Mohan, C. K. und Ranka, S. (1997): *Elements of Artificial Neural Networks*. Cambridge, USA: MIT Press. ISBN: 978-0-262-13328-9.
- [90] Montavon, G., Orr, G. und Müller, K.-R. (2012): *Neural Networks: Tricks of the Trade*. Second Edition. Lecture Notes in Computer Science 7700. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-642-35288-1.
- [91] Müller-Eping, T., Lichtenberg, G. und Vogelmann, V. (2017): »Fault Detection Algorithms Based on Decomposed Tensor Representations for Qualitative Models«. In: *IFAC-PapersOnLine* Bd. 50, Nr. 1: *20th IFAC World Congress*, Toulouse, Frankreich, S. 5622–5629. DOI: 10.1016/j.ifacol.2017.08.1109.

- [92] Müller, T., Kruppa, K., Lichtenberg, G. und Réhault, N. (2015): »Fault Detection with Qualitative Models Reduced by Tensor Decomposition Methods«. In: *IFAC-PapersOnLine* Bd. 48, Nr. 21: *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 2015)*, Paris, Frankreich, S. 416–421. DOI: 10.1016/j.ifacol.2015.09.562.
- [93] Müller, T. und Lichtenberg, G. (2016): »Fault Detection with CP-Decomposed Qualitative Models«. In: *IFAC-PapersOnLine* Bd. 49, Nr. 5: *4th IFAC Conference on Intelligent Control and Automation Sciences (ICONS 2016)*, Reims, Frankreich, S. 309–314. DOI: 10.1016/j.ifacol.2016.07.131.
- [94] Müller, T., Réhault, N. und Rist, T. (2013): »A Qualitative Modeling Approach for Fault Detection and Diagnosis on HVAC Systems«. In: *Proceedings of the 13th International Conference for Enhanced Building Operations*, Montreal, Kanada.
- [95] Neidig, J. (2007): *An Automata Theoretic Approach to Modular Diagnosis of Discrete-Event Systems*. Norderstedt: Books on Demand GmbH. ISBN: 978-3-8334-9801-5.
- [96] Neumann, C., Jacob, D., Burhenne, S., Florita, A. R., Burger, E. und Schmidt, F. (2011): *Mod-Ben - Modellbasierte Methoden für die Fehlererkennung und Optimierung im Gebäudebetrieb*. Endbericht. Freiburg: Fraunhofer-Institut für Solare Energiesysteme ISE. DOI: 10.2314/GBV:668908173.
- [97] Nickel, M. (2013): *scikit-tensor Library (Version 0.1)*. Verfügbar online. URL: <https://github.com/mnick/scikit-tensor>.
- [98] *OBSERVE - Optimierung und Betriebsführung komplexer Gebäudeenergieversorgungsanlagen* (2015–2018). Gefördert durch das Bundesministerium für Wirtschaft und Energie (BMWi) mit dem Förderkennzeichen: 03ET1225 A-D. URL: <http://www.ob-serve.de>.
- [99] Oseledets, I. (2011): »Tensor-Train Decomposition«. In: *SIAM Journal on Scientific Computing* Bd. 33, Nr. 5, S. 2295–2317. DOI: 10.1137/090752286.
- [100] Oseledets, I. (2016a): *TT-Toolbox (Version 2.2.2)*. Verfügbar online. URL: <https://github.com/oseledets/TT-Toolbox>.
- [101] Oseledets, I. (2016b): *TTpy - Python implementation of the Tensor Train (TT) -Toolbox*. Verfügbar online. URL: <https://github.com/oseledets/ttpy>.
- [102] Oseledets, I. und Tyrtysnikov, E. (2010): »TT-Cross Approximation for Multidimensional Arrays«. In: *Linear Algebra and its Applications* Bd. 432, Nr. 1, S. 70–88. DOI: 10.1016/j.laa.2009.07.024.
- [103] Panisson, A. (2013): *nep.py*. Verfügbar online. URL: <https://gist.github.com/panisson/7719245>.
- [104] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. und Duchesnay, E. (2011): »Scikit-learn: Machine Learning in Python«. In: *Journal of Machine Learning Research* Bd. 12, S. 2825–2830. Software online verfügbar: scikit-learn.org.
- [105] Qi, Y., Comon, P. und Lim, L.-H. (2016): »Uniqueness of Nonnegative Tensor Approximations«. In: *IEEE Transactions on Information Theory* Bd. 62, Nr. 4, S. 2170–2183. DOI: 10.1109/TIT.2016.2532906.

- [106] Réhault, N., Böhme, G. und Müller, T. (2015): »Energetischer Gebäudebetrieb in der Praxis und neue Methoden für eine energieeffiziente Betriebsführung«. In: *Energieeffizienz in Gebäuden - Jahrbuch 2015*. Hrsg. von J. Pöschk. Berlin: VME - Verlag und Medienservice Energie, S. 229–238. ISBN: 978-3-936062-11-3.
- [107] Réhault, N., Ohr, F., Zehnle, S., Müller, T., Rist, T., Jacob, D., Lichtenberg, G., Pangalos, G., Kruppa, K., Schmidt, F., Arkadius, Z., Harmsen, A. und Sewe, E. (2014): *Modellbasierte Qualitätssicherung des energetischen Gebäudebetriebs (ModQS)*. Freiburg: Fraunhofer-Institut für Solare Energiesysteme ISE. URL: <http://www.modqs.de>.
- [108] Réhault, N., Zehnle, S., Müller-Eping, T., Rist, T., Wystrcil, D., Merk, J., Lichtenberg, G., Kruppa, K., Lautenschlager, B., Harmsen, A., Sewe, E., Harmsen, P., Kirchner, R., Boeck, M., Buchholz, J., Przybilla, H., Wolfram, A. und Resse, K. (2019): *OBSERVE - Optimierung und Betriebsführung komplexer Gebäudeenergieversorgungsanlagen: Abschlussbericht*. Freiburg: Fraunhofer-Institut für Solare Energiesysteme ISE. DOI: 10.2314/KXP:1669312283.
- [109] Rokach, L. und Maimon, O. (2015): *Data Mining with Decision Trees: Theory and Applications*. Second Edition. Series in Machine Perception and Artificial Intelligence 81. Singapore: World Scientific. ISBN: 978-981-4590-07-5.
- [110] Saas, M. (2014): »Modellierung und Validierung raumluftechnischer Anlagen«. Masterarbeit. Hochschule Mannheim.
- [111] Saika, T. (2015): *ncp.py*. Verfügbar online. URL: <https://gist.github.com/tonmoy-saikia/e3ccc0d1b73dc82be54d>.
- [112] Schein, J., Bushby, S. T., Castro, N. S. und House, J. M. (2006): »A Rule-Based Fault Detection Method for Air Handling Units«. In: *Energy and Buildings* Bd. 38, Nr. 12, S. 1485–1492. DOI: 10.1016/j.enbuild.2006.04.014.
- [113] Schröder, J. (2003): *Modelling, State Observation and Diagnosis of Quantised Systems*. Lecture Notes in Control and Information Sciences 282. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-540-46086-2.
- [114] Shashua, A. und Hazan, T. (2005): »Non-Negative Tensor Factorization with Applications to Statistics and Computer Vision«. In: *Proceedings of the 22Nd International Conference on Machine Learning*, New York, USA, S. 792–799. DOI: 10.1145/1102351.1102451.
- [115] Specht, K., Bulander, R. und Gohout, W. (2014): *Statistik für Wirtschaft und Technik*. 2. aktualisierte und erweiterte Auflage. Berlin, Boston: De Gruyter Oldenbourg. ISBN: 978-3-11-035496-6.
- [116] *Statistics and Machine Learning Toolbox* (2018). URL: <https://de.mathworks.com/products/statistics.html>.
- [117] Sterling, R., Struß, P., Febres, J., Sabir, U. und Keane, M. (2014): »From Modelica Models to Fault Diagnosis in Air Handling Units«. In: *Proceedings of the 10th International Modelica Conference*, Lund, Schweden, S. 447–454. DOI: 10.3384/ecp14096447.
- [118] *TOPAS - Tools for Continuous Building Performance Auditing* (2015–2018). Gefördert durch das EU Förderprogramm Horizon 2020 mit dem Fördervertrag Nr. 676760. URL: <https://www.topas-eeb.eu>.
- [119] Tucker, L. R. (1966): »Some Mathematical Notes on Three-Mode Factor Analysis«. In: *Psychometrika* Bd. 31, Nr. 3, S. 279–311. DOI: 10.1007/BF02289464.

- [120] Unbehauen, H. (2008): *Regelungstechnik I - Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme, Fuzzy-Regelsysteme*. 15., überarb. und erw. Aufl. Wiesbaden: Vieweg + Teubner. ISBN: 978-3-8348-0497-6.
- [121] Van der Linden, F. (2014): »General Fault Triggering Architecture to Trigger Model Faults in Modelica Using a Standardized Blockset«. In: *Proceedings of the 10th International Modelica Conference*, Lund, Schweden, S. 427–436. DOI: 10.3384/ecp14096427.
- [122] Van Overschee, P. und De Moor, B. (1994): »N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems«. In: *Automatica* Bd. 30, Nr. 1, S. 75–93. DOI: 10.1016/0005-1098(94)90230-5.
- [123] Vannieuwenhoven, N. (2017): »Condition numbers for the tensor rank decomposition«. In: *Linear Algebra and its Applications* Bd. 535, S. 35–86. DOI: 10.1016/j.laa.2017.08.014.
- [124] Venkatasubramanian, V., Rengaswamy, R., Yin, K. und Kavuri, S. N. (2003): »A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods«. In: *Computers & Chemical Engineering* Bd. 27, Nr. 3, S. 293–311. DOI: 10.1016/S0098-1354(02)00160-6.
- [125] Vervliet, N., Debals, O., Sorber, L. und Lathauwer, L. D. (2014): »Breaking the Curse of Dimensionality Using Decompositions of Incomplete Tensors: Tensor-Based Scientific Computing in Big Data Analysis«. In: *IEEE Signal Processing Magazine* Bd. 31, Nr. 5, S. 71–79. DOI: 10.1109/MSP.2014.2329429.
- [126] Vervliet, N., Debals, O., Sorber, L., Van Barel, M. und De Lathauwer, L. (2016a): *Tensorlab 3.0*. Verfügbar online. URL: <http://www.tensorlab.net>.
- [127] Vervliet, N., Debals, O., Sorber, L., Van Barel, M. und De Lathauwer, L. (2016b): *Tensorlab User Guide, Release 3.0*. URL: <ftp://ftp.esat.kuleuven.be/pub/stadius/nvervliet/tensorlab3.0.pdf>.
- [128] Waldmann, K.-H. und Stocker, U. M. (2013): *Stochastische Modelle. Eine anwendungsorientierte Einführung*. 2. überarb. Auflage, Berlin, Heidelberg: Springer. ISBN: 978-3-642-32911-1.
- [129] Weinhold, L. (2014): »Statistical Analysis and Application of Decision Trees for Fault Detection in Building Operation«. Bachelorarbeit. Universität Ulm.
- [130] Welling, M. und Weber, M. (2001): »Positive Tensor Factorization«. In: *Pattern Recognition Letters* Bd. 22, Nr. 12, S. 1255–1261. DOI: 10.1016/S0167-8655(01)00070-8.
- [131] Werner, M. (2008): *Signale und Systeme: Lehr- und Arbeitsbuch mit MATLAB®-Übungen und Lösungen*. 3. Auflage. Vieweg + Teubner. ISBN: 978-3-8348-0233-0.
- [132] Wong, K.-C. (2015): »A Short Survey on Data Clustering Algorithms«. In: *2015 Second International Conference on Soft Computing and Machine Intelligence (ISCMI)*, Hong Kong, S. 64–68. DOI: 10.1109/ISCMI.2015.10.



Symbole und Symbolkonventionen

Tabelle I: Mengenkonventionen

Bedeutung	Konvention
Menge der reellen Zahlen	\mathbb{R}
Menge der ganzen Zahlen	\mathbb{Z}
Menge der natürlichen Zahlen inkl. Null	\mathbb{N}
Menge der nicht-negativen reellen Zahlen	$\mathbb{R}_{\geq 0}$
Menge der natürlichen Zahlen ohne Null	$\mathbb{N}_{>0}$
Menge der reellen Zahlen von 0 bis 1 (Einheitsintervall)	$[0, 1] \subseteq \mathbb{R}$
Menge der reellen Zahlen 0 und 1	$\{0, 1\} \subseteq \mathbb{R}$
Alle weiteren Mengen	kalligrafisch, wie $\mathcal{V}, \mathcal{Y}, \mathcal{Z}$

Tabelle II: Symbolkonventionen

Bedeutung	Konvention	Beispiele
Skalare	klein oder groß, kursiv	m, T
Vektoren, Tupel	klein, fett	$\mathbf{p}, \boldsymbol{\alpha}$
Matrizen	groß, fett	\mathbf{A}, \mathbf{B}
Tensoren	groß, serifenlos	\mathbf{E}, \mathbf{F}
Skalarfolgen	groß, kursiv	$V(0 \dots T) = (v(0), v(1), \dots, v(T))$
Vektorfolgen	groß, kursiv, fett	$\mathbf{P}(0 \dots T) = (\mathbf{p}(0), \mathbf{p}(1), \dots, \mathbf{p}(T))$
Mengen von Skalaren	groß, kalligrafisch	$\mathcal{U} = \{1, 2, \dots, M\}$
Mengen von Tupeln	groß, kalligrafisch, fett	$\mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_m \subseteq \mathbb{N}^q$
Vektorfunktionen	klein, serifenlos, fett	$\mathbf{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$
Automatenfunktionen	groß, Euler-Skript	$\mathcal{L} : \mathcal{N}_z \times \mathcal{N}_w \times \mathcal{N}_z \times \mathcal{N}_v \longrightarrow [0, 1]$
Geschlossene Intervalle	in eckigen Klammern	$[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$
Halboffene Intervalle	in eckigen Klammern	$]a, b] = \{x \in \mathbb{R} \mid a < x \leq b\}$
Quantisierte Werte	in eckigen Klammern	$[\mathbf{u}], [\mathbf{Y}], [\mathbf{x}]$
Einheiten	lateinisch	K, m

Tabelle III: Akzente, Symbole und Produkte

Symbol	Konvention/Bedeutung	Beispiele
$\underline{\cdot}$	Permutierte(r) Matrix/Tensor	$\underline{\mathbf{L}}, \underline{\mathbf{L}}$
$\overline{\cdot}$	Multiindex	$y_{i\overline{jk}}, y(i, \overline{jk})$
$\hat{\cdot}$	Durch Tensordekomposition approximierte Größen	$\hat{\mathbf{L}}, \hat{\mathbf{L}}, \hat{\mathbf{p}}$
$\tilde{\cdot}$	Größen des qualitativen Modells	$\tilde{\mathcal{L}}, \tilde{\mathbf{L}}, \tilde{\mathbf{p}}$
$:$	Index-Operator	$a_{i:}, a_{:j}, \mathbf{L}(:, w, :, v)$
$^{\top}$	Transponierte(r) Vektor/Matrix	$\mathbf{a}^{\top}, \mathbf{B}^{\top}, \mathbf{x} = (x_1 \ x_2 \ x_3)^{\top}$
$\ \cdot\ _{\text{F}}$	Frobeniusnorm	$\ \mathbf{A}\ _{\text{F}} := \sqrt{\sum_{i=1}^I \sum_{j=1}^J a_{ij} ^2}$
$\ \cdot\ _1$	Summennorm	$\ \mathbf{a}\ _1 := \sum_{i=1}^I a_i $
$[[\cdot]]$	Kruskal-Tensor / CP-Repräsentation	$[[\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]]$
$\langle\langle\cdot\rangle\rangle$	TT-Repräsentation	$\langle\langle \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H} \rangle\rangle$
$ $	Bedingung	$\mathcal{L}(\bar{z}, w z, v), \mathcal{F}(\bar{z} z)$
\forall	Für alle (Allquantor)	$\forall a, b \in \mathbb{R} : a + b = b + a$
\exists	Es existiert (Existenzquantor)	$\exists a \in \mathbb{R} : a + 5 = 7$
\subseteq	Teilmenge	$\mathbb{R}_{\geq 0} \subseteq \mathbb{R}$
\supseteq	Obermenge	$\mathbb{R} \supseteq \mathbb{R}_{\geq 0}$
\cup	Vereinigungsmenge	$\mathcal{A} \cup \mathcal{B} = \{x x \in \mathcal{A} \vee x \in \mathcal{B}\}$
\cap	Schnittmenge	$\mathcal{A} \cap \mathcal{B} = \{x x \in \mathcal{A} \wedge x \in \mathcal{B}\}$
\emptyset	Leere Menge	$\mathcal{A} = \{1, 2\}, \mathcal{B} = \{3, 4\}, \mathcal{A} \cap \mathcal{B} = \emptyset$
\times	Kartesisches Produkt	$\mathcal{A} \times \mathcal{B} = \{(a, b) a \in \mathcal{A} \wedge b \in \mathcal{B}\}$
\cdot	Matrizenprodukt	$I \times K$ -Matrix, $\mathbf{Z} \in \mathbb{R}^{I \times J \times K}$ $\mathbf{A} = \mathbf{X} \cdot \mathbf{Y}, \mathbf{X} \cdot \mathbf{y} = \mathbf{a}$
\circ	Dyadisches Vektorprodukt	$\mathbf{C} = \mathbf{x} \circ \mathbf{y}, \mathbf{Y} = \bigcirc_{i=1}^I \mathbf{x}^{(i)}$
\otimes	Kronecker-Produkt	$\mathbf{M} = \mathbf{A} \otimes \mathbf{B}, \mathbf{y} = \bigotimes_{j=1}^J \mathbf{z}^{(j)}$
\odot	Kontrahiertes Modus-($P 1$) Produkt	$\mathbf{Z} = \mathbf{E} \odot \mathbf{F}, \mathbf{A} = \bigodot_{k=1}^K \mathbf{Z}^{(k)}$
\otimes	Hadamard-Produkt	$\mathbf{M} = \mathbf{A} \otimes \mathbf{B}, \mathbf{Y} = \bigotimes_{s=1}^S \mathbf{H}^{(s)}$
$\bar{\times}_{(n)}$	Modus- n Tensor-Vektor-Produkt	$\mathbf{U} = \mathbf{Z} \bar{\times}_{(2)} \mathbf{y}$
$\langle\cdot \cdot\rangle_{(\cdot \cdot)}$	Kontrahiertes Tensorprodukt	$\mathbf{A} = \langle\mathbf{X} \mathbf{Y}\rangle_{(1,2 5,3)}$

Tabelle IV: Abkürzungen

Abkürzung	Bedeutung
ALS	Alternating Least Square (Algorithmus)
CP	C ANDECOMP (C ANonical D ECOMPosition) P ARAFAC (P ARAllel Factors)
E/A	Eingabe/Ausgabe, Eingang/Ausgang
HALS	Hierarchical Alternating Least Square Algorithmus
MU	Multiplicative Update Algorithmus
NNCP	Nicht-negative CP-Dekomposition
Pr	Wahrscheinlichkeit
QuaMo	Qualitatives Modell
SA	Stochastischer Automat
SP	Stochastischer Prozess
TT	Tensor-Train

Tabelle V: Lateinische Symbole

Symbol	Bedeutung
A	Faktormatrix der Folgezustände bei der CP-Dekomposition
\mathcal{A}	Stochastischer Automat
\mathcal{A}	Stochastischer Automat mit vektoriellen Ein-, Ausgaben (und Zuständen)
\mathcal{A}^a	Autonomer stochastischer Automat
\mathcal{A}^a	Autonomer stochastischer Automat mit vektoriellen Zuständen
$\tilde{\mathcal{A}}$	Qualitatives Modell
$\tilde{\mathcal{A}}$	Qualitatives Modell mit komp. quantisierten Ein-, Ausgängen (und Zuständen)
B	Faktormatrix der Ausgaben/Ausgänge bei der CP-Dekomposition
B	Verhaltenssignal des SA
\tilde{B}	Verhaltenssignal des QuaMo
\hat{B}	CP-/TT-approximiertes Verhaltenssignal des QuaMo
b	Element des Verhaltenssignals des SA
\tilde{b}	Element des Verhaltenssignals des QuaMo
\hat{b}	Element des CP-/TT-approximierten Verhaltenssignals des QuaMo
\mathcal{B}	Verhalten des SA
\mathcal{B}_m	Verhalten des QuaMo
\mathcal{B}_q	Verhalten des quantisierten Systems
C	Faktormatrix der Zustände bei der CP-Dekomposition
D	Faktormatrix der Eingaben/Eingänge bei der CP-Dekomposition
E	TT-Kern der Ausgaben
E_x	Quantisierungsfunktion der Zustände

Symbol	Bedeutung
\mathcal{E}_u	Partitionierung des Eingangsraums
\mathcal{E}_y	Partitionierung Partition des Ausgangsraums
\mathcal{E}_x	Partitionierung des Zustandsraums
\mathcal{F}	Übergangsrelation des stochastischen Automaten
F	TT-Kern der Eingaben/Eingänge
\mathbf{f}	Zustandsüberföhrungsfunktion eines wertkontinuierlichen Systems
\mathcal{G}	Ausgaberation des stochastischen Automaten
G	TT-Kern der Folgezustände
\mathbf{g}	Ausgabefunktion eines wertkontinuierlichen Systems
H	TT-Kern der Zustände
\mathbf{K}	Hilfsmatrix für die Komposition von Submatrizen
k	diskrete Zeit
\mathcal{L}	Verhaltensrelation des SA
$\tilde{\mathcal{L}}$	Verhaltensrelation des QuaMo
\mathbf{L}	Submatrix des Verhaltenstensors des SA
$\underline{\mathbf{L}}$	Submatrix des permutierten Verhaltenstensors des SA
$\hat{\mathbf{L}}$	Submatrix des CP-approximierten Verhaltenstensors des SA
$\hat{\underline{\mathbf{L}}}$	Submatrix des TT-approximierten Verhaltenstensors des SA
\mathbf{L}	Verhaltenstensor des SA
$\underline{\mathbf{L}}$	Permutierter Verhaltenstensor des SA
$\hat{\mathbf{L}}$	CP-approximierter Verhaltenstensor des SA
$\hat{\underline{\mathbf{L}}}$	TT-approximierter Verhaltenstensor des SA
$\tilde{\mathbf{L}}$	Verhaltenstensor des QuaMo
$\underline{\tilde{\mathbf{L}}}$	Permutierter Verhaltenstensor des QuaMo
$\hat{\tilde{\mathbf{L}}}$	CP-approximierter Verhaltenstensor des QuaMo
$\hat{\underline{\tilde{\mathbf{L}}}}$	TT-approximierter Verhaltenstensor des QuaMo
l	Element des Verhaltenstensors des SA
\underline{l}	Element des permutierten Verhaltenstensors des Sa
\hat{l}	Element des CP-approximierten Verhaltenstensors des SA
$\hat{\underline{l}}$	Element des TT-approximierten Verhaltenstensors des SA
M	Anzahl der Eingabe-/Eingangswerte des SA/SP/QuaMo
m	Anzahl der physikalischen Eingangsgrößen
N	Anzahl der Zustandswerte des SA/SP/QuaMo
n	Anzahl der physikalischen Zustandsgrößen
\mathbf{p}	Wahrscheinlichkeitsvektor
\mathbf{p}_v	Wahrscheinlichkeitsvektor der Eingaben des SA
\mathbf{p}_w	Wahrscheinlichkeitsvektor der Ausgaben des SA
\mathbf{p}_z	Wahrscheinlichkeitsvektor der Zustände des SA

Symbol	Bedeutung
$\tilde{\mathbf{p}}_v$	Wahrscheinlichkeitsvektor der qualitativen Eingänge
$\tilde{\mathbf{p}}_w$	Wahrscheinlichkeitsvektor der qualitativen Ausgänge
$\tilde{\mathbf{p}}_z$	Wahrscheinlichkeitsvektor der qualitativen Zustände
\mathbf{P}_v	Folge von Wahrscheinlichkeitsvektoren der Eingaben des SA
\mathbf{P}_w	Folge von Wahrscheinlichkeitsvektoren der Ausgaben des SA
\mathbf{P}_z	Folge von Wahrscheinlichkeitsvektoren der Zustände des SA
$\tilde{\mathbf{P}}_v$	Folge von Wahrscheinlichkeitsvektoren qualitativer Eingänge
$\tilde{\mathbf{P}}_w$	Folge von Wahrscheinlichkeitsvektoren qualitativer Ausgänge
$\tilde{\mathbf{P}}_z$	Folge von Wahrscheinlichkeitsvektoren qualitativer Zustände
Pr	Wahrscheinlichkeit
Q	Anzahl der Ausgabe-/Ausgangswerte des SA/SP/QuaMo
q	Anzahl der physikalischen Ausgangsgrößen
R	Rang der CP-Dekomposition
t	kontinuierliche Zeit
T	Zeithorizont
T_a	Abtastzeit
T_b	Beobachterzeitschrittdauer
\mathbf{u}	Eingangsvektor, Eingangsgröße, Eingang
$[\mathbf{u}]$	Quantisierter Eingangsvektor
$[\mathbf{u}]$	Komponentenweise quantisierter Eingangsvektor
\mathbf{U}	Folge von Eingangsvektoren
$[\mathbf{U}]$	Folge quantisierter Eingangsvektoren
$[\mathbf{U}]$	Folge komponentenweise quantisierter Eingangsvektoren
\mathcal{U}	Menge qualitativer Eingänge, Qualitativer Eingangsraum
\mathcal{U}	Menge qualitativer Eingangsvektoren/-tupel
v	Eingabe des SA/SP, qualitativer Eingang
\mathbf{v}	Eingabevektor/-tupel des SA/SP, qualitativer Eingangsvektor/-tupel
V	Eingabefolge des SA/SP, Zufallsvariable der Eingabe des SA/SP
\mathbf{V}	Folge von Eingabevektoren des SA/SP
\mathcal{V}	Menge der Eingaben des SA/SP
\mathcal{V}	Menge von Eingabevektoren/-tupeln des SA/SP
w	Ausgabe des SA/SP, qualitativer Ausgang
\mathbf{w}	Ausgabevektor/-tupel des SA/SP, qualitativer Ausgangsvektor/-tupel
W	Ausgabefolge des SA/SP, Zufallsvariable der Ausgabe des SA/SP
\mathbf{W}	Folge von Ausgabevektoren des SA/SP
\mathcal{W}	Menge der Ausgaben des SA/SP
\mathcal{W}	Menge von Ausgabevektoren/-tupeln des SA/SP
\mathbf{x}	Zustandsvektor, Zustandsgröße, Zustand

Symbol	Bedeutung
$[\mathbf{x}]$	Quantisierter Zustandsvektor
\mathbf{x}	Komponentenweise quantisierter Zustandsvektor
\mathbf{X}	Folge von Zustandsvektoren
$[\mathbf{X}]$	Folge quantisierter Zustandsvektoren
$[\mathbf{X}]$	Folge komponentenweise quantisierter Zustandsvektoren
\mathcal{X}	Menge qualitativer Zustände, Qualitativer Zustandsraum
\mathcal{X}	Menge qualitativer Zustandsvektoren/-tupel
\mathbf{y}	Ausgangsvektor, Ausgangsgröße, Ausgang
$[\mathbf{y}]$	Quantisierter Ausgangsvektor
\mathbf{y}	Komponentenweise quantisierter Ausgangsvektor
\mathbf{Y}	Folge von Ausgangsvektoren
$[\mathbf{Y}]$	Folge quantisierter Ausgangsvektoren
$[\mathbf{Y}]$	Folge komponentenweise quantisierter Ausgangsvektoren
\mathcal{Y}	Menge der qualitativen Ausgänge, Qualitativer Ausgangsraum
\mathcal{Y}	Menge qualitativer Ausgangsvektoren/-tupel
z	Zustand des SA/SP, qualitativer Zustand
\mathbf{z}	Zustandsvektor/-tupel des SA/SP, qualitativer Zustandsvektor/-tupel
\bar{z}	Folgezustand des SA/SP, qualitativer Folgezustand
$\bar{\mathbf{z}}$	Folgezustandsvektor/-tupel des SA/SP, qual. Folgezustandsvektor/-tupel
Z	Zustandsfolge des SA/SP, Zufallsvariable des Zustands des SA/SP
\mathbf{Z}	Folge von Zustandsvektoren des SA/SP
\mathcal{Z}	Menge der Zustände des SA/SP
\mathcal{Z}	Menge von Zustandsvektoren/-tupeln des SA/SP

Tabelle VI: Griechische Symbole

Symbol	Bedeutung
α	Hilfsvektor
β	Hilfsvektor
γ	Anzahl (zu speichernder) Werte des Verhaltenstensor
γ_{CP}	Anzahl (zu speichernder) Werte der CP-Dekomposition
γ_{TT}	Anzahl (zu speichernder) Werte der TT-Dekomposition
ε	Approximationsgenauigkeit der-TT Dekomposition
Θ	Tupel qualitativer Größen
λ	Gewichtungsvektor der CP-Dekomposition
ξ_{CP}	Reduktionsfaktor der CP-Dekomposition
ξ_{TT}	Reduktionsfaktor der TT-Dekomposition
ρ	Hilfsvektor
τ	Schwellwert für die Fehlererkennung
ϕ	Fenster des gleitenden Mittelwerts
Ψ	Hilfsvektor
Ω	Hilfsvektor

B

Beweise

B.1 Beweis der Proposition 7.1

Es sei $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ der Verhaltenstensor mit den Elementen $l(i, j, s, t)$, wobei die Folgezustände und Zustände Werte in $i \in \{1, \dots, N\}$ und $s \in \{1, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j \in \{1, \dots, Q\}$ und die der Eingaben $t \in \{1, \dots, M\}$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit den Komponenten $p_z(k, s)$, $\mathbf{p}_v(k) \in \{0, 1\}^M$ der Eingabewahrscheinlichkeitsvektor mit Komponenten $p_v(k, t)$ und $\mathbf{p}_w(k) \in \{0, 1\}^Q$ der Ausgabewahrscheinlichkeitsvektor mit Komponenten $p_w(k, j)$.

Beweis: Es ist

$$\begin{aligned} & \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle \in [0, 1]^N \\ &= \left\langle \mathbf{L} \mid (\mathbf{p}_w(k) \circ \mathbf{p}_z(k) \circ \mathbf{p}_v(k)) \right\rangle \\ &= \left\langle \mathbf{L} \mid \mathbf{U}(k) \right\rangle. \end{aligned} \tag{B.1}$$

Der aus den dyadischen Produkten der Vektoren $\mathbf{p}_w(k) \in \{0, 1\}^Q$, $\mathbf{p}_z(k) \in [0, 1]^N$ und $\mathbf{p}_v(k) \in \{0, 1\}^M$ resultierende Produkttensor $\mathbf{U}(k)$ ist dementsprechend ein Tensor der Größe $Q \times N \times M$ mit Elementen

$$u(j, s, t)(k) = p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \quad \forall j, s, t.$$

Das kontrahierte Tensorprodukt aus dem Verhaltenstensor $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ und dem Produkttensor $\mathbf{U}(k) \in [0, 1]^{Q \times N \times M}$ ergibt nach Definition 6.15 einen Vektor der Länge N .

Elementweise notiert erhält man damit

$$\begin{aligned} & \left\langle \mathbf{L} \mid \mathbf{U}(k) \right\rangle \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(1, j, s, t) \cdot u(j, s, t)(k) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(2, j, s, t) \cdot u(j, s, t)(k) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(N, j, s, t) \cdot u(j, s, t)(k) \end{pmatrix} \in [0, 1]^N \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(1, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(2, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(N, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \end{pmatrix} \in [0, 1]^N. \end{aligned} \tag{B.2}$$

Sind nun die Verteilungen der Eingabe- und Ausgabevektoren so, dass diese nur an der $v(k)$ -ten und $w(k)$ -ten Position den Wert eins annehmen, also

$$\mathbf{p}_v(k) = \begin{pmatrix} p_v(k, 1) \\ \vdots \\ p_v(k, v(k)) \\ \vdots \\ p_v(k, M) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^M, \quad \mathbf{p}_w(k) = \begin{pmatrix} p_w(k, 1) \\ \vdots \\ p_w(k, w(k)) \\ \vdots \\ p_w(k, Q) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^Q,$$

dann gilt für das in (B.2) auftretende Produkt

$$p_w(k, j) \cdot p_v(k, t) = \begin{cases} 1 & \text{für } j = w(k) \text{ und } t = v(k), \\ 0 & \text{sonst.} \end{cases}$$

Damit entfällt in Gleichung (B.2) die Summation über die Indizes j und t und die Tensorelemente $l(i, j, s, t)$ können direkt mit $w(k)$ und $v(k)$ indiziert werden. Damit erhält man

$$\begin{aligned} & \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(1, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(2, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(N, j, s, t) \cdot p_w(k, j) \cdot p_z(k, s) \cdot p_v(k, t) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{s=1}^N l(1, w(k), s, v(k)) \cdot p_z(k, s) \\ \sum_{s=1}^N l(2, w(k), s, v(k)) \cdot p_z(k, s) \\ \vdots \\ \sum_{s=1}^N l(N, w(k), s, v(k)) \cdot p_z(k, s) \end{pmatrix} \\ &= \begin{pmatrix} l(1, w(k), 1, v(k)) & l(1, w(k), 2, v(k)) & \dots & l(1, w(k), N, v(k)) \\ l(2, w(k), 1, v(k)) & l(2, w(k), 2, v(k)) & \dots & l(2, w(k), N, v(k)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N, w(k), 1, v(k)) & l(N, w(k), 2, v(k)) & \dots & l(N, w(k), N, v(k)) \end{pmatrix} \cdot \mathbf{p}_z(k) \\ &= \mathbf{L}(:, w(k), :, v(k)) \cdot \mathbf{p}_z(k) \in [0, 1]^N. \end{aligned}$$

□

B.2 Beweis der Proposition 9.1

Es sei $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ der Verhaltenstensor mit Elementen $l(i, j, s, t)$, wobei die Folgezustände Werte in $i \in \{1, \dots, N\}$ und die Zustände Werte in $s \in \{1, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j \in \{1, \dots, Q\}$ und die der Eingaben $t \in \{1, \dots, M\}$ gilt. Für eine *exakte* CP-Dekomposition gilt dann nach Definition 8.6

$$\mathbf{L} = [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!] = \sum_{r=1}^R \lambda_r \mathbf{a}_{:r} \circ \mathbf{b}_{:r} \circ \mathbf{c}_{:r} \circ \mathbf{d}_{:r},$$

mit der elementweisen Notation

$$l(i, j, s, t) = \sum_{r=1}^R \lambda_r a_{ir} b_{jr} c_{sr} d_{tr}.$$

Dabei sind $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}$ die Faktormatrizen und $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$ der Gewichtungsvektor. Damit kann die Proposition 9.1 wie folgt bewiesen werden:

Beweis: Es ist

$$\begin{aligned}\alpha(k) &= \lambda \otimes (\mathbf{b}(j(k), :) \otimes \mathbf{d}(t(k), :))^T \\ &= \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_R \end{pmatrix} \otimes \begin{pmatrix} b_{j(k)1} d_{t(k)1} \\ b_{j(k)2} d_{t(k)1} \\ \vdots \\ b_{j(k)R} d_{t(k)1} \end{pmatrix} = \begin{pmatrix} \lambda_1 b_{j(k)1} d_{t(k)1} \\ \lambda_2 b_{j(k)2} d_{t(k)2} \\ \vdots \\ \lambda_R b_{j(k)R} d_{t(k)R} \end{pmatrix}.\end{aligned}$$

Dann ist

$$\begin{aligned}\llbracket \alpha(k); \mathbf{A}, \mathbf{C} \rrbracket &= \sum_{r=1}^R \alpha_r(k) \mathbf{a}_{:r} \circ \mathbf{c}_{:r} \\ &= \sum_{r=1}^R \alpha_r(k) \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{Nr} \end{pmatrix} \circ \begin{pmatrix} c_{1r} \\ c_{2r} \\ \vdots \\ c_{Nr} \end{pmatrix} \\ &= \sum_{r=1}^R \alpha_r(k) \begin{pmatrix} a_{1r}c_{1r} & a_{1r}c_{2r} & \cdots & a_{1r}c_{Nr} \\ a_{2r}c_{1r} & a_{2r}c_{2r} & \cdots & a_{2r}c_{Nr} \\ \vdots & \vdots & \ddots & \vdots \\ a_{Nr}c_{1r} & a_{Nr}c_{2r} & \cdots & a_{Nr}c_{Nr} \end{pmatrix} \\ &= \begin{pmatrix} \sum_{r=1}^R \alpha_r(k) a_{1r}c_{1r} & \sum_{r=1}^R \alpha_r(k) a_{1r}c_{2r} & \cdots & \sum_{r=1}^R \alpha_r(k) a_{1r}c_{Nr} \\ \sum_{r=1}^R \alpha_r(k) a_{2r}c_{1r} & \sum_{r=1}^R \alpha_r(k) a_{2r}c_{2r} & \cdots & \sum_{r=1}^R \alpha_r(k) a_{2r}c_{Nr} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r=1}^R \alpha_r(k) a_{Nr}c_{1r} & \sum_{r=1}^R \alpha_r(k) a_{Nr}c_{2r} & \cdots & \sum_{r=1}^R \alpha_r(k) a_{Nr}c_{Nr} \end{pmatrix} \\ &= \begin{pmatrix} \sum_{r=1}^R \lambda_r a_{1r} b_{j(k)r} c_{1r} d_{t(k)r} & \sum_{r=1}^R \lambda_r a_{1r} b_{j(k)r} c_{2r} d_{t(k)r} & \cdots & \sum_{r=1}^R \lambda_r a_{1r} b_{j(k)r} c_{Nr} d_{t(k)r} \\ \sum_{r=1}^R \lambda_r a_{2r} b_{j(k)r} c_{1r} d_{t(k)r} & \sum_{r=1}^R \lambda_r a_{2r} b_{j(k)r} c_{2r} d_{t(k)r} & \cdots & \sum_{r=1}^R \lambda_r a_{2r} b_{j(k)r} c_{Nr} d_{t(k)r} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{r=1}^R \lambda_r a_{Nr} b_{j(k)r} c_{1r} d_{t(k)r} & \sum_{r=1}^R \lambda_r a_{Nr} b_{j(k)r} c_{2r} d_{t(k)r} & \cdots & \underbrace{\sum_{r=1}^R \lambda_r a_{Nr} b_{j(k)r} c_{Nr} d_{t(k)r}}_{= l(N, j(k), N, t(k))} \end{pmatrix} \\ &= \begin{pmatrix} l(1, j(k), 1, t(k)) & l(1, j(k), 2, t(k)) & \cdots & l(1, j(k), N, t(k)) \\ l(2, j(k), 1, t(k)) & l(2, j(k), 2, t(k)) & \cdots & l(2, j(k), N, t(k)) \\ \vdots & \vdots & \ddots & \vdots \\ l(N, j(k), 1, t(k)) & l(N, j(k), 2, t(k)) & \cdots & l(N, j(k), N, t(k)) \end{pmatrix} \\ &= \mathbf{L}(:, j(k), :, t(k)) \in [0, 1]^{N \times N}.\end{aligned}$$

□

B.3 Beweis der Proposition 9.2

Es sei $\mathbf{L} \in [0, 1]^{N \times Q \times N \times M}$ der Verhaltenstensor mit Elementen $l(i, j, s, t)$, wobei die Folgezustände Werte in $i \in \{1, \dots, N\}$ und die Zustände Werte in $s \in \{1, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j \in \{1, \dots, Q\}$ und die der Eingaben $t \in \{1, \dots, M\}$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit Komponenten $p_z(k, s)$, $\mathbf{p}_v(k) \in [0, 1]^M$ der Eingabewahrscheinlichkeitsvektor mit Komponenten $p_v(k, t)$ und $\mathbf{p}_w(k) \in [0, 1]^Q$ der Ausgabe-wahrscheinlichkeitsvektor mit Komponenten $p_w(k, j)$. Für eine *exakte* CP-Dekomposition gilt dann nach Definition 8.6

$$\mathbf{L} = \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \rrbracket = \sum_{r=1}^R \lambda_r \mathbf{a}_{:r} \circ \mathbf{b}_{:r} \circ \mathbf{c}_{:r} \circ \mathbf{d}_{:r}$$

mit der elementweisen Notation

$$l(i, j, s, t) = \sum_{r=1}^R \lambda_r a_{ir} b_{jr} c_{sr} d_{tr} .$$

Dabei sind $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{B} \in \mathbb{R}_{\geq 0}^{Q \times R}$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{D} \in \mathbb{R}_{\geq 0}^{M \times R}$ die Faktormatrizen und $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$ der Gewichtsvektor. Damit kann die Proposition 9.1 wie folgt bewiesen werden:

Beweis: Es ist

$$\begin{aligned} & \mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\mathbf{B}^\top \cdot \mathbf{p}_w(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\mathbf{D}^\top \cdot \mathbf{p}_v(k) \right) \right) \\ &= \mathbf{A} \cdot \left(\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_R \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{21} & \cdots & b_{Q1} \\ b_{12} & b_{22} & \cdots & b_{Q2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1R} & b_{2R} & \cdots & b_{QR} \end{pmatrix} \cdot \begin{pmatrix} p_w(k, 1) \\ p_w(k, 2) \\ \vdots \\ p_w(k, Q) \end{pmatrix} \right) \\ & \quad \otimes \begin{pmatrix} c_{11} & c_{21} & \cdots & c_{N1} \\ c_{12} & c_{22} & \cdots & c_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1R} & c_{2R} & \cdots & c_{NR} \end{pmatrix} \cdot \begin{pmatrix} p_z(k, 1) \\ p_z(k, 2) \\ \vdots \\ p_z(k, N) \end{pmatrix} \\ & \quad \otimes \begin{pmatrix} d_{11} & d_{21} & \cdots & d_{M1} \\ d_{12} & d_{22} & \cdots & d_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1R} & d_{2R} & \cdots & d_{MR} \end{pmatrix} \cdot \begin{pmatrix} p_v(k, 1) \\ p_v(k, 2) \\ \vdots \\ p_v(k, M) \end{pmatrix} \right) \\ &= \mathbf{A} \cdot \left(\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_R \end{pmatrix} \otimes \begin{pmatrix} \sum_{j=1}^Q b_{j1} p_w(k, j) \\ \sum_{j=1}^Q b_{j2} p_w(k, j) \\ \vdots \\ \sum_{j=1}^Q b_{jR} p_w(k, j) \end{pmatrix} \otimes \begin{pmatrix} \sum_{s=1}^N c_{s1} p_z(k, s) \\ \sum_{s=1}^N c_{s2} p_z(k, s) \\ \vdots \\ \sum_{s=1}^N c_{sR} p_z(k, s) \end{pmatrix} \otimes \begin{pmatrix} \sum_{t=1}^M d_{t1} p_v(k, t) \\ \sum_{t=1}^M d_{t2} p_v(k, t) \\ \vdots \\ \sum_{t=1}^M d_{tR} p_v(k, t) \end{pmatrix} \right) \\ &= \mathbf{A} \cdot \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_1 b_{j1} c_{s1} d_{t1} p_w(k, j) p_z(k, s) p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_2 b_{j2} c_{s2} d_{t2} p_w(k, j) p_z(k, s) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_R b_{jR} c_{sR} d_{tR} p_w(k, j) p_z(k, s) p_v(k, t) \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1R} \\ a_{21} & a_{22} & \cdots & a_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NR} \end{pmatrix} \cdot \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_1 b_{j1} c_{s1} d_{t1} p_w(k, j) p_z(k, s) p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_2 b_{j2} c_{s2} d_{t2} p_w(k, j) p_z(k, s) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \lambda_R b_{jR} c_{sR} d_{tR} p_w(k, j) p_z(k, s) p_v(k, t) \end{pmatrix} \\
&= \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \sum_{r=1}^R \lambda_r a_{1r} b_{jr} c_{sr} d_{tr} p_w(k, j) p_z(k, s) p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \sum_{r=1}^R \lambda_r a_{2r} b_{jr} c_{sr} d_{tr} p_w(k, j) p_z(k, s) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M \underbrace{\sum_{r=1}^R \lambda_r a_{Nr} b_{jr} c_{sr} d_{tr}}_{= l(N, j, s, t)} p_w(k, j) p_z(k, s) p_v(k, t) \end{pmatrix} \\
&= \begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(1, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(2, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(N, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \end{pmatrix} \in [0, 1]^N. \tag{B.3}
\end{aligned}$$

Der Ausdruck (B.3) entspricht der elementweisen Notation (B.2) des kontrahierten Tensorprodukts (B.1) im Beweis B.1. Aus der Gleichheit zwischen (B.1) und (B.2) im Beweis B.1 folgt damit unmittelbar

$$\begin{aligned}
&\begin{pmatrix} \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(1, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(2, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{s=1}^N \sum_{t=1}^M l(N, j, s, t) p_w(k, j) p_z(k, s) p_v(k, t) \end{pmatrix} \\
&= \left\langle \underline{\mathbf{L}} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_z(k), \mathbf{p}_v(k) \rrbracket \right\rangle \in [0, 1]^N.
\end{aligned}$$

□

B.4 Beweis der Proposition 9.3

Es sei $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ der permutierte Verhaltenstensor mit Elementen $\underline{l}(j, t, i, s)$, wobei die Folgezustände und Zustände Werte in $i \in \{1, \dots, N\}$ und $s \in \{1, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j \in \{1, \dots, Q\}$ und die der Eingaben $t \in \{1, \dots, M\}$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit Komponenten $p_z(k, s)$, $\mathbf{p}_v(k) \in \{0, 1\}^M$ der Eingabewahrscheinlichkeitsvektor mit Komponenten $p_v(k, t)$ und $\mathbf{p}_w(k) \in \{0, 1\}^Q$ der Ausgabewahrscheinlichkeitsvektor mit Komponenten $p_w(k, j)$.

Beweis: Es ist

$$\begin{aligned}
&\left\langle \underline{\mathbf{L}} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)} \tag{B.4} \\
&= \left\langle \underline{\mathbf{L}} \mid (\mathbf{p}_w(k) \circ \mathbf{p}_v(k) \circ \mathbf{p}_z(k)) \right\rangle_{(1,2,4|1,2,3)} \\
&= \left\langle \underline{\mathbf{L}} \mid \mathbf{U}(k) \right\rangle_{(1,2,4|1,2,3)}
\end{aligned}$$

Der aus den dyadischen Produkten der Vektoren $\mathbf{p}_w(k) \in \{0, 1\}^Q$, $\mathbf{p}_v(k) \in \{0, 1\}^M$ und $\mathbf{p}_z(k) \in [0, 1]^N$ resultierende Produkttensor $\mathbf{U}(k)$ ist dementsprechend ein Tensor der Größe $Q \times M \times N$ mit Elementen

$$u(j, t, s)(k) = p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \quad \forall j, t, s.$$

Das kontrahierte Tensorprodukt aus dem Verhaltenstensor $\underline{\mathbf{L}} \in [0, 1]^{Q \times M \times N \times N}$ und dem Produkttensor $\mathbf{U}(k) \in [0, 1]^{Q \times M \times N}$ ergibt nach Definition 6.15 wieder einen Vektor der Länge N . Elementweise notiert erhält man damit

$$\begin{aligned} \left\langle \underline{\mathbf{L}} \mid \mathbf{U}(k) \right\rangle_{(1,2,4|1,2,3)} &= \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, 1, s) \cdot u(j, t, s)(k) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, 2, s) \cdot u(j, t, s)(k) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, N, s) \cdot u(j, t, s)(k) \end{pmatrix} \in [0, 1]^N \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, 1, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, 2, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N l(j, t, N, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \end{pmatrix} \in [0, 1]^N . \end{aligned} \quad (\text{B.5})$$

Sind nun die Verteilungen der Eingabe- und Ausgabevektoren so, dass diese nur an der $v(k)$ -ten und $w(k)$ -ten Position den Wert eins annehmen, also

$$\mathbf{p}_v(k) = \begin{pmatrix} p_v(k, 1) \\ \vdots \\ p_v(k, v(k)) \\ \vdots \\ p_v(k, M) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^M, \quad \mathbf{p}_w(k) = \begin{pmatrix} p_w(k, 1) \\ \vdots \\ p_w(k, w(k)) \\ \vdots \\ p_w(k, Q) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^Q,$$

dann gilt für das in (B.5) auftretende Produkt

$$p_w(k, j) \cdot p_v(k, t) = \begin{cases} 1 & \text{für } j = w(k) \text{ und } t = v(k), \\ 0 & \text{sonst.} \end{cases}$$

Damit entfällt in Gleichung (B.5) die Summation über die Indizes j und t und die Tensorelemente $\underline{l}(j, t, i, s)$ können direkt mit $w(k)$ und $v(k)$ indiziert werden. Damit erhält man

$$\begin{aligned} &\begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 1, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 2, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, N, s) \cdot p_w(k, j) \cdot p_v(k, t) \cdot p_z(k, s) \end{pmatrix} \in [0, 1]^N \\ &= \begin{pmatrix} \sum_{s=1}^N \underline{l}(w(k), v(k), 1, s) \cdot p_z(k, s) \\ \sum_{s=1}^N \underline{l}(w(k), v(k), 2, s) \cdot p_z(k, s) \\ \vdots \\ \sum_{s=1}^N \underline{l}(w(k), v(k), N, s) \cdot p_z(k, s) \end{pmatrix} \\ &= \begin{pmatrix} \underline{l}(w(k), v(k), 1, 1) & \underline{l}(w(k), v(k), 1, 2) & \dots & \underline{l}(w(k), v(k), 1, N) \\ \underline{l}(w(k), v(k), 2, 1) & \underline{l}(w(k), v(k), 2, 2) & \dots & \underline{l}(w(k), v(k), 2, N) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{l}(w(k), v(k), N, 1) & \underline{l}(w(k), v(k), N, 2) & \dots & \underline{l}(w(k), v(k), N, N) \end{pmatrix} \cdot \mathbf{p}_z(k) \\ &= \mathbf{L}(w(k), v(k), :, :) \cdot \mathbf{p}_z(k) \in [0, 1]^N . \end{aligned}$$

□

B.5 Beweis der Proposition 9.4

Es sei $\underline{L} \in [0, 1]^{Q \times M \times N \times N}$ der permutierte Verhaltenstensor mit Elementen $\underline{L}(j, t, i, s)$, wobei die Folgezustände und Zustände Werte in $i \in \{1, \dots, N\}$ und $s \in \{1, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j \in \{1, \dots, Q\}$ und die der Eingaben $t \in \{1, \dots, M\}$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit Komponenten $p_z(k, s)$, $\mathbf{p}_v(k) \in [0, 1]^M$ der Eingabewahrscheinlichkeitsvektor mit Komponenten $p_v(k, t)$ und $\mathbf{p}_w(k) \in [0, 1]^Q$ der Ausgabewahrscheinlichkeitsvektor mit Komponenten $p_w(k, j)$.

Für eine *exakte* TT Dekomposition gilt dann nach Definition 8.11

$$\underline{L} = \mathbf{E} \odot \mathbf{F} \odot \mathbf{G} \odot \mathbf{H}$$

mit der elementweisen Notation

$$\underline{L}(j, t, i, s) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1tr_2} g_{r_2ir_3} h_{r_3s1}.$$

Dabei sind $\mathbf{E} \in \mathbb{R}^{1 \times Q \times R_1}$, $\mathbf{F} \in \mathbb{R}^{R_1 \times M \times R_2}$, $\mathbf{G} \in \mathbb{R}^{R_2 \times N \times R_3}$ und $\mathbf{H} \in \mathbb{R}^{R_3 \times N \times 1}$ die TT-Kerne.

Beweis: Es ist

$$\begin{aligned} (\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) &= \left(\sum_{j=1}^Q e_{1j1} p_w(k, j) \quad \sum_{j=1}^Q e_{1j2} p_w(k, j) \cdots \sum_{j=1}^Q e_{1jR_1} p_w(k, j) \right) \in \mathbb{R}^{1 \times R_1}, \\ (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) &= \begin{pmatrix} \sum_{t=1}^M f_{1t1} p_v(k, t) & \sum_{t=1}^M f_{1t2} p_v(k, t) \cdots \sum_{t=1}^M f_{1tR_2} p_v(k, t) \\ \sum_{t=1}^M f_{2t1} p_v(k, t) & \sum_{t=1}^M f_{2t2} p_v(k, t) \cdots \sum_{t=1}^M f_{2tR_2} p_v(k, t) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{t=1}^M f_{R_1t1} p_v(k, t) & \sum_{t=1}^M f_{R_1t2} p_v(k, t) \cdots \sum_{t=1}^M f_{R_1tR_2} p_v(k, t) \end{pmatrix} \in \mathbb{R}^{R_1 \times R_2}, \\ (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k)) &= \begin{pmatrix} \sum_{s=1}^N h_{1s1} p_z(k, s) \\ \sum_{s=1}^N h_{2s1} p_z(k, s) \\ \vdots \\ \sum_{s=1}^N h_{R_3s1} p_z(k, s) \end{pmatrix} \in \mathbb{R}^{R_3 \times 1}. \end{aligned}$$

Dann ist

$$\begin{aligned} &(\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \\ &= \left(\sum_{j=1}^Q e_{1j1} p_w(k, j) \quad \sum_{j=1}^Q e_{1j2} p_w(k, j) \cdots \sum_{j=1}^Q e_{1jR_1} p_w(k, j) \right) \\ &\odot \begin{pmatrix} \sum_{t=1}^M f_{1t1} p_v(k, t) & \sum_{t=1}^M f_{1t2} p_v(k, t) \cdots \sum_{t=1}^M f_{1tR_2} p_v(k, t) \\ \sum_{t=1}^M f_{2t1} p_v(k, t) & \sum_{t=1}^M f_{2t2} p_v(k, t) \cdots \sum_{t=1}^M f_{2tR_2} p_v(k, t) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{t=1}^M f_{R_1t1} p_v(k, t) & \sum_{t=1}^M f_{R_1t2} p_v(k, t) \cdots \sum_{t=1}^M f_{R_1tR_2} p_v(k, t) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{r_1=1}^{R_1} e_{1jr_1} f_{r_1t1} p_w(k, j) p_v(k, t) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{r_1=1}^{R_1} e_{1jr_1} f_{r_1t2} p_w(k, j) p_v(k, t) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{r_1=1}^{R_1} e_{1jr_1} f_{r_1tR_2} p_w(k, j) p_v(k, t) \end{pmatrix}^T \in \mathbb{R}^{1 \times R_2}. \end{aligned} \tag{B.6}$$

Die in (B.6) auftretenden Summen und Produkte werden mit $\Sigma := \sum_{j=1}^Q \sum_{t=1}^M \sum_{r_1=1}^{R_1}$ und $\bar{P} := p_w(k, j) p_v(k, t)$ abgekürzt.

Dann ist

$$\begin{aligned} & (\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \\ &= \begin{pmatrix} \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 11} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 12} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 1R_3} \bar{P} \\ \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 21} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 22} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 2R_3} \bar{P} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 N1} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 N2} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 NR_3} \bar{P} \end{pmatrix} \in \mathbb{R}^{1 \times N \times R_3}. \end{aligned}$$

Setzt man nun alle Terme zusammen, erhält man

$$\begin{aligned} & (\mathbf{E} \bar{\times}_{(2)} \mathbf{p}_w(k)) \odot (\mathbf{F} \bar{\times}_{(2)} \mathbf{p}_v(k)) \odot \mathbf{G} \odot (\mathbf{H} \bar{\times}_{(2)} \mathbf{p}_z(k)) \\ &= \begin{pmatrix} \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 11} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 12} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 1R_3} \bar{P} \\ \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 21} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 22} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 2R_3} \bar{P} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 N1} \bar{P} & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 N2} \bar{P} & \cdots & \Sigma \sum_{r_2=1}^{R_2} e_{1jr_1} f_{r_1 tr_2} g_{r_2 NR_3} \bar{P} \end{pmatrix} \odot \begin{pmatrix} \sum_{s=1}^N h_{1s1} p_z(k, s) \\ \sum_{s=1}^N h_{2s1} p_z(k, s) \\ \vdots \\ \sum_{s=1}^N h_{R_3s1} p_z(k, s) \end{pmatrix} \\ &= \begin{pmatrix} \Sigma \sum_{s=1}^N \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 1r_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \\ \Sigma \sum_{s=1}^N \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 2r_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \\ \vdots \\ \Sigma \sum_{s=1}^N \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 Nr_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 1r_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 2r_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} e_{1jr_1} f_{r_1 tr_2} g_{r_2 Nr_3} h_{r_3s1} p_w(k, j) p_v(k, t) p_z(k, s) \end{pmatrix} \in [0, 1]^{1 \times N \times 1} \\ & \quad \quad \quad = \underline{l}(j, t, N, s) \\ &= \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 1, s) p_w(k, j) p_v(k, t) p_z(k, s) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 2, s) p_w(k, j) p_v(k, t) p_z(k, s) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, N, s) p_w(k, j) p_v(k, t) p_z(k, s) \end{pmatrix} \in [0, 1]^{1 \times N \times 1}. \end{aligned} \tag{B.7}$$

Der Ausdruck (B.7) kann als Spaltenvektor der Länge N interpretiert werden und entspricht der elementweisen Notation (B.5) des kontrahierten Tensorprodukts (B.4) im Beweis B.4. Aus der Gleichheit zwischen (B.4) und (B.5) im Beweis B.4 folgt damit unmittelbar

$$\begin{aligned} & \begin{pmatrix} \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 1, s) p_w(k, j) p_v(k, t) p_z(k, s) \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, 2, s) p_w(k, j) p_v(k, t) p_z(k, s) \\ \vdots \\ \sum_{j=1}^Q \sum_{t=1}^M \sum_{s=1}^N \underline{l}(j, t, N, s) p_w(k, j) p_v(k, t) p_z(k, s) \end{pmatrix} \in [0, 1]^N \\ &= \left\langle \underline{l} \mid \llbracket \mathbf{p}_w(k), \mathbf{p}_v(k), \mathbf{p}_z(k) \rrbracket \right\rangle_{(1,2,4|1,2,3)} \in [0, 1]^N. \end{aligned}$$

□

B.6 Beweis der Proposition 10.1

Gegeben sei der Verhaltenstensor $\mathbf{L} \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m}$ mit den Elementen $l(i, j_1, j_2, \dots, j_q, s, t_1, t_2, \dots, t_m)$, wobei die Folgezustände und Zustände Werte in $i \in \{1, 2, \dots, N\}$ und $s \in \{1, 2, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j_e \in \{1, 2, \dots, Q_e\}$, für $e = 1, 2, \dots, q$ und für die Werte der Eingaben $t_h \in \{1, 2, \dots, M_h\}$ für $h = 1, \dots, m$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit den Komponenten $p_z(k, s)$ während die Eingabewahrscheinlichkeitsvektoren $\mathbf{p}_v^{(h)}(k) \in \{0, 1\}^{M_h}$ mit den Komponenten $p_v^{(h)}(k, t_h)$ für $h = 1, \dots, m$ und die Ausgabewahrscheinlichkeitsvektoren $\mathbf{p}_w^{(e)}(k) \in \{0, 1\}^{Q_e}$ mit Komponenten $p_w^{(e)}(k, j_e)$ für $e = 1, 2, \dots, q$ gegeben sind.

Beweis: Es ist

$$\left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w^{(1)}(k), \dots, \mathbf{p}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle \in [0, 1]^N \quad (\text{B.8})$$

$$\begin{aligned} &= \left\langle \mathbf{L} \mid (\mathbf{p}_w^{(1)}(k) \circ \dots \circ \mathbf{p}_w^{(q)}(k) \circ \mathbf{p}_z(k) \circ \mathbf{p}_v^{(1)}(k) \circ \dots \circ \mathbf{p}_v^{(m)}(k)) \right\rangle \quad (\text{B.9}) \\ &= \left\langle \mathbf{L} \mid \mathbf{U}(k) \right\rangle \end{aligned}$$

Der aus den dyadischen Produkten in (B.9) resultierende Produkttensor $\mathbf{U}(k)$, ist ein Tensor der Größe $Q_1 \times \dots \times Q_q \times N \times M_1 \times \dots \times M_m$ mit Elementen

$$u(j_1, \dots, j_q, s, t_1, \dots, t_m)(k) = p_w^{(1)}(k, j_1) \dots p_w^{(q)}(k, j_q) p_z(k, s) p_v^{(1)}(k, t_1) \dots p_v^{(m)}(k, t_m) .$$

Das kontrahierte Tensorprodukt des Verhaltenstensors $\mathbf{L} \in [0, 1]^{N \times Q_1 \times \dots \times Q_q \times N \times M_1 \times \dots \times M_m}$ und des Tensors $\mathbf{U}(k) \in [0, 1]^{Q_1 \times \dots \times Q_q \times N \times M_1 \times \dots \times M_m}$ ergibt nach Definition 6.15 einen Vektor der Länge N .

Elementweise notiert erhält man damit

$$\begin{aligned} &\left\langle \mathbf{L} \mid \mathbf{U}(k) \right\rangle \\ &= \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(1, j_1, \dots, j_q, s, t_1, \dots, t_m) u(j_1, \dots, j_q, s, t_1, \dots, t_m)(k) \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(2, j_1, \dots, j_q, s, t_1, \dots, t_m) u(j_1, \dots, j_q, s, t_1, \dots, t_m)(k) \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(N, j_1, \dots, j_q, s, t_1, \dots, t_m) u(j_1, \dots, j_q, s, t_1, \dots, t_m)(k) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(1, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(2, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(N, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \end{pmatrix} \in [0, 1]^N , \quad (\text{B.10}) \end{aligned}$$

wobei in (B.10) die Abkürzung

$$\bar{P} := p_w^{(1)}(k, j_1) \dots p_w^{(q)}(k, j_q) p_z(k, s) p_v^{(1)}(k, t_1) \dots p_v^{(m)}(k, t_m)$$

verwendet wurde. Sind nun die Verteilungen der Ausgabe- und Eingabevektoren so, dass diese für $e = 1, \dots, q$ und $h = 1, \dots, m$ nur an der $w_e(k)$ -ten und $v_h(k)$ -ten Position den Wert eins annehmen, also

$$\mathbf{p}_w^{(e)}(k) = \begin{pmatrix} p_w^{(e)}(k, 1) \\ \vdots \\ p_w^{(e)}(k, w_e(k)) \\ \vdots \\ p_w^{(e)}(k, Q_e) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^{Q_e}, \quad \mathbf{p}_v^{(h)}(k) = \begin{pmatrix} p_v^{(h)}(k, 1) \\ \vdots \\ p_v^{(h)}(k, v_h(k)) \\ \vdots \\ p_v^{(h)}(k, M_h) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^{M_h},$$

dann gilt für das in (B.10) auftretende Produkt

$$p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m) = \begin{cases} 1 & \text{für } j_1 = w_1(k), \dots, j_q = w_q(k) \text{ und} \\ & t_1 = v_1(k), \dots, t_m = v_m(k), \\ 0 & \text{sonst.} \end{cases} \quad (\text{B.11})$$

Damit entfällt in (B.10) das Produkt (B.11) und die Summation über die Indizes j_1, \dots, j_q und t_1, \dots, t_m , womit die Tensorelemente $l(i, j_1, \dots, j_q, s, t_1, \dots, t_m)$ mit $w_1(k), \dots, w_q(k)$ und $v_1(k), \dots, v_m(k)$ indiziert werden können.

Damit erhält man

$$\begin{aligned} & \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(1, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(2, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} l(N, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \end{pmatrix} \in [0, 1]^N \\ &= \begin{pmatrix} \sum_{s=1}^N l(1, w_1(k), \dots, w_q(k), s, v_1(k), \dots, v_m(k)) p_z(k, s) \\ \sum_{s=1}^N l(2, w_1(k), \dots, w_q(k), s, v_1(k), \dots, v_m(k)) p_z(k, s) \\ \vdots \\ \sum_{s=1}^N l(N, w_1(k), \dots, w_q(k), s, v_1(k), \dots, v_m(k)) p_z(k, s) \end{pmatrix} \\ &= \begin{pmatrix} l(1, w_1(k), \dots, w_q(k), 1, v_1(k), \dots, v_m(k)) \dots l(1, w_1(k), \dots, w_q(k), N, v_1(k), \dots, v_m(k)) \\ l(2, w_1(k), \dots, w_q(k), 1, v_1(k), \dots, v_m(k)) \dots l(2, w_1(k), \dots, w_q(k), N, v_1(k), \dots, v_m(k)) \\ \vdots \\ l(N, w_1(k), \dots, w_q(k), 1, v_1(k), \dots, v_m(k)) \dots l(N, w_1(k), \dots, w_q(k), N, v_1(k), \dots, v_m(k)) \end{pmatrix} \cdot \mathbf{p}_z(k) \\ &= \mathbf{L}(:, w_1(k), \dots, w_q(k), :, v_1(k), \dots, v_m(k)) \cdot \mathbf{p}_z(k) \\ &= \mathbf{L}(:, \mathbf{w}(k), :, \mathbf{v}(k)) \cdot \mathbf{p}_z(k) \in [0, 1]^N. \end{aligned}$$

□

B.7 Beweis der Proposition 10.2

Gegeben sei der Verhaltenstensor $\mathbf{L} \in [0, 1]^{N \times Q_1 \times Q_2 \times \dots \times Q_q \times N \times M_1 \times M_2 \times \dots \times M_m}$ mit den Elementen $l(i, j_1, j_2, \dots, j_q, s, t_1, t_2, \dots, t_m)$, wobei die Folgezustände und Zustände Werte in $i \in \{1, 2, \dots, N\}$ und $s \in \{1, 2, \dots, N\}$ annehmen können, während für die Werte der Ausgaben $j_e \in \{1, 2, \dots, Q_e\}$,

für $e = 1, 2, \dots, q$ und für die Werte der Eingaben $t_h \in \{1, 2, \dots, M_h\}$ für $h = 1, \dots, m$ gilt. Außerdem sei $\mathbf{p}_z(k) \in [0, 1]^N$ der Zustandswahrscheinlichkeitsvektor mit den Komponenten $p_z(k, s)$ während die Eingabewahrscheinlichkeitsvektoren $\mathbf{p}_v^{(h)}(k) \in [0, 1]^{M_h}$ mit den Komponenten $p_v^{(h)}(k, t_h)$ für $h = 1, \dots, m$ und die Ausgabewahrscheinlichkeitsvektoren $\mathbf{p}_w^{(e)}(k) \in [0, 1]^{Q_e}$ mit Komponenten $p_w^{(e)}(k, j_e)$ für $e = 1, 2, \dots, q$ gegeben sind. Für eine exakte CP-Dekomposition gilt nun

$$\begin{aligned} \mathbf{L} &= \llbracket \boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_q, \mathbf{C}, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_m \rrbracket \\ &= \sum_{r=1}^R \lambda_r \mathbf{a}_{:r} \circ \mathbf{b}_{:r}^{(1)} \circ \dots \circ \mathbf{b}_{:r}^{(q)} \circ \mathbf{c}_{:r} \circ \mathbf{d}_{:r}^{(1)} \circ \dots \circ \mathbf{d}_{:r}^{(m)}, \end{aligned}$$

mit der elementweisen Notation

$$l(i, j_1, \dots, j_q, s, t_1, \dots, t_m) = \sum_{r=1}^R \lambda_r a_{ir} b_{j_1 r}^{(1)} \dots b_{j_q r}^{(q)} c_{sr} d_{t_1 r}^{(1)} \dots d_{t_m r}^{(m)}.$$

Dabei sind $\mathbf{A} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{B}_e \in \mathbb{R}_{\geq 0}^{Q_e \times R}$, $\mathbf{C} \in \mathbb{R}_{\geq 0}^{N \times R}$, $\mathbf{D}_h \in \mathbb{R}_{\geq 0}^{M_h \times R}$ die Faktormatrizen und $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^R$ der Gewichtungsvektor.

Beweis: Für den Ausdruck

$$\mathbf{A} \cdot \left(\boldsymbol{\lambda} \otimes \left(\bigotimes_{e=1}^q \mathbf{B}_e^\top \cdot \mathbf{p}_w^{(e)}(k) \right) \otimes \left(\mathbf{C}^\top \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{h=1}^m \mathbf{D}_h^\top \cdot \mathbf{p}_v^{(h)}(k) \right) \right),$$

kann man Schrittweise zeigen:

$$\begin{aligned} \mathbf{C}^\top \cdot \mathbf{p}_z(k) &= \begin{pmatrix} c_{11} & c_{21} & \dots & c_{N1} \\ c_{12} & c_{12} & \dots & c_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1R} & c_{2R} & \dots & c_{NR} \end{pmatrix} \cdot \begin{pmatrix} p_z(k, 1) \\ p_z(k, 2) \\ \vdots \\ p_z(k, N) \end{pmatrix} = \begin{pmatrix} \sum_{s=1}^N c_{s1} p_z(k, s) \\ \sum_{s=1}^N c_{s2} p_z(k, s) \\ \vdots \\ \sum_{s=1}^N c_{sR} p_z(k, s) \end{pmatrix}, \\ \bigotimes_{e=1}^q \mathbf{B}_e^\top \cdot \mathbf{p}_w^{(e)}(k) &= \bigotimes_{e=1}^q \begin{pmatrix} b_{11}^{(e)} & b_{21}^{(e)} & \dots & b_{Q_e 1}^{(e)} \\ b_{12}^{(e)} & b_{12}^{(e)} & \dots & b_{Q_e 2}^{(e)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1R}^{(e)} & b_{2R}^{(e)} & \dots & b_{Q_e R}^{(e)} \end{pmatrix} \cdot \begin{pmatrix} p_w^{(e)}(k, 1) \\ p_w^{(e)}(k, 2) \\ \vdots \\ p_w^{(e)}(k, Q_e) \end{pmatrix} \\ &= \bigotimes_{e=1}^q \begin{pmatrix} \sum_{j_e=1}^{Q_e} b_{j_e 1}^{(e)} p_w^{(e)}(k, j_e) \\ \sum_{j_e=1}^{Q_e} b_{j_e 2}^{(e)} p_w^{(e)}(k, j_e) \\ \vdots \\ \sum_{j_e=1}^{Q_e} b_{j_e R}^{(e)} p_w^{(e)}(k, j_e) \end{pmatrix} \end{aligned} \quad (\text{B.12})$$

$$\begin{aligned} \bigotimes_{h=1}^m \mathbf{D}_h^\top \cdot \mathbf{p}_v^{(h)}(k) &= \begin{pmatrix} \sum_{j_1=1}^{Q_1} \dots \sum_{j_q=1}^{Q_q} b_{j_1 1}^{(1)} \dots b_{j_q 1}^{(q)} p_w^{(1)}(k, j_1) \dots p_w^{(q)}(k, j_q) \\ \sum_{j_1=1}^{Q_1} \dots \sum_{j_q=1}^{Q_q} b_{j_1 2}^{(1)} \dots b_{j_q 2}^{(q)} p_w^{(1)}(k, j_1) \dots p_w^{(q)}(k, j_q) \\ \vdots \\ \sum_{j_1=1}^{Q_1} \dots \sum_{j_q=1}^{Q_q} b_{j_1 R}^{(1)} \dots b_{j_q R}^{(q)} p_w^{(1)}(k, j_1) \dots p_w^{(q)}(k, j_q) \end{pmatrix}, \\ &= \begin{pmatrix} \sum_{t_1=1}^{M_1} \dots \sum_{t_m=1}^{M_m} d_{t_1 1}^{(1)} \dots d_{t_m 1}^{(m)} p_v^{(1)}(k, t_1) \dots p_v^{(m)}(k, t_m) \\ \sum_{t_1=1}^{M_1} \dots \sum_{t_m=1}^{M_m} d_{t_1 2}^{(1)} \dots d_{t_m 2}^{(m)} p_v^{(1)}(k, t_1) \dots p_v^{(m)}(k, t_m) \\ \vdots \\ \sum_{t_1=1}^{M_1} \dots \sum_{t_m=1}^{M_m} d_{t_1 R}^{(1)} \dots d_{t_m R}^{(m)} p_v^{(1)}(k, t_1) \dots p_v^{(m)}(k, t_m) \end{pmatrix}. \end{aligned} \quad (\text{B.13})$$

Somit erhält man

$$\begin{aligned}
& \mathbf{A} \cdot \left(\lambda \otimes \left(\bigotimes_{e=1}^q \mathbf{B}_e^T \cdot \mathbf{p}_w^{(e)}(k) \right) \otimes \left(\mathbf{C}^T \cdot \mathbf{p}_z(k) \right) \otimes \left(\bigotimes_{h=1}^m \mathbf{D}_h^T \cdot \mathbf{p}_v^{(h)}(k) \right) \right) \\
&= \mathbf{A} \cdot \left(\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_R \end{pmatrix} \otimes \begin{pmatrix} \sum_{j_1=1}^{Q_1} \cdots \sum_{j_q=1}^{Q_q} b_{j_1 1}^{(1)} \cdots b_{j_q 1}^{(q)} p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) \\ \sum_{j_1=1}^{Q_1} \cdots \sum_{j_q=1}^{Q_q} b_{j_1 2}^{(1)} \cdots b_{j_q 2}^{(q)} p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) \\ \vdots \\ \sum_{j_1=1}^{Q_1} \cdots \sum_{j_q=1}^{Q_q} b_{j_1 R}^{(1)} \cdots b_{j_q R}^{(q)} p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) \end{pmatrix} \right. \\
&\quad \otimes \begin{pmatrix} \sum_{s=1}^N c_{s1} p_z(k, s) \\ \sum_{s=1}^N c_{s2} p_z(k, s) \\ \vdots \\ \sum_{s=1}^N c_{sR} p_z(k, s) \end{pmatrix} \\
&\quad \otimes \begin{pmatrix} \sum_{t_1=1}^{M_1} \cdots \sum_{t_m=1}^{M_m} d_{t_1 1}^{(1)} \cdots d_{t_m 1}^{(m)} p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m) \\ \sum_{t_1=1}^{M_1} \cdots \sum_{t_m=1}^{M_m} d_{t_1 2}^{(1)} \cdots d_{t_m 2}^{(m)} p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m) \\ \vdots \\ \sum_{t_1=1}^{M_1} \cdots \sum_{t_m=1}^{M_m} d_{t_1 R}^{(1)} \cdots d_{t_m R}^{(m)} p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m) \end{pmatrix} \Bigg) \\
&= \mathbf{A} \cdot \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \lambda_1 b_{j_1 1}^{(1)} \cdots b_{j_q 1}^{(q)} c_{s1} d_{t_1 1}^{(1)} \cdots d_{t_m 1}^{(m)} \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \lambda_2 b_{j_1 2}^{(1)} \cdots b_{j_q 2}^{(q)} c_{s2} d_{t_1 2}^{(1)} \cdots d_{t_m 2}^{(m)} \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \lambda_R b_{j_1 R}^{(1)} \cdots b_{j_q R}^{(q)} c_{sR} d_{t_1 R}^{(1)} \cdots d_{t_m R}^{(m)} \bar{P} \end{pmatrix} \\
&= \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R \lambda_r a_{1r} \cdots b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{sr} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R \lambda_r a_{2r} \cdots b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{sr} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R \lambda_r a_{Nr} \cdots b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{sr} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \end{pmatrix} \\
&\quad \underbrace{\hspace{15em}}_{= l(N, j_1, \dots, j_q, s, t_1, \dots, t_m)} \\
&= \begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(1, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(2, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(N, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \end{pmatrix} \in [0, 1]^N, \quad (\text{B.14})
\end{aligned}$$

wobei $\bar{P} := p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) p_z(k, s) p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m)$ gilt. Der Vektor (B.14) entspricht dabei der elementweisen Notation des kontrahierten Tensorprodukts (B.8) im Beweis B.6. Aus der Gleichheit zwischen (B.8) und (B.10) im Beweis B.6 folgt somit unmittelbar

$$\begin{pmatrix} \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(1, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(2, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \\ \vdots \\ \sum_{j_1, \dots, j_q=1}^{Q_1, \dots, Q_q} \sum_{s=1}^N \sum_{t_1, \dots, t_m=1}^{M_1, \dots, M_m} \sum_{r=1}^R l(N, j_1, \dots, j_q, s, t_1, \dots, t_m) \bar{P} \end{pmatrix} \in [0, 1]^N$$

$$= \left\langle \mathbf{L} \mid \llbracket \mathbf{p}_w^{(1)}(k), \dots, \mathbf{p}_w^{(q)}(k), \mathbf{p}_z(k), \mathbf{p}_v^{(1)}(k), \dots, \mathbf{p}_v^{(m)}(k) \rrbracket \right\rangle \in [0, 1]^N.$$

□

B.8 Beweis der Proposition 10.3

Gegeben sei der Verhaltenstensor $\mathbf{L} \in [0, 1]^{N_1 \times N_2 \times \cdots \times N_n \times N_1 \times N_2 \times \cdots \times N_n}$ mit den Elementen $l(i_1, i_2, \dots, i_n, s_1, s_2, \dots, s_n)$ mit $i_g \in \{1, \dots, N_g\}$ für $g = 1, \dots, n$ und $s_u \in \{1, \dots, N_u\}$ für $u = 1, \dots, n$. Außerdem sei

$$\mathbf{L} = \llbracket \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n, \mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n \rrbracket$$

die exakte CP-Dekomposition des Verhaltenstensors mit den Faktormatrizen $\mathbf{A}_g \in \mathbb{R}_{\geq 0}^{N_g \times R}$ und $\mathbf{C}_u \in \mathbb{R}_{\geq 0}^{N_u \times R}$, sodass

$$l(i_1, i_2, \dots, i_n, s_1, s_2, \dots, s_n) = \sum_{r=1}^R \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_n r}^{(n)} c_{s_1 r}^{(1)} c_{s_2 r}^{(2)} \cdots c_{s_n r}^{(n)}$$

gilt. Außerdem seien $\mathbf{p}_z^{(u)}(k) \in [0, 1]^{N_u}$ die Zustandswahrscheinlichkeitsvektoren zur Zeit k mit den Komponenten $p_z^{(u)}(k, s_u)$ für $u = 1, \dots, n$. Gezeigt werden soll, dass

$$\mathbf{p}_z^{(g)}(k+1) = \mathbf{A}_g \cdot \left(\lambda \otimes \left(\bigotimes_{\alpha \neq g} \sum_{x_\alpha=1}^{N_\alpha} (\mathbf{a}_{x_\alpha}^{(\alpha)})^\top \right) \otimes \left(\bigotimes_{u=1}^n \mathbf{C}_u^\top \cdot \mathbf{p}_z^{(u)}(k) \right) \right)$$

zum gleichen Ergebnis führt wie Gleichung (10.56).

Beweis: Es ist

$$\begin{aligned} & \lambda \otimes \left(\bigotimes_{\alpha \neq g} \sum_{x_\alpha=1}^{N_\alpha} (\mathbf{a}_{x_\alpha}^{(\alpha)})^\top \right) \\ &= \lambda \otimes \left(\bigotimes_{\alpha \neq g} \begin{pmatrix} \sum_{x_\alpha=1}^{N_\alpha} a_{x_\alpha 1}^{(\alpha)} \\ \sum_{x_\alpha=1}^{N_\alpha} a_{x_\alpha 2}^{(\alpha)} \\ \vdots \\ \sum_{x_\alpha=1}^{N_\alpha} a_{x_\alpha R}^{(\alpha)} \end{pmatrix} \right) \\ &= \begin{pmatrix} \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \lambda_1 a_{x_1 1}^{(1)} \cdots a_{x_{g-1} 1}^{(g-1)} a_{x_{g+1} 1}^{(g+1)} \cdots a_{x_n 1}^{(n)} \\ \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=2}^{N_{g+1}} \cdots \sum_{x_n=2}^{N_n} \lambda_2 a_{x_1 2}^{(1)} \cdots a_{x_{g-1} 2}^{(g-1)} a_{x_{g+1} 2}^{(g+1)} \cdots a_{x_n 2}^{(n)} \\ \vdots \\ \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=R}^{N_{g+1}} \cdots \sum_{x_n=R}^{N_n} \lambda_R a_{x_1 R}^{(1)} \cdots a_{x_{g-1} R}^{(g-1)} a_{x_{g+1} R}^{(g+1)} \cdots a_{x_n R}^{(n)} \end{pmatrix} \end{aligned} \quad (\text{B.15})$$

und

$$\bigotimes_{u=1}^n \mathbf{C}_u^\top \cdot \mathbf{p}_z^{(u)} = \begin{pmatrix} \sum_{s_1=1}^{N_1} \cdots \sum_{s_n=1}^{N_n} c_{s_1 1}^{(1)} \cdots c_{s_n 1}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \sum_{s_1=1}^{N_1} \cdots \sum_{s_n=1}^{N_n} c_{s_1 2}^{(1)} \cdots c_{s_n 2}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \sum_{s_1=1}^{N_1} \cdots \sum_{s_n=1}^{N_n} c_{s_1 R}^{(1)} \cdots c_{s_n R}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix}. \quad (\text{B.16})$$

Im Folgenden gilt nun die Abkürzung:

$$\Sigma := \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_n=1}^{N_n}.$$

Dann ist

$$\begin{aligned} & \mathbf{A}_g \cdot \left(\lambda \otimes \left(\bigotimes_{\alpha \neq g} \sum_{x_\alpha=1}^{N_\alpha} (\mathbf{a}_{x_\alpha}^{(\alpha)})^\top \right) \otimes \left(\bigotimes_{u=1}^n \mathbf{C}_u^\top \cdot \mathbf{p}_z^{(u)}(k) \right) \right) \\ & \mathbf{A}_g \cdot \begin{pmatrix} \Sigma \lambda_1 a_{x_1 1}^{(1)} \cdots a_{x_{g-1} 1}^{(g-1)} a_{x_{g+1} 1}^{(g+1)} \cdots a_{x_n 1}^{(n)} c_{s_1 1}^{(1)} \cdots c_{s_n 1}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma \lambda_2 a_{x_1 2}^{(1)} \cdots a_{x_{g-1} 2}^{(g-1)} a_{x_{g+1} 2}^{(g+1)} \cdots a_{x_n 2}^{(n)} c_{s_1 2}^{(1)} \cdots c_{s_n 2}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma \lambda_R a_{x_1 R}^{(1)} \cdots a_{x_{g-1} R}^{(g-1)} a_{x_{g+1} R}^{(g+1)} \cdots a_{x_n R}^{(n)} c_{s_1 R}^{(1)} \cdots c_{s_n R}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\ & = \begin{pmatrix} a_{11}^{(g)} & a_{12}^{(g)} & \cdots & a_{1R}^{(g)} \\ a_{21}^{(g)} & a_{22}^{(g)} & \cdots & a_{2R}^{(g)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_g 1}^{(g)} & a_{N_g 2}^{(g)} & \cdots & a_{N_g R}^{(g)} \end{pmatrix} \cdot \begin{pmatrix} \Sigma \lambda_1 a_{x_1 1}^{(1)} \cdots a_{x_{g-1} 1}^{(g-1)} a_{x_{g+1} 1}^{(g+1)} \cdots a_{x_n 1}^{(n)} c_{s_1 1}^{(1)} \cdots c_{s_n 1}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma \lambda_2 a_{x_1 2}^{(1)} \cdots a_{x_{g-1} 2}^{(g-1)} a_{x_{g+1} 2}^{(g+1)} \cdots a_{x_n 2}^{(n)} c_{s_1 2}^{(1)} \cdots c_{s_n 2}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma \lambda_R a_{x_1 R}^{(1)} \cdots a_{x_{g-1} R}^{(g-1)} a_{x_{g+1} R}^{(g+1)} \cdots a_{x_n R}^{(n)} c_{s_1 R}^{(1)} \cdots c_{s_n R}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\ & = \begin{pmatrix} \Sigma \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{1r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{2r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{N_g r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\ & \quad \underbrace{\hspace{15em}}_{f(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n)} \\ & = \begin{pmatrix} \Sigma f(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma f(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma f(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix}. \end{aligned}$$

Dabei ist $f(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n)$ der $(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n)$ -te Eintrag des Übergangstensors \mathbf{F} .

Damit folgt

$$\begin{aligned}
& \begin{pmatrix} \Sigma f(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma f(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma f(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\
&= \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_{g-1}=1}^{N_{g-1}} \sum_{s_{g+1}=1}^{N_{g+1}} \cdots \sum_{s_n=1}^{N_n} p_z^{(1)}(k, s_1) \cdots \\
&\quad p_z^{(g-1)}(k, s_{g-1}) \cdot p_z^{(g+1)}(k, s_{g+1}) \cdots p_z^{(n)}(k, s_n) \quad (\text{B.17}) \\
& \cdot \begin{pmatrix} f(x_1 \dots x_{g-1} 1 x_{g+1} \dots x_n s_1 s_{g-1} 1 s_{g+1} \dots s_n) \cdots f(x_1 \dots x_{g-1} 1 x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n) \\ f(x_1 \dots x_{g-1} 2 x_{g+1} \dots x_n s_1 s_{g-1} 1 s_{g+1} \dots s_n) \cdots f(x_1 \dots x_{g-1} 2 x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n) \\ \vdots \quad \ddots \quad \vdots \\ f(x_1 \dots x_{g-1} N_g x_{g+1} \dots x_n s_1 s_{g-1} 1 s_{g+1} \dots s_n) \cdots f(x_1 \dots x_{g-1} N_g x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n) \end{pmatrix} \\
& \cdot \begin{pmatrix} p_z^{(g)}(k, 1) \\ p_z^{(g)}(k, 2) \\ \vdots \\ p_z^{(g)}(k, N_g) \end{pmatrix}.
\end{aligned}$$

Durch Gleichung (10.66) gilt für den yz -ten Eintrag, der in (B.17) auftretenden Matrix, die Gleichheit

$$\begin{aligned}
& f(x_1, \dots, x_{g-1}, y, x_{g+1}, \dots, x_n, s_1, s_{g-1}, z, s_{g+1}, \dots, s_n) \\
&= f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{yz}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)}.
\end{aligned}$$

Daraus folgt dann

$$\begin{aligned}
& \begin{pmatrix} \Sigma f(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \Sigma f(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \Sigma f(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, s_1, \dots, s_n) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\
&= \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_{g-1}=1}^{N_{g-1}} \sum_{s_{g+1}=1}^{N_{g+1}} \cdots \sum_{s_n=1}^{N_n} p_z^{(1)}(k, s_1) \cdots \\
&\quad p_z^{(g-1)}(k, s_{g-1}) \cdot p_z^{(g+1)}(k, s_{g+1}) \cdots p_z^{(n)}(k, s_n) \\
& \cdot \begin{pmatrix} f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{11}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \cdots f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{1N_g}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \\ f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{21}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \cdots f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{2N_g}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \\ \vdots \quad \ddots \quad \vdots \\ f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{N_g 1}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \cdots f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{N_g N_g}^{(g)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)} \end{pmatrix} \\
& \cdot \begin{pmatrix} p_z^{(g)}(k, 1) \\ p_z^{(g)}(k, 2) \\ \vdots \\ p_z^{(g)}(k, N_g) \end{pmatrix} = (\star). \quad (\text{B.18})
\end{aligned}$$

In jedem Eintrag der Matrix steht außerdem die Summe von Produkten

$$\sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_{g-1}=1}^{N_{g-1}} \sum_{s_{g+1}=1}^{N_{g+1}} \cdots \sum_{s_n=1}^{N_n} p_z^{(1)}(k, s_1) \cdots p_z^{(g-1)}(k, s_{g-1}) p_z^{(g+1)}(k, s_{g+1}) \cdots p_z^{(n)}(k, s_n) f_{x_1 s_1}^{(1)} \cdots f_{x_{g-1} s_{g-1}}^{(g-1)} f_{x_{g+1} s_{g+1}}^{(g+1)} \cdots f_{x_n s_n}^{(n)}.$$

Diese Summe kann für $h \in \{1, \dots, g-1, g+1, \dots, n\}$ kürzer notiert werden, dass heißt es ist

$$\sum_{x_h=1}^{N_h} \sum_{s_h=1}^{N_h} f_{x_h s_h}^{(h)} p_z^{(h)}(k, s_h) \quad \forall h.$$

Für diese Summe gilt nun

$$\begin{aligned} \sum_{x_h=1}^{N_h} \sum_{s_h=1}^{N_h} f_{x_h s_h}^{(h)} p_z^{(h)}(k, s_h) &= \sum_{x_h=1}^{N_h} \sum_{s_h=1}^{N_h} \Pr(Z_h(k+1) = x_h \mid Z_h(k) = s_h) \Pr(Z_h(k) = s_h) \\ &= \sum_{x_h=1}^{N_h} \sum_{s_h=1}^{N_h} \frac{\Pr(Z_h(k+1) = x_h, Z_h(k) = s_h)}{\Pr(Z_h(k) = s_h)} \Pr(Z_h(k) = s_h) \\ &= \sum_{x_h=1}^{N_h} \sum_{s_h=1}^{N_h} \Pr(Z_h(k+1) = x_h, Z_h(k) = s_h) \\ &= 1, \end{aligned}$$

da die Wahrscheinlichkeiten *aller* Folgezustände x_h und Zustände s_h , für alle Werte von $h \in \{1, \dots, g-1, g+1, \dots, n\}$ aufsummiert werden. Damit erhält man in (B.18):

$$(\star) = \begin{pmatrix} f_{11}^{(g)} & f_{12}^{(g)} & \cdots & f_{1N_g}^{(g)} \\ f_{21}^{(g)} & f_{22}^{(g)} & \cdots & f_{2N_g}^{(g)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N_g 1}^{(g)} & f_{N_g 2}^{(g)} & \cdots & f_{N_g N_g}^{(g)} \end{pmatrix} \cdot \begin{pmatrix} p_z^{(g)}(k, 1) \\ p_z^{(g)}(k, 2) \\ \vdots \\ p_z^{(g)}(k, N_g) \end{pmatrix} = \mathbf{F}_g \cdot \mathbf{p}_z^{(g)}(k) = \mathbf{p}_z^{(g)}(k+1) \in [0, 1]^{N_g}.$$

□

B.9 Beweis der Proposition 10.4

Es sei $\mathbf{L} \in [0, 1]^{N_1 \times N_2 \times \cdots \times N_n \times Q_1 \times Q_2 \times \cdots \times Q_q \times N_1 \times N_2 \times \cdots \times N_n \times M_1 \times M_2 \times \cdots \times M_m}$ der Verhaltenstensor mit Elementen $l(i_1, i_2, \dots, i_n, j_1, j_2, \dots, j_q, s_1, s_2, \dots, s_n, \dots, t_1, t_2, \dots, t_m)$ wobei die Folgezustände die Werte $i_g \in \{1, \dots, N_g\}$ für $g = 1, \dots, n$ und die Zustände Werte in $s_u \in \{1, \dots, N_u\}$ für $u = 1, \dots, n$ annehmen können. Für die Werte der Ausgaben gilt $j_e \in \{1, \dots, Q_e\}$ für $e = 1, 2, \dots, q$ und für die Werte der Eingaben $t_h \in \{1, \dots, M_h\}$ für $h = 1, 2, \dots, m$. Außerdem seien $\mathbf{p}_z^{(u)}(k) \in [0, 1]^{N_u}$ die Zustandswahrscheinlichkeitsvektoren zur Zeit k mit Komponenten $p_z^{(u)}(k, s_u)$, $\mathbf{p}_v^{(h)}(k) \in \{0, 1\}^{M_h}$ die Eingabewahrscheinlichkeitsvektoren mit Komponenten $p_v^{(h)}(k, t_h)$ und $\mathbf{p}_w^{(e)}(k) \in \{0, 1\}^{Q_e}$ die Ausgabewahrscheinlichkeitsvektoren mit Komponenten $p_w^{(e)}(k, j_e)$.

Beweis: Es ist

$$\begin{aligned} &\mathbf{A}_g \cdot \left(\lambda \otimes \left(\bigotimes_{\alpha \neq g} \sum_{x_\alpha=1}^{N_\alpha} (\mathbf{a}_{x_\alpha}^{(\alpha)})^\top \right) \otimes \left(\bigotimes_{e=1}^q \mathbf{B}_e^\top \cdot \mathbf{p}_w^{(e)}(k) \right) \otimes \left(\bigotimes_{u=1}^n \mathbf{C}_u^\top \cdot \mathbf{p}_z^{(u)}(k) \right) \otimes \left(\bigotimes_{h=1}^m \mathbf{D}_h^\top \cdot \mathbf{p}_v^{(h)}(k) \right) \right) \\ &= \begin{pmatrix} \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{x_{g+1} r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \\ \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{x_{g+1} r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \\ \vdots \\ \sum_{r=1}^R \lambda_r a_{x_1 r}^{(1)} \cdots a_{x_{g-1} r}^{(g-1)} a_{x_{g+1} r}^{(g)} a_{x_{g+1} r}^{(g+1)} \cdots a_{x_n r}^{(n)} b_{j_1 r}^{(1)} \cdots b_{j_q r}^{(q)} c_{s_1 r}^{(1)} \cdots c_{s_n r}^{(n)} d_{t_1 r}^{(1)} \cdots d_{t_m r}^{(m)} \bar{P} \end{pmatrix} \\ &\quad \underbrace{\hspace{15em}}_{l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1, \dots, t_m)} \end{aligned}$$

$$= \begin{pmatrix} \Sigma l(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \\ \Sigma l(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \\ \vdots \\ \Sigma l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \end{pmatrix} \quad (\text{B.19})$$

Dabei ist $l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m)$ der $(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m)$ -te Eintrag des Verhaltenstensor \mathbf{L} . In obiger Gleichung gelten außerdem die Abkürzungen

$$\Sigma := \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{j_1=1}^{Q_1} \cdots \sum_{j_q=1}^{Q_q} \sum_{s_1=1}^{N_1} \cdots \sum_{s_n=1}^{N_n} \sum_{t_1=1}^{M_1} \cdots \sum_{t_m=1}^{M_m},$$

$$\bar{P} := p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m).$$

Sind nun die Verteilungen der Aus- und Eingabevektoren so, dass diese für $e = 1, \dots, q$ und $h = 1, \dots, m$, nur an der $w_e(k)$ -ten und $v_h(k)$ -ten Position den Wert eins annehmen, also

$$\mathbf{p}_w^{(e)}(k) = \begin{pmatrix} p_w^{(e)}(k, 1) \\ \vdots \\ p_w^{(e)}(k, w_e(k)) \\ \vdots \\ p_w^{(e)}(k, Q_e) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^{Q_e}, \quad \mathbf{p}_v^{(h)}(k) = \begin{pmatrix} p_v^{(h)}(k, 1) \\ \vdots \\ p_v^{(h)}(k, v_h(k)) \\ \vdots \\ p_v^{(h)}(k, M_h) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \{0, 1\}^{M_h},$$

dann gilt für das in (B.19) auftretende Produkt

$$p_w^{(1)}(k, j_1) \cdots p_w^{(q)}(k, j_q) p_v^{(1)}(k, t_1) \cdots p_v^{(m)}(k, t_m) = \begin{cases} 1 & \text{für } j_1 = w_1(k), \dots, j_q = w_q(k) \text{ und} \\ & t_1 = v_1(k), \dots, t_m = v_m(k), \\ 0 & \text{sonst.} \end{cases} \quad (\text{B.20})$$

Damit entfällt in (B.19) das Produkt (B.20) sowie die Summation über die Indizes j_1, \dots, j_q und t_1, \dots, t_m und die Tensorelemente können direkt mit $w_1(k), \dots, w_q(k)$ und $v_1(k), \dots, v_m(k)$ indiziert werden. Damit erhält man

$$\begin{pmatrix} \Sigma l(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \\ \Sigma l(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \\ \vdots \\ \Sigma l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, j_1, \dots, j_q, s_1, \dots, s_n, t_1 \dots, t_m) \bar{P} \end{pmatrix} \\ = \begin{pmatrix} \sum l(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k) \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \sum l(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k) \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \sum l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k) \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix}$$

wobei

$$\Sigma := \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_{g-1}=1}^{N_{g-1}} \sum_{s_{g+1}=1}^{N_{g+1}} \cdots \sum_{s_n=1}^{N_n} \quad (\text{B.21})$$

gilt. Damit folgt

$$\begin{aligned}
 & \begin{pmatrix} \sum l(x_1, \dots, x_{g-1}, 1, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k), \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \sum l(x_1, \dots, x_{g-1}, 2, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k), \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \\ \vdots \\ \sum l(x_1, \dots, x_{g-1}, N_g, x_{g+1}, \dots, x_n, w_1(k), \dots, w_q(k), s_1, \dots, s_n, v_1(k), \dots, v_m(k)) p_z^{(1)}(k, s_1) \cdots p_z^{(n)}(k, s_n) \end{pmatrix} \\
 &= \sum_{x_1=1}^{N_1} \cdots \sum_{x_{g-1}=1}^{N_{g-1}} \sum_{x_{g+1}=1}^{N_{g+1}} \cdots \sum_{x_n=1}^{N_n} \sum_{s_1=1}^{N_1} \cdots \sum_{s_{g-1}=1}^{N_{g-1}} \sum_{s_{g+1}=1}^{N_{g+1}} \cdots \sum_{s_n=1}^{N_n} p_z^{(1)}(k, s_1) \cdots \\
 & \quad p_z^{(g-1)}(k, s_{g-1}) \cdot p_z^{(g+1)}(k, s_{g+1}) \cdots p_z^{(n)}(k, s_n) \tag{B.22}
 \end{aligned}$$

$$\begin{aligned}
 & \cdot \begin{pmatrix} l_{x_1 \dots x_{g-1} 1 x_{g+1} \dots x_n w_1(k) \dots w_q(k) s_1 s_{g-1} 1 s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \cdots l_{x_1 \dots x_{g-1} 1 x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \\ l_{x_1 \dots x_{g-1} 2 x_{g+1} \dots x_n w_1(k) \dots w_q(k) s_1 s_{g-1} 1 s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \cdots l_{x_1 \dots x_{g-1} 2 x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \\ \vdots \quad \quad \quad \ddots \quad \quad \quad \vdots \\ l_{x_1 \dots x_{g-1} N_g x_{g+1} \dots x_n w_1(k) \dots w_q(k) s_1 s_{g-1} 1 s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \cdots l_{x_1 \dots x_{g-1} N_g x_{g+1} \dots x_n s_1 s_{g-1} N_g s_{g+1} \dots s_n v_1(k) \dots v_m(k)} \end{pmatrix} \\
 & \cdot \begin{pmatrix} p_z^{(g)}(k, 1) \\ p_z^{(g)}(k, 2) \\ \vdots \\ p_z^{(g)}(k, N_g) \end{pmatrix} \\
 &= \begin{pmatrix} l_g(1, w_g(k), 1, v_g(k)) & l_g(1, w_g(k), 2, v_g(k)) & \cdots & l_g(1, w_g(k), N_g, v_g(k)) \\ l_g(2, w_g(k), 1, v_g(k)) & l_g(2, w_g(k), 2, v_g(k)) & \cdots & l_g(2, w_g(k), N_g, v_g(k)) \\ \vdots & \vdots & \ddots & \vdots \\ l_g(N_g, w_g(k), 1, v_g(k)) & l_g(N_g, w_g(k), 2, v_g(k)) & \cdots & l_g(N_g, w_g(k), N_g, v_g(k)) \end{pmatrix} \cdot \mathbf{p}_z^{(g)}(k) \tag{B.23}
 \end{aligned}$$

$$= \mathbf{L}_g(:, w_g(k), :, v_g(k)) \cdot \mathbf{p}_z^{(g)}(k) \in [0, 1]^{N_g} . \tag{B.24}$$

Beweisschritt (B.19) erhält man aus der elementweisen Multiplikation der bereits in (B.12) und (B.13) im Beweis der Proposition B.7 berechneten Terme, sowie den Termen (B.15) und (B.16) im Beweis der Proposition B.8. Der Schritt von (B.22) nach (B.23) folgt aufgrund der stochastische Unabhängigkeit sowie der Beziehung (10.78) und der daraus resultierenden Gleichheit

$$\begin{aligned}
 & l(x_1, x_2, \dots, x_n, w_1(k), w_2(k), \dots, w_q(k), s_1, s_2, \dots, s_n, v_1(k), v_2(k), \dots, v_m(k)) \\
 &= l_1(x_1, w_1(k), s_1, v_1(k)) \cdot l_2(x_2, w_2(k), s_2, v_2(k)) \cdots l_n(x_n, w_q(k), s_n, v_m(k)) .
 \end{aligned}$$

Die in (B.22) auftretende Summe von Produkten entfällt, weil diese zu einem Wert von 1 führt, da über alle möglichen Zustände und Folgezustände des in Abhängigkeit der Ein- und Ausgaben selektierten Subtensors summiert wird.

□

Diese Arbeit liefert einen Beitrag zur technischen Fehlererkennung auf Grundlage der Zustandsbeobachtung technischer Systeme durch qualitative Modelle. Als qualitatives Modell wird ein stochastischer Automat verwendet, mit dem das qualitative Verhalten eines Prozesses anhand diskreter Wahrscheinlichkeitsverteilungen beschrieben wird. Das Verhalten des Prozesses wird dabei nur näherungsweise abgebildet, wobei jedoch die grundlegende Systemdynamik erhalten bleibt. Stochastische Automaten und qualitative Modelle leiden dabei unter dem Problem, dass ihr Speicherbedarf mit einer steigenden Anzahl der abzubildenden Systemgrößen exponentiell ansteigt. Zur Lösung dieses Komplexitätsproblems wird ein neuer Ansatz verfolgt, bei dem die Übergangswahrscheinlichkeiten des stochastischen Automaten in Tensoren gespeichert werden. Dabei wird eine moderne Methode zur Modellreduktion vorgestellt, die auf der Dekomposition der multilinearen Tensorstruktur basiert. Die vorgestellten Algorithmen, welche auf der speicherreduzierten dekomponierten Tensorrepräsentation definiert sind, ermöglichen die effiziente Realisierung der qualitativen Fehlererkennung.