

**Oliver C. Schrempf**

Stochastische Behandlung  
von Unsicherheiten in  
kaskadierten  
dynamischen Systemen





Oliver C. Schrempf

**Stochastische Behandlung von Unsicherheiten in kaskadierten dynamischen Systemen**

**Karlsruhe Series on Intelligent Sensor-Actuator-Systems**  
**Volume 1**

ISAS | Universität Karlsruhe (TH)  
Intelligent Sensor-Actuator-Systems Laboratory

*Edited by Prof. Dr.-Ing. Uwe D. Hanebeck*

# **Stochastische Behandlung von Unsicherheiten in kaskadierten dynamischen Systemen**

von  
Oliver C. Schrempf



---

universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)  
Fakultät für Informatik, 2008

## Impressum

Universitätsverlag Karlsruhe  
c/o Universitätsbibliothek  
Straße am Forum 2  
D-76131 Karlsruhe  
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz  
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2008  
Print on Demand

ISSN: 1867-3813  
ISBN: 978-3-86644-287-0







# **Stochastische Behandlung von Unsicherheiten in kaskadierten dynamischen Systemen**

zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik  
der Universität Fridericiana zu Karlsruhe (TH)

**genehmigte**

**Dissertation**

von

**Oliver C. Schrempf**

aus Gelnhausen

Tag der mündlichen Prüfung: 20. Juni 2008

Erster Gutachter: Prof. Dr.-Ing. Uwe D. Hanebeck

Zweiter Gutachter: Prof. Dr.-Ing. Jürgen Beyerer



# Danksagung

Diese Arbeit entstand in den Jahren 2004 bis 2007 am Lehrstuhl für Intelligente Sensor-Aktor-Systeme (ISAS) an der Universität Karlsruhe (TH). Dies wäre nicht möglich gewesen ohne ein besonderes Umfeld und eine Menge von Personen die mir auf meinem Weg zur Seite standen.

Ich möchte Herrn Prof. Dr.-Ing. U. D. Hanebeck für die ausgezeichneten Arbeitsbedingungen und konstruktive Betreuung danken. Herrn Prof. Dr.-Ing. Beyerer gilt mein herzlicher Dank für die Übernahme des Korreferats.

Besonderer Dank gilt meinen Kollegen am ISAS Patrick Rößler, Dietrich Brunn, Florian Weißel, Frederik Beutler, Marco Huber, Vesa Klumpp, Kathrin Roberts, Felix Sawo, Olga Feiermann, Antonia Pérez Arias, Peter Krauthausen und Marcus Baum. Hier gab es immer Raum für offene Diskussionen und aufmunternde Worte in schwierigen Situationen. Die gegenseitige Unterstützung werde ich immer in guter Erinnerung behalten.

Für die gute Zusammenarbeit und den Ideenaustausch möchte ich auch Andreas Schmid vom IPR und Jennifer Sander vom IES danken.

Sehr entscheidend hat auch die Zusammenarbeit mit Studenten zum Erfolg meiner Arbeit beigetragen. Hierbei möchte ich besonders Anne Hanselmann, Kurt Volker Krieg, David Albrecht, Andreas Niess und Thomas Krekeler danken.

Die gute Atmosphäre am ISAS wurde besonders durch die nicht-wissenschaftlichen Mitarbeiter mitgeprägt. Mein Dank gilt hier dem Sekretariat vertreten durch Frau Gambichler und Frau Murr-Grobe, sowie der Werkstatt mit Frau Oberle, Herrn Bleier, Herrn Rihm und Herrn Merkle.

Ich danke meinen Eltern, die mich in allen Entscheidungen meines Lebens unterstützt und mir immer Rückhalt gegeben haben. Mein großer Dank gilt aber meine lieben Frau Sabrina ohne deren Unterstützung und Geduld die Vollendung dieser Arbeit nur sehr schwer möglich gewesen wäre. Ihr widme ich daher diese Arbeit.

Linsengericht, im Oktober 2008

Oliver Schrempf

*Für Sabrina*

# Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Verzeichnis der Beispiele	IX
Notation	XI
Zusammenfassung	1
<b>1 Einleitung</b>	<b>3</b>
1.1 Problemformulierung . . . . .	4
1.2 Aufbau der Arbeit . . . . .	5
<b>I Kaskadierte Systeme</b>	<b>7</b>
<b>2 Kaskadierte Systeme</b>	<b>9</b>
2.1 Stand der Forschung . . . . .	10
2.2 Zustandsschätzung in kaskadierten Systemen . . . . .	11
2.3 Statische und dynamische Systeme . . . . .	14
2.4 Herausforderungen . . . . .	14
<b>3 Systemapproximation</b>	<b>17</b>
3.1 Stand der Forschung . . . . .	17
3.2 Dichterepräsentation . . . . .	18
3.3 Approximation der Systemfunktion . . . . .	19
3.4 Gaussian Mixtures . . . . .	20
3.5 Dirac-Mixture-Repräsentation von diskreten Dichten . . . . .	22

3.6	Hybride bedingte Dichten . . . . .	23
3.7	Kompatibilität . . . . .	28
3.7.1	Kontinuierliche Dichte . . . . .	29
3.7.2	Exakte Beobachtung . . . . .	29
3.7.3	Approximierte Dichte . . . . .	29
<b>4</b>	<b>Dichteapproximation</b>	<b>31</b>
4.1	Stand der Technik . . . . .	31
4.2	Dirac-Mixture-Dichten . . . . .	32
4.3	Inferenz . . . . .	34
4.3.1	Vorwärtsinferenz . . . . .	35
4.3.2	Rückwärtsinferenz . . . . .	35
4.4	Kompatibilität . . . . .	36
4.4.1	Kontinuierliche Dichte . . . . .	37
4.4.2	Exakte Beobachtung . . . . .	37
4.4.3	Approximierte Dichte . . . . .	37
4.5	Optimale Approximation . . . . .	37
4.5.1	Optimierung der Gewichte bei gegebenen Positionen . . . . .	39
4.5.2	Optimierung der Positionen bei festen Gewichten . . . . .	41
4.5.3	Vollständige Optimierung . . . . .	51
4.6	Dynamisches System als Kaskade . . . . .	59
4.6.1	Prädiktor . . . . .	59
<b>5</b>	<b>Modellidentifikation</b>	<b>71</b>
5.1	Stand der Technik . . . . .	71
5.2	Dichteschätzung . . . . .	72
5.2.1	Schätzverfahren . . . . .	73
<b>II</b>	<b>Anwendung: Intentionserkennung</b>	<b>83</b>
<b>6</b>	<b>Konzept und Fallstudie</b>	<b>85</b>
6.1	Stand der Technik . . . . .	85

6.2	Vorgehen . . . . .	86
6.3	Fallstudie . . . . .	92
<b>7</b>	<b>Modelle und Evaluationsumgebung</b>	<b>97</b>
7.1	Entkopplung und Komplexitätsreduktion . . . . .	97
7.1.1	Exakte Entkopplung . . . . .	99
7.1.2	Reduktion . . . . .	101
7.1.3	Vergleich der Modelle . . . . .	103
7.2	Experimente in der virtuellen Realität mit Techniken der weiträumigen Telepräsenz	105
7.2.1	Intentionserkennungsmodell mit integrierten Aktionserkennern . . . . .	111
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>119</b>
	<b>Literaturverzeichnis</b>	<b>123</b>





# Abbildungsverzeichnis

2.1	Kaskadiertes System . . . . .	9
2.2	Blockdiagrammdarstellung eines Beispielsystems mit 3 Teilsystemen . . . . .	11
2.3	Schematische Darstellung eines Subsystems eines kaskadierten Systems . . . . .	12
3.1	Konturplot einer Dichtefunktion mit drei achsenausgerichteten Komponenten . .	19
3.2	Hybrides kaskadiertes Systems . . . . .	25
3.3	Evaluation der Systemapproximationsmethode . . . . .	27
3.4	Approximation des nichtlinearen Modells . . . . .	27
3.5	Vergleich der Zustandsdichten bei gegeben die Beobachtungen . . . . .	28
4.1	Approximation einer Standardnormalverteilung durch eine Dirac-Mixture-Funktion	40
4.2	Approximation einer Gleichverteilung durch eine Dirac-Mixture-Funktion . . . .	43
4.3	Approximation eine Gaussdichte mit unterschiedlicher Komponentenanzahl . . .	46
4.4	Einfacher Progressionsablauf . . . . .	48
4.5	Progressive Approximation einer Gaussian Mixture Dichte . . . . .	50
4.6	Approximation einer Gaussian Mixture Dichte durch Dirac-Mixture-Funktionen	50
4.7	Verlauf eines Parametervektors mit $L = 15$ Komponenten. . . . .	51
4.8	Progressionsablauf für eine Gaussdichte . . . . .	54
4.9	Startlösung für Approximation einer Gaussdichte . . . . .	57
4.10	Endergebnis der Approximation einer Gaussdichte . . . . .	57
4.11	Progressive Approximation einer Gaussian-Mixture mit 3 Komponenten . . . . .	58
4.12	Approximationsergebnis für eine Gaussian-Mixture mit drei Komponenten . . .	59
4.13	Darstellung einer rekursiven Prädiktion als kaskadiertes System . . . . .	60
4.14	Ausschnitt aus der in Abbildung 4.13 . . . . .	62

4.15	Approximationsfehler einer Dirac-Mixture Approximation . . . . .	64
4.16	Approximationsfehler in den Momenten einer Dichte . . . . .	65
4.17	Zeitliche Entwicklung eines chaotischen Systems . . . . .	68
4.18	Prädiktionsschritte für ein chaotisches System . . . . .	69
5.1	Blockdiagramm der progressiven Dichteschätzung . . . . .	75
5.2	Progressionsverlauf für eine Dichteschätzung . . . . .	79
5.3	Resultat und Fehler einer progressiven Dichteschätzung . . . . .	79
5.4	Schätzung einer Gaussian-Mixture-Dichte mit drei Komponenten aus 30 zufälligen Realisierungen . . . . .	80
5.5	Quadratischer Fehler der Dichteschätzung . . . . .	81
6.1	Vennendiagramm zur Definition von Situation und Kontext . . . . .	88
6.2	Modell des Menschen für die Intentionserkennung . . . . .	89
6.3	Generisches Modell für die Intentionserkennung . . . . .	89
6.4	Dynamisches Bayesnetzmodell für die Intentionserkennung . . . . .	90
6.5	Blockdiagrammmodell für die Intentionserkennung . . . . .	90
6.6	Schätzer für das generische Intentionserkennungsmodell . . . . .	91
6.7	Videoaufzeichnung einer Person in der Küche . . . . .	92
6.8	Graphische Oberfläche zur Präsentation transkribierter Aktionsfolgen . . . . .	94
6.9	Menschliche Intentionsschätzung aus transkribierten Aktionsdaten . . . . .	94
6.10	Modellbasierte Intentionsschätzung aus transkribierten Aktionsdaten . . . . .	96
7.1	Kaskadiertes System für Intentions-Aktions Relationen. . . . .	98
7.2	Head Mounted Display NVisor SX der Firma NVIS . . . . .	106
7.3	Lautsprecher des akustischen Trackingsystems . . . . .	107
7.4	HMD mit angebrachtem Kopftracker . . . . .	107
7.5	Ein Handtrackingmodul . . . . .	108
7.6	Gesamtaufbau des Trackingsystems . . . . .	108
7.7	Virtuelle Küche . . . . .	109
7.8	Visualisierung der Intentionsschätzung . . . . .	110
7.9	Kommunikationsstruktur des Evaluationssystems . . . . .	110

7.10 Bedingte hybride Dichte für Annäherung an Roboter . . . . .	113
7.11 DBN Struktur des oberen kaskadierten Teilsystems . . . . .	114
7.12 Aufnahme der Handposition zur Identifikation des Messrauschens . . . . .	115
7.13 Rauschidentifikation des Trackingsystems . . . . .	116
7.14 Modell für die Erkennung der Aktion <b>Halten eines Gegenstands</b> . . . . .	116
7.15 Ergebnisse der Intentionsschätzung . . . . .	118



# Verzeichnis der Beispiele

Beispiel 3.1:	Hybrides kaskadiertes System . . . . .	25
Beispiel 4.1:	Approximation einer Standardnormalverteilung an festen Positionen . . . . .	40
Beispiel 4.2:	Approximation einer Gleichverteilung mit festen Gewichten . . . . .	42
Beispiel 4.3:	Approximation einer Gaussdichte mit festen Gewichten . . . . .	45
Beispiel 4.4:	Progressive Approximation mit festen Gewichten . . . . .	47
Beispiel 4.4:	Progressive Approximation mit festen Gewichten (Fortsetzung) . . . . .	49
Beispiel 4.5:	Approximation einer unnormierten Gaussdichte mit freien Gewichten und Positionen . . . . .	56
Beispiel 4.6:	Approximation einer Gaussian-Mixture-Dichte mit freien Gewichten und Positionen . . . . .	58
Beispiel 4.7:	Monotonie des Approximationsfehlers für steigende Anzahl von Komponenten.	64
Beispiel 4.8:	Prädiktion eines chaotischen Systems. . . . .	67
Beispiel 5.1:	Approximation einer Dichte aus optimalen Realisierungen . . . . .	78
Beispiel 5.2:	Approximation einer Dichte aus zufälligen Samples . . . . .	80
Beispiel 5.3:	Vergleich des progressiven Dichteschätzverfahrens mit dem EM-Algorithmus	81
Beispiel 7.1:	Enkopplung und Modellreduktion . . . . .	99
Beispiel 7.1:	Enkopplung und Modellreduktion (Fortsetzung) . . . . .	100
Beispiel 7.1:	Enkopplung und Modellreduktion (Fortsetzung) . . . . .	100
Beispiel 7.1:	Enkopplung und Modellreduktion (Fortsetzung) . . . . .	101
Beispiel 7.1:	Enkopplung und Modellreduktion (Fortsetzung) . . . . .	102
Beispiel 7.1:	Enkopplung und Modellreduktion (Fortsetzung) . . . . .	103



# Notation

## Konventionen

$x$	Skalar
$\hat{x}$	Mittelwert einer Zufallsvariablen $\mathbf{x}$ .
$E\{\mathbf{x}\}$	Erwartungswert einer Zufallsvariable $\mathbf{x}$ .
$x_k$	Skalar zum Zeitpunkt $k$ .
$\underline{x}$	Spaltenvektor
$\mathbf{x}$	Zufallsvariable
$\underline{x}_k$	Vektor zum Zeitpunkt $k$ .
$\tilde{y}$	Konkrete Beobachtung oder Messwert.
$\mathbf{A}$	Matrizen werden groß und fett geschrieben.
$\mathbb{N}$	Menge der natürlichen Zahlen.
$\mathbb{R}$	Menge der reellen Zahlen.
$\sim$	Verteilungsoperator. Z.B. bedeutet $\mathbf{x} \sim \mathcal{U}$ , dass $\mathbf{x}$ gemäß der Verteilung $\mathcal{U}$ verteilt ist.
■	Ende eines Beispiels oder einer Beschreibung zur Referenzanwendung.
□	Ende eines Beweises.

## Funktionssymbole

$\mathcal{N}(x)$	Gaussdichte
$\delta(x)$	Dirac-Delta Funktion
$\sigma(x)$	Sprungfunktion
$\text{erf}(x)$	Gauß'sche Fehlerfunktion





## Zusammenfassung

In dieser Arbeit werden theoretische und praktische Aspekte der stochastischen Unsicherheitsbetrachtung in kaskadierten dynamischen Systemen behandelt. Die dabei verfolgte Idee besteht darin, Systeme *bottom-up* aus sehr einfachen Teilsystemen aufzubauen, indem der Ausgang eines Systems als Eingang eines folgenden Systems verwendet wird. Hierbei entstehen verdeckte Zustände die gleichzeitig den Ausgang des einen und den Eingang eines anderen Teilsystems beschreiben und dadurch im Allgemeinen nicht direkt beobachtet werden können. Das Ziel ist es, für solche Systemkaskaden eine stochastische Zustandsschätzung durchzuführen und basierend auf den Beobachtungen von Ein- und Ausgängen der Gesamtkaskade auf die verdeckten Zustände zu schließen. Dabei kann, wie in dieser Arbeit gezeigt wird, die Struktur der Kaskade verwendet werden, um die Zustandsschätzung lokal in den Teilsystemen durchzuführen und die resultierende Schätzung an die benachbarten Systeme weiterzugeben. Dadurch entsteht eine durchgehende Unsicherheitsbetrachtung über die eine Zustandsschätzung für das Gesamtsystem ermöglicht wird.

Die Ein- und Ausgänge der Teilsysteme, die hier betrachtet werden, können kontinuierlich oder diskret sein. Daraus ergeben sich hybride Systeme für deren Behandlung eine einheitliche Dichterepräsentation vorgestellt wird. Einen weiteren Schwerpunkt bildet die Behandlung kontinuierlicher Teilsysteme deren Systemfunktion eine nicht-lineare Abbildung von Eingang auf Ausgang beschreibt. Für solche Teilsysteme ist eine exakte Berechnung der Zustandsschätzung häufig nicht möglich, da sich der Typ der Zustandsdichte im Allgemeinen ändert und die Komplexität so anwächst, dass eine Weiterbehandlung in der Kaskade unmöglich ist.

In der Arbeit werden zwei approximative Ansätze vorgestellt, mit deren Hilfe das Schätzproblem für nichtlineare kontinuierliche Teilsysteme gelöst werden kann. Der erste Ansatz beruht auf der Approximation der Systemfunktion durch Mixture-Dichten mit achsenausgerichteten Komponenten. Diese Methode ermöglicht es, hybride bedingte Dichten mit Hilfe von Dirac- und Gaussian-Mixture-Komponenten zu modellieren. Ein Vorteil dieses neuartigen Ansatzes besteht zunächst darin, dass die Vorwärts- und Rückwärtsinferenz aufgrund der Systemapproximation geschlossen lösbar sind, und die Anzahl der Komponenten, die zur Darstellung der posterioren Dichte benötigt werden, begrenzt ist.

Das zweite vorgestellte Verfahren basiert auf der Approximation der zu schätzenden Dichtefunktion selbst, durch eine parametrische Dichtebeschreibung in Form einer Dirac-Mixture-Dichte. Diese Approximation wird als Optimierungsverfahren durchgeführt, indem ein Gütemaß zwischen der wahren Dichte und der Approximation minimiert wird. Die Vorteile dieses Ansatzes bestehen darin, dass auch hier die Inferenz analytisch geschlossen gelöst werden kann und die Anzahl der Komponenten der posterioren Dichte durch die Anzahl der Komponenten der approximierten Dichte begrenzt wird. Diese Vorgehensweise ist verwandt zum Ansatz der Partikelfilter.

Der große Vorteil des hier vorgestellten Verfahrens besteht allerdings darin, dass die Approximation der Dichte deterministisch ist, d. h. die Approximation liefert immer die selbe (optimale) Repräsentierung, im Gegensatz zu Monte-Carlo-Methoden wo die Approximation durch zufälliges Ziehen aus der Dichte entsteht. Daraus erwachsen zwei weitere Vorteile: Zum einen ist die Konvergenzrate der systematischen Dirac-Mixture-Approximation wesentlich höher, zum anderen lässt sich dadurch die optimale Anzahl der benötigten Komponenten bestimmen.

Als eine mögliche Anwendung von kaskadierten Systemen wird in Teil II der Arbeit die Intentionserkennung in der Mensch-Roboter-Kooperation betrachtet. Diese Arbeit enthält nach aktuellem Kenntnisstand die erste systemtheoretische Definition der Intentionserkennung. Hierzu wird ein generisches Modell eingeführt, das auf der Idee der kaskadierten Systeme beruht. Dieses Modell kann für konkrete Anwendungen leicht instanziiert werden. Dazu wird ein Verfahren vorgestellt, welches zeigt, wie hierfür Expertenwissen eingesetzt werden kann. Die Methode wurde in einer Evaluationsumgebung mit menschlichen Probanden erfolgreich getestet.

## KAPITEL 1

# Einleitung

Diese Arbeit beschäftigt sich mit der Kaskadierung einfacher, kausaler Systemmodelle zu einem größeren Gesamtsystem. Solche Systemmodelle besitzen Ein- und Ausgänge und beschreiben deren Zusammenhang. Da die Eingänge als Ursache und der Ausgang als Wirkung angesehen werden können spricht man auch von gerichteten kausalen Modellen. Diese einfachen Systemmodelle lassen sich zu einer Kaskade kombinieren, indem der Ausgang des einen Systems als Eingang eines anderen Systems fungiert. Somit lassen sich aus einer Reihe von einfachen, lokalen Systembeschreibungen, wie in einem Baukasten, größere Systembeschreibungen „zusammenbauen“. Die daraus entstehende Struktur kann beim Umgang mit dem Gesamtsystem genutzt werden. So kann beispielsweise bei der Zustandsschätzung, wie in dieser Arbeit betrachtet, der Berechnungsaufwand verringert werden. Dies geschieht durch eine lokale Auswertung der Einzelsysteme in der Kaskade und die sukzessive Fortschreibung deren Auswirkung auf die benachbarten Systeme. In dieser Arbeit werden stochastische Teilsystembeschreibungen verwendet, um Unsicherheiten, wie beispielsweise endogene oder exogene Rauscheinflüsse, in den Systemmodellen quantifizieren zu können. Diese Unsicherheiten der einzelnen Teilsysteme haben allerdings bei der Zustandsschätzung einen, zum Teil erheblichen Einfluss auf benachbarte Systeme in der Kaskade. Daher müssen diese Unsicherheiten bei der Behandlung der Teilsysteme durchgehend berücksichtigt werden.

Bei der Betrachtung kaskadierter Systeme ergeben sich verschiedene Herausforderungen. Zum einen können für die verschiedenen Teilsysteme unterschiedliche Systembeschreibungen verwendet werden, für die unterschiedliche Eingaberepräsentationen benötigt werden, um eine effiziente Weiterverarbeitung zu gewährleisten. Zum anderen können in der Kaskade sowohl kontinuierliche als auch diskrete Größen zur Beschreibung der Ein- und Ausgänge verwendet werden, wodurch die Gesamtkaskade ein hybrides System darstellt. Eine besonders große Herausforderung ist die Betrachtung wertekontinuierlicher Teilsysteme die durch ein nichtlineares Systemmodell beschrieben werden. Hierbei ist eine exakte Berechnung der Zustandsschätzung häufig nicht möglich, da die Komplexität der Zustandsbeschreibung im Allgemeinen stark anwächst, was eine Weiterverarbeitung der Information in der Kaskade praktisch unmöglich macht.

Der Ansatz der kaskadierten Systeme soll in dieser Arbeit in der Intentionserkennung für die Mensch-Roboter-Kooperation verwendet werden. Dies stellt nicht den einzig möglichen Anwendungsbereich dar, aber zur Intentionserkennung wird ein Modell des Menschen benötigt, welches

sich aus kausalen Teilsystemen zusammensetzen lässt. Hierbei lässt sich also die Kaskadierung sehr gut anwenden, wie im zweiten Teil der Arbeit gezeigt wird.

### 1.1 Problemformulierung

In dieser Arbeit werden Methoden entwickelt mit deren Hilfe kleine kausale Systembeschreibungen zu einem Gesamtsystem zusammengesetzt werden können. Dies wird im Folgenden als Kaskadierung von Systemen bezeichnet. Dabei ist es unerheblich, ob die Teilsysteme diskret oder kontinuierlich, linear oder nichtlinear sind. Die Teilsysteme unterliegen einer gewissen Unsicherheit und werden daher stochastisch, mit Hilfe bedingter Dichten oder Systemabbildungen mit stochastischem Rauschanteil, beschrieben. Diese Unsicherheit der Teilsysteme hat natürlich einen Einfluss auf die benachbarten Teilsysteme. Daher werden die Ein- und Ausgänge mit Hilfe von Zufallsvariablen beschrieben.

Bei der Kaskadierung von Systemen entstehen so genannte verdeckte Zustände. Dies sind Zustände, die gleichzeitig den Ausgang eines Teilsystems und den Eingang eines benachbarten Teilsystems beschreiben. Ziel dieser Arbeit ist es, basierend auf Beobachtungen der freien Ein- und Ausgänge auf diese verdeckten Zustände zu schließen. Hierbei spricht man auch von Zustandsschätzung. Das Ergebnis dieser Zustandsschätzung wird in Form einer Wahrscheinlichkeitsdichtefunktion über die zu schätzende Zufallsvariablen angegeben.

Ein möglicher Ansatz um diese Zustandsschätzung durchzuführen, wäre die Betrachtung des Zustands des Gesamtsystems in Form einer Verbunddichte über alle Zufallsvariablen. Die Zustände der einzelnen verdeckten Zustände könnten dann durch Marginalisierung gewonnen werden. Dieser Ansatz kann allerdings sehr schnell zu komplex werden, weshalb er in der Praxis schlecht anzuwenden ist. In dieser Arbeit wird daher eine Methode zur Zustandsschätzung vorgestellt, die die Struktur der Kaskade von Teilsystemen gezielt ausnutzt. Hierzu werden die Methoden der Bayes'schen Inferenz verwendet, um die einzelnen Systeme separat auszuwerten und das Ergebnis dann sukzessive an die folgenden Teilsysteme weiterzureichen. Dies erlaubt es, die Schätzmethode jeweils konzentriert, nur auf die Teilsysteme, anzuwenden, was eine deutliche Verringerung der Berechnungskomplexität bedeutet.

Eine besondere Schwierigkeit stellt hierbei die Betrachtung nichtlinearer kontinuierlicher Teilsysteme dar. Für solche Systeme existieren oft keine analytische Dichterepräsentation, die das Ergebnis der Zustandsschätzung mit einer endlichen Anzahl von Parametern beschreiben. Im Allgemeinen ändert sich hierbei der Typ der Dichte und die Komplexität der Beschreibung nimmt zu. Daher werden in dieser Arbeit Methoden entwickelt, die dieses Problem approximativ lösen. Grundsätzlich kommen dazu zwei Möglichkeiten in Frage. Zum einen kann die Systemfunktion approximiert werden, um eine geschlossene Lösung der Zustandsschätzung zu ermöglichen, zum anderen kann die Dichte selbst approximiert werden bevor sie als Eingang für die Zustandsschätzung des in der Kaskade folgenden Systems verwendet wird. In beiden

Ansätzen ist es auf jeden Fall das Ziel, die Komplexität der resultierenden Dichte zu begrenzen, um eine Berechenbarkeit zu gewährleisten. Beide Ansätze sollen interoperabel sein, das heißt, das Ergebnis des einen Verfahrens soll nahtlos als Eingabe für das andere Verfahren in der Kaskade genutzt werden können, um eine durchgehende Unsicherheitsbetrachtung zu ermöglichen.

## 1.2 Aufbau der Arbeit

Die vorliegende Arbeit ist in zwei Teile gegliedert. Der erste Teil beschäftigt sich zunächst mit der generischen Betrachtung von kaskadierten Systemen und den Herausforderungen die dieser Ansatz mit sich bringt. Der zweite Teil behandelt die Problemstellung der Intentionserkennung mit Hilfe von kaskadierten Systemen. Die Arbeit ist daher wie folgt strukturiert:

Teil I befasst sich mit Behandlung von Unsicherheiten in kaskadierten Systemen. In Kapitel 2 wird die Idee der Kaskadierung von Teilsystemen dargelegt. Dabei liegt der Fokus auf der Vorgehensweise bei der stochastischen Zustandsschätzung im Hinblick auf eine durchgehende Unsicherheitsbehandlung in der Kaskade. Es wird dazu ein generisches Berechnungsschema vorgestellt. Am Ende des Kapitels werden die Herausforderungen aufgezeigt, für die in den folgenden Kapitel Lösungen erarbeitet werden. Im Mittelpunkt steht hierbei das Problem der Zustandsschätzung für nichtlineare kontinuierliche und hybride Systeme. Als Beiträge zur Lösung dieses Problems werden in Kapitel 3 und 4 Verfahren zur System- und Dichteapproximation vorgestellt, die interoperabel innerhalb einer Systemkaskade eingesetzt werden können und eine durchgehende Unsicherheitsbehandlung ermöglichen. Abgeschlossen wird der erste Teil durch einen Beitrag zur Modellidentifikation in Kapitel 5. Hier wird ein neuartiges Verfahren zur Dichteschätzung aus Datensamples vorgestellt.

Der zweite Teil der Arbeit beschäftigt sich mit der Behandlung der Intentionserkennung durch kaskadierte Systeme. Dazu wird in Kapitel 6 zunächst ein generischer Ansatz präsentiert, gefolgt von einer Fallstudie bei der die Fähigkeit von Menschen zur Intentionserkennung untersucht wurde. Einige konkrete Modelle werden in Kapitel 7 vorgestellt. Dies beinhaltet auch eine Methode zur Entkopplung und Komplexitätsreduktion. Die Beiträge in diesem Teil sind das generische Modell, Methoden zur Komplexitätsreduktion bei diskreten Modellen sowie ein prototypischer Aufbau einer Evaluationsumgebung.

Eine Zusammenfassung und ein Ausblick werden zusammen mit der Bewertung der vorgestellten Verfahren in Kapitel 8 gegeben.

Der Stand der Technik wird jeweils zu Beginn der jeweiligen Kapitel beleuchtet.

In der Arbeit werden an einigen Stellen englische Fachausdrücke verwendet, da eine adäquate deutsche Übersetzung in der Literatur entweder nicht existiert oder sich bisher nicht durchgesetzt hat.



**Teil I**

# **Kaskadierte Systeme**

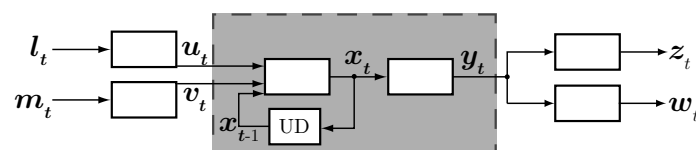




## Kaskadierte Systeme

Unter kaskadierten Systemen soll hier die Verbindung von kleinen, überschaubaren Systemen zu einer größeren Systembeschreibung verstanden werden. Dazu werden die Ausgänge von Teilsystemen mit den Eingängen anderer Teilsysteme verbunden, um so die Wirkung der einzelnen Teilsysteme aufeinander zu beschreiben. Dabei entsteht eine kaskadische Topologie von Teilsystemen, die durch graphische Modelle beschrieben werden kann. Jedes dieser Teilsysteme kann mehrere Eingänge aber nur einen Ausgang haben und wird durch ein Systemmodell, im Allgemeinen eine Funktion, beschrieben. Dieses Systemmodell beschreibt, wie die Eingänge auf den Ausgang abgebildet werden. Modelle für komplexere Systeme, wie Systeme mit internem Zustand oder Systeme mit mehreren Ausgängen können durch eine Kaskadierung solcher einfachen Systeme erzeugt werden. Da selbst die Teilsysteme im Allgemeinen nicht exakt beschrieben werden können, beispielsweise aufgrund endogener oder exogener Rauscheinflüsse, werden hier stochastische Systembeschreibungen verwendet, um die bleibende Unsicherheit zu quantifizieren. Diese stochastischen Modelle können nach [44] als probabilistisches Modell in Form einer bedingten Dichte oder als generatives Modell in Form einer Systemfunktion mit stochastischem Rauschterm beschrieben werden.

Die Gesamtkaskade selbst kann mehrere Eingänge und mehrere Ausgänge haben, was von der Topologie der Verbindung der Teilsysteme abhängt. Durch die Verbindung der Teilsysteme entstehen darüber hinaus so genannte verdeckte Zustände. Dies sind Zustände, die gleichzeitig den Ausgang eines Teilsystems und den Eingang eines folgenden Teilsystems beschreiben. Die Menge der verdeckten Zustände beschreibt den Gesamtzustand des kaskadierten Systems. Ziel ist es, basierend auf Beobachtungen der Ein- und Ausgänge der Kaskade auf den Gesamtzustand, also auf die verdeckten Zustände zu schließen. Dies wird als Zustandsschätzung bezeichnet. Ein Beispiel eines kaskadiertes Systems ist in Abbildung 2.1 zu sehen.



**Abbildung 2.1:** Ein kaskadiertes System aus sechs Teilsystemen. Das Gesamtsystem hat zwei Ein- und zwei Ausgänge. Die Teilkaskade im grauen Kasten beschreibt ein System mit internem Zustand, wobei UD eine Einheitsverzögerung darstellt.

## 2.1 Stand der Forschung

Ein populärer Ansatz für die stochastische Zustandsschätzung ist die Bayes'sche Methode. Das Besondere dabei ist die Möglichkeit der Verwendung so genannter *a priori* oder priorer Wahrscheinlichkeiten. Da diese Wahrscheinlichkeiten auch empirisch angenommen werden können, wird auch häufig von einem Grad persönlicher Überzeugung (engl. „degree of belief“ ) gesprochen. Dies steht im Gegensatz zum frequentistischen Ansatz der Stochastik, bei dem keine empirischen Größen erlaubt sind. Der große Vorteil der Bayes'schen Methode besteht darin, dass die *a priori* Wahrscheinlichkeit aufgrund von Beobachtungen oder Messungen in *a posteriori* Wahrscheinlichkeiten überführt werden können. Man spricht hier auch von Inferenz oder Fusion. Diese Methode ist heute in einer Vielzahl von Anwendungen vertreten. Ein Beispiel hierfür ist die maschinelle Spracherkennung, wo der Bayes'sche Ansatz durch Hidden-Markov-Modelle verfolgt wird [86]. Auch in der Bildverarbeitung werden Bayes'sche Methoden zur Analyse verwendet [46]. Im Bereich der Robotik ist der Ansatz ebenfalls weit verbreitet. Hier wird er beispielsweise zur Lokalisierung [17] und zum Aufbau von Karten unbekannter Umgebungen [16] verwendet. Darüber hinaus wird die Bayes'sche Methode im Bereich der agentenbasierten Informationsfusion verwendet [91, 10]. Ebenfalls sehr populär ist die Methode im Bereich der künstlichen Intelligenz [89].

Kaskadierte Systeme lassen sich sehr gut durch Graphen darstellen. Hierbei wird im Allgemeinen sehr gut sichtbar, welche Ausgänge gleichzeitig Eingänge anderer Teilsysteme sind. Da in dieser Arbeit stochastische Modelle verwendet werden sollen, gibt es eine enge Verwandtschaft von kaskadierten Systemen zu den so genannten graphischen Modellen.

Der wohl bekannteste Vertreter der graphischen Modelle in der Literatur sind die so genannten Bayesnetze (engl. *Bayesian Networks*). Diese Methode wurde populär durch ein Buch von Judea Pearl [78] und hat eine große Anzahl von Publikationen nach sich gezogen [21, 79]. Die Idee der Bayesnetze beruht auf dem Prinzip, Zufallsvariablen als Knoten in einem Graphen zu repräsentieren und deren Abhängigkeiten untereinander durch gerichtete Kanten zu beschreiben. Wurzelknoten beschreiben dabei Zufallsvariablen mit *a priori* Wahrscheinlichkeiten.

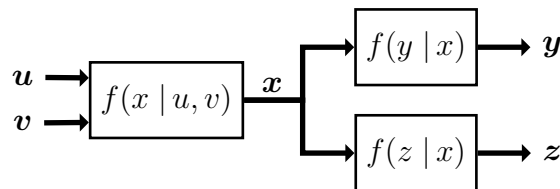
Der Vorteil der Bayesnetzrepräsentierung ist die Fokussierung auf die Variablen. Es fällt Systemmodellierern häufig leichter, zuerst die relevanten Variablen als Knoten zu identifizieren und anschließend die Abhängigkeitsstruktur einzuzeichnen. Ein Nachteil dieser Darstellung ist allerdings, dass die relevanten Systeme nicht direkt sichtbar sind. Häufig wird fälschlicherweise angenommen, dass jede Kante des Graphen für ein System, also für eine bedingte Wahrscheinlichkeit steht. In der Literatur sind Bayesnetze häufig auf diskrete Zufallsvariablen beschränkt. Für Netze mit kontinuierlichen Größen werden bis auf wenig Ausnahmen [29] nur lineare Abhängigkeiten [78] betrachtet.

Eine weitere Möglichkeit der graphischen Modellierung aus dem Bereich der Signalverarbeitung stellen die so genannten Faktorgraphen dar. Sie sind sehr gut in den Arbeiten von Loeliger [65, 57] beschrieben. Faktorgraphen beschreiben das Gesamtsystem durch ungerichtete Graphen. Die Knoten sind Rechtecke und Kreise, wobei die Rechtecke für Funktionen und die Kreise

für Variablen stehen. Ein Faktor besteht aus allen Variablen, die zu einer Funktion gehören. Im hier behandelten Fall sind die Funktionen bedingte Wahrscheinlichkeiten oder bedingte Dichtefunktionen und die Faktoren bestehen aus allen Variablen, die an der jeweiligen bedingten Wahrscheinlichkeit beteiligt sind.

In dieser Arbeit soll eine in der Systemtheorie weit verbreitete graphische Darstellung von Systemen, so genannte Blockdiagramme, verwendet werden, die den Faktorgraphen sehr ähnlich sind. Blockdiagramme werden im Allgemeinen dazu verwendet, um Signalflüsse in Systemen darzustellen.

Die verwendeten Graphen bestehen in der Regel aus Rechtecken und gerichteten Kanten, wobei die Rechtecke Systeme beschreiben und die Kanten den Signalfluss von einem System zum nächsten. Die Blockdiagrammdarstellung eines Beispielsystems ist in Abbildung 2.2 zu sehen. In der hier verwendeten Variante werden in den Systemblöcken die probabilistischen Modelle in



**Abbildung 2.2:** Blockdiagrammdarstellung eines Beispielsystems mit 3 Teilsystemen.

Form von bedingten Dichten notiert. Sie bilden jeweils Dichten am Eingang des Teilsystems auf Dichten am Ausgang des Teilsystems ab. An den Kanten stehen die jeweiligen Zufallsvariablen, über denen die Dichten definiert sind. Die Pfeile deuten an, dass es sich hierbei um gerichtete Modelle handelt. In der hier verwendeten Notation sieht man sehr deutlich, dass  $\mathbf{u}$  und  $\mathbf{v}$  Eingänge und  $\mathbf{y}$  und  $\mathbf{z}$  Ausgänge des Gesamtsystems sind.  $\mathbf{x}$  ist demnach ein verdeckter, interner Zustand des Gesamtsystems, der nicht direkt beobachtet werden kann.

Der Hauptunterschied der klassischen Ansätze wie Bayesnetze und Faktorgraphen zum hier verwendeten Vorgehen liegt in der Herangehensweise an den Modellaufbau. Bei Bayesnetzen und Faktorgraphen ist es das Ziel, ein komplexes System in Teilsysteme zu zerlegen, um dann effizient die Unabhängigkeiten der Teilsysteme ausnutzen zu können. Im hier vorgestellten Ansatz geht es vielmehr darum, bestehende, lokale Teilsysteme miteinander zu koppeln und eine durchgehende Behandlung der Unsicherheiten bei der Betrachtung der Teilsysteme zu gewährleisten. Dieser Ansatz kann somit als *Bottom-up*-Methode bezeichnet werden wohingegen die klassischen Ansätze im Allgemeinen eine *Top-down*-Strategie verfolgen.

## 2.2 Zustandsschätzung in kaskadierten Systemen

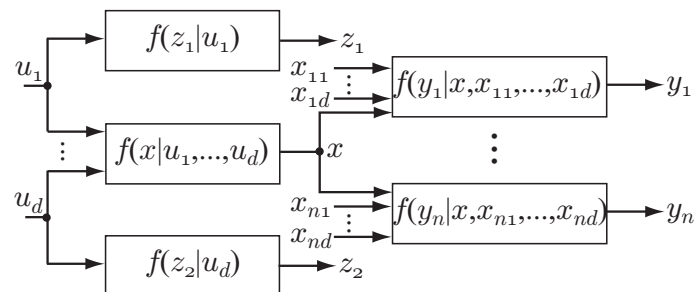
Das Ziel der in dieser Arbeit vorgestellten Methoden ist die Berechnung der verdeckten Zustände die durch eine Kaskadierung entstehen, basierend auf den Beobachtungen der Ein- und Ausgänge

der Gesamtkaskade. Dabei soll die Unsicherheit der einzelnen Teilsysteme in der Berechnung durchgehend mitberücksichtigt werden. Da die Berechnung der einzelnen Zustände nicht durch eine Berechnung der Verbunddichte über alle Zustände gleichzeitig erfolgen soll, wird ein Verfahren zur sukzessiven Berechnung der Teilzustände verwendet. Der Berechnungsaufwand wird hierdurch auf eine lokale Betrachtung der Teilsysteme beschränkt, deren Ergebnis dann als Information über den Ein- bzw. Ausgang des in der Kaskade folgenden Systems verwendet wird.

Zur Berechnung einer Dichte über die Ausgabe bei gegebener Eingabe bzw. über die Eingabe bei gegebener Ausgabe eines Teilsystems wird die Methode der Bayes'schen Inferenz [44] verwendet. Diese erlaubt das Ziehen von Schlussfolgerungen basierend auf unsicheren (stochastischen) Informationen, Inferenz genannt. Da die Systembeschreibungen immer eine kausale Richtung vom Eingang zum Ausgang haben, gibt es eine so genannte Vorwärts- und eine Rückwärtsinferenz. Die Vorwärtsinferenz schließt vom Eingang auf den Ausgang, die Rückwärtsinferenz vom Ausgang auf den Eingang. Bei der Vorwärtsinferenz kann sowohl ein konkreter Wert oder eine Wahrscheinlichkeitsdichtefunktion am Eingang anliegen. Das Ergebnis ist bei einem mit Unsicherheit behafteten System immer eine Dichtefunktion. Bei der Rückwärtsinferenz liegt im Allgemeinen nur ein konkreter Wert des Ausgangs vor, woraus eine Dichte über den Eingang berechnet wird. Im Fall der Kaskadierung von Systemen ist es jedoch typisch, dass der Ausgang aufgrund der unsichereren Information eines Nachbarsystems durch eine Wahrscheinlichkeitsdichte beschrieben wird. Dies muss im Folgenden explizit betrachtet werden. Das Ergebnis der Rückwärtsinferenz ist bei unsicherheitsbehafteten Systemen ebenfalls eine Dichte.

Im Folgenden wird nun ein generisches rekursives Berechnungsverfahren für kaskadierte Systeme vorgestellt das lediglich auf der Bayes'schen Vorwärts- und Rückwärtsinferenz basiert und direkt auf Wahrscheinlichkeitsdichtefunktionen arbeitet und keinerlei Annahme bezüglich der graphischen Struktur macht. Es wird lediglich ausgeschlossen, dass die Kaskade Zyklen enthält. Dieser Ansatz wurde auf der *Fusion 2006* [101] vorgestellt.

Abbildung 2.3 zeigt eine Kaskade der Teilsysteme, die einen Einfluss auf die Berechnung einer Dichte über  $\mathbf{x}$  haben. Die Variable  $d$  bezeichnet den Eingangsgrad des jeweiligen Teilsystems,



**Abbildung 2.3:** Schematische Darstellung eines Subsystems eines kaskadierten Systems.

also die Anzahl der Eingänge. Dieser kann von Teilsystem zu Teilsystem unterschiedlich sein.

Das Interesse gilt nun der Dichte über  $\mathbf{x}$ , die von den Teilsystemen links und rechts beeinflusst wird. Die Information, die vom linken Teilsystem kommt, wird mit  $f^p(x)$  bezeichnet,

die Information von rechts mit  $f^L(x)$ <sup>1</sup>.  $f^p(x)$  und  $f^L(x)$  werden als voneinander unabhängig<sup>2</sup> angenommen, woraufhin sich die Dichte als

$$f(x) = c f^p(x) f^L(x)$$

berechnen lässt.  $c$  ist hierbei eine Normierungskonstante.

Die Dichtefunktion  $f^p(x)$  von links ist das Ergebnis der Bayes'schen Vorwärtsinferenz. In dem betrachteten System ist dies

$$f^p(x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x|u_1, \dots, u_d) \prod_{i=1}^d f(u_i) \backslash_x du_1 \cdots du_d . \quad (2.1)$$

In dieser Gleichung beschreibt  $f(u_i) \backslash_x$  die Dichte über  $u_i$ , bei der die Information, die von  $\mathbf{x}$  kommt, ignoriert wird. Würde man die volle Dichte  $f(u_i)$  berücksichtigen, hätte dies zur Folge, dass die Information über  $\mathbf{x}$ , die ja auch die Information aus den rechten Teilsystemen subsumiert, doppelt verwendet würde. Sie wäre somit in  $f^p(x)$  als auch in  $f^L(x)$  enthalten.

Die Dichte  $f^L(x)$ , die aus dem rechten Teil von  $\mathbf{x}$  kommt, berechnet sich aus dem Produkt der Dichten  $f_j^L(x)$

$$f^L(x) = \prod_{j=1}^n f_j^L(x) .$$

Diese resultieren aus der Rückwärtsinferenz der Teilsysteme, die  $\mathbf{x}$  als Eingang haben. Die Anzahl der Teilsysteme die  $\mathbf{x}$  als Eingang haben wird mit  $n$  bezeichnet. Die jeweiligen Rückwärtsinferenzen werden durch

$$f_j^L(x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_j|x, x_{j1}, \dots, x_{jd}) f^L(y_j) \prod_{k=1}^d f^p(x_{jk}) dy_j dx_{j1} \cdots dx_{jd} \quad (2.2)$$

berechnet. Auch hier wird die Information von links ignoriert, um eine doppelte Verwendung zu vermeiden. Daher tritt in der Formel kein  $f^p(x)$  auf.

Bei der expliziten Beobachtung einer Variable, wie beispielsweise  $\mathbf{y}_j = \check{y}_j$ , wird

$$f(y_j = \check{y}_j) = f^p(y_j = \check{y}_j) = f^L(y_j = \check{y}_j) = \delta(y_j - \check{y}_j)$$

gesetzt. Daraus folgt

$$\begin{aligned} f_j^L(x) &= f_j^L(x|\check{y}_j) \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_j|x, x_{j1}, \dots, x_{jd}) \delta(y_j - \check{y}_j) \prod_{k=1}^d f^p(x_{jk}) dy_j dx_{j1} \cdots dx_{jd} . \end{aligned}$$

Durch Ausnutzung der Ausblendeigenschaft von  $\delta(\cdot)$  erhält man

$$f_j^L(x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(\check{y}_j|x, x_{j1}, \dots, x_{jd}) \prod_{k=1}^d f^p(x_{jk}) dx_{j1} \cdots dx_{jd} .$$

1 Die Bezeichnungen stammen aus der Literatur für stochastische Filterverfahren, wobei  $p$  für *predicted* (vorhergesagt) und  $L$  für *Likelihood* steht.

2 Dies beruht auf der strukturellen Annahme, dass das linke und das rechte Teilsystem, gegeben  $\mathbf{x}$  voneinander unabhängig sind.

Bei unbeobachteten Ausgängen des Gesamtsystems, wird  $f_j^L(y_j) = 1$  gesetzt.

Mit diesen Berechnungsvorschriften lassen sich die Dichten über alle Variablen eines kaskadierten Systems rekursiv berechnen.

### 2.3 Statische und dynamische Systeme

Über Kaskaden von Systemen lassen sich sowohl statische als auch zeitdiskrete dynamische Systeme beschreiben. Für die Zustandsschätzung wird ein dynamisches System im Allgemeinen durch ein System- und ein Messmodell beschrieben [44]. Das Systemmodell beschreibt den zeitlichen Fortschritt des Systemzustands. Das Messmodell beschreibt, wie eine messbare Größe aus dem Systemzustand folgt. Diese Teilsysteme werden im Allgemeinen als voneinander unabhängig angenommen und in separaten Schritten (Prädiktions- und Filterschritt) berechnet, da die aktuelle Messung nur vom aktuellen Zustand abhängt. Auch hierbei handelt es sich, wie bei allgemeinen kaskadierten Systemen, um eine lokale Berechnung. Wenn die Beschreibung eines Gesamtsystems innerhalb eines Zeitschritts aus einer Kaskade besteht, kann nicht mehr von einem Messmodell im eigentlichen Sinne gesprochen werden. So entsteht eine Abhängigkeitskaskade, die entweder ein statisches System oder einen Teil eines dynamischen Systems beschreibt.

An jeden verdeckten Zustand lässt sich ein Teilsystem anhängen, das die Entwicklung in die Zukunft in Abhängigkeit vom aktuellen Zustand beschreibt. Daraus lässt sich auch eine Kaskade über mehrere Zeitschritte erzeugen. Eine Kombination aus einer Abhängigkeitskaskade mit einer Kaskade über verschiedene Zeitschritte ist genauso möglich. Zur Berechnung eines solchen dynamischen Gesamtsystems mit Kaskaden innerhalb eines Zeitschritts wird in jedem Fall das gleiche Verfahren angewendet: Die Inferenzen werden immer nur lokal berechnet, d.h. vom Eingang zum Ausgang eines Teilsystems bzw. umgekehrt. Dabei ist lediglich zu beachten, dass Rückwärtsinferenzen über Zeitschritte bei der Zustandsschätzung nicht nötig sind. Dies ist lediglich bei der Betrachtung von Zustandsfolgen, also Sequenzen von Zuständen, notwendig, die aber nicht Gegenstand dieser Arbeit sind.

### 2.4 Herausforderungen

In der Herleitung des Berechnungsschemas wurden lediglich generische Systeme betrachtet. Das bedeutet, es wurden keine Annahmen über den Typ der Variablen oder der Systeme gemacht. Wenn es allerdings um konkrete Systeme geht, ergeben sich weitere Herausforderungen.

Einen ganz erheblichen Einfluss auf die Handhabbarkeit von kaskadierten Systemen hat der Typ der involvierten Zufallsvariablen. Diese können diskret oder kontinuierlich sein. Im Allgemeinen können aber auch diskrete und kontinuierliche Variablen in ein und demselben Teilsystem vorkommen. Dabei spricht man von hybriden Systemen.

In solchen Systemen ergeben sich vielfältige Herausforderungen, für die in dieser Arbeit Lösungsmöglichkeiten aufgezeigt werden. Eine weitere Herausforderung, die im Folgenden behandelt wird, ist die Berechnung der Bayes'schen Vorwärts- und Rückwärtsinferenz für kontinuierliche und hybride Teilsystemen der Gesamtkaskade, die nichtlineare Abhängigkeiten aufweisen. Es ist bekannt [44], dass sich für diese Fälle die Inferenz im Allgemeinen nicht analytisch geschlossen lösen lässt. Das bedeutet, dass keine analytische Dichtebeschiebung mit einer endlichen Anzahl von Parametern existiert, mit der das Ergebnis der Inferenz exakt dargestellt werden kann. Im Allgemeinen ändert sich der Typ der Dichte und die Komplexität der Dichte steigt. Daher werden in den Kapiteln 3 und 4 zwei neue Verfahren vorgestellt, die das Inferenzproblem approximativ lösen.

Das erste Verfahren verwendet dazu eine Approximation der Systemfunktionen. Diese Approximation muss dabei so gewählt werden, dass der nichtlineare Charakter der Systemfunktion erhalten bleibt. Dieser Ansatz ist einer einfachen Linearisierung deutlich überlegen. Ferner wird für die Behandlung von hybriden Systemen eine einheitliche Dichterepräsentation benötigt. Dafür wird eine Mischung aus Dirac- und Gaussfunktionen verwendet um hybride bedingte Dichten zu realisieren. Das Resultat einer Schätzung ist hier immer eine kontinuierliche, parametrische Dichterepräsentation in Form einer Gaussian-Mixture-Dichte oder einer Dirac-Mixture-Dichte, wobei letztere eine kontinuierliche Repräsentation einer diskreten Dichte darstellt.

Das zweite Verfahren basiert auf einer systematischen Approximation der relevanten Dichtefunktionen zur Lösung des Inferenzproblems. Hierbei wird die wahre Dichte durch einen Dichtetyp approximiert, der eine analytisch geschlossene Lösung der Inferenz erlaubt. Die Approximation der Dichte wird hier als Optimierungsproblem behandelt.

Die beiden vorgestellten Verfahren sind so ausgelegt, dass sie innerhalb einer Kaskade kombiniert werden können. Das bedeutet, dass das Ergebnis einer Inferenz für ein System durch Approximation der Systemfunktion als Eingang für ein folgendes System dienen kann, bei dem die Inferenz durch Dichteapproximation berechnet wird und umgekehrt. Wichtig ist hierbei, dass die Ausgabe des einen Schätzers kompatibel zum Eingang des anderen Schätzers ist. Bei den hier vorgestellten Verfahren handelt es sich um diskrete Approximationen von Dichtefunktionen sowie um parametrische Darstellungen kontinuierlicher Dichtefunktionen. Beide sind approximative Repräsentationen der jeweils wahren Dichtefunktion. Durch das Einsetzen der jeweils vorliegenden Dichtebeschiebung in den folgenden Schätzer wird eine durchgehende Unsicherheitsbetrachtung erreicht, da versucht wird, so wenig wie möglich Information über die wahre Dichte zu verlieren. Für die beiden im Folgenden vorgestellten Ansätze wird jeweils gezeigt, dass die geschlossene Vorwärts- bzw. Rückwärtsinferenz für alle vorkommenden Dichterepräsentationen möglich ist. Dabei wird eine Beschränkung auf die Fälle von Dirac-Mixture-Dichten und Gaussian-Mixture-Dichten vorgenommen. Die Vor- und Nachteile der beiden Ansätze werden in Kapitel 8 betrachtet.





# Systemapproximation

In diesem Kapitel wird eine geschlossene analytische Lösung der Vorwärts- und Rückwärtsinferenz für hybride nichtlineare Teilsysteme durch Approximation der Systemfunktion gezeigt. Die Probleme, die dabei gelöst werden müssen, lassen sich in zwei Kategorien aufteilen:

Um mit hybriden Systemen arbeiten zu können, wird eine einheitliche Repräsentation der Unsicherheiten benötigt. Diskrete Systeme werden im Allgemeinen durch Wahrscheinlichkeitstabellen modelliert, kontinuierliche durch Wahrscheinlichkeitsdichtefunktionen. Es wird also eine vereinheitlichende Repräsentation gesucht, mit deren Hilfe die Formeln der Bayes'schen Inferenz aus Kapitel 2 instanziiert werden können.

Das zweite Problem ist die Berechnung der Inferenz für nichtlineare kontinuierliche Teilsysteme. Der Ansatz, der zur Lösung dieser Probleme verwendet werden soll, basiert auf einer Approximation der jeweiligen Systemfunktion durch eine geeignete Repräsentation. Hierzu wird die Systemfunktion im Folgenden als bedingte Dichte dargestellt, welche durch eine Mischung von achsenausgerichteten Dichtefunktionen approximiert wird. Dies ermöglicht eine geschlossene Lösung der Inferenz, bei der der Typ der Dichte erhalten bleibt und die Anzahl der benötigten Parameter zur Beschreibung der Dichte begrenzt ist.

### 3.1 Stand der Forschung

Eine besonders einfache Methode zur Inferenzberechnung durch Approximation besteht in der Diskretisierung des Zustandsraums. Dies zieht eine Diskretisierung der jeweiligen Systemfunktion nach sich, auf die dann diskrete Verfahren wie beispielsweise das Wonhamfilter [110] angewendet werden können. Der Nachteil hierbei ist allerdings, dass der Zustandsraum häufig zu grob oder zu fein diskretisiert wird. Im ersten Fall führt dies zu großen Fehlern im Schätzergebnis im zweiten Fall werden die Berechnungen zu aufwendig.

Ein Standardverfahren zur Approximation nichtlinearer Systemfunktionen ist die Linearisierung der Systemfunktion. Sie wird beispielsweise im populären Erweiterten Kalman Filter (EKF) verwendet [75]. Dieser Ansatz ist allerdings für Systeme mit sehr starker Nichtlinearität nur bedingt geeignet. Unter anderem, weil sich das Verfahren auf die Berechnung des ersten Moments und des zweiten zentralen Moments beschränkt, was zur Beschreibung der Dichtefunktionen bei

nichtlinearen Systemen häufig nicht ausreicht, da häufig multimodale Dichten entstehen. Eine Verringerung des Fehlers durch die Linearisierung wird in verschiedenen Erweiterungen des EKF erzielt. Dabei sind unter anderem das Iterated extended Kalman Filter (IEKF) [102] und das Second-Order Extended Kalman Filter [47] zu nennen. Das IEKF verringert den Linearisierungsfehler durch mehrfache Ausführung des EKF Filterschritts. Das Second-Order Extended Kalman Filter verwendet eine Taylorreihenentwicklung zweiter Ordnung der Systemfunktion. Eine Verbesserung der Performance gegenüber dem EKF ist aber nicht in allen Fällen garantiert. Zudem ist es rechenaufwendiger und die zweite Ableitung muss nicht immer existieren oder muss numerisch ermittelt werden.

In diesem Kapitel soll ein Verfahren verwendet werden, welches ein probabilistisches Systemmodell mit Hilfe einer geeigneten Dichtefunktion wie beispielsweise einer Gaussian Mixture [29, 49] approximiert. Da das probabilistische Modell durch eine bedingte Dichte beschrieben wird, kann es nicht exakt durch eine Dichte approximiert werden. Auf einem festen Intervall lässt sich die bedingte Dichte allerdings beliebig genau durch Gaussian Mixtures beschreiben.

Für die Behandlung hybrider kaskadierter Systeme gibt es in der Literatur zu Bayesnetzen eine Fülle von Veröffentlichungen [60, 61, 73, 62]. Allerdings beschränken sich diese Methoden auf lineare Abhängigkeiten und additives Gauss'sches Rauschen. Es werden dabei lediglich das erste Moment und das zweite Zentralmoment der kontinuierlichen Dichten betrachtet.

Eine Methode, die dem hier vorgestellten Verfahren sehr ähnlich ist, aber im Allgemeinen nur auf rein kontinuierliche Probleme angewendet wird, ist das *Hybrid Density Filter* [50, 51].

Der Ansatz, der in diesem Kapitel nun vorgestellt werden soll, beruht auf Arbeiten die auf der *CDC 2004* [96], auf dem *IFAC World Congress 2004* [98] und auf der *Fusion 2006* [101] vorgestellt wurden. Diese Methoden lassen sich sowohl für kaskadierte Systeme als auch für allgemeine nichtlineare hybride Filterverfahren nutzen.

### 3.2 Dichterepräsentation

Als Dichterepräsentation sollen, sowohl für den kontinuierlichen als auch für den diskreten Teil, Beschreibungen verwendet werden, die universelle Approximatoren darstellen. Im Kontinuierlichen sind dies beispielsweise Gaussian-Mixture-Funktionen [104]. Im Diskreten werden die Zustände durch Dirac-Delta-Funktionen in einen kontinuierliche Raum abgebildet. Diese Darstellung wird auch Dirac-Mixture-Funktion genannt.

Zunächst wird aber erst einmal angenommen, dass eine generische Mixtur der Form

$$f(\cdot) = \sum_{i=1}^M \alpha_i g_i(\cdot)$$

zur Repräsentierung beliebiger Dichten verwendet wird.  $g_i(\cdot)$  ist hier eine Wahrscheinlichkeitsdichtefunktion und  $M$  die Anzahl der Komponenten in der Mixtur.  $\alpha_i$  sind positive Gewichtungsfaktoren mit der Eigenschaft  $\sum_{i=1}^M \alpha_i = 1$ . Da im Folgenden, durch die Behandlung bedingter Dichten, mehrdimensionale Funktionen betrachtet werden, müssen multivariate Mixturen vom Typ

$$f(x_1, \dots, x_n) = \sum_{i=1}^M \alpha_i g_i(x_1, \dots, x_n)$$

verwendet werden.

Um die Vereinfachung der Inferenzschritte zu ermöglichen, werden Mixturen mit so genannten achsenausgerichteten Komponenten verwendet. Dies bedeutet, dass sich jede Komponente  $g_i(x_1, \dots, x_n)$  in das Produkt

$$g_i(x_1, \dots, x_n) = f_{i,1}(x_1) \cdot \dots \cdot f_{i,n}(x_n)$$

zerlegen lässt. Die vollständige Mixtur kann dann als

$$f(x_1, \dots, x_n) = \sum_{i=1}^M \alpha_i \prod_{j=1}^n f_{i,j}(x_j)$$

geschrieben werden. Die Funktion selbst, also die Summierung der Komponenten, ist im Allgemeinen nicht achsenausgerichtet! Ein Beispiel dafür ist in Abbildung 3.1 zu sehen. Dies ist der Konturplot einer Dichtemixtur mit drei Komponenten. Jede Komponente für sich ist

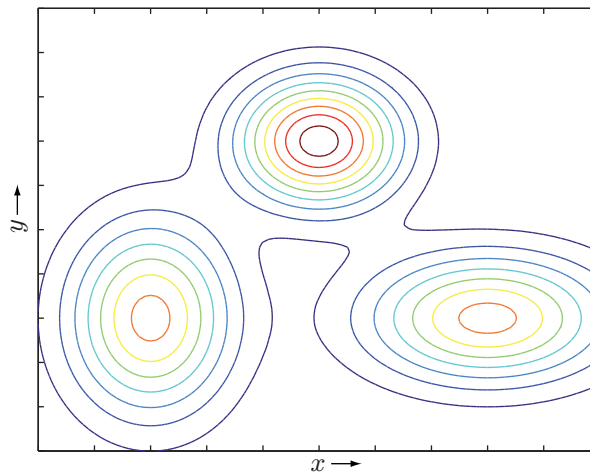


Abbildung 3.1: Konturplot einer Dichtefunktion mit drei achsenausgerichteten Komponenten.

symmetrisch zur  $x$ - und zur  $y$ -Achse. Die Gesamtdichte ist aber nicht achsenausgerichtet!

### 3.3 Approximation der Systemfunktion

Die Idee besteht nun darin, eine bedingte Dichtefunktion durch eine solche Mixtur

$$f(x|u_1, \dots, u_d) \approx \sum_{i=1}^M \gamma_i f_i(x) \prod_{j=1}^d f_{i,j}(u_j) \quad (3.1)$$

mit achsenausgerichteten Komponenten zu approximieren. Der Grund für diesen Schritt liegt in der Tatsache, dass sich durch diese Approximation die Formeln des in Abschnitt 2.2 vorgestellten rekursiven Berechnungsverfahrens weiter vereinfachen lassen.

Die Vorwärtsinferenz ist nach (2.1) gegeben durch

$$f^p(x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x|u_1, \dots, u_d) \prod_{i=1}^d f(u_i) \backslash_x du_1 \cdots du_d .$$

Setzt man die Approximation (3.1) ein, so erhält man

$$f^p(x) \approx \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \sum_{i=1}^M \gamma_i f_i(x) \prod_{j=1}^d f_{i,j}(u_j) \prod_{k=1}^d f(u_k) \backslash_x du_1 \cdots du_d .$$

Durch Hineinziehen der Integrale in die Summe und Umsortieren erhält man die Form

$$f^p(x) \approx \sum_{i=1}^M \gamma_i f_i(x) \prod_{j=1}^d \int_{-\infty}^{\infty} f_{i,j}(u_j) f(u_j) \backslash_x du_j . \quad (3.2)$$

Ähnliches gilt für die Rückwärtsinferenz. Sie ist in (2.2) gegeben durch

$$f_j^L(x) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_j|x, x_{j1}, \dots, x_{jd}) f^L(y_j) \prod_{k=1}^d f^p(x_{jk}) dy_j dx_{j1} \cdots dx_{jd} .$$

Durch Einsetzen der Approximation der Systemfunktion (3.1) erhält man

$$f_j^L(x) \approx \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left( \sum_{i=1}^M \gamma_i f_i(y_j) f_i(x) \prod_{k=1}^d f_{i,k}(x_{jk}) \right) f^L(y_j) \prod_{k=1}^d f^p(x_{jk}) dy_j dx_{j1} \cdots dx_{jd} .$$

Auch hier lassen sich die Integrale in die Summe ziehen und umsordieren und man erhält

$$f_j^L(x) \approx \sum_{i=1}^M \gamma_i f_i(x) \int_{-\infty}^{\infty} f_i(y_j) f^L(y_j) dy_j \prod_{k=1}^d \int_{-\infty}^{\infty} f_{i,k}(x_{jk}) f^p(x_{jk}) dx_{jk} . \quad (3.3)$$

Im Folgenden werden nun Dichtetypen für den kontinuierlichen, den diskreten und den hybriden Fall angegeben, für die sich diese Integrale analytisch lösen lassen.

### 3.4 Gaussian Mixtures

Für den rein kontinuierlichen Fall lässt sich die bedingte Dichte durch eine Gaussian Mixture beschreiben. Für den Fall einer Gaussian Mixture mit achsenausgerichteten Komponenten, der hier betrachtet wird, erhält man somit

$$f(x|u_1, \dots, u_d) = \sum_{i=1}^M \gamma_i \mathcal{N}_i(x, \mu_i, \sigma_i) \prod_{j=1}^d \mathcal{N}_{i,j}(u_j, \mu_{ij}, \sigma_{ij}) ,$$

wobei  $\mathcal{N}(\cdot)$  eine Gaussdichte bezeichnet. Die Erwartungswerte der jeweiligen Komponenten sind durch  $\mu$  und die Standardabweichungen durch  $\sigma$  gegeben. Durch Einsetzen dieser Funktion in (3.2) ergibt sich

$$f^p(x) = \sum_{i=1}^M \gamma_i \mathcal{N}_i(x, \mu_i, \sigma_i) \prod_{j=1}^d \int_{-\infty}^{\infty} \mathcal{N}_{i,j}(u_j, \mu_{ij}, \sigma_{ij}) \mathcal{N}(u_j, \mu_j, \sigma_j) \backslash_x du_j .$$

Dieses Integral über das Produkt von Gaussdichten lässt sich analytisch lösen und man erhält

$$f^p(x) = \sum_{i=1}^M \gamma_i \mathcal{N}_i(x, \mu_i, \sigma_i) \prod_{j=1}^d \underbrace{\mathcal{N}(\mu_{ij}, \mu_j \backslash_x, \sqrt{\sigma_{ij}^2 + \sigma_j^2 \backslash_x})}_{c_{u_j}} ,$$

was letztendlich wieder einer Gaussian Mixture

$$f^p(x) = \sum_{i=1}^M \alpha_i \mathcal{N}_i(x, \mu_i, \sigma_i)$$

mit

$$\alpha_i = \gamma_i \prod_{j=1}^d c_{u_j}$$

entspricht.

Dasselbe gilt für die Rückwärtsinferenz. Durch Einsetzen der Gaussian Mixture erhält man mit

$$f_j^L(x) = \sum_{i=1}^M \gamma_i \mathcal{N}_i(x) \underbrace{\int_{-\infty}^{\infty} \mathcal{N}_i(y_j) \mathcal{N}^L(y_j) dy_j}_{c_{y_j}} \prod_{k=1}^d \underbrace{\int_{-\infty}^{\infty} \mathcal{N}_{i,k}(x_{jk}) \mathcal{N}^p(x_{jk}) dx_{jk}}_{c_{x_{jk}}}$$

wieder eine Gaussian Mixture der Form

$$f_j^L(x) = \sum_{i=1}^M \beta_i \mathcal{N}_i(x, \mu_i, \sigma_i) .$$

Die Parameter der Gaussdichten wurden hier aus Gründen der Übersicht weggelassen. Die Gewichte werden hier durch

$$\beta_i = \gamma_i c_{y_j} \prod_{k=1}^d c_{x_{jk}}$$

berechnet. Ebenfalls wurde hier angenommen, dass  $f^p(u_j) = \mathcal{N}(u_j)$  eine einzelne Gaussdichte ist. Im Allgemeinen kann es sich auch hier um eine Gaussian Mixture handeln. Dasselbe gilt für  $f^L(y_j)$ .

Der große Vorteil dieses Verfahrens besteht nun darin, dass der Typ der Dichte über die Inferenz erhalten bleibt. Das bedeutet, egal ob Vorwärts- oder Rückwärtsinferenz, bei einer Gaussian Mixture als Prioren ist die Posteriorie wieder eine Gaussian Mixture<sup>1</sup>. Darüber hinaus ist bemerkenswert, dass die Anzahl der Komponenten in der Posteriorien begrenzt ist, nämlich durch die

<sup>1</sup> unter der Zusatzannahme von achsenausgerichteten Komponenten für die Systemapproximation

Anzahl der Komponenten, die für die Approximation der Systemfunktion verwendet wurden. Ferner gibt die geschlossene Lösung unmittelbar die Parameter der posterioren Dichte an.

Um die Marginaldichte über  $\mathbf{x}$  zu berechnen, müssen die Dichten  $f^p(x)$  und  $f^L(x)$  multipliziert werden. Dies ist bei Gaussian Mixtures analytisch möglich und man erhält wieder eine Gaussian Mixture als Ergebnis. Lediglich die Anzahl der Komponenten steigt. Sie ist das Produkt der Komponentenzahlen von  $f^p(x)$  und  $f^L(x)$ . Da häufig eine so hohe Anzahl von Komponenten nicht benötigt wird, um eine gute Darstellung der wahren Dichte zu erhalten, wurde auf der *Fusion 2005* eine Methode vorgestellt, die die Multiplikation mit einer gleichzeitigen Reduktion der Komponentenanzahl ausführt [95]. Dazu wird ein progressives Verfahren verwendet, das den einen Multiplikator graduell in das Produkt überführt und die Komponenten dabei anpasst.

Es bleibt die Frage, wie Parameter für eine gute Approximation der Systemfunktion bestimmt werden können? Für einfache Systeme können die Parameter mit etwas Erfahrung häufig durch „Hinsehen“ angegeben werden. Eine Methode zur optimalen Berechnung der Parameter der Approximation bei gegebenem generativen Systemmodell, wird in [49] gezeigt. Hier wird zunächst eine lineare Systemfunktion optimal approximiert und dann progressiv in die gewünschte nicht-lineare Funktion überführt. Eine Idee für eine datengetriebene Gewinnung der hier benötigten Approximation ist in Kapitel 4 gegeben.

### 3.5 Dirac-Mixture-Repräsentation von diskreten Dichten

Für diskrete Systeme wird als Dichterepräsentierung eine so genannte Dirac-Mixture-Dichte verwendet. Es handelt sich dabei um eine Abbildung der diskreten Zustände in einen kontinuierlichen Zustandsraum. Eine Dirac-Mixture-Dichte ist gegeben durch

$$f(x) = \sum_{i=1}^{|\mathbf{x}|} \alpha_i \delta(x - x_i) \ ,$$

wobei  $\delta(\cdot)$  die Dirac-Delta-Funktion ist.  $|\mathbf{x}|$  bezeichnet die Anzahl der diskreten Zustände von  $\mathbf{x}$  und  $x_i$  sind die Zustände von  $\mathbf{x}$  abgebildet auf  $\mathbb{R}$ . Ohne Beschränkung der Allgemeinheit gilt  $x_i = i$  für  $i = 1, \dots, |\mathbf{x}| \in \mathbb{N}$ . Eine Dirac-Mixture-Funktion kann auch zur Approximation einer kontinuierlichen Dichte verwendet werden. Dies wird in Kapitel 4 gezeigt. In diesem Fall ist  $x_i$  nicht nur auf die natürlichen Zahlen  $\mathbb{N}$  beschränkt.

Wird die Dirac-Mixture-Funktion als Dichterepräsentation in die Vorwärtsinferenz (3.2) eingesetzt, so erhält man

$$f^p(x) = \sum_{i=1}^M \gamma_i \left[ \sum_{k=1}^{|\mathbf{x}|} \alpha_{ik} \delta(x - k) \right] \prod_{j=1}^d \int_{\mathbb{R}} \left[ \sum_{l=1}^{|\mathbf{u}_j|} \alpha_{jl} \delta(u_j - l) \right] \left[ \sum_{l=1}^{|\mathbf{u}_j|} \beta_{jl} \delta(u_j - l) \right] du_j \ .$$

Durch Ausnutzung der Ausblendeigenschaft und Umsortieren erhält man eine Dirac-Mixture-Funktion

$$f^p(x) = \sum_{i=1}^M \alpha_i \left[ \sum_{k=1}^{|\mathbf{x}|} \alpha_{ik} \delta(x - k) \right] \ ,$$

wobei die Gewichte durch

$$\alpha_i = \gamma_i \prod_{j=1}^d \sum_{l=1}^{|\mathbf{y}_j|} \alpha_{lj} \beta_{lj}$$

berechnet werden<sup>2</sup>. Die ausführlichere Betrachtung ist in [101] gegeben.

Setzt man die Dirac-Mixture-Beschreibungen in die Rückwärtsinferenz (3.3) ein, so erhält man

$$\begin{aligned} f_j^L(x) = & \sum_{i=1}^M \gamma_i \left[ \sum_{l=1}^{|\mathbf{x}|} \alpha_{il} \delta(x-l) \right] \int_{\mathbb{R}} \left[ \sum_{l=1}^{|\mathbf{y}_j|} \alpha_{il} \delta(y_j-l) \right] \left[ \sum_{l=1}^{|\mathbf{y}_j|} \beta_{il} \delta(y_j-l) \right] dy_j \\ & \cdot \prod_{k=1}^d \int_{\mathbb{R}} \left[ \sum_{l=1}^{|\mathbf{x}_{jk}|} \alpha_{kl} \delta(x_{jk}-l) \right] \left[ \sum_{l=1}^{|\mathbf{x}_{jk}|} \beta_{kl} \delta(x_{jk}-l) \right] dx_{jk} . \end{aligned}$$

Auch hier erhält man nach Ausnutzung der Ausblendeigenschaft und Umsortieren wieder eine Dirac Mixture der Form

$$f_j^L(x) = \sum_{i=1}^M \alpha_i \left[ \sum_{l=1}^{|\mathbf{x}|} \alpha_{il} \delta(x-l) \right] .$$

Die Gewichte lassen sich durch

$$\alpha_i = \gamma_i \left( \sum_{l=1}^{|\mathbf{y}_j|} \alpha_{il} \beta_{il} \right) \prod_{k=1}^d \sum_{l=1}^{|\mathbf{x}_{jk}|} \alpha_{kl} \beta_{kl}$$

berechnen.

Diese Schreibweise ist eine exakte Abbildung der Inferenz für diskrete Systeme in den kontinuierlichen Zustandsraum. Dies ermöglicht nun eine einheitliche Behandlung von kontinuierlichen und diskreten Dichten in der Zustandsschätzung, was im nächsten Schritt zur Behandlung hybrider Teilsysteme führt.

### 3.6 Hybride bedingte Dichten

Im Folgenden werden nun zuerst einige Spezialfälle für bedingte Dichten betrachtet, aus denen dann sukzessive das generische hybride Modell hergeleitet wird. Die Spezialfälle betrachten ein einfaches System, das aus zwei Zufallsvariablen  $\mathbf{x}$  und  $\mathbf{y}$  und dem Modell  $f(\mathbf{y}|\mathbf{x})$  besteht. Dieses Modell wird approximiert durch

$$f(\mathbf{y}|\mathbf{x}) \approx \sum_{i=1}^M \gamma_i f_i(\mathbf{x}) f_i(\mathbf{y}) .$$

<sup>2</sup> Es gilt zu beachten, dass hier Diracfunktionen, also Distributionen multipliziert werden. Das ist in diesem Fall möglich, da es sich um ein *normiertes Produkt über eine Summe von Diracfunktionen* handelt, wodurch eine Konvergenz gewährleistet wird. Dies lässt sich beispielsweise zeigen, indem für die Diracfunktionen Gaussdichten eingesetzt werden und dann ein Grenzübergang  $\sigma \rightarrow 0$  der Standardabweichung durchgeführt wird.

Für den ersten Fall wird angenommen, dass  $\mathbf{x}$  und  $\mathbf{y}$  diskret sind. Daher gilt

$$f(y|x) \approx \sum_{i=1}^M \gamma_i \left[ \sum_{j=1}^{|\mathbf{x}|} \alpha_{ij} \delta(x-j) \right] \left[ \sum_{j=1}^{|\mathbf{y}|} \beta_{ij} \delta(y-j) \right] .$$

Um die Anwendung dieser Darstellung zu erklären wird folgendes Beispiel verwendet: Seien  $\mathbf{x}$  und  $\mathbf{y}$  binär und  $f(y|x)$  durch die Wahrscheinlichkeitstabelle

	$\mathbf{y} = 1$	$\mathbf{y} = 2$
$\mathbf{x} = 1$	0.3	0.7
$\mathbf{x} = 2$	0.8	0.2

gegeben. Um diese bedingte Dichte als Dirac Mixture zu beschreiben, werden  $M = 2$  Komponenten benötigt, deren Parameter durch

$$\begin{array}{cccc} \alpha_{11} = 1 & \alpha_{12} = 0 & \beta_{11} = 0.3 & \beta_{12} = 0.7 \\ \alpha_{21} = 0 & \alpha_{22} = 1 & \beta_{21} = 0.8 & \beta_{22} = 0.2 \end{array}$$

gegeben sind. Die  $\gamma$ -Parameter können alle auf Eins gesetzt werden, da die bedingte Dichte nicht normiert sein muss.

Für den rein diskreten Fall sieht man, dass es sich um eine exakte Darstellung der bedingten Dichte handelt und nicht um eine Approximation.

Als nächstes wird der Fall betrachtet, das  $\mathbf{x}$  diskret und  $\mathbf{y}$  kontinuierlich ist. Dies führt zu der Darstellung

$$f(y|x) \approx \sum_{i=1}^M \gamma_i \left[ \sum_{j=1}^{|\mathbf{x}|} \alpha_{ij} \delta(x-j) \right] \mathcal{N}(y, \mu_i, \sigma_i) .$$

Für diesen Fall werden so viele Komponenten benötigt, dass

$$M \bmod |\mathbf{x}| = 0$$

gilt, da alle Zustände von  $\mathbf{x}$  berücksichtigt werden müssen. Daraus folgt, dass die Parameter für eine binäre Variable  $\mathbf{x}$  durch

$$\begin{array}{cc} \alpha_{11} = 1 & \alpha_{12} = 0 \\ \alpha_{21} = 0 & \alpha_{22} = 1 \\ \vdots & \vdots \\ \alpha_{M-1,1} = 1 & \alpha_{M-1,2} = 0 \\ \alpha_{M1} = 0 & \alpha_{M2} = 1 \end{array}$$

gegeben sind. Die Parameter der Gausskomponenten müssen dann dem Problem entsprechend angepasst werden, beispielsweise mit der Methode aus [49].



Als nächster Fall wird  $\mathbf{x}$  kontinuierlich und  $\mathbf{y}$  diskret betrachtet. Hier gilt

$$f(\mathbf{y} | \mathbf{x}) \approx \sum_{i=1}^M \gamma_i N(\mathbf{x}, \mu_i, \sigma_i) \left[ \sum_{j=1}^{|\mathbf{y}|} \alpha_{ij} \delta(\mathbf{y} - j) \right] .$$

Für diesen Fall ist die Wahl der Komponentenanzahl stark von der zu approximierenden bedingten Dichte abhängig. Das liegt daran, dass  $f(\mathbf{y} | \mathbf{x})$  über  $\mathbf{x}$  im Allgemeinen keine Dichte ist. Da diese Funktion aber durch eine Gaussian-Mixture-Dichte approximiert werden soll, muss ein Intervall festgelegt werden, innerhalb dessen die Approximation durchgeführt wird. Ansonsten würde eine unendlich hohe Anzahl von Gausskomponenten benötigt.

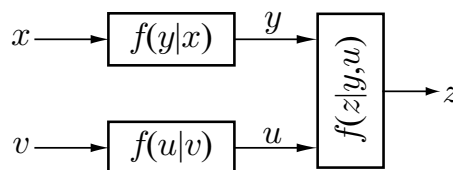
Um den allgemeinen Fall einer hybriden Dichte zu beschreiben, wird die Funktion  $f(\mathbf{y} | x_1, \dots, x_m)$  betrachtet. Hier soll eine Zufallsvariable  $\mathbf{y}$  beliebigen Typs von einer Menge diskreter Zufallsvariablen  $\mathbf{x}_1, \dots, \mathbf{x}_n$  und einer Menge kontinuierlicher Zufallsvariablen  $\mathbf{x}_{n+1}, \dots, \mathbf{x}_m$  abhängen. Dann lässt sich eine hybride bedingte Dichte approximieren durch

$$f(\mathbf{y} | x_1, \dots, x_m) \approx \sum_{i=1}^M \gamma_i f_i(\mathbf{y}) \left( \prod_{k=1}^n \sum_{j=1}^{|\mathbf{x}_k|} \alpha_{ij} \delta(x_k - j) \right) \prod_{k=n+1}^m N(x_k, \mu_i, \sigma_i) .$$

$f_i(\mathbf{y})$  ist dann entsprechend des Typs von  $\mathbf{y}$  eine Gauss- oder eine Dirac-Mixture-Komponente.

### Beispiel 3.1: Hybrides kaskadiertes System

In diesem Beispiel wird ein kaskadiertes System betrachtet, dessen Systemfunktionen nach der oben behandelten Methode approximiert werden. Diese Kaskade hat zwei Eingänge  $x$  und  $v$ , einen Ausgang  $z$  sowie zwei verdeckte Zustände  $\mathbf{y}$  und  $\mathbf{u}$  und ist in Abbildung 3.2 zu sehen. Die Eingänge sind diskret und der Ausgang ist kontinuierlich. Der Zustand  $\mathbf{y}$  ist kontinuierlich und der Zustand  $\mathbf{u}$  ist diskret.



**Abbildung 3.2:** Hybrides kaskadiertes Systems mit zwei diskreten Eingängen und einem kontinuierlichen Ausgang. Der Zustand  $\mathbf{y}$  ist kontinuierlich und der Zustand  $\mathbf{u}$  ist diskret.

Die diskreten Variablen sind jeweils binär und die Parameter der Teilsysteme sind wie folgt gegeben: Für die zwei Eingänge  $x$  und  $v$  wird eine priore Verteilung  $P(x = 1) = 0.4$  und  $P(x = 2) = 0.6$  beziehungsweise  $P(v = 1) = 0.3$  und  $P(v = 2) = 0.7$  angenommen.

Die Abhängigkeit der binären Variable  $\mathbf{u}$  von  $v$  ist durch die Wahrscheinlichkeitstabelle

$v$	$P(\mathbf{u} = 1   v = v)$	$P(\mathbf{u} = 2   v = v)$
1	0.5	0.5
2	0.3	0.7

gegeben.

Die Zufallsvariable  $y$  ist kontinuierlich und hängt von der diskreten Zufallsvariable  $x$  ab. Daher muss für jeden Zustand von  $x$  eine kontinuierliche Dichte über  $y$  gegeben sein. Aus Gründen der Übersicht wird in diesem Beispiel jeweils eine Gaussverteilung über  $y$  mit unterschiedlichen Parametern angenommen. Die Parameter sind durch

$x$	$\mu_y$	$\sigma_y$
1	-2	1
2	2	1

gegeben.

Die Abhängigkeit von  $y$  und  $z$  wird als linear, aber für jeden Zustand von  $u$  als verschieden angenommen. Die generativen Modelle hierfür sind

$$\begin{array}{|l|l|} \hline \mathbf{u} = 1 & z = 2 + y + w_z \\ \hline \mathbf{u} = 2 & z = -2 + y + w_z \\ \hline \end{array},$$

wobei  $w_z$  ein standardnormalverteilter additiver Rauschterm ist. Die entsprechenden probabilistischen Modelle sind demnach gegeben durch

$$\begin{array}{|l|} \hline f(z|y, \mathbf{u} = 1) = N(z, 2 + y, 1) \\ \hline f(z|y, \mathbf{u} = 2) = N(z, -2 + y, 1) \\ \hline \end{array}.$$

Um den neuen Ansatz zu verwenden, wird dieses Modell durch eine Gaussian Mixture mit 50 Komponenten approximiert. Die hierfür verwendeten Funktionen sind

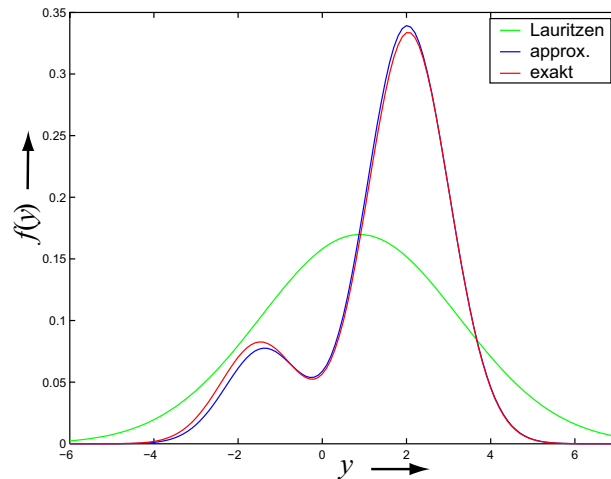
$$f(z|y, \mathbf{u} = 1) \approx \sum_{j=-25}^{25} \frac{1}{50} \cdot N(y, 2 + 0.5j, 1)N(z, 0.5j, 1)$$

und

$$f(z|y, \mathbf{u} = 2) \approx \sum_{j=-25}^{25} \frac{1}{50} \cdot N(y, -2 + 0.5j, 1)N(z, 0.5j, 1).$$

Die Qualität des Verfahrens wird nun mit der exakten Lösung und einem Verfahren aus der Literatur nach Lauritzen [61] verglichen. Dazu wird nun die Beobachtung  $\{v = 1, z = 3\}$  angenommen und die resultierende Dichte über  $y$  betrachtet. Das Ergebnis ist in Abbildung 3.3 zu sehen.

Man sieht hier deutlich, dass das vorgestellte Verfahren, basierend auf der Approximation der Systemfunktionen, sehr nahe an der wahren Dichte liegt. Das Ergebnis nach der Methode von Lauritzen hingegen ist deutlich ungenauer, da hier nur erstes Moment und zweites Zentralmoment der Dichte betrachtet werden.

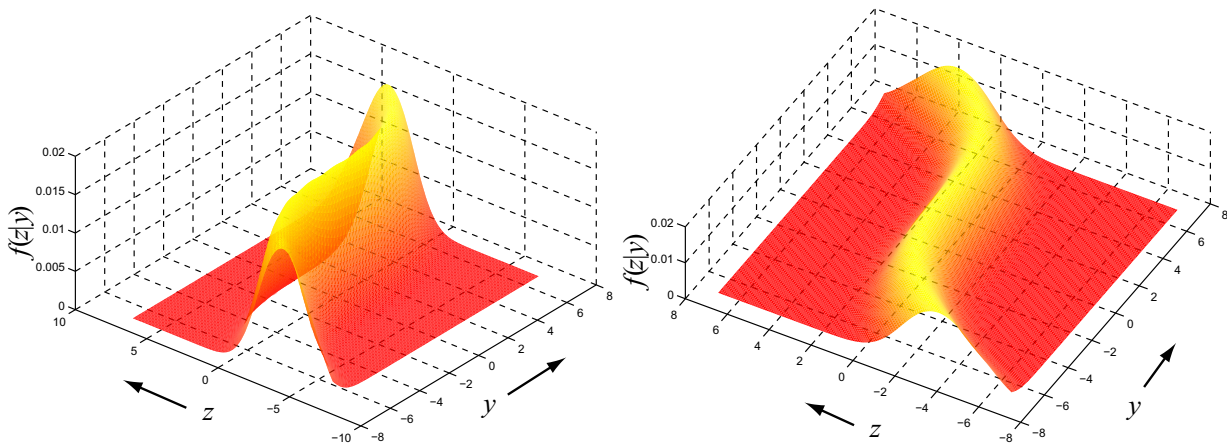


**Abbildung 3.3:** Gegeben die Beobachtung  $\mathbf{v} = 1$ ,  $\mathbf{z} = 3$  wird die exakte Marginaldichte über  $\mathbf{y}$  mit dem Resultat des vorgestellten Verfahrens und der Methode nach Lauritzen verglichen.

In diesem Beispiel wurde keine nichtlineare Abhängigkeit behandelt, da die Methode nach Lauritzen mit diesem Fall nicht umgehen kann. Um zu zeigen, dass die neue Methode es erlaubt mit Nichtlinearitäten umzugehen wird im Folgenden das Modell  $f(z|y, u)$  ausgetauscht. Dabei wird nun das generative Modell

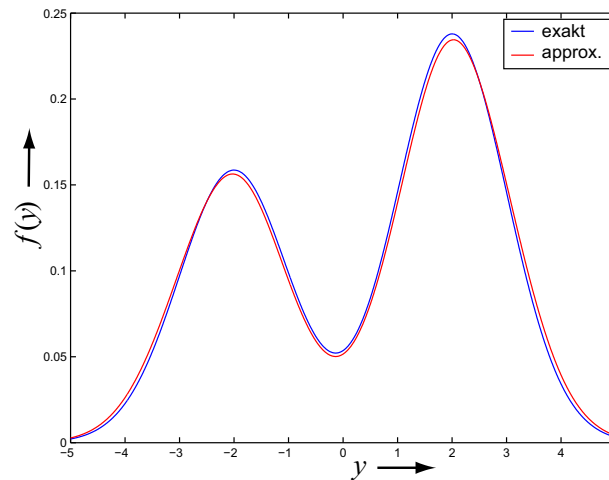
$\mathbf{u} = 1$	$z = 0.01 \cdot y^3 + w_z$
$\mathbf{u} = 2$	$z = 0.01 \cdot (-y)^3 + w_z$

angenommen. Die Approximation dieses Systemmodell ist aus zwei verschiedenen Perspektiven in Abbildung 3.4 für  $\mathbf{u} = 1$  gezeigt.



**Abbildung 3.4:** Approximation des nichtlinearen Modells  $f(z|y, u)$  für  $\mathbf{u} = 1$  aus zwei unterschiedlichen Perspektiven.

Die resultierende Zustandsdichte über  $y$  für die Beobachtung  $\{v = 1, z = 3\}$  ist in Abbildung 3.5 gezeigt.



**Abbildung 3.5:** Vergleich der Zustandsdichten über  $y$  für ein nichtlineares Modell  $f(z|y, u)$  gegeben die Beobachtung  $v = 1, z = 3$ .

Auch hier sieht man, dass das Ergebnis des vorgestellten Verfahrens sehr nahe an der wahren Dichte liegt. ■

Das hier vorgestellte Verfahren eignet sich sehr gut für hybride kaskadierte Systeme, die viele diskrete Anteile haben. In Teil II wird dieses Verfahren daher auch für das Beispiel der Intentionserkennung eingesetzt, da hier viele symbolische Beschreibungen wie Intentionen oder Aktionen durch diskrete Variablen beschrieben werden. Die darüber hinaus benötigten kontinuierlichen Bewegungs- und Messmodelle können mit Hilfe dieses Verfahrens nahtlos in das Gesamtmodell integriert werden.

### 3.7 Kompatibilität

Wie bereits in Abschnitt 2.4 angesprochen, ist es das Ziel den in diesem Kapitel vorgestellten Ansatz und das im folgenden Kapitel verwendete Verfahren interoperabel zu halten. Das bedeutet, dass beide Methoden innerhalb einer Kaskade angewendet werden können. Da die Vorwärts- und Rückwärtsinferenz über ein Teilsystem jeweils für sich abgeschlossen sind, hängt die Kompatibilität der Verfahren lediglich davon ab, ob das jeweilige Verfahren die Ausgabe des anderen Verfahrens als Eingabe verarbeiten kann. Der Fokus liegt hierbei auf der Verarbeitung von Dichten über kontinuierliche Variablen, da die Darstellung von diskreten Dichten durch Dirac-Mixture-Funktion immer exakt ist und für beide Verfahren anwendbar ist.

Für den kontinuierlichen Fall können jedoch verschiedene Dichtebeschreibungen aufgrund der unterschiedlichen Verfahren auftreten. Diese werden im Folgenden aufgezählt und im Hinblick auf die in diesem Kapitel vorgestellte Methode, bezüglich Vorwärts- und Rückwärtsinferenz, betrachtet. In Kapitel 4 wird diese Betrachtung für das andere Verfahren angestellt.

### 3.7.1 Kontinuierliche Dichte

Der Fall, dass eine kontinuierliche Dichte als Eingabe für die Vorwärts- bzw. Rückwärtsinferenz vorliegt, wurde in diesem Kapitel als der Standard behandelt. Hier wurde im Speziellen die Form der Gaussian-Mixture-Dichte als Eingang betrachtet. Für beide Inferenzrichtungen wurde in Abschnitt 3.4 gezeigt, dass das Ergebnis jeweils wieder eine Gaussian-Mixture-Dichte ist, deren Parameter geschlossen berechnet werden können.

### 3.7.2 Exakte Beobachtung

Eine exakte Beobachtung kann entweder für den Eingang oder für den Ausgang eines Teilsystems bestehen und wird in Form eines Diracimpulses an der Stelle der Beobachtung beschrieben. Dies wurde bereits in der Herleitung des Berechnungsschemas in Abschnitt 2.2 eingeführt. Sowohl für die Vorwärts- als auch für die Rückwärtsinferenz ist das Resultat in diesem Fall, bei Anwendung des in diesem Kapitel eingeführten Verfahrens, wieder eine Gaussian Mixture, für die die Parameter in geschlossener Form angegeben werden können.

### 3.7.3 Approximierte Dichte

Der für die Kompatibilität zum in Kapitel 4 vorgestellten Verfahren wichtigste Fall ist das Auftreten von Dichteapproximationen in Form einer Dirac-Mixture-Funktion. Wie diese zustande kommen wird in Kapitel 4 beschrieben. Hier soll nur betrachtet werden, ob das in diesem Kapitel vorgestellte Verfahren mit einer solchen Dichterepräsentation zurecht kommt. Die Antwort ist ja, denn die Dirac-Mixture-Funktion ist letztendlich lediglich eine Verallgemeinerung einer vorliegenden exakten Beobachtung. Es handelt sich hierbei um eine Summe von gewichteten Diracimpulsen, wobei für jeden dieser Impulse als Ergebnis der Vorwärts- oder Rückwärtsinferenz eine Gaussian Mixture entsteht. Das Gesamtergebnis ist eine Summe solcher Gaussian Mixtures, die zu einer Dichte zusammengefasst werden können.

Die Kompatibilität zum Verfahren aus Kapitel 4 ist somit gegeben und das hier beschriebene Verfahren kann im Sinne einer durchgehenden Unsicherheitsbehandlung innerhalb einer Kaskade von Teilsystemen eingesetzt werden.



# Dichteapproximation

Ziel dieses Kapitels ist es, das Inferenzproblem für nichtlineare kontinuierliche Systeme durch Approximation der Dichtefunktion zu lösen. Dabei soll eine parametrische Beschreibung zur Repräsentation der Dichte verwendet werden. Der verwendete Dichtetyp soll sich bei der Inferenz nicht ändern. Darüber hinaus soll die Komplexität der Dichtebeschreibung in Bezug auf die Anzahl der verwendeten Parameter nicht steigen oder zumindest zu Lasten der Approximationsgüte begrenzt werden können.

## 4.1 Stand der Technik

In der Literatur sind vor allem im Bereich der nichtlinearen Filter viele Dichterepräsentationen bekannt. Hier sind beispielsweise Edgeworth Expansionen [20] oder Exponentialdichten [18] zu nennen. Dabei handelt es sich um Reihenentwicklungen. Da diese Entwicklungen abgebrochen werden müssen, handelt es sich bei der beschriebenen Funktion nicht immer um eine korrekte Dichte, da, wie im Beispiel von Edgeworth Expansionen, negative Werte angenommen werden können.

Ein in der Literatur sehr weit verbreiteter Ansatz ist die Verwendung von Gaussian-Mixture-Funktionen zur Dichtebeschreibung [4]. Hierbei handelt es sich um einen universellen Approximator [104], der bei der Verwendung von positiven Gewichtungsfaktoren immer eine korrekte Dichte beschreibt. Die Eigenschaft des universellen Approximators gilt allerdings im Allgemeinen nur für Gaussian Mixtures mit unendlich vielen Komponenten. Eine Methode zur Filterung mit Gaussian Mixtures ist das Progressive Bayes Verfahren [42]. Hier wird das systematische Approximieren der Dichte in ein Optimierungsproblem überführt, das dann progressiv gelöst wird. Dieses Verfahren ist eine theoretische Basis für das in diesem Kapitel vorgestellte Verfahren.

Ein weiterer sehr populärer Ansatz in der Literatur sind Monte-Carlo-Methoden, zu denen auch die so genannten *Partikelfilter* [74, 11, 28, 27] gehören. Hier wird die Dichte durch eine Menge von Realisierungen, so genannte Partikel, repräsentiert. Dies ermöglicht eine punktweise analytische Auswertung der Inferenzintegrale. Da die Partikel zufällig aus der Dichte gezogen werden, handelt es sich hierbei um eine Monte-Carlo-Integration. Da die Samples aus Komplexitätsgründen im Allgemeinen nicht aus der wahren Dichte gezogen werden können, kommen

häufig so genannte *Proposal-Dichten* zum Einsatz, was man auch als *Importance Sampling* [36] bezeichnet. Ein Problem bei Partikelfiltern ist, dass die Partikel mit der Zeit aussterben können, da sie an ungünstigen Stellen plaziert werden und somit abgewichtet werden müssen. Dem wird beispielsweise mit so genannten *Bootstrap-Methoden* [38] begegnet. Ein spezieller Typ sind so genannte regularisierte Partikelfilter [70, 6], die aus einer kontinuierlichen Approximation der posterioren Dichte die Realisierungen ziehen. Damit produzieren sie glattere Resultate.

Die Monte-Carlo-Methode hat jedoch einige Probleme, die durch das zufällige Ziehen von Realisierungen verursacht werden. So sind die Ergebnisse im Allgemeinen nicht deterministisch. Das heißt, dass das mehrfache Ausführen des gleichen Inferenzschrittes bei identischen Vorbedingungen zu unterschiedlichen Ergebnissen führt. Darüber hinaus wird eine große Anzahl von Partikeln benötigt, um eine gute Darstellung der Dichte zu erhalten. Die Anzahl der Partikel kann aber aufgrund des Indeterminismus im Allgemeinen nicht im Vorhinein festgelegt werden.

Um diesen Problemen zu begegnen, wurden so genannte *Quasi Random Sequenzen* [56, 67] als Alternative zum zufälligen Ziehen von Realisierungen vorgeschlagen. Eine bekannte Sequenz ist die Niederreiter-Reihe [71]. Ein Filter, das auf Quasi Random Sequenzen beruht wurde in [83] vorgestellt.

Ein weiteres sehr bekanntes Verfahren, das auf einer punktweisen Auswertung beruht ist das *Unscented Kalman Filter* (UKF) [54, 55]. Hier wird das Integral der Vorwärtsinferenz an deterministischen Stellen, so genannten  $\Sigma$ -Punkten ausgewertet. Die Wahl dieser Punkte liefert allerdings nur im linearen Fall exakte Ergebnisse.

Das in diesem Kapitel vorgestellte Verfahren vereint einige positive Eigenschaften der oben genannten Methoden. Die Idee beruht auf einer punktweisen Darstellung der Dichte, die deterministisch und optimal ist. Dazu wird der Dichtetyp der Dirac-Mixture, der bereits in Kapitel 3 zur Darstellung von diskreten Dichtefunktionen verwendet wurde, für die Approximation kontinuierlicher Dichten verwendet. Diese Approximation wird durch Minimierung eines Gütemaßes in ein Optimierungsproblem überführt, was ein deterministisches Ergebnis in Vorwärts- und Rückwärtsinferenz ermöglicht. Ein weiterer Effekt ist die Möglichkeit, die Anzahl der benötigten Parameter für eine frei wählbare Approximationsgüte zu bestimmen.

## 4.2 Dirac-Mixture-Dichten

Eine besondere Form der Dichtebeschreibung stellt die Summe gewichteter Diracfunktionen dar. Diese Repräsentierung wird als *Dirac-Mixture-Dichte* bezeichnet. Die Dichtefunktion ist gegeben durch

$$f(x) = \sum_{i=1}^L \omega_i \delta(x - x_i) , \quad (4.1)$$

wobei  $\omega_i$  Gewichtungsfaktoren sind und  $\delta(\cdot)$  für die Dirac-Delta-Funktion steht.



Die Dirac-Delta-Funktion, oder kurz Diracfunktion, ist im Allgemeinen definiert durch

$$\delta(x) = \lim_{\sigma \rightarrow 0} \mathcal{N}(x, 0, \sigma) , \quad (4.2)$$

also eine Gaussdichte deren Standardabweichung  $\sigma$  gegen Null geht. Als besondere Eigenschaften der Diracfunktion gelten

$$\delta(x) = \begin{cases} \text{undef,} & x = 0 \\ 0, & \text{sonst} \end{cases} ,$$

sowie die Ausblendeigenschaft

$$\int_{a-\epsilon}^{a+\epsilon} f(x) \delta(x - a) dx = f(a) .$$

Da ferner gilt, dass  $\delta(\pm\infty) = 0$  und  $\int_{-\infty}^{\infty} \delta(x) dx = 1$ , handelt es sich bei der Diracfunktion um eine normierte Dichtefunktion.

Eine Dirac-Mixture-Dichte ist somit eine parametrische Dichtefunktion mit dem Parametervektor

$$\underline{\eta} = \left[ \omega_1 \quad \dots \quad \omega_L \quad x_1 \quad \dots \quad x_L \right]^T ,$$

wobei  $\omega_i$  die Gewichte und  $x_i$  die Positionen der jeweiligen Diracfunktionen sind.

Um zu gewährleisten, dass es sich um eine gültige Dichte handelt, müssen alle Gewichtungsfaktoren

$$\omega_i \geq 0$$

positiv sein für  $i = 1, \dots, L$ . Außerdem muss, um eine normierte Dichte zu erhalten, die Summe der Gewichte Eins ergeben

$$\sum_{i=1}^L \omega_i = 1 .$$

Da sich die Diracfunktion aus der Gaussdichte definiert und eine Gaussian Mixture ein universeller Approximator ist [104], handelt es sich bei der Dirac Mixture ebenfalls um einen universellen Approximator. Allerdings gilt auch hier, dass im Allgemeinen unendlich viele Komponenten benötigt werden.

Eine Dirac-Mixture-Darstellung erlaubt eine geschlossene Berechnung der Momente aus den Parametern.

Für die Momente gilt

$$\begin{aligned}
 E_i &= E\{\mathbf{x}^i\} \\
 &= \int_{-\infty}^{\infty} x^i f(x) dx \\
 &= \int_{-\infty}^{\infty} x^i \sum_{j=1}^L \omega_j \delta(x - x_j) dx \\
 &= \sum_{j=1}^L \omega_j \int_{-\infty}^{\infty} x^i \delta(x - x_j) dx \\
 &= \sum_{j=1}^L \omega_j x_j^i .
 \end{aligned}$$

Der Mittelwert  $\hat{x} = E\{\mathbf{x}\}$  einer Dirac-Mixture-Dichte ist also gegeben durch  $\sum_{j=1}^L \omega_j x_j^i$ .

Die Zentralmomente lassen sich berechnen durch

$$\begin{aligned}
 C_i &= E\{(\mathbf{x} - \hat{x})^i\} \\
 &= \int_{-\infty}^{\infty} (x - \hat{x})^i f(x) dx \\
 &= \int_{-\infty}^{\infty} (x - \hat{x})^i \sum_{j=1}^L \omega_j \delta(x - x_j) dx \\
 &= \sum_{j=1}^L \omega_j \int_{-\infty}^{\infty} (x - \hat{x})^i \delta(x - x_j) dx \\
 &= \sum_{j=1}^L \omega_j (x_j - \hat{x})^i .
 \end{aligned}$$

Ähnliche Beschreibungen für Dichten werden im Bereich der Monte-Carlo-Methoden oder Partikelfilter benutzt [27]. Die Parameter repräsentieren dort Stichproben die zufällig aus einer Dichte gezogen werden. Ein solcher Ansatz wird in dieser Arbeit nicht verfolgt! Die neuartige Idee besteht vielmehr darin, eine komplexe Wahrscheinlichkeitsdichte optimal hinsichtlich eines Gütemaßes durch eine Dirac Mixture zu approximieren. Ein Verfahren dazu wird in Abschnitt 4.5 vorgestellt.

### 4.3 Inferenz

Der große Vorteil der Darstellung einer Dichte durch eine Dirac-Mixture-Funktion liegt darin, dass es für diesen speziellen Dichtetyp möglich ist, die Bayes'sche Vorwärts- und Rückwärtsinferenz aus Abschnitt 2 für beliebige Systeme analytisch geschlossen zu berechnen. Dies soll in den folgenden zwei Abschnitten gezeigt werden. Es werden dazu wieder die Formeln aus Abschnitt 2.2 instanziiert. Zur besseren Übersicht wird im Folgenden ein Teilsystem mit nur einem Eingang behandelt. Dies lässt sich aber ohne Probleme erweitern.

### 4.3.1 Vorwärtsinferenz

Die Bayes'sche Vorwärtsinferenz ist nach (2.1) gegeben durch die Gleichung

$$f^p(x) = \int_{-\infty}^{\infty} f(x|u)f(u) du .$$

Ist nun die priore Dichtefunktion  $f(u)$  in Form einer Dirac-Mixture-Funktion

$$f(u) = \sum_{i=1}^N \omega_i \delta(u - u_i) \quad (4.3)$$

gegeben, so kann diese in (2.1) eingesetzt werden, woraus

$$f^p(x) = \int_{-\infty}^{\infty} f(x|u) \sum_{i=1}^N \omega_i \delta(u - u_i) du$$

folgt. Dieses Integral lässt sich unter Ausnutzung der Ausblendeigenschaft der Diracfunktion lösen und man erhält

$$f^p(x) = \sum_{i=1}^N \omega_i f(x|u_i) .$$

Die Funktion  $f(x|u_i)$  ist eine Dichte über  $\mathbf{x}$ , da  $u_i$  ein konstanter Wert ist. Es lässt sich daher auch von einem Schnitt durch die bedingte Dichte  $f(x|u)$  an der Stelle  $u_i$  sprechen.

Im Falle eines Systems mit additivem Rauschen, beschrieben durch die Funktion

$$\mathbf{x} = g(u) + \mathbf{w} ,$$

wobei  $\mathbf{w}$  nach  $f^w(w)$  verteilt ist, ergibt sich das probabilistische Modell

$$f(x|u) = f^w(x - g(u)) .$$

Als analytische Lösung der Vorwärtsinferenz für dieses System erhält man somit

$$f^p(x) = \sum_{i=1}^N \omega_i f^w(x - g(u_i)) .$$

Wenn das Rauschen gaussverteilt ist, so ist das Ergebnis der Vorwärtsinferenz eine Gaussian-Mixture-Dichte. Diese Vorgehensweise funktioniert gleichermaßen auch für andere Rauschcharakteristika wie beispielsweise multiplikatives Rauschen.

Eine praktische Anwendung dieser analytischen Vorwärtsinferenz wird in Abschnitt 4.6.1 gezeigt.

### 4.3.2 Rückwärtsinferenz

Die Bayes'sche Rückwärtsinferenz ist nach (2.2) gegeben durch

$$f^L(x) = \int_{-\infty}^{\infty} f(y|x)f^L(y) dy . \quad (4.4)$$

wobei  $f^L(y)$  die Beobachtung über  $\mathbf{y}$  beschreibt. Für den Fall, dass eine exakte Beobachtung  $\delta(y - \tilde{y})$  vorliegt, wurde in Abschnitt 2.2 bereits gezeigt, dass

$$f^L(x) = f(\tilde{y}|x) \tag{4.5}$$

gilt. Handelt es sich um eine unsichere Beobachtung, beispielsweise als Ergebnis der Rückwärtsinferenz eines folgenden Systems in der Kaskade, so kann diese in Form einer Dirac-Mixture-Dichte

$$f^L(y) = \sum_{i=1}^M \omega_i \delta(y - y_i)$$

gegeben sein. Diese lässt sich ebenfalls in (2.2) einsetzen und man erhält

$$f^L(x) = \sum_{i=1}^M \omega_i f(y_i|x) .$$

In beiden Fällen ist das Ergebnis eine kontinuierliche Likelihoodfunktion. Um nun den Zustand über  $\mathbf{x}$  zu erhalten muss  $f^L(x)$  mit der prioren Dichte  $f^p(x)$  multipliziert werden. Ist die priore Dichte  $f^p(x)$  als Dirac-Mixture-Funktion wie in (4.3) gegeben, so lässt sich die posteriore Dichte  $f(x)$  berechnen, indem die Likelihoodfunktion  $f^L(x)$  an den Stellen  $x_i$  ausgewertet wird, an denen die Dirac Komponenten von  $f^p(x)$  sitzen. Dies ist sinnvoll, da die Priore an allen anderen Stellen den Wert Null hat und daher auch keinen Beitrag für die Posteriore liefert. Die posteriore Dichte ist somit in diesem Fall wieder eine Dirac-Mixture-Funktion. Der Beitrag der Likelihoodfunktion wird daher in den Gewichten der neu entstehenden Dirac-Mixture-Dichte

$$f(x) = \sum_{i=1}^{N'} \bar{\omega}_i \delta(x - x_i)$$

berücksichtigt mit

$$\bar{\omega}_i = c \cdot \omega_i \cdot f^L(x_i) ,$$

wobei  $\omega_i$  das Gewicht aus der prioren Dichte und  $c$  eine Normierungskonstante ist. Diese Konstante lässt sich berechnen durch

$$c = \frac{1}{\sum_{i=1}^{N'} \omega_i \cdot f^L(x_i)} .$$

Im Fall einer exakten Beobachtung gilt  $N' = N$ , im Fall einer unsicheren Beobachtung  $N' = N \cdot M$ .

## 4.4 Kompatibilität

Bevor nun in Abschnitt 4.5 gezeigt wird, wie eine optimale Approximation einer kontinuierlichen Dichte durch Dirac-Mixture-Dichten berechnet werden kann, soll hier die Kompatibilität der Inferenz mit Dichteapproximation zur Inferenz mit Systemapproximation aus Kapitel 3 untersucht werden. Dazu werden wieder die drei Fälle wie in Kapitel 3 betrachtet.

### 4.4.1 Kontinuierliche Dichte

Durch Verwendung des Verfahrens der Systemapproximation aus Kapitel 3 für die Inferenz auf Vorgänger- oder Nachfolgersysteme in der Kaskade erhält man als Eingang für die Inferenz kontinuierliche Dichtebeschreibungen. Dies sind hier im Allgemeinen Gaussian Mixtures. Um diese Dichten weiterverarbeiten zu können, müssen sie in eine Form gebracht werden, die zur hier vorgestellten Inferenzmethode passen. D.h. sie müssen durch eine Dirac-Mixture-Dichte approximiert werden. Dies ist mit der in Abschnitt 4.5 vorgestellten Methode möglich. Daher kann das hier behandelte Verfahren zusammen mit dem Verfahren aus Kapitel 3 in einer Kaskade verwendet werden und die durchgehende Unsicherheitsbetrachtung bleibt gewahrt.

### 4.4.2 Exakte Beobachtung

Die Behandlung einer exakten Beobachtung wurde für die Rückwärtsinferenz in Form von  $f^L(y - \hat{y})$  bereits gezeigt. Für den Fall der Vorwärtsinferenz stellt die exakte Beobachtung einen Spezialfall zur Eingabe einer Dirac-Mixture-Dichte dar, da es sich nur um eine Dirac-Komponente handelt.

### 4.4.3 Approximierte Dichte

Die Eingabe einer Dirac-Mixture-Dichte für die Inferenz stellt gewissermaßen die native Form für diese Inferenzmethode dar und bereitet daher keine Probleme.

## 4.5 Optimale Approximation

In diesem Abschnitt wird nun ein Vorgehen präsentiert, mit dessen Hilfe sich eine geeignete Approximation durch Dirac-Mixture-Dichten berechnen lässt, um die vorgestellten Inferenzschritte anwenden zu können. Das Ziel ist es, das Approximationsproblem als Optimierungsproblem zu formulieren. Gegeben sei dazu eine kontinuierliche Dichtefunktion  $\tilde{f}(x)$ . Diese Dichte wird auch als *wahre* Dichte bezeichnet und soll durch eine Dirac-Mixture-Dichte der Form

$$f(x, \underline{\eta}) = \sum_{i=1}^L \omega_i \delta(x - x_i)$$

approximiert werden. Wie bereits in Abschnitt 4.2 beschrieben, ist  $\underline{\eta}$  der Parametervektor der Dirac-Mixture

$$\underline{\eta} = \left[ \omega_1 \quad \dots \quad \omega_L \quad x_1 \quad \dots \quad x_L \right]^T$$

mit  $\omega_i$  als Gewichte und  $x_i$  als Positionen der jeweiligen Diracfunktionen.

Die Idee, eine Approximation als Optimierungsproblem aufzufassen, beruht darauf, dass ein Gütemaß  $G(\cdot)$  zwischen der wahren Dichte und ihrer Approximation

$$G\left(\tilde{f}(x), f(x, \underline{\eta})\right)$$

minimiert wird. Die optimale Approximation im Hinblick auf das gewählte Gütemaß ist dann gegeben durch den Parametervektor für den gilt

$$\hat{\underline{\eta}} = \arg \min_{\underline{\eta}} G \left( \tilde{f}(x), f(x, \underline{\eta}) \right) . \quad (4.6)$$

In der Literatur sind viele verschiedene Gütemaße zum Vergleich von Dichten bekannt. Als besonders anschaulich ist hier das integralquadratische Maß

$$G(\underline{\eta}) = \int_{-\infty}^{\infty} \left( \tilde{f}(x) - f(x, \underline{\eta}) \right)^2 dx$$

zu nennen [104, 52]. Dieses Gütemaß beschreibt die Ähnlichkeit der *Form* der Dichten.

Ein in der Literatur sehr populäres Maß ist die Kullback-Leibler Divergenz [58]

$$G(\underline{\eta}) = \int_{-\infty}^{\infty} f(x, \underline{\eta}) \log \left( \frac{f(x, \underline{\eta})}{\tilde{f}(x)} \right) dx .$$

Diese ist allerdings aufgrund der fehlenden Symmetrieeigenschaft kein Distanzmaß. Es gibt jedoch Ansätze zur Symmetrierung [53].

Allen Gütemaßen, die auf Dichten definiert sind ist gemein, dass sie für die Verwendung mit Dirac-Mixtures nicht geeignet sind. Dies liegt daran, dass die Dirac-Mixture an den meisten Stellen den Wert Null annimmt und somit ein Vergleich an diesen Stellen keinen Sinn ergibt. Der Ansatz der hier nun verfolgt wird, besteht stattdessen darin, ein Gütemaß im Verteilungsraum zu betrachten. Die Verteilungsfunktion einer Dichte  $f(x)$  ist gegeben durch das Integral

$$F(x) = \int_{-\infty}^x f(u) du .$$

Die Verteilungsfunktion der Diracfunktion ist ein Sprung, die so genannte Heavisidefunktion

$$\int_{-\infty}^x \delta(u) du = H(x) .$$

Diese ist definiert durch

$$H(x) = \begin{cases} 0, & x < 0 \\ \frac{1}{2}, & x = 0 \\ 1, & x > 0 \end{cases} .$$

Die Verteilungsfunktion einer Dirac-Mixture ergibt sich somit zu

$$F(x) = \sum_{i=1}^L \omega_i H(x - x_i) \quad (4.7)$$

und wird auch als Heaviside-Mixture bezeichnet. Diese nimmt Werte zwischen Eins und Null an und lässt sich mit anderen Verteilungsfunktionen mittels Gütemaß vergleichen.

Ein einfaches und weit verbreitetes Gütemaß im Verteilungsraum ist die Kolmogorov-Smirnov Distanz [84] gegeben durch

$$G(\underline{\eta}) = \max_{-\infty < x < \infty} \left| \tilde{F}(x) - F(x, \underline{\eta}) \right| .$$

Sie gibt den maximalen absoluten Fehler zwischen zwei Verteilungsfunktionen an.

Ein Gütemaß, welches die Form der Verteilungsfunktionen betrachtet, ist die Cramér-von Mises Distanz [12]

$$G(\underline{\eta}) = \int_{-\infty}^{\infty} r(x) \left( \tilde{F}(x) - F(x, \underline{\eta}) \right)^2 dx , \quad (4.8)$$

wobei  $r(x)$  eine Gewichtungsfunktion ist. Im Folgenden wird dieses Gütemaß für die Optimierung verwendet.

Um nun eine optimale Approximation bezüglich des geeigneten Gütemaßes, wie in (4.6) gegeben, zu erhalten, wird die Nullstelle der Ableitung nach dem Parametervektor

$$\frac{\partial G(\underline{\eta})}{\partial \underline{\eta}} = 0$$

gesucht.

Der Rest dieses Abschnitts gliedert sich nun in die Spezialfälle Optimierung der Gewichte bei gegebenen Positionen, Optimierung der Positionen bei festen Gewichten sowie die vollständige Optimierung von Positionen und Gewichten der Dirac-Mixture.

#### 4.5.1 Optimierung der Gewichte bei gegebenen Positionen

Zuerst wird ein besonders einfacher Spezialfall betrachtet. Die Positionen der Dirac-Mixture-Funktion  $x_1, \dots, x_L$  werden hier als gegeben und fest angenommen. Der Parametervektor

$$\underline{\eta} = \left[ \omega_1 \quad \dots \quad \omega_L \right]^T ,$$

besteht somit nur aus den Gewichten.

Die Ableitung der Cramér-von Mises Distanz zwischen der wahren Verteilung  $\tilde{F}(x)$  und der Approximation  $F(x, \underline{\eta})$  ergibt nun

$$\frac{\partial G(\underline{\eta})}{\partial \underline{\eta}} = \begin{bmatrix} \frac{\partial G(\underline{\eta})}{\partial \omega_1} \\ \vdots \\ \frac{\partial G(\underline{\eta})}{\partial \omega_L} \end{bmatrix} .$$

Durch Einsetzen der Verteilungsfunktion der Approximation (4.7) ergeben sich die Ableitung nach den einzelnen Gewichten zu

$$\begin{aligned} \frac{\partial G(\underline{\eta})}{\partial \omega_i} &= 2 \int_{-\infty}^{\infty} r(x) \left[ \tilde{F}(x) - F(x, \underline{\eta}) \right] H(x - x_i) dx \\ &= 2 \int_{x_i}^{\infty} r(x) \left[ \tilde{F}(x) - F(x, \underline{\eta}) \right] dx . \end{aligned} \quad (4.9)$$

Durch Nullsetzen der Funktion und Einsetzen der Verteilungsfunktion erhält man

$$\int_{x_i}^{\infty} r(x) \tilde{F}(x) dx = \sum_{j=1}^L \omega_j \int_{x_i}^{\infty} r(x) H(x - x_j) dx .$$

Die Gewichtungsfunktion  $r(x)$  kann nun so gewählt werden, dass, wie bei einer Fensterfunktion, nur der relevante Teil der wahren Dichte betrachtet wird, beispielsweise einen Bereich in dem der Großteil der Wahrscheinlichkeitsmasse liegt. Eine einfache Gewichtungsfunktion

$$r(x) = \begin{cases} 1, & -c < x < c \\ 0, & \text{sonst} \end{cases} \quad (4.10)$$

erfüllt diesen Zweck.  $c$  ist hierbei eine Konstante. Durch Einsetzen von (4.10) ergibt sich nun

$$\int_{x_i}^c \tilde{F}(x) dx = \sum_{j=1}^L \omega_j \int_{x_i}^c H(x - x_j) dx . \quad (4.11)$$

Es verbleiben also lediglich bestimmte Integrale, die sich für die rechte Seite lösen lassen zu

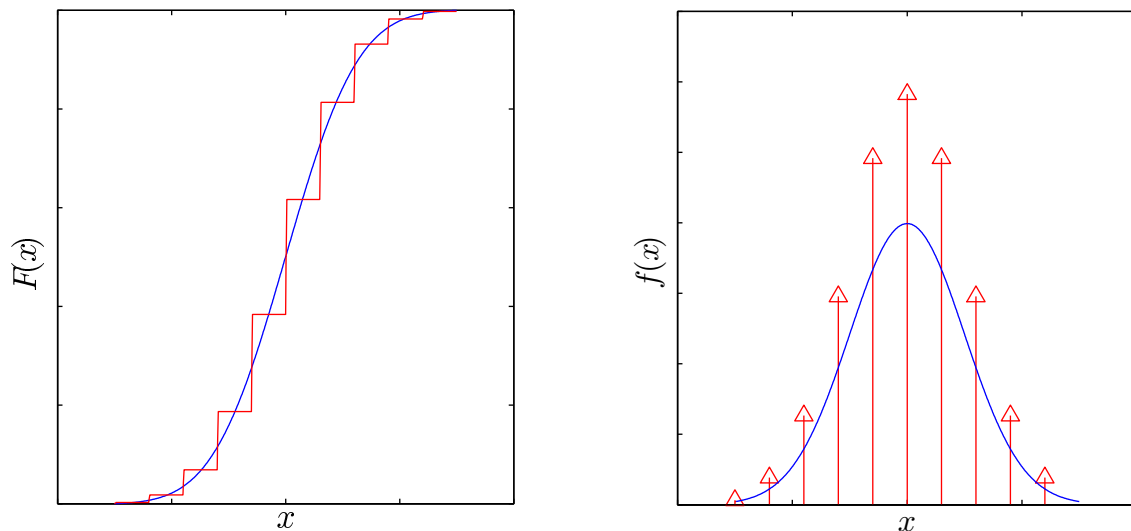
$$\int_{x_i}^c \tilde{F}(x) dx = c \sum_{j=1}^L \omega_j - x_i \sum_{j=1}^i \omega_j - \sum_{j=i+1}^L \omega_j x_j .$$

Dies ergibt für die Gewichte  $\omega_i$  ein lineares Gleichungssystem

$$\begin{bmatrix} \int_{x_1}^c \tilde{F}(x) dx \\ \int_{x_2}^c \tilde{F}(x) dx \\ \int_{x_3}^c \tilde{F}(x) dx \\ \vdots \\ \int_{x_L}^c \tilde{F}(x) dx \end{bmatrix} = \begin{bmatrix} c - x_1 & c - x_2 & c - x_3 & \cdots & c - x_L \\ c - x_2 & c - x_2 & c - x_3 & \cdots & c - x_L \\ c - x_3 & c - x_3 & c - x_3 & \cdots & c - x_L \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c - x_L & c - x_L & c - x_L & \cdots & c - x_L \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \vdots \\ \omega_L \end{bmatrix}$$

### Beispiel 4.1: Approximation einer Standardnormalverteilung an festen Positionen

Dieses Beispiel zeigt die Approximation einer Standardnormalverteilung mit  $L = 10$  Komponenten in der Dirac-Mixture. Die Konstante  $c$  wurde hier auf  $c = 3$  gesetzt. Die Positionen  $x_i$  für  $i = 1, \dots, 10$  sind im Intervall  $[-c, c]$  äquidistant.



**Abbildung 4.1:** Approximation einer Standardnormalverteilung durch eine Dirac-Mixture-Funktion mit festen Positionen. Auf der linken Seite sind die Verteilung der wahren Funktion in blau und die Verteilung der Approximation in rot gegeben. Auf der rechten Seite sind die entsprechenden Dichtefunktionen zu sehen.





Das Ergebnis des Beispiels zeigt sehr schön, wie sich die Gewichte gemäß dem Verlauf der Dichte anpassen. In der Verteilung sieht man, wie der Approximationsfehler symmetrisch zur wahren Dichte angeordnet ist, was dem Minimum der Gütefunktion entspricht.

#### 4.5.2 Optimierung der Positionen bei festen Gewichten

In diesem Abschnitt werden nun die Positionen der Dirac-Mixture optimiert unter der Annahme, dass die Gewichte identisch und fest sind. Die folgenden Herleitungen stammen aus einer Veröffentlichung auf der *CDC 2006* [93].

Der Parametervektor der Dirac-Mixture-Funktion ist gegeben durch

$$\underline{\eta} = [x_1, x_2, \dots, x_L]^T . \quad (4.12)$$

Er besteht also nur aus Positionen. Für die Positionen wird eine Ordnung

$$x_1 < x_2 < \dots < x_{L-1} < x_L$$

angenommen. Die Gewichte sind identisch  $\omega_i = \frac{1}{L}$  für  $i = 1, \dots, L$ . Als Gütemaß wird die Cramér-von Mises Distanz verwendet, diesmal allerdings mit  $r(\cdot) = 1$

$$G(\underline{\eta}) = \int_{-\infty}^{\infty} \left( \tilde{F}(x) - F(x, \underline{\eta}) \right)^2 dx . \quad (4.13)$$

**Satz 4.1** Die optimalen Parameter  $x_i$  für  $i = 1, \dots, L$  einer Dirac-Mixture-Approximation  $f(x, \underline{\eta})$  nach (4.1) mit dem Parametervektor  $\underline{\eta}$  nach (4.12) einer gegebenen Dichte  $\tilde{f}(x)$  bei gegebenem Gütemaß (4.13) erhält man durch Lösen der Gleichung

$$\tilde{F}(x_i) = \frac{2i - 1}{L} \quad (4.14)$$

für  $i = 1, \dots, L$ .

BEWEIS: Als notwendige Bedingung für das Minimum des Gütemaßes muss die Nullstelle der Ableitung des Gütemaßes nach dem Parametervektor

$$\frac{\partial G(\underline{\eta})}{\partial \underline{\eta}} = 0$$

bestimmt werden. Für die einzelnen Parameter  $x_i$  erhält man

$$\frac{\partial G(\underline{\eta})}{\partial x_i} = -2 \int_{-\infty}^{\infty} \left( \tilde{F}(x) - \sum_{j=1}^L w_j H(x - x_j) \right) \delta(x - x_i) dx$$

für  $i = 1, \dots, L$ . Durch Ausnutzung der Ausblendeigenschaft der Diracfunktion erhält man

$$\frac{\partial G(\underline{\eta})}{\partial x_i} = -2 \left( \tilde{F}(x_i) - \sum_{j=1}^L w_j H(x_i - x_j) \right) . \quad (4.15)$$

Nach Auswertung der Heavisidefunktionen  $H(x_i - x_j)$  und Nullsetzen der Gleichung erhält man das gewünschte Ergebnis

$$\tilde{F}(x_i) - \frac{2i - 1}{2L} \stackrel{!}{=} 0$$

für  $i = 1, \dots, L$ . □

Das Gleichungssystem (4.14) ist allerdings nur für Spezialfälle geschlossen analytisch lösbar. Es folgt daher zuerst eine Betrachtung solcher Spezialfälle, gefolgt von einer Lösung für den allgemeinen Fall.

### Gleichverteilung

Der erste Spezialfall, der eine geschlossen Lösung von (4.14) ermöglicht, ist die Gleichverteilungsdichte. Ohne Beschränkung der Allgemeinheit nehmen wir eine Gleichverteilungsdichte gegeben durch

$$\tilde{f}(x) = \begin{cases} 0, & x < 0 \\ 1, & 0 \leq x < 1 \\ 0, & x \geq 1 \end{cases} \quad (4.16)$$

an. Die entsprechende Verteilungsfunktion ist somit gegeben durch

$$\tilde{F}(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < 1 \\ 1, & x \geq 1 \end{cases} .$$

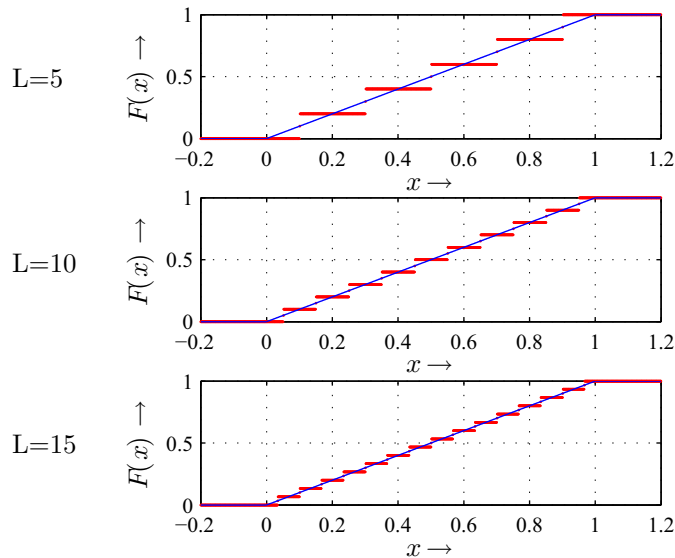
Durch Einsetzen in (4.14) erhält man unmittelbar die Lösung

$$x_i = \frac{2i - 1}{2L} \quad (4.17)$$

für  $i = 1, \dots, L$ .

### Beispiel 4.2: Approximation einer Gleichverteilung mit festen Gewichten

Dieses Beispiel zeigt in Abbildung 4.2 die Lösung von (4.14) für eine Gleichverteilung im Intervall  $[0, 1]$  für  $L = 5, L = 10$  und  $L = 15$  Komponenten in der Dirac Mixture.



**Abbildung 4.2:** Approximation einer Gleichverteilung für unterschiedliche Anzahlen von Komponenten  $L = 5, L = 10$  und  $L = 15$ . In blau ist jeweils die wahre Verteilungsfunktion und in rot die Approximation angegeben.

■

Es ist offensichtlich, dass diese Approximation für  $L \rightarrow \infty$  gegen die wahre Dichte konvergiert. Diese Aussage wird im Folgenden etwas formaler gefasst. Die Beweise sind jeweils in [93] gegeben.

**Satz 4.2** Das Distanzmaß  $G(\underline{\eta})$  zwischen der Gleichverteilungsfunktion  $\tilde{F}(x)$  und ihrer Dirac-Mixture-Approximation mit  $L$  Komponenten ist

$$G(\underline{\eta}) = \frac{1}{12} \frac{1}{L^2}$$

und geht quadratisch gegen Null, wenn die Anzahl der Komponenten  $L$  steigt.

BEWEIS: Das Distanzmaß zwischen  $\tilde{F}(x)$  und  $F(x, \underline{\eta})$  ist in diesem Fall gegeben durch

$$G(\underline{\eta}) = \int_0^1 \left( x - \sum_{i=1}^L \frac{1}{L} H \left( x - \frac{2i-1}{2L} \right) \right)^2 dx .$$

Dies ist äquivalent zu

$$G(\underline{\eta}) = \sum_{i=1}^{2L} \left( \int_0^{\frac{1}{2L}} x^2 dx \right) .$$

Die Auswertung des Integrals ergibt

$$G(\underline{\eta}) = \sum_{i=1}^{2L} \frac{1}{3} \frac{1}{(2L)^3} = \frac{1}{3} \frac{1}{(2L)^2}$$

was den Beweis abschließt. □

Der Erwartungswert und die Varianz der wahren Dichte  $\tilde{f}(x)$  betragen

$$E_{\tilde{f}}\{x\} = \frac{1}{2}$$

und

$$E_{\tilde{f}} \{ (x - E_{\tilde{f}}\{x\}) \} = \frac{1}{12} .$$

**Lemma 4.1** *Der Erwartungswert einer Dirac-Mixture-Approximation einer Gleichverteilungsdichte ist (unabhängig von der Anzahl der Komponenten  $L$ ) gegeben durch*

$$E_{\tilde{f}}\{x\} = \sum_{i=1}^L \frac{1}{L} \frac{2i-1}{2L} = \frac{1}{2} .$$

BEWEIS:

$$E_f\{x\} = \int_{-\infty}^{\infty} x \left( \sum_{i=1}^L \omega_i \delta\left(x - \frac{2i-1}{2L}\right) \right) dx$$

Daraus folgt durch Anwendung der Ausblendeigenschaft

$$E_f\{x\} = \sum_{i=1}^L \frac{1}{L} \frac{2i-1}{2L} .$$

Durch Vereinfachung erhält man

$$E_f\{x\} = \frac{L(L+1) - L}{2L^2} = \frac{1}{2} .$$

□

Somit ist der Erwartungswert der Approximation gleich dem Erwartungswert der wahren Dichte, unabhängig von der Anzahl der Komponenten.

**Lemma 4.2** *Die Varianz der Dirac-Mixture-Approximation ist gegeben durch*

$$E_f \{ (x - E_f\{x\})^2 \} = \frac{L^2 - 1}{12L^2} .$$

BEWEIS: Die Varianz ist gegeben durch

$$\begin{aligned} E_f \{ (x - E_f\{x\})^2 \} &= E_f \left\{ \left( x - \frac{1}{2} \right)^2 \right\} \\ &= E_f \left\{ x^2 - x + \frac{1}{4} \right\} \\ &= \left( \sum_{i=1}^L \frac{1}{L} \frac{(2i-1)^2}{(2L)^2} \right) - \frac{1}{4} . \end{aligned}$$

Da es sich hierbei um eine arithmetische Reihe handelt, lässt sich der Ausdruck zu

$$\frac{4L^2 - 1}{12L^2} - \frac{1}{4}$$

vereinfachen, woraus unmittelbar das Ergebnis folgt. □

Die Varianz der Approximation konvergiert somit gegen die Varianz der wahren Dichte, wenn die Anzahl der Komponenten gegen unendlich geht.

## Gaussdichte

Als weiterer Spezialfall für den die Gleichung (4.14) geschlossen lösbar ist wird nun die Gaussdichte gegeben durch

$$\tilde{f}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2} \right\}$$

betrachtet, wobei  $\mu$  der Mittelwert und  $\sigma$  die Standardabweichung ist. Die Verteilungsfunktion der Gaussdichte kann mit Hilfe der Fehlerfunktion  $\operatorname{erf}(\cdot)$  dargestellt werden als

$$\tilde{F}(x) = \frac{1}{2} \operatorname{erf} \left( \frac{x - \mu}{\sqrt{2}\sigma} \right) + \frac{1}{2} .$$

Die erf-Funktion liegt im Allgemeinen tabelliert vor [19], oder sie wird, wie beispielsweise in MATLAB, approximativ berechnet [22].

**Lemma 4.3** Die Parameter einer Dirac-Mixture-Approximation einer Gaussdichte sind

$$x_i = \sqrt{2}\sigma \operatorname{erf}^{-1} \left( \frac{2i - 1 - L}{L} \right) + \mu$$

für  $i = 1, \dots, L$ .

BEWEIS: Einsetzen der Verteilungsfunktion in (4.14) ergibt

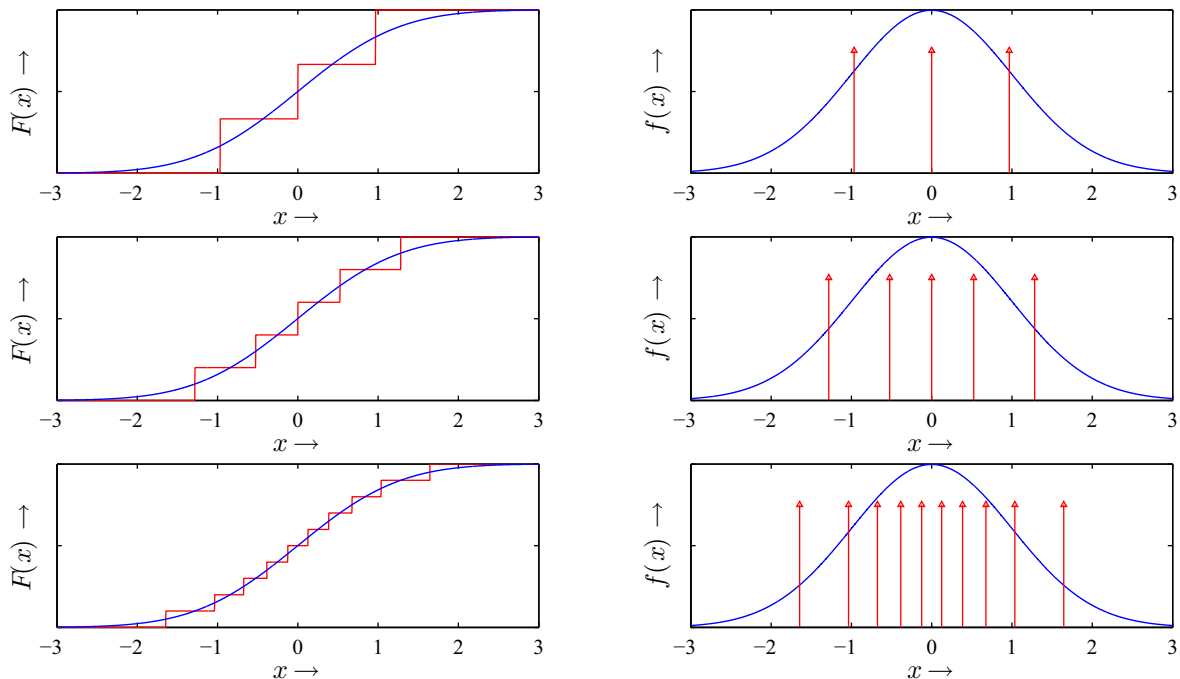
$$\frac{1}{2} \operatorname{erf} \left( \frac{x - \mu}{\sqrt{2}\sigma} \right) + \frac{1}{2} = \frac{2i - 1}{2L} ,$$

woraus das Ergebnis unmittelbar folgt. □

Die inverse Fehlerfunktion  $\operatorname{erf}^{-1}$  ist beispielsweise in MATLAB gegeben durch die Funktion `erfinv(·)`.

### Beispiel 4.3: Approximation einer Gaussdichte mit festen Gewichten

Dieses Beispiel zeigt in Abbildung 4.3 die Approximation einer Gaussdichte durch eine Dirac-Mixture-Funktion mit festen Gewichten.



**Abbildung 4.3:** Approximation einer Gaussdichte durch eine Dirac-Mixture-Funktion mit unterschiedlichen Anzahlen von Komponenten  $L = 3, L = 5$  und  $L = 10$ . Links sind jeweils die wahre Verteilungsfunktion in blau und die Approximation in rot angegeben. Rechts sind die entsprechenden Dichtefunktionen zu sehen.

Die Approximation wurde mit  $L = 3, L = 5$  und  $L = 10$  durchgeführt. Man sieht deutlich, wie sich die Dirac-Komponenten im Bereich der größten Wahrscheinlichkeitsmasse der wahren Dichte drängen. Außerdem wirkt sich die Symmetrie der wahren Dichte als eine Symmetrie in der Approximation aus. ■

### Allgemeiner Fall

Für den allgemeinen Fall, in dem eine beliebige komplexe kontinuierliche Dichtefunktion  $\tilde{f}(x)$  vorliegt, ist (4.14) nicht geschlossen lösbar. Dies liegt vor allem daran, dass die Verteilungsfunktion  $\tilde{F}(x)$  häufig nicht analytisch invertierbar ist. Daher wird hier nun ein progressives numerisches Verfahren zur Lösung dieses Problems vorgeschlagen. Dieses Verfahren wurde bereits erfolgreich für die Approximation einer wahren Dichte durch Gaussian-Mixtures angewendet [43, 42, 41].

Die Hauptidee des Verfahrens besteht darin, die Approximation progressiv herbeizuführen. D.h. es wird zuerst ein „einfaches“ Approximationsproblem gelöst, welches dann sukzessive in das eigentliche (komplexe) Approximationsproblem überführt wird. Dieses kontinuierliche Überführen des einen Problems in das andere erlaubt es, die Lösung graduell anzupassen und so schlussendlich die Gesamtlösung zu erhalten.

Anschaulich lässt sich das Verfahren so beschreiben, dass zu Beginn eine „einfache“ Dichte  $s(x)$  approximiert wird, für die eine optimale Lösung bereits bekannt ist oder leicht gefunden werden kann. Kandidaten hierfür sind beispielsweise die Gleichverteilungsdichte oder die Gaussdichte

für die bereits geschlossene Lösungen hergeleitet wurden. Um die graduelle Überführung der Approximation zu ermöglichen wird ein so genannter Progressionsparameter  $\gamma$  eingeführt, der die wahre Dichte  $\tilde{f}(x)$  parametrisiert. Ohne Beschränkung der Allgemeinheit wird angenommen, dass dieser Parameter Werte im Bereich  $\gamma \in [0, 1]$  annimmt, und zwar so, dass  $\tilde{f}(x, \gamma = 0)$  der einfachen Dichte  $s(x)$  und  $\tilde{f}(x, \gamma = 1)$  der wahren Dichte  $\tilde{f}(x)$  entspricht. Es gilt also

$$\tilde{f}(x, \gamma = 0) = s(x)$$

und

$$\tilde{f}(x, \gamma = 1) = \tilde{f}(x) .$$

Wie der Parameter  $\gamma$  die wahre Dichte parametrisiert, um die obige Forderung zu gewährleisten, ist nicht festgelegt. Daher sind verschiedene Progressionsabläufe denkbar. Hier soll nun zunächst der besonders einfache Ablauf

$$\tilde{f}(x, \gamma) = (1 - \gamma)s(x) + \gamma\tilde{f}(x) \tag{4.18}$$

betrachtet werden. Weitere Möglichkeiten werden in Abschnitt 4.5.3 betrachtet.

Da auch hier die Approximation im Verteilungsraum berechnet werden soll, muss auch der entsprechende Verlauf

$$\tilde{F}(x, \gamma) = (1 - \gamma)S(x) + \gamma\tilde{F}(x)$$

betrachtet werden, wobei  $S(x)$  die Verteilungsfunktion der einfachen Dichte  $s(x)$  beschreibt. Ein Beispiel für den hier betrachteten Progressionsablauf wird nun gegeben.

#### Beispiel 4.4: Progressive Approximation mit festen Gewichten

Als wahre Dichte wird eine Gaussian Mixture mit zwei Komponenten betrachtet, gegeben durch

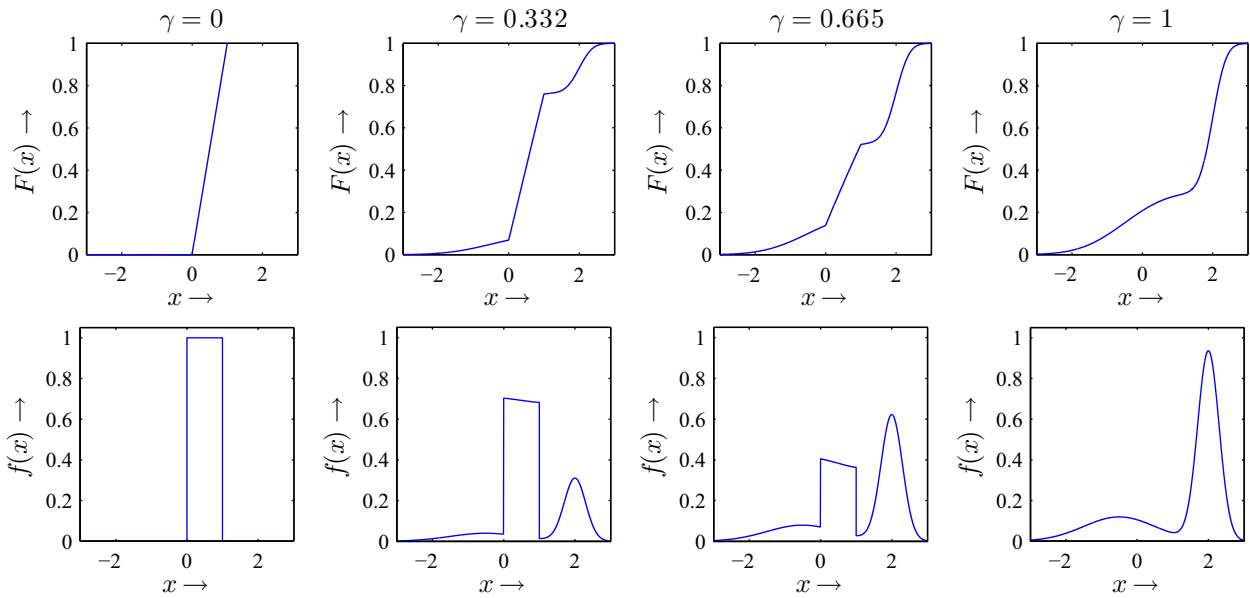
$$\tilde{f}(x) = \omega_1\mathcal{N}(x, \mu_1, \sigma_1) + \omega_2\mathcal{N}(x, \mu_2, \sigma_2)$$

mit

$$\begin{aligned} \omega_1 &= 0.3, \omega_2 = 0.7, \\ \mu_1 &= -0.5, \mu_2 = 2.0, \\ \sigma_1 &= 1, \sigma_2 = 0.3 . \end{aligned}$$

Als einfache Dichte  $s(x)$  wird die Gleichverteilung wie in (4.16) verwendet.

Der Progressionsablauf für  $\gamma = 0$  nach  $\gamma = 1$  ist in Abbildung 4.4 zu sehen.



**Abbildung 4.4:** Einfacher Progressionsablauf für Verteilungen (oben) und Dichtefunktion (unten) für  $\gamma = 0$  (links) nach  $\gamma = 1$  (rechts).



Für den gesamten Verlauf der Progression soll nun das Minimum des Gütemaßes verfolgt werden. D.h. für einen kleinen Schritt des Progressionsparameters  $\gamma$  in Richtung 1 ändert sich die zu approximierende Funktion  $\tilde{F}(x, \gamma)$  geringfügig, woraufhin die Parameter der approximierenden Funktion  $F(x, \underline{\eta})$  angepasst werden müssen, um im Minimum von  $G(\underline{\eta})$  zu bleiben.

Zur Verfolgung der Nullstelle mit Hilfe der Parameter wird die progressive Version  $\tilde{F}(x, \gamma)$  von  $\tilde{F}(x)$  in (4.14) eingesetzt und die partielle Ableitung nach  $\gamma$  berechnet. Da aufgrund der Progression die Positionen der Diracimpulse nach  $x_i = x_i(\gamma)$  implizit von  $\gamma$  abhängen, hängt  $\tilde{F}(x, \gamma)$  sowohl explizit, als auch implizit von  $\gamma$  ab. Die Ableitung von (4.14) ergibt somit

$$\frac{\partial \tilde{F}(x_i, \gamma)}{\partial \gamma} + \frac{\partial \tilde{F}(x_i, \gamma)}{\partial x_i} \frac{\partial x_i(\gamma)}{\partial \gamma} = 0 \quad ,$$

für  $i = 1, \dots, L$ . Dies lässt sich mit

$$\frac{\partial \tilde{F}(x_i, \gamma)}{\partial x_i} = \tilde{f}(x_i, \gamma)$$

zu

$$-\frac{\partial \tilde{F}(x_i, \gamma)}{\partial \gamma} = \tilde{f}(x_i, \gamma) \dot{x}_i(\gamma)$$

vereinfachen, wobei  $\dot{x}_i(\gamma)$  die Ableitung von  $x_i$  nach dem Progressionsparameter  $\gamma$  bezeichnet.

Das resultierende Differentialgleichungssystem kann in Vektor-Matrix Schreibweise angegeben werden durch

$$\underline{b}(\underline{\eta}(\gamma), \gamma) = \mathbf{P}(\underline{\eta}(\gamma), \gamma) \cdot \dot{\underline{\eta}}(\gamma) \tag{4.19}$$



mit

$$\underline{b}(\underline{\eta}(\gamma), \gamma) = \begin{bmatrix} -\frac{\partial \tilde{F}(x_1, \gamma)}{\partial \gamma} \\ -\frac{\partial \tilde{F}(x_2, \gamma)}{\partial \gamma} \\ \vdots \\ -\frac{\partial \tilde{F}(x_L, \gamma)}{\partial \gamma} \end{bmatrix}$$

und

$$\mathbf{P}(\underline{\eta}(\gamma), \gamma) = \begin{bmatrix} \tilde{f}(x_1, \gamma) & 0 & \cdots & 0 \\ 0 & \tilde{f}(x_2, \gamma) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & & \tilde{f}(x_L, \gamma) \end{bmatrix}.$$

Für die spezielle Progression gegeben in (4.18), erhält man

$$\frac{\partial \tilde{F}(x_i, \gamma)}{\partial \gamma} = \tilde{F}(x_i(\gamma)) - S(x_i(\gamma)).$$

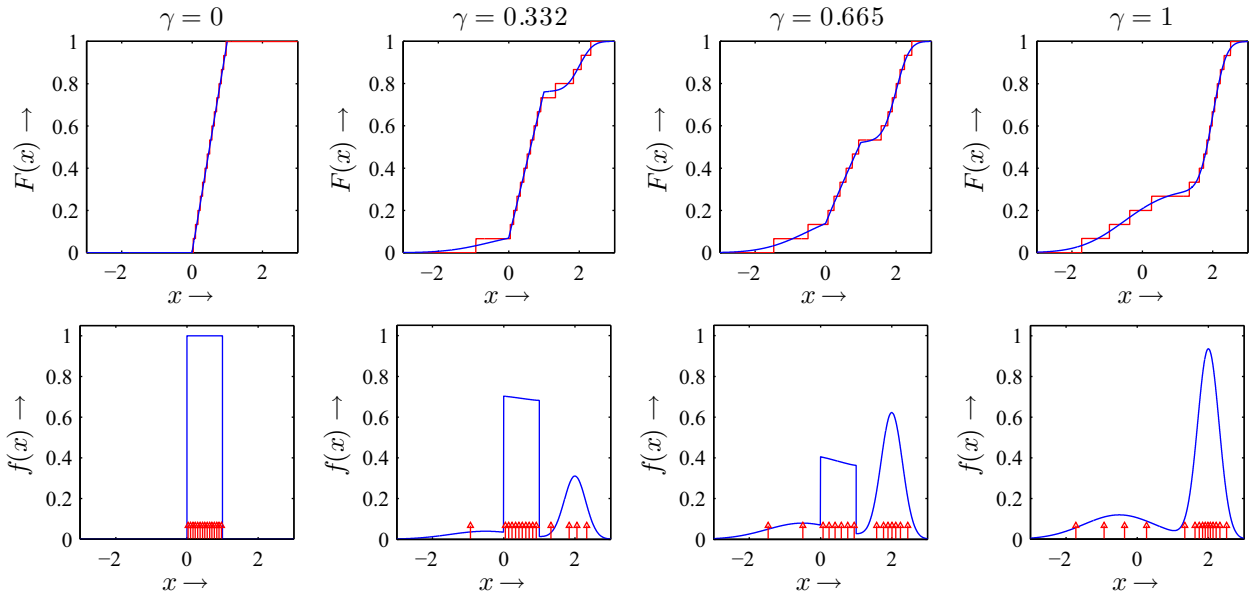
Daraus ergibt sich  $\underline{b}(\underline{\eta}(\gamma), \gamma)$  mit

$$\underline{b}(\underline{\eta}(\gamma), \gamma) = \begin{bmatrix} S(x_1(\gamma)) - \tilde{F}(x_1(\gamma)) \\ S(x_2(\gamma)) - \tilde{F}(x_2(\gamma)) \\ \vdots \\ S(x_L(\gamma)) - \tilde{F}(x_L(\gamma)) \end{bmatrix}.$$

#### Beispiel 4.4: Progressive Approximation mit festen Gewichten (Fortsetzung)

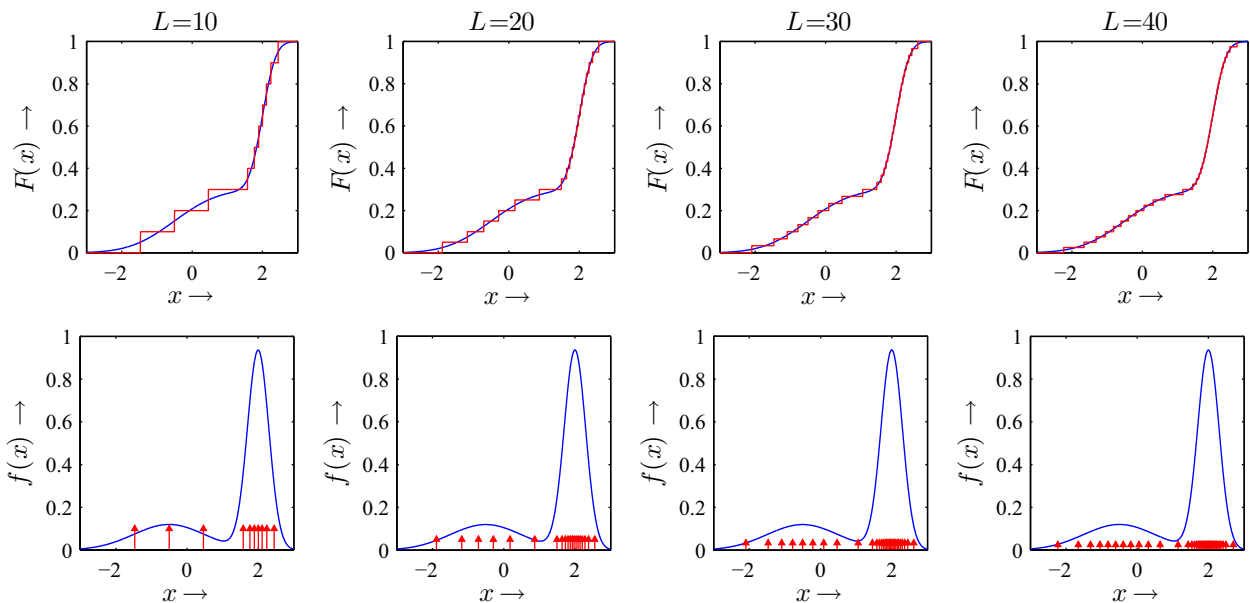
Abbildung 4.5 zeigt nun die Lösung des Differentialgleichungssystems (DGL) (4.19) für  $L = 15$  Dirac Komponenten. Die Höhe der Dirac-Komponenten entspricht der Gewichtung. Das DGL-System kann mit einem beliebigen numerischen Lösungsverfahren wie einem impliziten Eulerverfahren [40] oder Runge-Kutta Methoden [26] gelöst werden. Ein speziell auf das hier vorliegende Problem zugeschnittener Löser wird in Abschnitt 4.5.3 vorgestellt.

Da  $\tilde{f}(x, \gamma)$  eine Gleichverteilung ist wird als Startlösung  $\underline{\eta}(\gamma = 0)$  für die Progression der Parametervektor aus (4.17), also die geschlossene Lösung für die Gleichverteilung, verwendet.



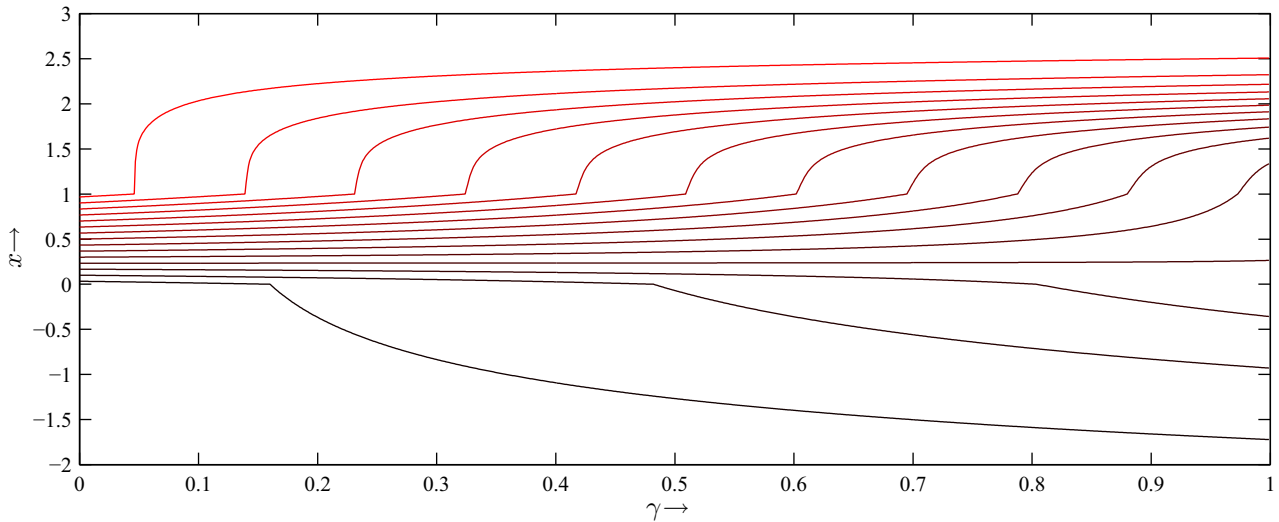
**Abbildung 4.5:** Progressive Approximation einer Gaussian Mixture Dichte (blau) mit zwei Komponenten für  $\gamma = 0 \dots 1$ . Die Approximationsdichte (rot) besteht aus  $L = 15$  Komponenten. Die Höhe der Dirac-Komponenten entspricht der Gewichtung. In der oberen Reihe ist der Progressionsverlauf in der Verteilung und in der unteren Reihe in der Dichtefunktion gezeigt.

Abbildung 4.6 zeigt das Endergebnis (für  $\gamma = 1$ ) der Approximation für  $L = 10, L = 20, L = 30$  und  $L = 40$  Dirac-Komponenten. Die unterschiedlichen Höhen der Dirac-Impulse resultieren aus den unterschiedlichen Komponentenzahlen, da die Gewichte mit  $\omega_i = \frac{1}{L}$  für  $i = 1, \dots, L$  festgelegt sind.



**Abbildung 4.6:** Approximation einer Gaussian-Mixture-Dichte durch Dirac-Mixtures mit  $L = 10, L = 20, L = 30$  und  $L = 40$  Komponenten.

Für eine feste Anzahl von Dirac-Komponenten lässt sich der Verlauf der Positionen über die Progression sehr gut verfolgen. Abbildung 4.7 zeigt den Verlauf der Positionen für  $L = 15$  Dirac-Komponenten für  $\gamma = [0, 1]$ . Es handelt sich hier also um die Funktion  $\underline{\eta}(\gamma)$ .



**Abbildung 4.7:** Verlauf des Parametervektors  $\underline{\eta}(\gamma)$  mit  $L = 15$  Komponenten. Jede Linie im Plot stellt den Verlauf einer Komponente dar.

■

Wie bei der geschlossenen löslichen Approximation mit festen Gewichten sieht man auch hier, wie sich die Positionen der Dirac-Mixture-Funktion im Bereich der größten Wahrscheinlichkeitsmasse der wahren Dichte drängen. Da es sich um eine global optimale Approximation handelt, werden die Bereiche mit geringerer Wahrscheinlichkeitsmasse aber nicht ausgespart.

### 4.5.3 Vollständige Optimierung

Die beste Approximationsgüte erreicht man, wenn der vollständige Parametervektor

$$\underline{\eta} = \left[ \omega_1 \quad \dots \quad \omega_L \quad x_1 \quad \dots \quad x_L \right]^T$$

also Positionen und Gewichte gleichzeitig, optimiert wird. Die folgenden Ergebnisse wurden auf der *Fusion 2006* [94] veröffentlicht.

Auch bei der vollständigen Optimierung soll wieder ein Gütemaß zwischen den Verteilungsfunktionen der wahren Dichte  $\tilde{F}(x)$  und der Approximationsdichte  $F(x, \underline{\eta})$  minimiert werden. Da nun aber Positionen und Gewichte optimiert werden sollen, muss für die notwendige Bedingung nach dem vollen Parametervektor abgeleitet werden. Da ferner hier keine geschlossene Lösung mehr möglich ist, wird wieder der progressive Ansatz, wie in Abschnitt 4.5.2 beschrieben,

verwendet. Es soll folglich gelten, dass

$$\frac{\partial G(\underline{\eta}, \gamma)}{\partial \underline{\eta}} = \begin{bmatrix} \frac{\partial G(\underline{\eta}, \gamma)}{\partial \omega} \\ \frac{\partial G(\underline{\eta}, \gamma)}{\partial x} \end{bmatrix} \stackrel{!}{=} 0 .$$

Im Folgenden wird wieder die Cramér-von Mises Distanz (4.8) als Gütemaß betrachtet. Die entsprechenden Ableitungen

$$\begin{aligned} \frac{\partial G(\underline{\eta}, \gamma)}{\partial \omega_i} &= \int_{-\infty}^{\infty} r(x) \left[ \tilde{F}(x, \gamma) - F(x) \right] H(x - x_i) dx \\ \frac{\partial G(\underline{\eta}, \gamma)}{\partial x_i} &= -\omega_i r(x_i) \left[ \tilde{F}(x, \gamma) - F(x_i) \right] \end{aligned}$$

wurden bereits in Abschnitt 4.5.1 bzw. 4.5.2 berechnet.

Durch Nullsetzen, Einsetzen der Gewichtungsfunktion (4.10) und Vereinfachen erhält man das Gleichungssystem bestehend aus

$$\int_{x_i}^c \tilde{F}(x, \gamma) dx = c \sum_{j=1}^L \omega_j - x_i \sum_{j=1}^i \omega_j - \sum_{j=i+1}^L \omega_j x_j \quad (4.20)$$

und

$$\tilde{F}(x_i, \gamma) = \sum_{j=1}^{i-1} \omega_j + \frac{\omega_i}{2} . \quad (4.21)$$

für  $i = 1, \dots, L$ . Diese beiden Bedingungen erlauben im Übrigen eine recht anschauliche graphische Interpretation. So bedeutet (4.20), dass die Flächen, die zwischen der Verteilungsfunktion der wahren Dichte und der Treppenfunktion entstehen, klein sein müssen. Dies ist auch in sofern klar, da diese Flächen den resultierenden Approximationsfehler beschreiben. Aus Bedingung (4.21) folgt, dass die Verteilungsfunktion der wahren Dichte immer durch die Mitten der Sprünge der Treppenfunktion gehen soll. Diese Bedingung gewährleistet die Formtreue in der Verteilungsfunktion.

Um das Minimum des Gütemaßes während der Progression zu verfolgen, wird die partielle Ableitung nach  $\gamma$  berechnet. Es sei an dieser Stelle noch einmal darauf hingewiesen, dass  $\omega_i = \omega_i(\gamma)$  und  $x_i = x_i(\gamma)$  implizit von  $\gamma$  abhängen, was bei der Ableitung berücksichtigt werden muss. Somit ergibt sich für die Ableitung von (4.20) nach  $\gamma$

$$\int_{x_i}^c \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx = c \sum_{j=1}^L \dot{\omega}_j - x_i \sum_{j=1}^i \dot{\omega}_j - \sum_{j=i+1}^L x_j \dot{\omega}_j + \tilde{F}(x_i, \gamma) \dot{x}_i - \dot{x}_i \sum_{j=1}^i \omega_j - \sum_{j=i+1}^L \omega_j \dot{x}_j$$

und für die Ableitung von (4.21) nach  $\gamma$

$$\frac{\partial \tilde{F}(x_i, \gamma)}{\partial \gamma} + \tilde{f}(x_i, \gamma) \dot{x}_i = \sum_{j=1}^{i-1} \dot{\omega}_j + \frac{\dot{\omega}_i}{2}$$

jeweils für  $i = 1, \dots, L$ .

Das resultierende System gewöhnlicher Differentialgleichungen lässt sich wieder in Vektor-Matrix Form als

$$\underline{b} = \mathbf{P}\underline{\dot{\eta}} , \tag{4.22}$$

mit

$$\underline{b} = \begin{bmatrix} \int_{x_1}^c \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx \\ \vdots \\ \int_{x_L}^c \frac{\partial \tilde{F}(x, \gamma)}{\partial \gamma} dx \\ \frac{\partial \tilde{F}(x_1, \gamma)}{\partial \gamma} \\ \vdots \\ \frac{\partial \tilde{F}(x_L, \gamma)}{\partial \gamma} \end{bmatrix}$$

und

$$\underline{\dot{\eta}} = \begin{bmatrix} \dot{\omega}_1 & \dots & \dot{\omega}_L & \dot{x}_1 & \dots & \dot{x}_L \end{bmatrix}^T$$

schreiben. Die Matrix  $\mathbf{P}$  ergibt sich zu

$\mathbf{P} =$

$$\begin{bmatrix} c - x_1 & c - x_2 & c - x_3 & \dots & c - x_L & -\omega_1 & -\omega_2 & \dots & -\omega_L \\ c - x_2 & c - x_2 & c - x_3 & \dots & c - x_L & \tilde{F}(x_1, \gamma) - \omega_1 & -\omega_2 & \dots & -\omega_L \\ c - x_3 & c - x_3 & c - x_3 & \dots & c - x_L & 0 & \tilde{F}(x_2, \gamma) - \sum_{i=1}^2 \omega_i & \dots & -\omega_L \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c - x_L & c - x_L & c - x_L & \dots & c - x_L & 0 & 0 & \dots & \tilde{F}(x_L, \gamma) - \sum_{i=1}^L \omega_i \\ \frac{1}{2} & 0 & 0 & \dots & 0 & -\tilde{f}(x_1, \gamma) & 0 & \dots & 0 \\ 1 & \frac{1}{2} & 0 & \dots & 0 & 0 & -\tilde{f}(x_2, \gamma) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & \frac{1}{2} & 0 & 0 & \dots & -\tilde{f}(x_L, \gamma) \end{bmatrix}$$

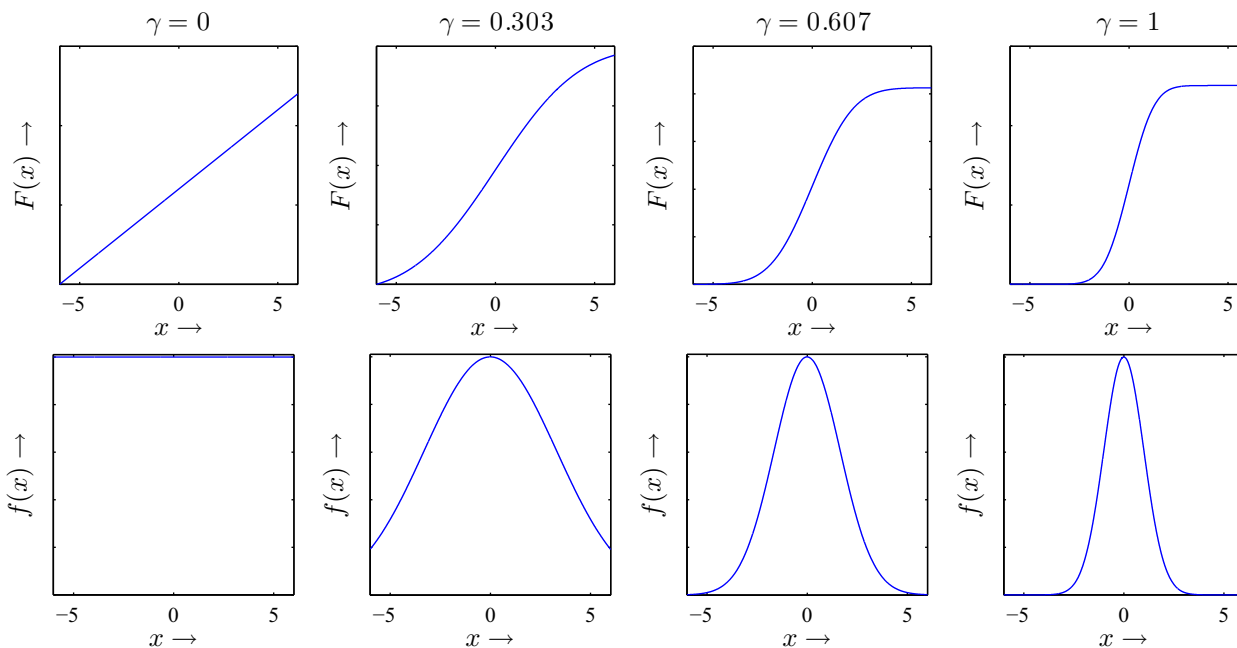
### Progression

Für die Lösung des Approximationsproblems soll ein anderer Progressionsablauf als der in Abschnitt 4.5.2 vorgestellte lineare Übergang verwendet werden. Ein Problem dieses Progressionsablaufs können die entstehenden Unstetigkeiten in der Dichtefunktion darstellen. Um diesem Problem entgegenzuwirken sollte der Progressionsablauf Unstetigkeiten vermeiden<sup>1</sup>. Ein solcher Verlauf lässt sich beispielsweise sehr einfach für Gaussdichten durch

$$\tilde{f}(x, \gamma) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left\{ -\frac{1}{2} \frac{x^2}{\left(\frac{1+\epsilon}{\gamma+\epsilon}\sigma\right)^2} \right\} \tag{4.23}$$

erzeugen, wobei  $\epsilon$  eine sehr kleine Konstante ist. Für  $\gamma = 0$  erhält man somit eine Gleichverteilung, für die eine geschlossene Lösung bekannt ist und die als Startlösung für die Progression eingesetzt werden kann.

<sup>1</sup> Es sei denn die wahre Dichte ist selbst unstetig!



**Abbildung 4.8:** Progressionsablauf für eine Gaussdichte (unten) und entsprechende Verteilungsfunktion (oben). Für  $\gamma = 0$  ergibt sich eine Gleichverteilung, für  $\gamma = 1$  die wahre Gaussdichte.

Der Verlauf dieser Progression ist in Abbildung 4.8 zu sehen. Dieser Progressionsablauf ist im Übrigen auch für Gaussian-Mixture-Dichten anwendbar, indem jede Komponente einzeln wie gezeigt parametrisiert wird.

### Löser

Da das System von Differentialgleichungen (4.22), wie bereits erwähnt, analytisch nicht lösbar ist, werden numerische Verfahren benötigt. Hier wird nun ein entsprechender Algorithmus vorgestellt, der auf das zu lösende Problem angepasst ist. Der Ansatz ist sehr effizient, da eine adaptive Schrittweitenregelung beim Lösen verwendet wird, die gleichzeitig auf die Einhaltung gewisser Fehlergrenzen achtet. Diese Fehlerüberwachung wird durch ein so genanntes Prädiktor-Korrektor Schema, auch bekannt als Adams-Bashforth-Moulton-Methode, realisiert. Dieses Verfahren wird im Folgenden erläutert und ist als Pseudocode in Algorithmus 4.1 angegeben.

Der Algorithmus beginnt bei  $\gamma = 0$ . Während des Lösungsprozesses wird  $\gamma$  sukzessive erhöht und der Parametervektor  $\underline{\eta}$  entsprechend angepasst. Da das Minimum von  $G(\underline{\eta}, \gamma)$  für jedes  $\gamma$  gesucht wird, muss  $\underline{\eta}$  so initialisiert werden, dass  $\underline{\eta}_0 = \arg_{\underline{\eta}} \min G(\underline{\eta}, \gamma = 0)$  gilt. Aufgrund der geschickten Parametrisierung von  $G(\underline{\eta})$  durch  $\gamma$  ist es im Allgemeinen möglich, eine solche Startlösung, wie bereits in Abschnitt 4.5.2 gezeigt, anzugeben.

In Zeile 3 des Algorithmus wird eine initiale Schrittweite  $\Delta\gamma$  für den Algorithmus festgelegt, die dann im Laufe des Verfahrens variiert wird.

**Algorithmus 4.1** Prädiktor–Korrektor Methode mit Schrittweitenregelung

---

```

1:  $\gamma := 0$ 
2:  $\underline{\eta}_0 := \underline{\eta}(\gamma = 0)$ 
3:  $\Delta\gamma := \gamma_{\text{step\_min}}$ 
4: repeat
5:    $\gamma := \gamma + \Delta\gamma$ 
6:    $\underline{\eta}_{\text{tmp}} := \text{Prediktor}(\underline{\eta}, \gamma)$ 
7:    $[\underline{\eta}_{\text{tmp}}, \text{success}] := \text{Corrector}(\underline{\eta}_{\text{tmp}})$ 
8:   if success then
9:      $\underline{\eta} := \underline{\eta}_{\text{tmp}}$ 
10:    Increase( $\Delta\gamma, \gamma_{\text{step\_max}}$ )
11:  else
12:     $\gamma := \gamma - \Delta\gamma$ 
13:    Decrease( $\Delta\gamma, \gamma_{\text{step\_min}}$ )
14:  end if
15: until  $\gamma == 1$ 

```

---

Der Hauptteil des Algorithmus besteht aus einer Schleife, in der  $\gamma$  sukzessive um  $\Delta\gamma$  erhöht wird bis  $\gamma = 1$  erreicht ist. In jedem Schleifendurchlauf wird zuerst ein Parametervektor  $\underline{\eta}_{\text{tmp}}$  für das zuvor erhöhte  $\gamma$  prädiziert. Die entsprechende Prädiktormethode wird im nächsten Abschnitt beschrieben.

Nach jedem Prädiktorschritt wird ein Korrektorschritt auf den prädizierten Parametervektor  $\underline{\eta}_{\text{tmp}}$  berechnet, um den Fehler des Prädiktors zu minimieren. Wenn der Korrektor schnell konvergiert, lässt sich daraus folgern, dass der Prädiktor nur einen kleinen Fehler verursacht hat. Daher kann die Schrittweite für den nächsten Schleifendurchlauf erhöht werden. Wenn der Korrektor jedoch nicht konvergiert, so war der Fehler des Prädiktors zu groß. Aus diesem Grund muss der letzte Prädiktorschritt rückgängig gemacht und die Schrittweite verringert werden. Die Korrektormethode wird im Anschluss an den Prädiktor beschrieben.

**Prädiktor** Der Prädiktor löst zuerst das *Gleichungssystem* (4.22) für  $\underline{\eta}$ . Das heißt, dass ein lineares Gleichungssystem für gegebenes  $\underline{\eta}$  und  $\gamma$  gelöst wird.

Der prädizierte Parametervektor wird dann berechnet durch

$$\underline{\eta}_{\text{tmp}} = \underline{\eta} + \Delta\gamma \underline{\dot{\eta}} .$$

Da  $\underline{\dot{\eta}}$  nur eine Richtung angibt wird hierbei ein Fehler erzeugt, der stark von der Schrittweite  $\Delta\gamma$  abhängt. Dieser Fehler muss im Korrektorschritt verringert werden.

**Korrektor** Der Korrektorschritt basiert auf dem Newtonverfahren zur Nullstellensuche. Gemäß diesem Ansatz wird die Gleichung

$$\mathbf{J}(\underline{\eta}_k) \underbrace{[\underline{\eta}_{k+1} - \underline{\eta}_k]}_{\Delta \underline{\eta}} = -\underline{h}(\underline{\eta}_k) \quad (4.24)$$

verwendet, wobei

$$\underline{h}(\underline{\eta}) = \frac{\partial G(\underline{\eta}, \gamma)}{\partial \underline{\eta}} = [h_1^{(1)}, \dots, h_1^{(L)}, h_2^{(0)}, \dots, h_2^{(L)}]^T$$

mit

$$h_1^{(i)}(\underline{\eta}) = \int_{x_i}^c \tilde{F}(x, \gamma) dx - c \sum_{j=1}^L \omega_j - x_i \sum_{j=1}^i \omega_j - \sum_{j=i+1}^L \omega_j x_j$$

und

$$h_2^{(i)}(\underline{\eta}) = \tilde{F}(x_i, \gamma) - \sum_{j=0}^{i-1} \omega_j - \frac{\omega_i}{2}$$

ist. Es ist zu beachten, dass  $\gamma$  in diesem Schritt konstant ist!

Die Jakobimatrix  $\mathbf{J}(\underline{\eta}_k)$  ist gegeben durch

$$\mathbf{J}(\underline{\eta}_k) = \frac{\partial \underline{h}(\underline{\eta})}{\partial \underline{\eta}^T} = \frac{\partial^2 G(\underline{\eta}, \gamma)}{\partial \underline{\eta} \partial \underline{\eta}^T}.$$

Die Gleichung (4.24) wird nun für  $\Delta \underline{\eta}$  gelöst, um einen rekursiven Ansatz

$$\underline{\eta}_{k+1} = \underline{\eta}_k - \Delta \underline{\eta}$$

zu erhalten. Wenn der initiale Parametervektor  $\underline{\eta}_0$  für diese Rekursion nahe dem wahren Parametervektor ist, dann konvergiert dieses Verfahren sehr schnell. Die Konvergenz kann festgestellt werden, wenn  $\Delta \underline{\eta} \rightarrow 0$  geht. In diesem Fall liefert die Methode *success* und den konvergierten Parametervektor  $\underline{\eta}$  zurück. Sollte die Rekursion nicht in wenigen Schritten konvergieren, so wird das Verfahren abgebrochen und kein Erfolg zurückgegeben.

### Beispiel 4.5: Approximation einer unnormierten Gaussdichte mit freien Gewichten und Positionen

In diesem Beispiel wird die progressive Approximation einer Standardnormalverteilung ohne Normierungsfaktor

$$\tilde{f}(x, \gamma) = \exp \left\{ -\frac{1}{2} \frac{x^2}{\left( \frac{1+\epsilon}{\gamma+\epsilon} \right)^2} \right\}$$

mit freien Gewichten und Positionen gezeigt. Dabei wird der Progressionsablauf aus (4.23) verwendet, dessen Verlauf in Abbildung 4.8 gezeigt ist.

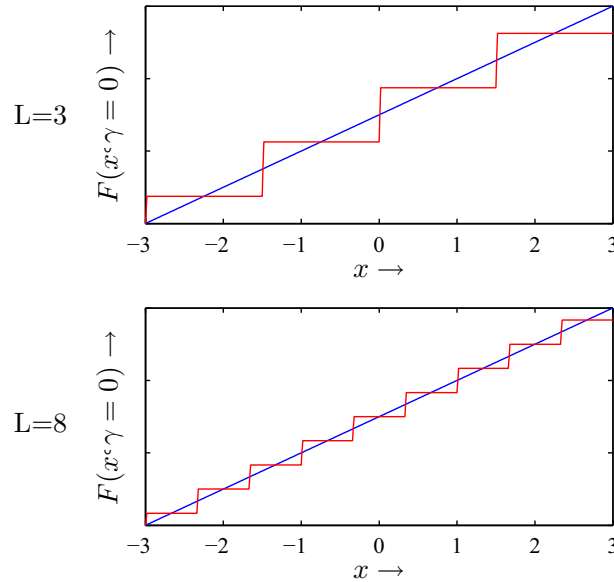
Da  $\tilde{f}(x, \gamma = 0) = 1$  eine Gleichverteilung mit  $\tilde{F}(x, \gamma = 0) = x$  ist, kann (4.17) grundsätzlich als Startlösung verwendet werden. Da  $\tilde{F}(-c, \gamma = 0)$  jedoch nicht Null ist, wird noch ein Kompensationsgewicht  $\omega_0$  an der Stelle  $x = -c$  benötigt. Für  $L$  Dirac-Komponenten ist eine gültige Startlösung



somit

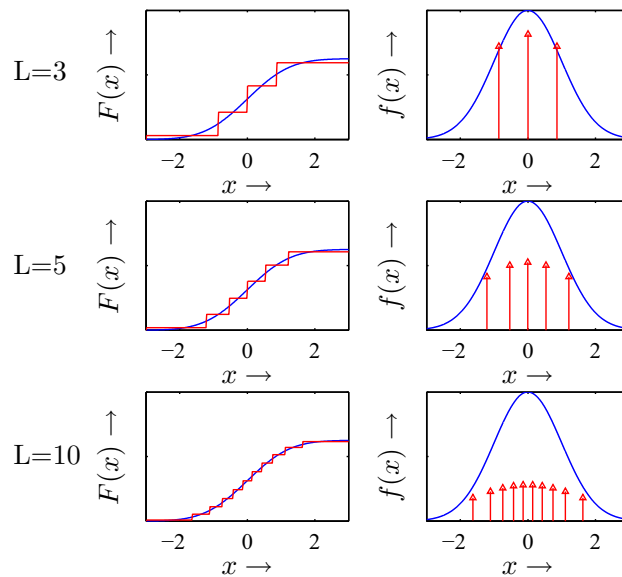
$$x_i = \frac{2ci}{L+1}, \omega_0 = \frac{c}{L+1}, \omega_i = \frac{2c}{L+1} \quad (4.25)$$

für  $i = 1, \dots, L$ . Für diese Parameter gilt, unabhängig von der Anzahl der Komponenten,  $G(\underline{\eta}, \gamma) = 0$ . Abbildung 4.9 zeigt diese Startlösung für  $L = 3$  und  $L = 8$  Komponenten.



**Abbildung 4.9:** Startlösung für Approximation einer Gaussdichte mit  $L = 3$  und  $L = 8$  Komponenten. In blau ist jeweils die wahre Verteilung und in rot die Approximation gegeben.

Die resultierende Approximation für  $L = 3, L = 5$  und  $L = 10$  Komponenten ist in Abbildung 4.10 zu sehen. Der Löser benötigte für die Berechnung des Differentialgleichungssystems 14, 25 bzw. 49 Schritte.



**Abbildung 4.10:** Endergebnis der Approximation einer Gaussdichte mit  $L = 3, L = 5$  und  $L = 10$  Komponenten. Die Verteilung ist in blau und die Approximation gegeben in rot gegeben.



Zum Abschluss wird nun ein Beispiel mit einer komplexeren wahren Dichte betrachtet.

**Beispiel 4.6: Approximation einer Gaussian-Mixture-Dichte mit freien Gewichten und Positionen**

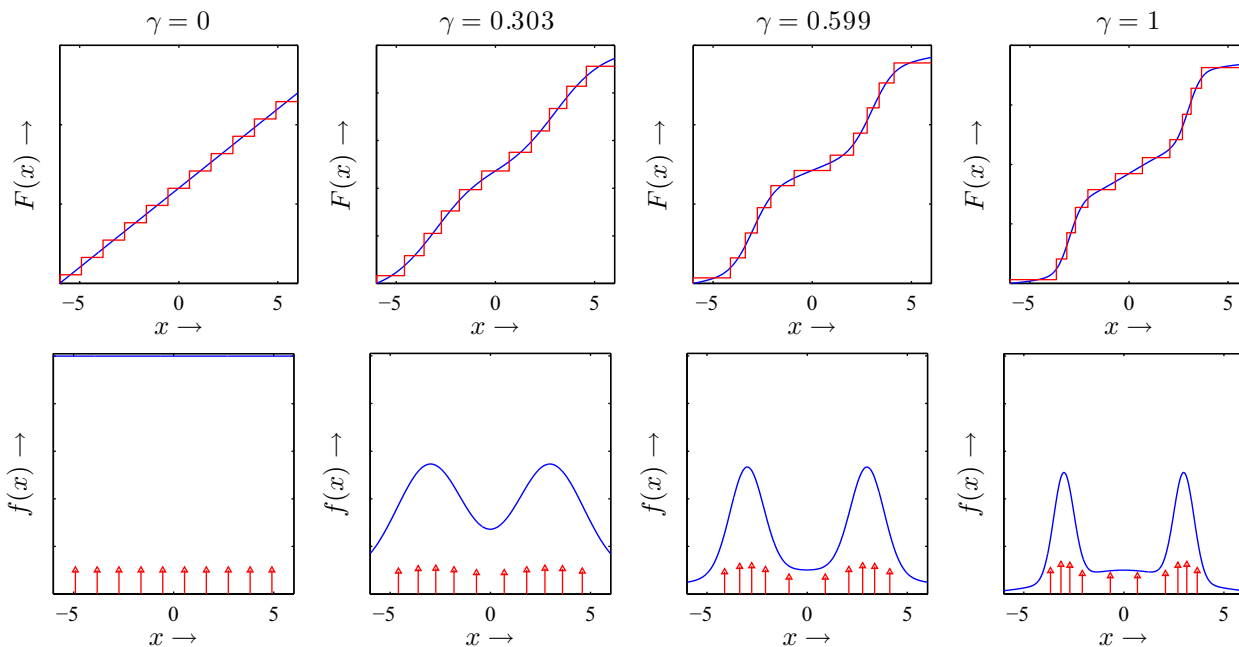
In diesem Beispiel wird eine Gaussian-Mixture-Dichte

$$\tilde{f}(x, \gamma) = \sum_{j=1}^M \alpha_j \mathcal{N}(x, \mu_j, \sigma_j, \gamma)$$

mit  $M = 3$  Komponenten approximiert. Die Parameter dieser Dichte sind gegeben durch

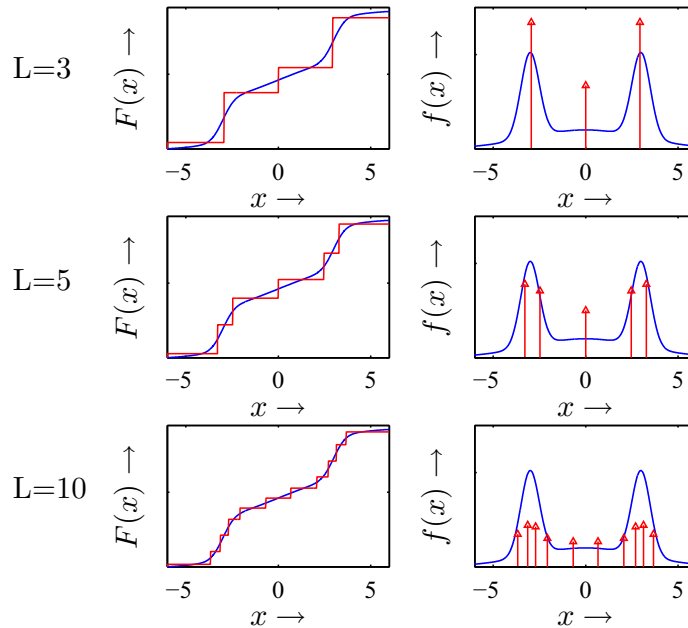
$$\begin{aligned} \underline{\alpha} &= [0.45, 0.1, 0.45]^T \\ \underline{\mu} &= [-3, 0, 3]^T \\ \underline{\sigma} &= [0.5, 3, 0.5]^T \end{aligned}$$

als Gewichte, Mittelwerte und Standardabweichung der jeweiligen Gausskomponenten. Die Parametrisierung durch  $\gamma$  erfolgt in jeder Komponente wie in (4.23) gezeigt. Der Progressionsablauf in Dichte und Verteilung zusammen mit der optimalen Approximation für den jeweiligen  $\gamma$ -Wert für  $L = 10$  Dirac-Komponenten ist in Abbildung 4.11 gezeigt.



**Abbildung 4.11:** Progressive Approximation einer Gaussian-Mixture mit 3 Komponenten für  $\gamma = 0 \dots 1$ . Die Approximationsdichte besteht aus  $L = 10$  Dirac-Komponenten.

Die fertige Approximation für  $L = 3$ ,  $L = 5$  und  $L = 10$  Dirac-Komponenten ist in Abbildung 4.12 gezeigt.



**Abbildung 4.12:** Approximationsergebnis für eine Gaussian-Mixture mit drei Komponenten durch eine Dirac-Mixture mit  $L = 3$ ,  $L = 5$  und  $L = 10$  Komponenten.

Der Löser benötigte hier 42, 66 bzw. 111 Schritte zur Berechnung des Ergebnisses. ■

## 4.6 Dynamisches System als Kaskade

In diesem Abschnitt wird nun als Ergebnis der Anwendung der Approximation durch Dirac-Mixture-Dichten ein Prädiktor für nichtlineare wertekontinuierliche stochastische Systeme vorgestellt. Diese Ergebnisse wurden auf der *ACC 2007* [100] veröffentlicht. Dabei wird gezeigt, wie eine rekursive Prädiktion als Kaskade zeitdiskreter Systemfunktionen angesehen werden kann. In Anlehnung an die Literatur zu dynamische Filtern wird die Bayes'sche Vorwärtsinferenz hier auch als Prädiktionsschritt bezeichnet. Dieser Abschnitt enthält unter anderem Überlegungen zur Bestimmung der optimalen Anzahl von Komponenten für die Approximationsdichte sowie zu der Frage an welcher Stelle im Zustandsraum eines komplexen Gesamtsystems approximiert werden soll. Insbesondere dieser letzte Punkt ist im Hinblick auf kaskadierte Systeme von entscheidender Bedeutung. Ferner werden Vergleiche mit „State-of-the-art“-Verfahren angestellt.

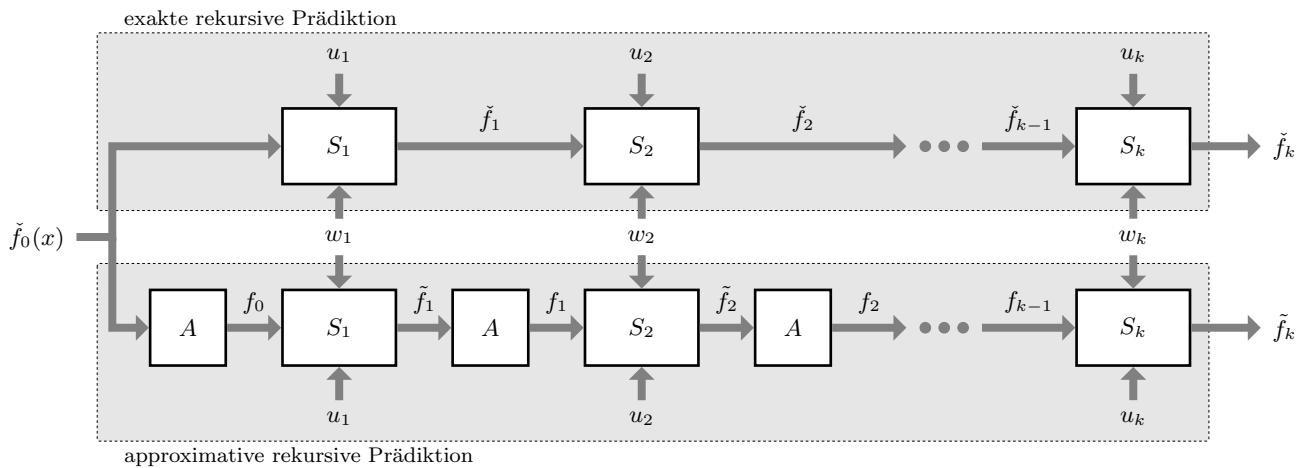
### 4.6.1 Prädiktor

In diesem Abschnitt wird die rekursive Fortpflanzung einer Wahrscheinlichkeitsdichte durch ein zeitdiskretes stochastisches nichtlineares dynamisches System

$$x_{k+1} = a_k(x_k, u_k, w_k)$$

betrachtet, wobei  $x_k$  für den Systemzustand,  $u_k$  für einen bekannten Eingang und  $a_k(\cdot)$  für eine zeitabhängige nichtlineare Abbildung stehen.  $w_k$  bezeichnet endogene wie exogene Rauschquellen, die das System beeinflussen und wird beschrieben durch eine Dichtefunktion  $f_k^w(w_k)$ . Da es

sich um ein zeitdiskretes System handelt, kann es auch als ein kaskadiertes System betrachtet werden, da der Ausgang in jedem Zeitschritt als Eingang für den nächsten Zeitschritt fungiert. Ausgehend von einer Dichte  $f_0^x(x_0)$  über den initialen Systemzustand  $x_0$  besteht das Ziel darin, die Entwicklung des Systems über einen endlichen Zeithorizont  $k = 1, \dots, N$ , für gegebene Eingänge  $u_k$ ,  $k = 0, \dots, N - 1$  zu berechnen. Dies entspricht der sequentiellen Berechnung der Wahrscheinlichkeitsdichten für  $f_k^x(x_k)$ ,  $k = 1, \dots, N - 1$  des Systemzustands und kann, wie im oberen Teil von Abbildung 4.13 zu sehen, als Kaskade von Systemen betrachtet werden.



**Abbildung 4.13:** Darstellung einer rekursiven Prädiktion als kaskadiertes System. Der obere Teil entspricht der sukzessiven exakten Bayes'schen Prädiktion. Die Dichten  $\tilde{f}$  bezeichnen die resultierenden exakten Dichten. Der untere Teil entspricht der approximativen rekursiven Prädiktion, die durch Approximation der Posteriors nach jedem Prädiktionsschritt entsteht.  $f$  bezeichnet approximierte Dichten.  $\tilde{f}$  bezeichnet Dichten, die aus einem exakten Prädiktionsschritt mit einer approximierten Prioren entstehen.

Verfahren aus der Literatur zur Behandlung solcher Systeme, wie beispielsweise Particle Filter, wurden bereits am Anfang des Kapitels vorgestellt. Die Idee, die hier nun verfolgt wird besteht darin, die Vorwärtsinferenz, beschrieben in Abschnitt 4.3.1, rekursiv anzuwenden, gefolgt von Approximationsschritten. Das generelle Vorgehen ist im unteren Teil von Abbildung 4.13 gezeigt.

Abbildung 4.13 zeigt den Vergleich der exakten Bayes'schen Prädiktion mit der approximativen Prädiktion. Da die exakte Berechnung, wie im oberen Teil gezeigt, im Allgemeinen nicht berechenbar ist, wird ein Approximationsschritt benötigt. Der Fokus liegt hier natürlich auf der Approximation durch Dirac-Mixture-Dichten. Wie der untere Teil der Abbildung zeigt, soll jeweils die posteriore Dichte aus einem Prädiktionsschritt durch eine Dirac-Mixture-Dichte approximiert werden, die dann als Priore für den nächsten Prädiktionsschritt dient. Alle Dichten  $f_k$  in Abbildung 4.13 sind somit Dirac-Mixtures.

Aufgrund des Systemrauschens  $w_k$  ist die Posteriore des Prädiktionsschritts im Allgemeinen von einem anderen Typ als die Priore. Dies ist durch die Bezeichnung  $\tilde{f}_k$  gekennzeichnet. Es handelt sich hier also normalerweise nicht um eine Dirac-Mixture. Daher muss die Posteriore optimal reapproximiert werden, um die Dichte weiterpropagieren zu können.

### Spezialfall additives Gauss'sches Rauschen

Im weiteren Verlauf wird nun der wichtige Spezialfall des additiven Gauss'schen Rauschens betrachtet. Das System lässt sich in diesem Fall beschreiben durch

$$x_{k+1} = a_k(x_k, u_k) + w_k ,$$

wobei

$$f_k^w(w_k) = \mathcal{N}(w_k, 0, \sigma_k^w)$$

mittelwertfrei und gaussverteilt ist. Das Bemerkenswerte an diesem, auch für die Praxis relevanten Fall ist, dass man durch Einsetzen des probabilistischen Modells

$$f(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}, a_k(x_k, u_k), \sigma_k^w)$$

in den Vorwärtsschritt

$$f_{k+1}^x(x_{k+1}) = \int_{-\infty}^{\infty} f(x_{k+1}|x_k) f_k^x(x_k) dx_k$$

bei  $f_k^x(x_k)$  geben durch eine Dirac-Mixture, als geschlossene Lösung eine Gaussian-Mixture-Dichte

$$f_{k+1}^x(x_{k+1}) = \sum_{i=1}^L \omega_i \mathcal{N}(x_{k+1}, a_k(x_i, u_k), \sigma_k^w)$$

erhält. Hierbei sind  $x_i$  für  $i = 1, \dots, L$  die Positionen der Dirac-Funktionen zum Zeitpunkt  $k$ .

Die Approximation einer Gaussian-Mixture wurde bereits in Beispiel 4.6 gezeigt. Somit lässt sich das Verfahren rekursiv, wie im unteren Teil von Abbildung 4.13 gezeigt, anwenden.

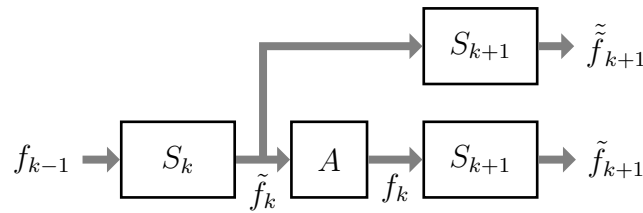
### Weitere Approximationskriterien

Bisher wurde bei der Approximation lediglich die priore Dichte für den folgenden Vorwärtsschritt aus der posterioren Dichte des vorangegangenen Vorwärtsschritts berechnet. Was dabei bisher außer Acht gelassen wurde, ist wie sich der Approximationsfehler, der in jedem Schritt entsteht, durch das kaskadierte System weiterentwickelt.

Es ist klar, dass für eine unendliche Anzahl von Dirac-Mixture-Komponenten die Dichten im System exakt approximiert werden können. Der Fehler der Approximation hängt somit in erster Linie von der Anzahl der verwendeten Komponenten ab.

Es werden nun verschiedene Strategien zur Bewertung des Effekts der Approximation einer prioren Dichte zum Zeitpunkt  $k$  vorgestellt. Diese Strategien konzentrieren sich auf die zukünftige Entwicklung des Systems und beschränken sich auf die Approximationsstruktur, d.h. die Positionen in der Kaskade von Einzelsystemen, an denen ein Vergleich zwischen einer wahren und einer aus der Approximation stammenden Dichte vorgenommen werden kann. Für diesen Vergleich werden entsprechende Gütemaße vorgeschlagen.

Zur Veranschaulichung der verschiedenen Möglichkeiten wird ein Ausschnitt aus der Gesamtkaskade in Abbildung 4.14 betrachtet der zwei Vorwärtsschritte umfasst.



**Abbildung 4.14:** Ausschnitt aus der in Abbildung 4.13 gezeigten Kaskade bestehend aus zwei Vorwärtsschritten und einer Approximation. Eingangs- und Rauschgröße wurden zur Vereinfachung weggelassen.

**Priore Optimierung** Der einfachste Optimierungsansatz ist der direkte Vergleich der wahren Dichte  $\tilde{f}_k$  mit der Approximation  $f_k$ . Dies entspricht im Wesentlichen dem bisher gezeigten Vorgehen. Was jedoch fehlt, ist die Berechnung der optimalen Komponentenzahl von  $f_k$ .

Dieser Ansatz ist sehr naheliegend, allerdings garantiert eine gute Approximation zum Zeitschritt  $k$  nicht, dass die zukünftige Entwicklung, die aus dieser Approximation resultiert, nahe der Entwicklung der wahren Dichte liegt.

**Vorausschau Optimierung** Ein wünschenswerterer Optimierungsansatz wäre eine Approximation im Zeitschritt  $k$  zu finden, so dass ein Gütemaß zwischen der wahren Dichte und der Fortschreibung der Approximation in der Zukunft des Systems minimiert wird. Idealerweise würde dies für eine sehr große oder gar unendliche Anzahl von Vorwärtsschritten ausgeführt. Aus Komplexitätsgründen ist eine solche Optimierung im Allgemeinen aber nicht durchführbar.

**Posteriore Optimierung** Eine praktikablere Lösung beschränkt sich auf ein paar wenige Vorwärtsschritte. Für einen einzelnen Vorwärtsschritt vergleicht man mittels Gütemaß die Dichten  $\tilde{f}_{k+1}$  und  $\tilde{\tilde{f}}_{k+1}$ .  $\tilde{\tilde{f}}_{k+1}$  ist die „exakte“ Dichte die aus der Approximation  $f_k$  durch den Vorwärtsschritt entsteht.  $\tilde{f}_{k+1}$  ist die exakte posteriore Dichte die aus der exakten prioren Dichte<sup>2</sup>  $\tilde{f}_k$  entsteht. Es wird also  $\tilde{f}_k$  so durch  $f_k$  approximiert, dass ein Gütemaß zwischen  $\tilde{f}_{k+1}$  und  $\tilde{\tilde{f}}_{k+1}$  minimiert wird. Dieser Ansatz ist wesentlich aufwändiger als der erste Ansatz, da ein exakter Vorwärtsschritt ohne Approximation gerechnet werden müsste. Allerdings wird hier die Entwicklung des Systems mitberücksichtigt.

**Priore Optimierung für beste Posteriore** Ein Optimierungsansatz, der mit den zur Verfügung stehenden Methoden sofort angewandt werden kann, stellt eine Erweiterung des ersten Ansatzes dar. Hier wird zunächst  $\tilde{f}_k$  durch  $f_k$  approximiert. Die Anzahl der Parameter von  $f_k$  wird dann jedoch durch einen Vergleich von  $\tilde{f}_{k+1}$  und  $\tilde{\tilde{f}}_{k+1}$  festgelegt. So wird die Entwicklung des Systems über einen Zeitschritt berücksichtigt.

Im Folgenden werden nun für den ersten und den letzten Ansatz praktische Algorithmen und Gütemaße vorgestellt.

<sup>2</sup> Dies ist die Dichte vor der Approximation

### Priore Optimierung

Zunächst wird ein Algorithmus zur Bestimmung der optimalen Anzahl von Komponenten für eine priore Dichte vorgestellt. Dabei kann das in Abschnitt 4.5.2 eingeführte Gütemaß  $G(\underline{\eta})$  aus (4.8) zum Vergleich der wahren Dichte und der Approximation verwendet werden. Das Vorgehen ist in Algorithmus 4.2 angegeben und wird im Folgenden näher erläutert.

---

**Algorithmus 4.2** Bestimmung der optimalen Anzahl der Komponenten für die Approximation der prioren Dichte.

---

```

1: Wähle maximalen Fehlerschwellwert  $G_{\max}$ 
2: Wähle initiale Anzahl von Komponenten  $L = L_0$ 
3: Wähle Suchschrittweite  $\Delta L$ 
4:  $\underline{\eta} = \text{approx}(L)$ 
5: while  $G(\underline{\eta}) > G_{\max}$  do
6:    $L = L + \Delta L$ 
7:    $\underline{\eta} = \text{approx}(L)$ 
8: end while
9:  $L_l = L - \Delta L$ 
10:  $L_u = L$ 
11: while  $L_u - L_l > 1$  do
12:    $L_t = L_l + \lfloor \frac{L_u - L_l}{2} \rfloor$ 
13:    $\underline{\eta} = \text{approx}(L_t)$ 
14:   if  $G(\underline{\eta}) > G_{\max}$  then
15:      $L_l = L_t$ 
16:   else
17:      $L_u = L_t$ 
18:   end if
19: end while

```

---

Zunächst werden zur Initialisierung des Verfahrens einige Werte festgelegt. Dies sind der maximale Approximationsfehler  $G_{\max}$ , eine initiale Anzahl von Komponenten, sowie eine Suchschrittweite  $\Delta L$ , gefolgt von einer initialen Approximation für die gewählte Anzahl von Komponenten.

Der Ansatz sucht nun in einer ersten Schleife eine Anzahl von Komponenten, für die die Approximation besser (kleiner) als der vorgegebene maximale Approximationsfehler ist. Bei dieser Suche wird die Anzahl der Komponenten jeweils um  $\Delta L$  erhöht.

Sobald der maximale Fehler unterschritten wurde, wird die Suche auf das letzte Intervall  $[L, L + \Delta L]$  begrenzt. Dabei wird mittels Intervallschachtelungsverfahren nach der kleinsten Anzahl von Komponenten gesucht, für die der Approximationsfehler kleiner als die Vorgabe ist.

Diese Art der Suche ist möglich, da das Approximationsverfahren deterministisch ist und der Approximationsfehler für steigende Anzahl von Komponenten monoton fallend ist. Dies wird im folgenden Beispiel verdeutlicht.

**Beispiel 4.7: Monotonie des Approximationsfehlers für steigende Anzahl von Komponenten.**

In diesem Beispiel wird die Monotonie des Approximationsfehlers für steigende Anzahl von Komponenten betrachtet und mit dem Approximationsfehler durch zufälliges Ziehen, wie bei Monte-Carlo-Methoden üblich, verglichen.

Als Beispiel wird eine Gaussian-Mixture-Dichte mit vier Komponenten und den Parametern

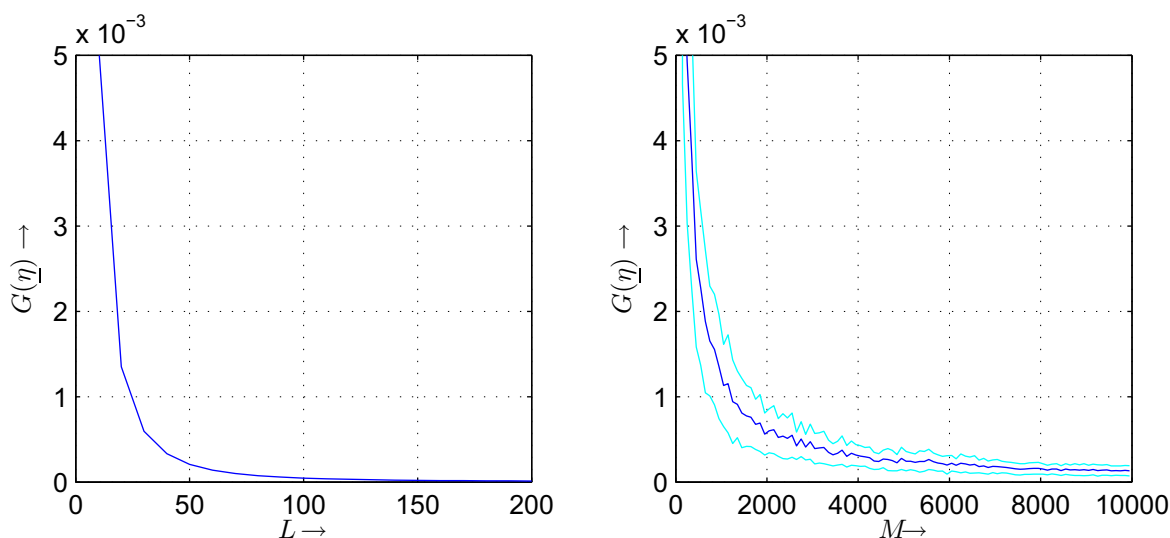
$$\underline{\alpha} = [0.3 \quad 0.3 \quad 0.2 \quad 0.2]^T$$

$$\underline{\mu} = [-3 \quad -1.5 \quad 0.5 \quad 2.5]^T$$

$$\underline{\sigma} = [0.8 \quad 0.4 \quad 1 \quad 1]^T$$

approximiert.

Abbildung 4.15 zeigt im linken Teil den Verlauf des Approximationsfehlers  $G(\eta)$  für die steigende Anzahl von Komponenten  $L$ .



**Abbildung 4.15:** Approximationsfehler  $G(\eta)$  für eine steigende Anzahl  $L$  von Komponenten in der Dirac-Mixture (links) und eine steigende Anzahl von Realisierungen  $M$  in einem zufallsbasierenden Ansatz (rechts). Der zufallsbasierte Ansatz wurde 200 mal wiederholt. Daher sind in hellblau  $\sigma$ -Grenzen angegeben.

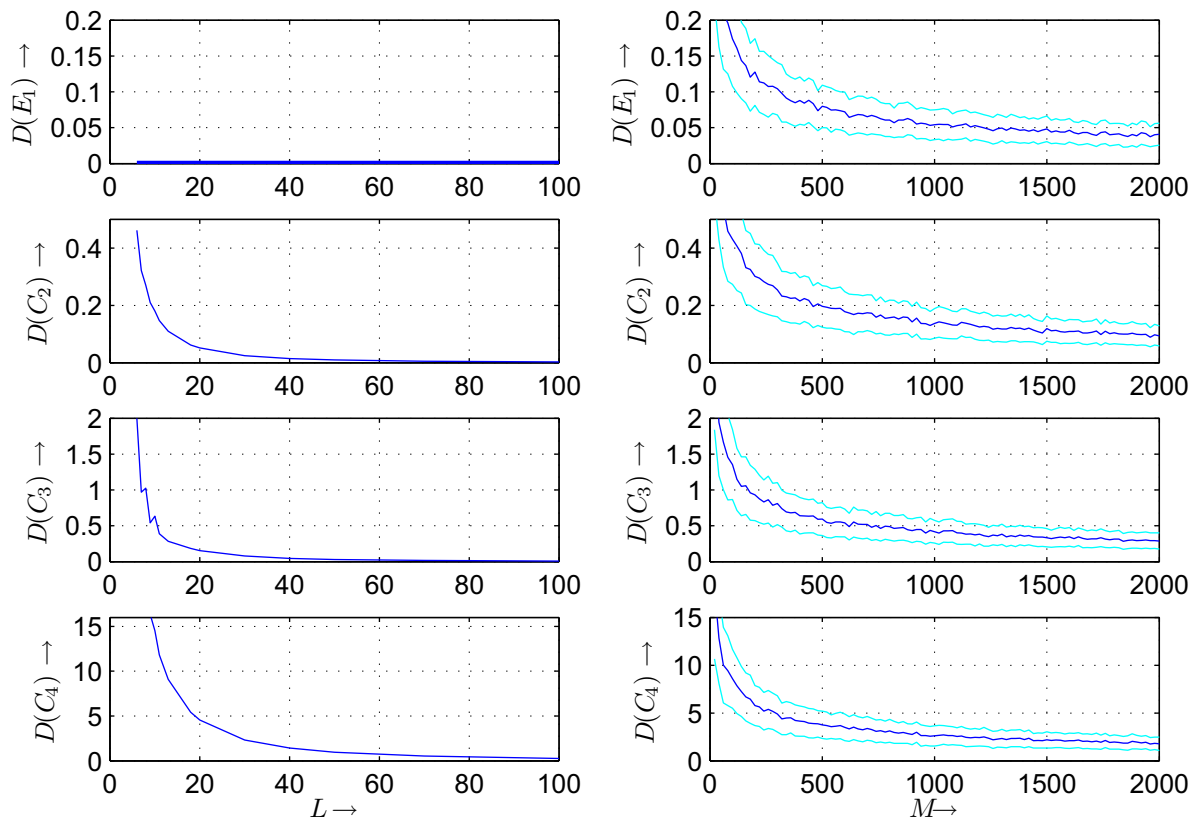
Zum Vergleich ist im rechten Teil von Abbildung 4.15 der Fehler für eine Darstellung der Dichte durch zufällige Realisierungen aus der Dichte<sup>3</sup> gezeigt. Um den statistischen Effekt dieser Methode zu berücksichtigen, wurde das zufällige Ziehen von  $M$  Realisierungen jeweils zweihundertmal durchgeführt. Der Plot zeigt in dunkel den mittleren Fehler und in hell die  $\sigma$ -Grenzen der Streuung.

Hieraus wird sehr gut deutlich, dass die Dirac-Mixture-Approximation deterministisch ist. D.h. die Parameter und der daraus resultierende Fehler sind im Gegensatz zum zufallsbasierten Ansatz immer identisch für eine bestimmte Anzahl von Komponenten. Ferner ist die Konvergenzrate viel höher, was an der Anzahl der verwendeten Komponenten in der  $L$ -Achse abzulesen ist.

<sup>3</sup> Auch Particles genannt



Dieses Verhalten ist auch für die Konvergenz der Momente, wie in Abbildung 4.16 gezeigt, zu beobachten.



**Abbildung 4.16:** Verlauf des quadratischen Fehlers des ersten Moments  $E_1$  sowie der zentralen Momente  $C_2, C_3$  und  $C_4$  für eine steigende Anzahl von Komponenten. Die linke Seite zeigt die Ergebnisse für eine Dirac-Mixture-Approximation, die rechte Seite die Ergebnisse für zufälliges Samplen aus der wahren Dichte.

Abbildung 4.16 zeigt den quadratischen Fehler

$$D(E_1) = |E_1^{\text{wahr}} - E_1^{\text{approx}}|$$

für das erste Moment  $E_1$  (Erwartungswert) sowie für die zentralen Momente  $C_2, C_3$  und  $C_4$ . Die Berechnung der jeweiligen Momente wurde bereits in Abschnitt 4.2 gezeigt.

Bemerkenswert ist hier, dass der Erwartungswert von Beginn an exakt approximiert wird und die Konvergenzrate auch für die zentralen Momente wesentlich höher ist als beim zufälligen Ziehen. ■

### Priore Optimierung für beste Posteriore

Die Approximation der prioren Dichte für eine optimale posteriore Dichte ist eine Erweiterung des zuvor vorgestellten Verfahrens. Hierbei sollen, wie in Abbildung 4.14 dargestellt, die posteriore Dichte  $\tilde{f}_{k+1}$  die aus der Approximation resultiert mit der wahren posterioren Dichte  $\tilde{\tilde{f}}_{k+1}$  verglichen werden. Das Verfahren ist in Algorithmus 4.3 als Pseudocode dargestellt.

**Algorithmus 4.3** Bestimmung der optimalen Anzahl der Komponenten für die Approximation der prioren Dichte im Hinblick auf eine optimale posteriore Dichte.

---

```

1: Wähle maximalen Fehlerschwellwert  $G_{\max}$ 
2: Wähle initiale Anzahl von Komponenten  $L = L_0$ 
3: Wähle Suchschrittweite  $\Delta L$ 
4:  $\underline{\kappa}_t = \text{predict}(\text{approx}(L_{\text{large}}))$ 
5:  $\underline{\kappa} = \text{predict}(\text{approx}(L))$ 
6:  $G = G(\underline{\kappa}_t, \underline{\kappa})$ 
7: while  $G > G_{\max}$  do
8:    $L = L + \Delta L$ 
9:    $\underline{\kappa} = \text{predict}(\text{approx}(L))$ 
10:   $G = G(\underline{\kappa}_t, \underline{\kappa})$ 
11: end while
12:  $L_l = L - \Delta L$ 
13:  $L_u = L$ 
14: while  $L_u - L_l > 1$  do
15:   $L_t = L_l + \lfloor \frac{L_u - L_l}{2} \rfloor$ 
16:   $\underline{\kappa} = \text{predict}(\text{approx}(L_t))$ 
17:   $G = G(\underline{\kappa}_t, \underline{\kappa})$ 
18:  if  $G > G_{\max}$  then
19:     $L_l = L_t$ 
20:  else
21:     $L_u = L_t$ 
22:  end if
23: end while

```

---

Die Initialisierung des Verfahrens ist zunächst identisch zu Algorithmus 4.2. Die Hauptidee der Erweiterung besteht darin, die Approximation zu verwenden, um einen Vorwärts- oder Prädiktionsschritt zu berechnen und dann den Vergleich der Posterioeren mittels Gütemaß durchzuführen. Für den Vergleich wird allerdings die wahre posteriore Dichte für einen Schritt  $\tilde{f}_{k+1}$  benötigt, die im Allgemeinen nicht geschlossen berechnet werden kann. Aufgrund der hohen Konvergenzrate der Dirac-Mixture-Approximation kann man jedoch davon ausgehen, dass für eine große Anzahl  $L_{\text{large}}$  von Komponenten ein Ergebnis sehr nahe an der wahren Dichte gefunden werden kann. Dieses Ergebnis wird als Referenz genutzt. Ein alternativer Ansatz, der ohne diese Approximation mit vielen Komponenten auskommt, wird weiter unten beschrieben.

Durch den Prädiktionsschritt ändert sich, wie bereits erwähnt, der Typ der Dichtebeschreibung. Im hier behandelten Fall von additivem Gaussrauschen im System, erhält man als Posteriore eine Gaussian-Mixture. Der Parametervektor dieser Dichte wird im Algorithmus mit

$$\underline{\kappa} = \begin{bmatrix} \underline{\alpha} & \underline{\mu} & \underline{\sigma} \end{bmatrix}^T$$

bezeichnet.  $\underline{\kappa}_t$  bezeichnet somit den Parametervektor der wahren Posterioeren  $\tilde{f}_{k+1}$ . Die Suche nach der kleinsten Anzahl von Komponenten deren Prädiktion einen Fehler kleiner als  $G_{\max}$  erzeugt, verläuft nun wie in Algorithmus 4.2. Der einzige Unterschied besteht darin, dass das Gütemaß nun auf den Posterioeren definiert ist. Da es sich hierbei um kontinuierliche Dichten handelt, kann beispielsweise das integralquadratische Gütemaß

$$G(\underline{\kappa}_t, \underline{\kappa}) = \int_{-\infty}^{\infty} \left( \tilde{f}_{k+1}(x, \underline{\kappa}_t) - \tilde{f}_{k+1}(x, \underline{\kappa}) \right)^2 dx$$

verwendet werden.

Eine Alternative zum Vergleich mit der wahren Posterioeren  $\tilde{f}_{k+1}$  besteht darin, die Änderung der Approximation  $\tilde{f}_{k+1}$  für eine steigende Anzahl von Komponenten zu betrachten. Aufgrund der fallenden Monotonie des Fehlers bei steigender Komponentenzahl kann nach einer Anzahl von Komponenten  $L$  gesucht werden, bei der sich der Unterschied

$$G \left( \tilde{f}_{k+1}(x, \underline{\kappa}_L), \tilde{f}_{k+1}(x, \underline{\kappa}_{L+n}) \right)$$

nicht mehr signifikant verändert, wobei  $\underline{\kappa}_L$  einen Parametervektor mit  $L$  Komponenten beschreibt.

Es folgt nun ein Beispiel für eine rekursive Prädiktion durch ein chaotisches System.

#### Beispiel 4.8: Prädiktion eines chaotischen Systems.

Es wird die Simulation des Verlaufs eines stochastischen nichtlinearen dynamischen Systems betrachtet, welches ein chaotisches Verhalten aufweist. Das System wurde vorgestellt in [85] und ist gegeben durch

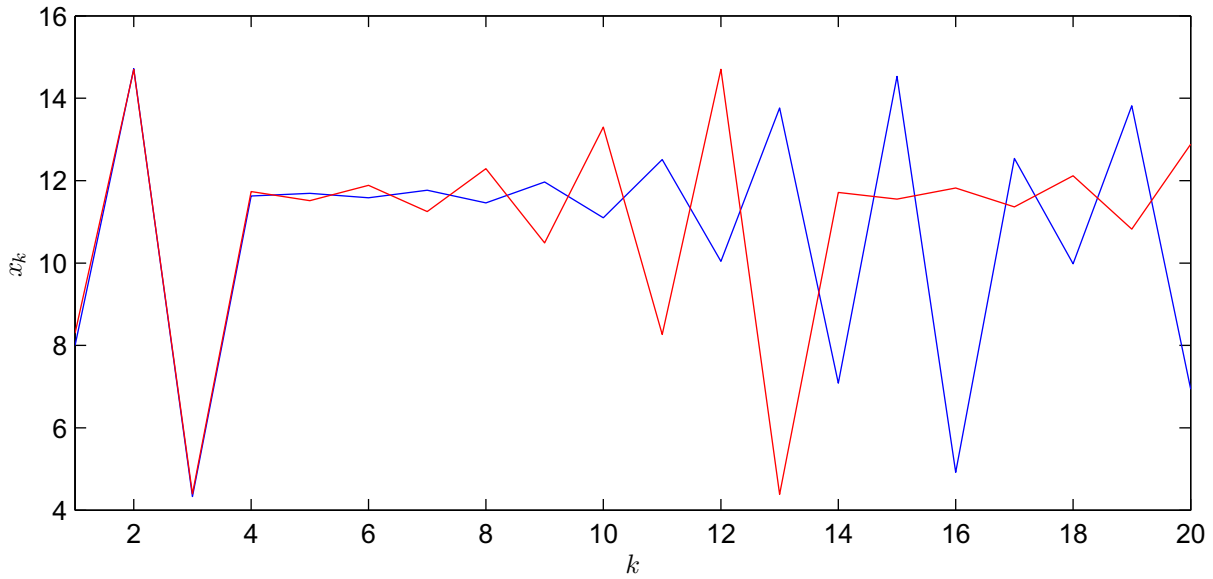
$$x_{k+1} = a(x_k) + w_k$$

mit einer zeitinvarianten Nichtlinearität

$$a(x_k) = 0.23 x_k (16 - x_k) ,$$

wobei  $w_k$  ein mittelwertfreier gaussverteilter Rauschterm mit einer Standardabweichung von  $\sigma_w = 0.4$  ist.

Das chaotische Verhalten des Systems bedeutet, dass eine kleine Änderung beim Startwert des Systems eine große Änderung im Verlauf des Systems verursacht, was in der Simulation in Abbildung 4.17 deutlich zu sehen ist. Hier wurden die Startwerte 8.0 und 8.3 gewählt.



**Abbildung 4.17:** Zeitliche Entwicklung eines chaotischen Systems bei nahe beieinanderliegenden Startwerten von 8.0 (blau) und 8.3 (rot).

Für die Betrachtung der Prädiktion wird nun von einem Startwert  $x_0 = 8.0$  ausgegangen, was zu einer ersten posterioren Dichte

$$\tilde{f}_1(x_1) = \mathcal{N}(x_1, 14.72, 0.4)$$

führt. Diese Dichte wird durch eine Dirac-Mixture-Dichte  $f_1(x_1)$  approximiert und in die Rekursion eingesetzt.

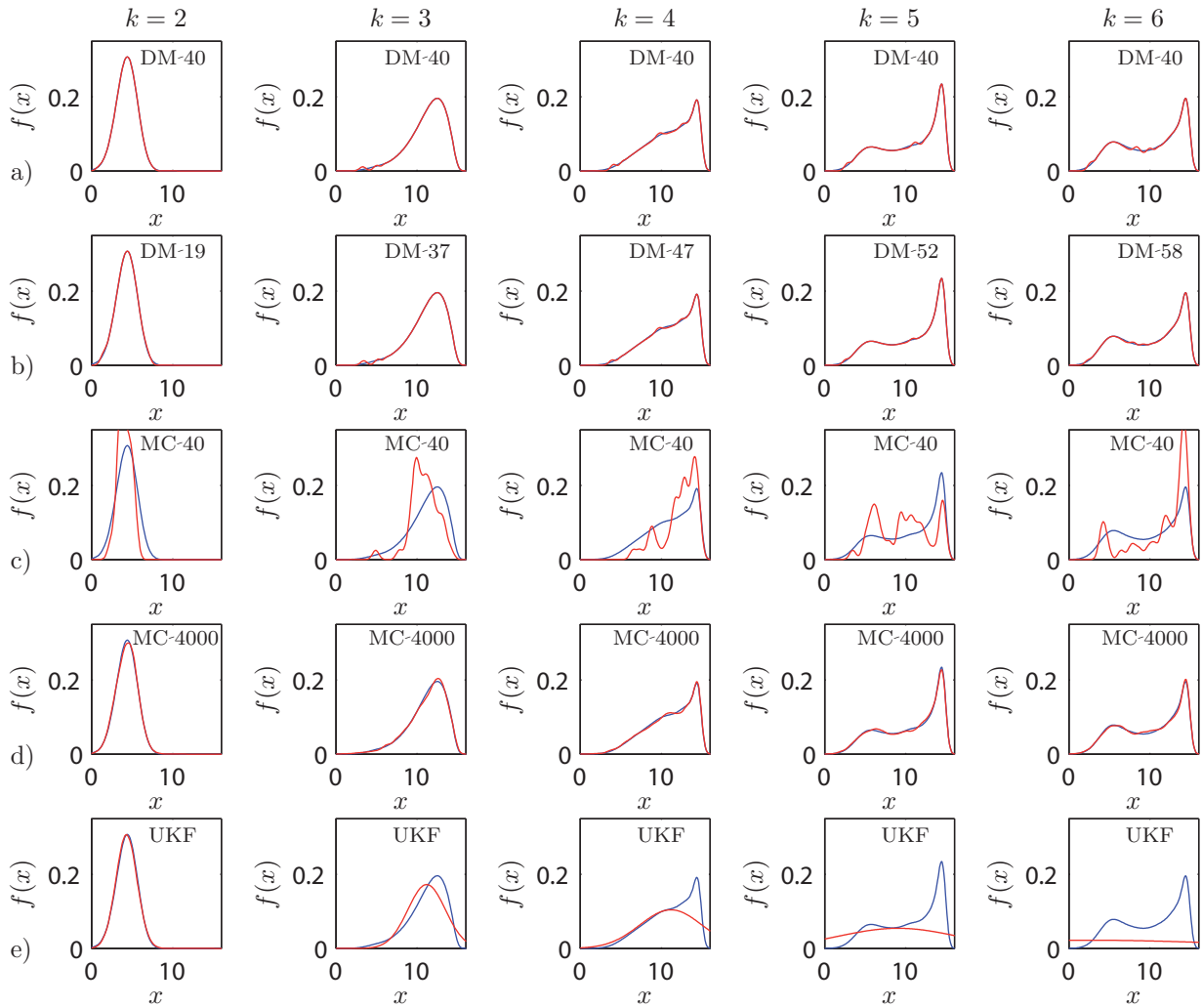
Im Folgenden werden 5 Prädiktionsschritte gezeigt, für die die wahren Dichten  $\tilde{f}_2(x_2)$  bis  $\tilde{f}_6(x_6)$  „brute force“ mittels numerischer Integration zum Vergleich berechnet wurden. Diese sind in Abbildung 4.18 jeweils in blau gezeigt.

Die wahre Dichte wird jeweils mit den resultierenden Dichten des Dirac-Mixture-Ansatzes a) und b), der Monte-Carlo-Methode c) und d) sowie des Unscented Kalman Filters (UKF) e) verglichen.

In der Zeile a) von Abbildung 4.18 wird das Ergebnis der Reapproximation der jeweiligen posterioren Gaussian-Mixture-Dichte durch eine Dirac Mixture mit  $L = 40$  Komponenten gezeigt. Hier wird keines der vorgestellten Verfahren zur Berechnung der optimalen Anzahl von Komponenten verwendet. Die rote Kurve zeigt jeweils die resultierende Gaussian Mixture des Prädiktionsschritts. Man sieht wie gut die Approximation für die einfache Dichte am Anfang gelingt. Mit steigender Komplexität der wahren Dichte wird jedoch ein geringer Fehler in  $k = 6$  sichtbar.

In Zeile b) wird das Verfahren zu Bestimmung der optimalen Anzahl von Komponenten der Priors für die beste Posteriore verwendet. Als maximaler Fehler wurde hier  $G_{\max} = 0.5\%$  gewählt. Dazu wurde ein normiertes Fehlermaß

$$G(\underline{\kappa}_t, \underline{\kappa}) = \frac{\int_{-\infty}^{\infty} \left( \tilde{f}_{k+1}(x, \underline{\kappa}_t) - \tilde{f}_{k+1}(x, \underline{\kappa}) \right)^2 dx}{\int_{-\infty}^{\infty} \left( \tilde{f}_{k+1}(x, \underline{\kappa}_t) \right)^2 dx + \int_{-\infty}^{\infty} \left( \tilde{f}_{k+1}(x, \underline{\kappa}) \right)^2 dx}$$



**Abbildung 4.18:** Fünf Prädiktionsschritte für ein chaotisches System mit einer Dirac-Mixture-Approximation mit a) einer festen Anzahl von 40 Komponenten b) einer optimalen Anzahl von Komponenten für einen maximalen Fehler in der posterioren Dichte von 0.5%, Monte-Carlo-Sampling mit c) 40 und d) 4000 Werten und e) dem Unscented Kalman Filter (UKF). In jedem Zeitschritt wird mit der wahren Dichte (blau) verglichen.

nach [42] verwendet.

Man sieht hier deutlich, dass für die einfache Dichte am Anfang wesentlich weniger Komponenten verwendet werden müssen um ein vergleichbares Ergebnis zu erlangen. Mit Zunahme der Komplexität der wahren Dichte steigt auch die Anzahl der benötigten Komponenten. Die Anzahl der Komponenten in den jeweiligen Schritten und die daraus resultierenden Fehler sind in der folgenden Tabelle zusammengefasst.

$k$	2	3	4	5	6
$L$	19	37	47	52	58
$G$	0.483%	0.444%	0.439%	0.425%	0.497%

In den Zeilen c) und d) sind die Ergebnisse bei Verwendung der Monte-Carlo-Methode gezeigt. Hier werden in jedem Schritt aus der Dichte zufällige Werte gezogen (gesampled). Der rote Graph zeigt jeweils die posteriore Dichte, die entsteht, wenn man diese Zufallswerte in den Vorwärtsschritt

einsetzt und die jeweilige Gaussian-Mixture-Dichte erhält. Bei Particle Filtern ist dieses Vorgehen zwar nicht üblich, da die Posterioe im Allgemeinen auch durch Partikel repräsentiert wird. Die Darstellung über kontinuierliche Dichten erleichtert allerdings hier den Vergleich.

Es ist deutlich zu sehen, dass das Ergebnis mit 40 Partikeln deutlich schlechter ist als mit 40 Dirac-Komponenten. Um vergleichbare Ergebnisse zu erhalten, muss eine hundertfach größere Anzahl von Partikeln verwendet werden wie in Zeile d) zu sehen.

In Zeile e) sind der Vollständigkeit halber noch die Ergebnisse bei Verwendung des Unscented Kalman Filter (UKF) gezeigt. Hier wird deutlich, dass die alleinige Verwendung des ersten Moments und des zweiten Zentralmoments nicht ausreicht, um die entstehenden Dichten eines solchen nichtlinearen Systems adäquat zu beschreiben. ■

Ein vollständiges Filter basierend auf Dirac-Mixture-Approximation wurde auf der *ICINCO 2007* [99] vorgestellt.

## KAPITEL 5

# Modellidentifikation

Zur Verwendung modellbasierter Verfahren wie der Bayes'schen Zustandsschätzung wird immer ein Modell des betrachteten Systems benötigt. Wenn es um die Modellierung physikalischer Zusammenhänge geht, sind diese Modelle häufig in einer generativen Form gegeben, wie sie direkt in dem Verfahren aus Kapitel 4 verwendet werden können. Stochastische Größen wie beispielsweise ein System- oder Messrauschen müssen jedoch häufig aus Messungen identifiziert werden.

Da die Rauschcharakteristik im Allgemeinen durch eine Wahrscheinlichkeitsdichte beschrieben wird, nennt man diese Rekonstruktion aus Daten auch Dichterekonstruktion oder Dichteschätzung [81]. Auch im Bereich des Lernens von Bayesnetzen wird die Dichteschätzung eingesetzt [88].

### 5.1 Stand der Technik

Dichteschätzverfahren können in zwei Familien unterteilt werden: parametrische Verfahren und nicht-parametrische Verfahren.

Nicht-parametrische Verfahren machen im Allgemeinen keine Annahme bezüglich des Typs der zu schätzenden Dichte. Der einfachste Ansatz in diesem Kontext sind Histogramme [35]. Hier werden die beobachteten Daten in Klassen, so genannte *Bins* eingeteilt und gezählt. Dies führt zu einer diskreten Repräsentation der Dichte. Eine kontinuierliche Beschreibung erhält man von so genannten *Kernel*-Schätzern. Bei diesem Ansatz wird eine kontinuierliche Dichtefunktion (Kernel) an den Positionen der beobachteten Daten platziert und aufsummiert. Der wohl bekannteste Vertreter dieser Verfahren ist der Parzen Schätzer [76]. Der verwendete Kernel ist häufig eine Gaußfunktion mit einer festen Standardabweichung die auch als Bandbreite bezeichnet wird.

Parametrische Schätzer machen im Gegensatz zu nicht-parametrischen Verfahren eine Annahme bezüglich des Typs der Dichte mit dem die wahre Dichte geschätzt werden soll. Dies können verschiedene parametrische Dichtetypen wie Gauss, Laplace, Gleichverteilung oder Mixturen aus diesen sein. Diese Annahme bedeutet allerdings nicht, dass die wahre Dichte von genau

diesem Typ sein muss. Die traditionelle Methode zur Schätzung der Parameter einer parametrischen Dichtefunktion aus Daten ist das *Maximum Likelihood* Verfahren, das auf R.A. Fisher zurückgeht [3]. Die Idee des Verfahrens besteht darin, die Parameter zu finden, die die beobachteten Daten mit der höchsten Wahrscheinlichkeit erklären. Das Problem des Verfahrens ist, dass die dabei verwendete Likelihoodfunktion häufig mehrere lokale Maxima besitzt und das Ergebnis der Suche nach dem globalen Maximum stark von der Wahl der Startparameter abhängt. Das bekannteste Verfahren zur Lösung des Maximum Likelihood Problems ist die *Expectation Maximization* (EM) Methode [25]. Dabei handelt es sich um einen iterativen Algorithmus, von dem bekannt ist, dass er in ein Maximum der Likelihoodfunktion konvergiert.

In diesem Kapitel wird ein neuartiges parametrisches Schätzverfahren vorgestellt, das auf den Ideen der Dichteapproximation aus Kapitel 4 beruht. Dabei wird allerdings das Argument herumgedreht. D. h. es wird eine Dirac-Mixture-Funktion durch eine kontinuierliche Funktion approximiert. Da dies wieder im Verteilungsraum geschieht, ist das Ergebnis vergleichbar mit Verfahren, die eine Kolmogorov-Smirnov-Test-Statistik [106] minimieren, da diese ebenfalls im Verteilungsraum definiert ist. In dem hier vorgestellten Verfahren wird jedoch ein anderes Gütemaß gewählt.

Darüber hinaus ist interessant, dass das vorgestellte Verfahren auch als *Curve Fitting* Problem [80] betrachtet werden kann. Dieses Problem ist für einfache Dichtefunktionen wie Gaussdichten leicht zu lösen. Für Gaussian-Mixtures, wie sie beispielsweise hier betrachtet werden, ist es allerdings ein aufwändiges Problem, für das im Folgenden eine Lösung vorgestellt wird.

## 5.2 Dichteschätzung

Es wird nun ein neues Verfahren zur Dichteschätzung betrachtet. Dieses Verfahren wurde auf der *Fusion 2007* publiziert [45] und beruht auf einem Approximationsansatz, der dem in Kapitel 4 verwendeten Verfahren ähnlich ist.

Im Folgenden wird eine kontinuierliche Dichtefunktion  $\tilde{f}(x)$  betrachtet. Die Charakteristik dieser Dichte ist unbekannt. Lediglich eine Menge von Realisierungen (*Samples*) dieser Dichte sind gegeben. Es wird angenommen, dass diese Realisierungen unabhängig und gemäß  $\tilde{f}(x)$  verteilt sind.

Das Ziel ist nun, eine parametrische Dichtebeschreibung  $f(x, \underline{\kappa})$  aus den Realisierungen zu bestimmen, die als Approximation von  $\tilde{f}(x)$  dient.  $\underline{\kappa}$  ist hier der Parametervektor von  $f(\cdot)$ . Als Approximationsdichte wird eine Mischdichte vom Typ

$$f(x, \underline{\kappa}) = \sum_{i=1}^M \omega_i f_i(x, \underline{\kappa}_i)$$

mit einer endlichen Anzahl von Komponenten angenommen.  $\omega_i$  sind Gewichte und  $f_i(x, \underline{\kappa}_i)$  parametrische Dichtefunktionen mit den Parametern  $\underline{\kappa}_i$  für  $i = 1, \dots, M$ . Die Gewichte sind



positiv und addieren sich zu Eins nach

$$\sum_{i=1}^M \omega_i = 1 .$$

Der Parametervektor ist somit durch

$$\underline{\kappa} = \left[ \omega_1, \omega_2, \dots, \omega_M, \underline{\kappa}_1^T, \underline{\kappa}_2^T, \dots, \underline{\kappa}_M^T \right]^T$$

gegeben.

Der besondere Ansatz des neuen Verfahrens besteht in der Interpretation der Realisierungen als Dirac-Mixture-Funktion

$$\tilde{f}(x, \underline{\tilde{\eta}}) = \sum_{j=1}^L \tilde{\omega}_j \delta(x - \tilde{x}_j) .$$

$\underline{\tilde{\eta}}$  ist hier ein Parametervektor der aus  $L$  Gewichten  $\tilde{\omega}_i$  und Positionen  $\tilde{x}_i$  der Realisierungen besteht. Da alle Realisierungen gleich wichtig sind, ergibt sich eine Gleichgewichtung mit  $\omega_i = \frac{1}{L}$  woraus

$$\tilde{f}(x, \underline{\tilde{\eta}}) = \frac{1}{L} \sum_{j=1}^L \delta(x - \tilde{x}_j) .$$

mit

$$\underline{\tilde{\eta}} = \left[ \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_L \right]^T .$$

folgt.

Die wahre Dichte  $\tilde{f}(x)$  ist im Allgemeinen nicht bekannt und kann daher auch nicht direkt approximiert werden. Da  $\tilde{f}(x, \underline{\tilde{\eta}})$  die einzige Information ist, die bezüglich der wahren Dichte vorliegt, soll diese durch  $f(x, \underline{\kappa})$  approximiert werden.

Häufig wird dieses Problem durch die Maximierung der Likelihoodfunktion  $f(\underline{\tilde{\eta}}|\underline{\kappa})$  für ein festes  $\underline{\tilde{\eta}}$  behandelt. Im Gegensatz dazu soll hier ein neues Verfahren angewendet werden, bei dem ein Gütemaß  $G$  zwischen der Dirac-Mixture  $\tilde{f}(x, \underline{\tilde{\eta}})$  und der Approximation  $f(x, \underline{\kappa})$  minimiert wird. Dadurch wird das Approximationsproblem wieder als Optimierungsproblem betrachtet. Ferner soll gewährleistet werden, dass das gefundene Minimum auch ein Minimum des dualen Problems ist. Unter dem dualen Problem ist die Approximation einer kontinuierlichen Dichte durch eine Dirac-Mixture-Dichte zu verstehen, wie sie im vorangegangenen Kapitel behandelt wurde. Diese Forderung erlaubt einen Übergang von der einen Repräsentation in die andere und zurück.

### 5.2.1 Schätzverfahren

Wie auch beim dualen Problem ist der Vergleich einer Dirac-Mixture mit einer kontinuierlichen Dichtefunktion im Dichteraum nicht sinnvoll. Aus diesem Grund wird der Ansatz, wie bereits zuvor, im Verteilungsraum betrachtet. Als Verteilungsfunktionen sind hier für die Dirac-Mixture

$$\tilde{F}(x, \underline{\tilde{\eta}}) = \frac{1}{L} \sum_{i=1}^L H(x - \tilde{x}_i)$$

und für die Approximationsfunktion

$$F(x, \underline{\kappa}) = \int_{-\infty}^x f(t, \underline{\kappa}) dt$$

gegeben. Das zu minimierende Gütemaß ist

$$G(\underline{\tilde{\eta}}, \underline{\kappa}) = \int_{-\infty}^{\infty} \left( \tilde{F}(x, \underline{\tilde{\eta}}) - F(x, \underline{\kappa}) \right)^2 dx .$$

Da ein Minimum dieses Gütemaßes im Hinblick auf das duale Problem gefunden werden soll, wird die partielle Ableitung nach  $\underline{\tilde{\eta}}$  gleich Null gesetzt, woraus die Bedingung

$$F(\tilde{x}_i, \underline{\kappa}) = \frac{2i - 1}{2L} \tag{5.1}$$

für  $i = 1, \dots, L$  entsteht. Ein vergleichbares Ergebnis wird auch in der optimalen Quantisierungstheorie gefunden [64].

Da die Anzahl der Realisierungen, gegeben durch die Dimension von  $\underline{\tilde{\eta}}$ , im Allgemeinen wesentlich größer ist als die Anzahl der Komponenten, die zur Approximation verwendet werden und durch  $\underline{\eta}$  gegeben sind, handelt es sich hier um ein überbestimmtes Gleichungssystem. Darüber hinaus handelt es sich hierbei im Allgemeinen um eine nichtlineare Gleichungssystem.

Das Lösen dieses nichtlinearen Gleichungssystems ist identisch mit dem Finden der Nullstelle der vektoriellen Funktion

$$\underline{g}(\underline{\kappa}) = \begin{bmatrix} F(\tilde{x}_1, \underline{\kappa}) - \frac{2 \cdot 1 - 1}{2L} \\ \vdots \\ F(\tilde{x}_L, \underline{\kappa}) - \frac{2 \cdot L - 1}{2L} \end{bmatrix} .$$

Die Verwendung von Standardverfahren wie Newtoniteration oder Gradientenabstieg liefert aber häufig keine befriedigenden Ergebnisse. Dies liegt an der Tatsache, dass die Konvergenz dieser Verfahren sehr stark vom initialen Startwert der Algorithmen abhängt. Um diesem Problem zu begegnen, soll auch hier eine Progressionstechnik verwendet werden.

Die Idee besteht darin, die Approximation mit einem Paar aus Parametervektoren  $\underline{\kappa}_{\text{Start}}$  und  $\underline{\eta}_{\text{Start}}$  zu beginnen, für die bekannt ist, dass sie bezüglich des Gütemaßes optimal zusammenpassen. D.h. die Distanz zwischen  $F(x, \underline{\kappa}_{\text{Start}})$  und  $\tilde{F}(x, \underline{\eta}_{\text{Start}})$  ist bereits minimal. Dabei gilt zu bedenken, dass  $\underline{\eta}_{\text{Start}}$  eine Konfiguration von Realisierungen repräsentiert, die im Allgemeinen völlig verschieden von der gewünschten Konfiguration der Realisierungen  $\underline{\tilde{\eta}}$  ist.

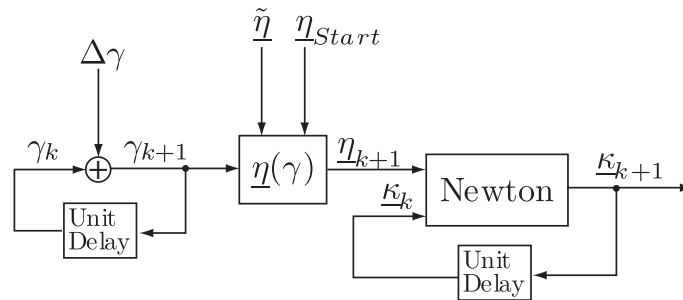
Ausgehend von diesem initialen Paar von Parametervektoren werden nun die Positionen der Realisierungen graduell in die gewünschten Positionen  $\underline{\tilde{\eta}}$  überführt. Dabei soll das Minimum des Gütemaßes durch Anpassung des Parametervektors der Approximationsfunktion  $\underline{\kappa}$  beibehalten werden.

Die Überführung der Realisierungspositionen von  $\underline{\eta}_{\text{Start}}$  nach  $\underline{\tilde{\eta}}$  geschieht durch einen Progressionsparameter  $\gamma$  der von 0 nach 1 geht. Die Progression kann dadurch als lineare Verschiebung

$$\underline{\eta}(\gamma) = \gamma \underline{\tilde{\eta}} + (1 - \gamma) \underline{\eta}_{\text{Start}}$$

formuliert werden.

Während  $\gamma$  kontinuierlich von 0 bis 1 erhöht wird, muss das Minimum der Gütefunktion durch Adaption des Parametervektors  $\underline{\kappa}$  verfolgt werden. Dazu kann beispielsweise ein Newtonverfahren [107] verwendet werden. Zur Berechnung dieser Parameteranpassung, wird  $\gamma$  immer um kleine Inkremente erhöht. Daraus ergibt sich ein schrittweises iteratives Verfahren, das in Abbildung 5.1 schematisch dargestellt ist.



**Abbildung 5.1:** Blockdiagramm der progressiven Dichteschätzung. Der Progressionsparameter  $\gamma$  wird in jedem Schritt um  $\Delta\gamma$  erhöht, woraus ein verändertes  $\eta_{k+1}$  resultiert.  $\underline{\kappa}$  wird durch ein Newtonverfahren angepasst, um das Minimum des Gütemaßes  $G$  zu verfolgen.

Jeder Schritt besteht aus zwei Phasen. Zuerst wird  $\gamma$  um eine Schrittweite  $\Delta\gamma$  erhöht. Anschließend erfolgt eine Anpassung des Parametervektors  $\underline{\kappa}$  um das Gütemaß  $G(\underline{\eta}(\gamma + \Delta\gamma), \underline{\kappa})$  zu minimieren.

Es kann angenommen werden, dass der aktuelle Parametervektor  $\underline{\kappa}$  ein globales Minimum der Gütefunktion vor der Erhöhung von  $\gamma$  beschreibt. Dies begründet sich in der Wahl des initialen Parameterpaars  $\underline{\kappa}_{Start}$  und  $\underline{\eta}_{Start}$ . Da  $\gamma$  nur um infinitesimal kleine Schritte  $\gamma + \Delta\gamma$  erhöht wird, wird auch das Minimum nur gering verschoben. Aus diesem Grund kann ein Newtonverfahren zur Anpassung des Parametervektors  $\underline{\kappa}$  verwendet werden. Der resultierende Parametervektor kann als Startwert für die nächste Anwendung des Newtonverfahrens eingesetzt werden.

Die verbleibende Frage ist nun, wie ein initiales Paar von Parametervektoren  $\underline{\kappa}_{Start}$  und  $\underline{\eta}_{Start}$  für  $\gamma = 0$  gefunden werden kann. Hier zeigt sich, dass die Berücksichtigung des dualen Problems von Vorteil war, denn hier wurde im vorangehenden Kapitel gezeigt, wie eine optimale Dirac-Mixture-Approximation für eine kontinuierliche Dichte berechnet werden kann. Folglich kann Algorithmus 4.1 verwendet werden, um für ein beliebiges  $\underline{\kappa}_{Start}$  das optimale  $\underline{\eta}_{Start}$  zu berechnen. Die besten Ergebnisse im Hinblick auf die anschließende Progression werden erreicht, indem die Dichte  $f(x, \underline{\kappa}_{Start})$  so gewählt wird, dass sie den Bereich, in dem sich die wahren Realisierungen  $\tilde{x}_i$  befinden, annähernd gleichverteilt abdeckt. Eine exakte Gleichverteilung ist im Allgemeinen nicht möglich, da es sich bei  $f(x, \underline{\kappa}_{Start})$  um eine Mischdichte mit endlicher Komponentenanzahl handelt.

Die progressive Approximationsmethode kann durch zwei Erweiterungen noch verbessert werden: durch eine Schrittweitenregelung, sowie durch eine Regelung der Genauigkeit des verwendeten Newtonverfahrens (Toleranzregelung).

Die Schrittweitenregelung adaptiert die Schrittweite  $\Delta\gamma$ . Dies ermöglicht maximal große Schritte, die immer noch eine Konvergenz des Newtonverfahrens erlauben. Dazu wird  $\Delta\gamma$  zu Beginn der Approximation auf einen minimalen Wert gesetzt. Wenn das folgende Newtonverfahren erfolgreich konvergiert, wird der resultierende Parametervektor  $\underline{\kappa}$  als Startwert für den nächsten Schritt akzeptiert und die Schrittweite erhöht. Dies geschieht bis eine maximale Schrittweite erreicht ist. Konvergiert das Newtonverfahren nicht, so wird der resultierende Parametervektor verworfen und  $\gamma$  auf  $\gamma - \Delta\gamma$  zurückgesetzt. Für den folgenden Schritt wird  $\Delta\gamma$  verringert. Dies geschieht bis zu einer minimalen Schrittweite.

Die Toleranzregelung beeinflusst das Konvergenzkriterium des Newtonverfahrens. Das Newtonverfahren gilt als konvergiert, wenn der Parametervektor sich um weniger als einen Wert  $\epsilon$  verändert, bevor eine maximale Anzahl von Iterationen erreicht ist, wobei  $\epsilon > 0$  aber sehr klein ist. Die Toleranzregelung beeinflusst dieses  $\epsilon$  wie folgt: Zu Beginn wird  $\epsilon$  auf einen minimalen Wert gesetzt. Sollte das Newtonverfahren nicht konvergieren und  $\Delta\gamma$  ist bereits minimal, so wird  $\epsilon$  erhöht. Wenn das Newtonverfahren konvergiert und  $\Delta\gamma$  ist bereits maximal, so wird  $\epsilon$  verringert.

Das gesamte Verfahren ist in Algorithmus 5.1 als Pseudocode angegeben.

### Spezialfall: Gaussian Mixture

Im Folgenden wird nun der Spezialfall der Dichteschätzung durch eine Gaussian-Mixture betrachtet. Das heißt, es werden die Parameter einer Funktion

$$f(x, \underline{\kappa}) = \sum_{j=1}^M \omega_j \mathcal{N}(x, \mu_j, \sigma_j),$$

also Gewichte  $\omega_j$ , Erwartungswerte  $\mu_j$  und Standardabweichungen  $\sigma_j$  der jeweiligen Komponenten aus den beobachteten Realisierungen bestimmt. Die entsprechende Verteilungsfunktion lautet

$$F(x, \underline{\kappa}) = \frac{1}{2} \sum_{j=1}^M \omega_j \operatorname{erf} \left( \frac{x - \mu_j}{\sqrt{2}\sigma_j} \right) + \frac{1}{2},$$

wobei  $\operatorname{erf}(\cdot)$  die Fehlerfunktion ist.

Durch Einsetzen in die Bedingung (5.1) erhält man das System nichtlinearer Gleichungen

$$\frac{1}{2} \sum_{j=1}^M \omega_j \operatorname{erf} \left( \frac{x_i - \mu_j}{\sqrt{2}\sigma_j} \right) + \frac{1}{2} = \frac{2i - 1}{2L}$$

für  $i = 1, \dots, L$ . Um dieses zu lösen, wird die Nullstelle der vektorwertigen Funktion

$$\underline{g}(\underline{\kappa}) = \begin{bmatrix} \frac{1}{2} \sum_{j=1}^M \omega_j \operatorname{erf} \left( \frac{x_1 - \mu_j}{\sqrt{2}\sigma_j} \right) + \frac{1}{2} - \frac{2 \cdot 1 - 1}{2L} \\ \vdots \\ \frac{1}{2} \sum_{j=1}^M \omega_j \operatorname{erf} \left( \frac{x_L - \mu_j}{\sqrt{2}\sigma_j} \right) + \frac{1}{2} - \frac{2 \cdot L - 1}{2L} \end{bmatrix}$$

---

**Algorithmus 5.1** Progressive Methode zur Dichteschätzung mit Schrittweiten- und Toleranzregelung.

---

```

1:  $\gamma := 0$ 
2:  $\underline{\kappa} := \text{Uniform}(M, \text{range}(\tilde{\eta}))$ 
3:  $\underline{\eta}_{\text{Start}} := \text{DiracApprox}(\underline{\kappa}, L)$  //aus Alg. 4.1
4:  $\epsilon := \epsilon_{\min}$ 
5:  $\Delta\gamma := \gamma_{\text{step\_min}}$ 
6: function  $\underline{\eta} = \underline{\eta}(\gamma)$ 
7:    $\underline{\eta} = \gamma\tilde{\eta} + (1 - \gamma)\underline{\eta}_{\text{Start}}$ 
8: end function
9: repeat
10:   $\gamma := \gamma + \Delta\gamma$ 
11:   $\underline{\eta} = \underline{\eta}(\gamma)$ 
12:   $[\underline{\kappa}_{\text{tmp}}, \text{success}] := \text{NewtonApproach}(\underline{\kappa}, \underline{\eta}, \epsilon)$ 
13:  if success then
14:     $\underline{\kappa} := \underline{\kappa}_{\text{tmp}}$ 
15:    if  $\Delta\gamma < \gamma_{\text{step\_max}}$  then
16:      Increase( $\Delta\gamma, \gamma_{\text{step\_max}}$ )
17:    else
18:      Decrease( $\epsilon, \epsilon_{\min}$ )
19:    end if
20:  else
21:     $\gamma := \gamma - \Delta\gamma$ 
22:    if  $\Delta\gamma > \gamma_{\text{step\_min}}$  then
23:      Decrease( $\Delta\gamma, \gamma_{\text{step\_min}}$ )
24:    else
25:      Increase( $\epsilon, \epsilon_{\max}$ )
26:    end if
27:  end if
28: until  $\gamma = 1$ 

```

---

gesucht. Hierzu wird das Newtonverfahren verwendet, dass in jeder Iteration

$$\underline{\kappa}_{k+1} = \underline{\kappa}_k - \left( \mathbf{J}_g(\underline{\kappa}_k)^T \mathbf{J}_g(\underline{\kappa}_k) \right)^{-1} \mathbf{J}_g(\underline{\kappa}_k)^T \cdot \underline{g}(\underline{\kappa}_k)$$

berechnet.  $\mathbf{J}_g(\underline{\kappa}_k)$  bezeichnet hierbei die Jakobimatrix der Funktion  $\underline{g}(\underline{\kappa})$ . Da die Jakobimatrix in dieser Anwendung im Allgemeinen nicht quadratisch ist, wird ein modifiziertes Newtonverfahren mit einer Pseudoinversen verwendet.

Die Jakobimatrix setzt sich aus drei Blöcken

$$\mathbf{J}_g(\underline{\kappa}_k) = [\mathbf{J}_1 \ \mathbf{J}_2 \ \mathbf{J}_3]$$

zusammen. Der erste Block ist gegeben durch

$$\mathbf{J}_1(i, j) = \frac{1}{2} \left[ \operatorname{erf} \left( \frac{x_i - \mu_j}{\sqrt{2}\sigma_j} \right) - \operatorname{erf} \left( \frac{x_i - \mu_M}{\sqrt{2}\sigma_M} \right) \right]$$

Dies ist eine  $(L \times M - 1)$ -Matrix, die die Ableitungen nach den Gewichten  $\omega_j$ ,  $j = 1, \dots, M - 1$  enthält. Das letzte Gewicht wird aus der Normierungsbedingung  $\omega_M = 1 - \sum_{j=1}^M \omega_j$  berechnet. Aus dieser Bedingung ergibt sich auch der zweite Term in  $\mathbf{J}_1$ .

Der zweite Block ist gegeben durch die  $L \times M$ -Matrix

$$\mathbf{J}_2(i, j) = -\omega_j \mathcal{N}(x_i, \mu_j, \sigma_j)$$

und beschreibt die Ableitung nach den Mittelwerten  $\mu_j$  für  $j = 1, \dots, M$ .

Der dritte Block ist ebenfalls eine  $L \times M$ -Matrix, die durch

$$\mathbf{J}_3(i, j) = -\omega_j \frac{x_i - \mu_j}{\sigma_j} \mathcal{N}(x_i, \mu_j, \sigma_j)$$

gegeben ist. Sie beschreibt die Ableitung nach den Standardabweichungen  $\sigma_j$  für  $j = 1, \dots, M$ .

Im Folgenden wird nun ein Beispiel für die progressive Approximation betrachtet.

### Beispiel 5.1: Approximation einer Dichte aus optimalen Realisierungen

In diesem Beispiel wird eine Gaussian-Mixture mit drei Komponenten und dem Parametervektor

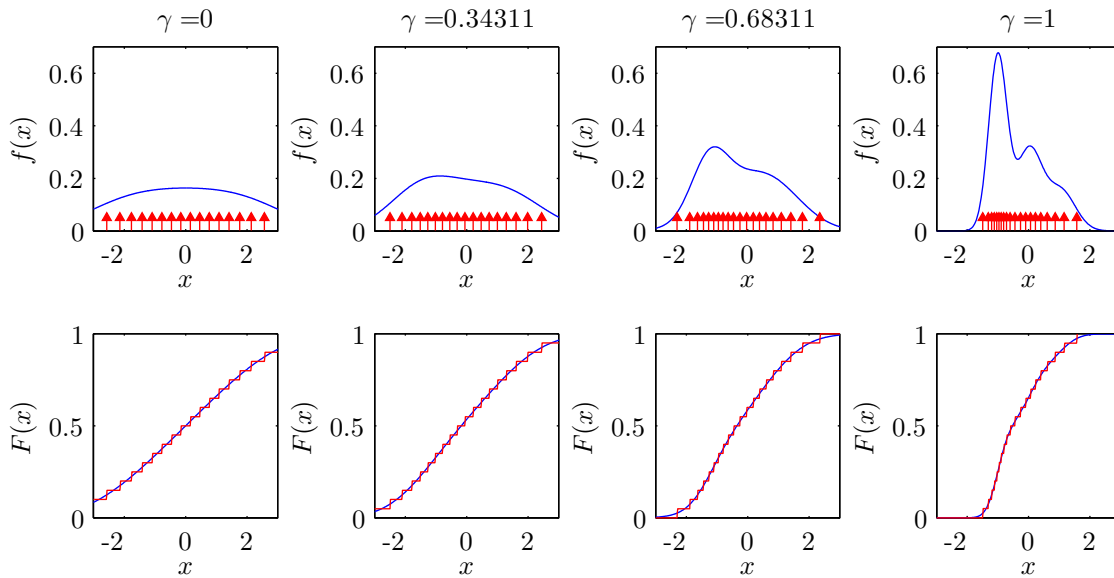
$$\tilde{\underline{\kappa}} = [0.5, 0.3, 0.2, -1, 0, 1, 0.3, 0.4, 0.5]^T$$

als wahre Dichte  $\tilde{f}(x)$  betrachtet.

Aus dieser Wahrscheinlichkeitsdichtefunktion werden nun Realisierungen generiert. Um zu zeigen, dass das vorgestellte Verfahren auch das duale Problem löst, werden diese Realisierungen zunächst mit dem Verfahren aus Abschnitt 4.5.2 generiert. Sie sind somit im Hinblick auf das Gütemaß optimal. Für das Beispiel werden  $L = 20$  Realisierungen generiert.

Die Progression startet mit einer Gaussian abdeckt. Dies ist der Bereich in dem die Realisierungen liegen. Der Parametervektor dieser Funktion ist gegeben durch

$$\underline{\kappa}_{\text{Start}} = \left[ \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, -2, 0, 2, 1.4, 1.4, 1.4 \right]^T .$$

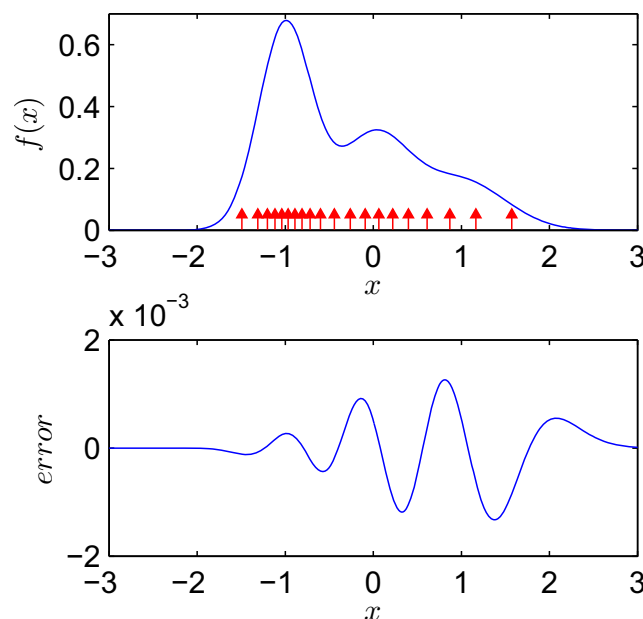


**Abbildung 5.2:** Progressionsverlauf für eine Dichteschätzung. Es wird eine Gaussian-Mixture mit 3 Komponenten aus 20 optimalen Realisierungen rekonstruiert. Die obere Zeile zeigt den Progressionsverlauf in der Dichtefunktion, die untere den Verlauf in der Verteilungsfunktion. In rot sind jeweils die Realisierungen und in blau die rekonstruierte Funktion zu sehen.

Aus diesem Startvektor werden die Startparameter der Dirac-Mixture-Funktion  $\underline{\eta}_{\text{Start}}$  mit dem Verfahren aus Kapitel 4.5.2 berechnet. Dadurch wird gewährleistet, dass sich  $G(\underline{\eta}_{\text{Start}}, \underline{\kappa}_{\text{Start}})$  in einem Minimum befindet.

Die Progression ist ausschnittsweise in Abbildung 5.2 zu sehen. Die Abbildung stellt den Verlauf der Progression für  $\gamma = 0$ ,  $\gamma = 1$  und zwei Zwischenschritte dar. Die obere Zeile zeigt den Progressionsverlauf in der Dichtefunktion, die untere den Verlauf in der Verteilungsfunktion.

Die resultierende Approximation sowie der Rekonstruktionsfehler sind in Abbildung 5.3 gezeigt. Als Fehler wird hier die Differenz  $\tilde{f}(x) - f(x, \underline{\kappa})$  betrachtet.



**Abbildung 5.3:** Resultat der progressiven Dichteschätzung (oben). Differenz zur wahren Dichtefunktion (unten).

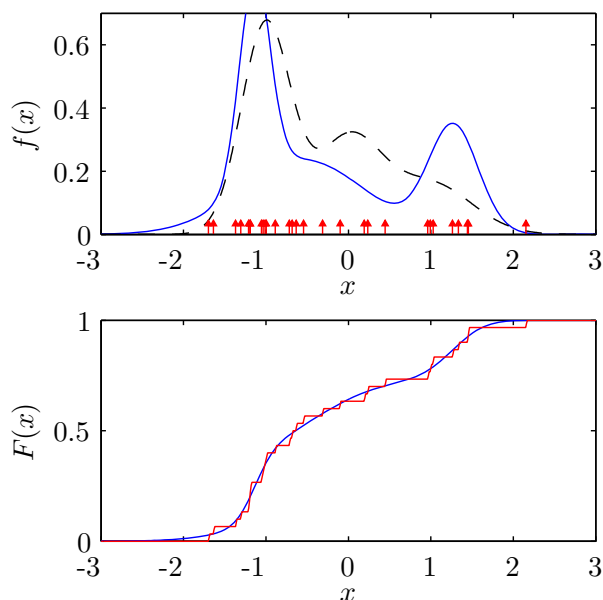
Es ist deutlich zu sehen, dass trotz der geringen Anzahl von Realisierungen der Beträgsfehler geringer als  $2 \times 10^{-3}$  bleibt, was eine hohe Genauigkeit bedeutet. ■

Das gezeigte Beispiel verwendete optimale Realisierungen, die durch das vorgestellte deterministische Verfahren berechnet wurden. Dies diente hauptsächlich der Veranschaulichung der Progression. In einem weiteren Beispiel werden nun zufällige Realisierungen generiert und zur progressiven Dichteschätzung verwendet.

**Beispiel 5.2: Approximation einer Dichte aus zufälligen Samples**

In diesem Beispiel wird dieselbe wahre Dichte  $\tilde{f}(x)$  wie im vorangegangenen Beispiel verwendet. Abweichend von diesem Beispiel werden nun 30 *zufällige* Realisierungen aus dieser Dichte gezogen.

Die Startkonfiguration  $\underline{\eta}_{\text{Start}}$  und  $\underline{\kappa}_{\text{Start}}$  ist identisch zum vorherigen Beispiel. Das Ergebnis der progressiven Dichteschätzung ist in Abbildung 5.4 zu sehen. Im oberen Teil dieser Abbildung wird der



**Abbildung 5.4:** Schätzung einer Gaussian-Mixture-Dichte mit drei Komponenten aus 30 zufälligen Realisierungen. Oben: Die wahre Dichte  $\tilde{f}(x)$  ist durch die schwarze gestrichelte Linie gezeigt, das Schätzergebn durch die blaue Linie. Unten: Die Stufenfunktion der Realisierungen aus der wahren Dichte sind in rot dargestellt und die geschätzte Verteilungsfunktion in blau.

Dichteraum und im unteren Teil der Verteilungsraum dargestellt. Die gestrichelte schwarze Linie im oberen Plot zeigt die wahre Dichte  $\tilde{f}(x)$ ; die blaue Linie zeigt die resultierende Schätzung  $f(x, \underline{\kappa})$ . Die Positionen der Realisierungen als Dirac Mixture  $\tilde{f}(x, \underline{\eta})$  sind durch die roten Pfeile angegeben. Die Verteilungsfunktionen  $\tilde{F}(x, \underline{\eta})$  und  $F(x, \underline{\kappa})$  sind im unteren Plot jeweils in rot und blau gezeigt. ■

Man sieht im unteren Teil von Abbildung 5.4 deutlich, dass die kontinuierliche Funktion nicht exakt durch die Stufen der Sprungfunktion geht, was bedeutet, dass die Bedingung (5.1) nicht exakt erfüllt ist. Dies liegt daran, dass es sich um ein nichtlineares überbestimmtes Gleichungssystem handelt, für das keine exakte Lösung existiert. Allerdings liefert das Verfahren den besten möglichen Parametervektor  $\underline{\kappa}$  im Sinne der kleinsten Fehlerquadrate.



Um die Qualität des Verfahrens beurteilen zu können, wird nun in einem weiteren Beispiel das Resultat mit den Ergebnis des häufig verwendeten EM-Algorithmus verglichen.

### Beispiel 5.3: Vergleich des progressiven Dichteschätzverfahrens mit dem EM-Algorithmus

Die wahre Dichte  $\tilde{f}(x)$  ist identisch zu den vorherigen Beispielen, ebenso wie die Startkonfiguration für das progressive Schätzverfahren.

Um die Ergebnisse der beiden Verfahren vergleichen zu können werden für verschiedene Anzahlen von Realisierungen jeweils 20 Sätze von zufälligen Realisierungen aus  $\tilde{f}(x)$  gezogen. Für jeden dieser 20 Sätze wird die Abweichung des Schätzergebnisses zur wahren Dichte durch

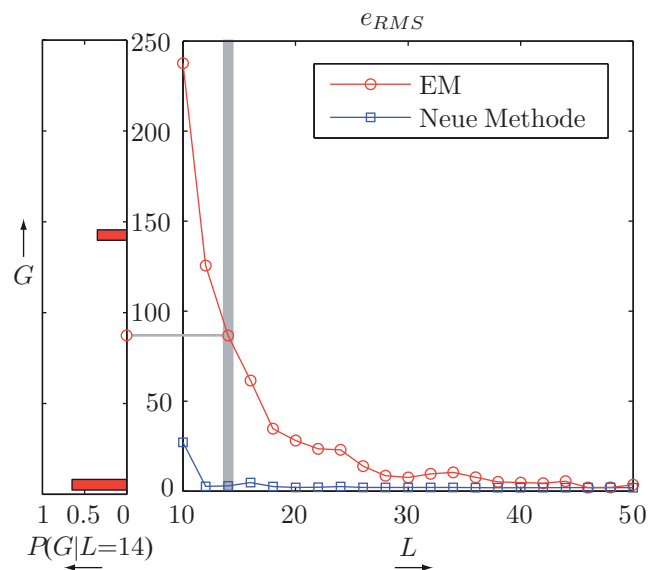
$$G(\tilde{\kappa}, \kappa_{EM}) = \int_{-\infty}^{\infty} \left( \tilde{f}(x) - f(x, \kappa_{EM}) \right)^2 dx$$

bzw.

$$G(\tilde{\kappa}, \kappa_P) = \int_{-\infty}^{\infty} \left( \tilde{f}(x) - f(x, \kappa_P) \right)^2 dx$$

berechnet.

Das quadratische Mittel der Fehler über alle 20 Sätze ( $e_{RMS}$ ) ist für eine steigende Anzahl von Realisierungen im rechten Teil von Abbildung 5.5 gezeigt. Es ist deutlich zu sehen, dass der Schätzfehler



**Abbildung 5.5:** Links: Histogramm der Verteilung von  $G(\tilde{\kappa}, \kappa_{EM})$  für  $L = 14$  Realisierungen. Rechts: Quadratisches Mittel der Schätzfehler für EM in rot und für das progressive Verfahren in blau.

des progressiven Verfahrens für kleine Anzahlen von Realisierungen im Mittel wesentlich geringer ist als der mittlere Schätzfehler des EM Verfahrens. Um zu verdeutlichen woran dies liegt, ist auf der linken Seite der Abbildung das Histogramm des Schätzfehlers  $G(\tilde{\kappa}, \kappa_{EM})$  für  $L = 14$  Realisierungen gezeigt. Man sieht hier deutlich, dass das EM Verfahren für einen hohen Prozentsatz von Realisierungskonfigurationen gegen ein lokales Minimum konvergiert. Dieses Verhalten ist beim progressiven Schätzer nicht zu beobachten.

■

Das hier vorgestellte Verfahren zur Dichteschätzung kann nun, wie bereits erwähnt, für die Identifikation stochastischer Rauscheinflüsse verwendet werden. Dies ist beispielsweise für die generativen Systemmodelle, wie sie durch das Verfahren in Kapitel 4 verwendet werden, hilfreich. In Teil II wird das Verfahren zur Identifikation von Messrauschen verwendet. Dazu wurden Realisierungsdaten eines akustischen Trackingsystems aufgezeichnet und mit der neuen Methode verarbeitet.

Durch eine Erweiterung des hier vorgestellten Dichteschätzverfahrens auf den mehrdimensionalen Fall lässt sich die Methode auch zur Identifikation von probabilistischen Systemmodellen, wie sie in Kapitel 3 verwendet werden, nutzen. Hierbei kann mit Hilfe des Verfahrens eine Verbunddichte über Ein- und Ausgang des Systems geschätzt werden. Aus dieser Verbunddichte lässt sich dann das probabilistische Modell durch einen progressiven Ansatz bestimmen. Dies wurde bereits in einer Studienarbeit [30] gezeigt.

## Teil II

# Anwendung: Intentionserkennung



# Konzept und Fallstudie

Als eine mögliche Anwendung für kaskadierte stochastische Systeme soll nun die Intentionserkennung in der Mensch-Roboter Kooperation betrachtet werden. Die Motivation zur Erkennung von menschlichen Intentionen durch einen Roboter entstammt dem Wunsch, einen Menschen in einer alltäglichen Umgebung optimal zu unterstützen. Ein Beispielszenario hierfür ist die Kooperation mit einem humanoiden Serviceroboter im Haushalt. Eine Erkennung der Intention durch den Roboter würde zum einen eine intuitivere Kommunikation mit dem Roboter erlauben, zum anderen würde eine implizite Kommandierung ermöglicht. Dies liegt in der Tatsache begründet, dass Menschen es von der Kooperation mit anderen Menschen gewohnt sind, deren Intentionen zu erkennen und das Gleiche von ihrem Gegenüber erwarten. Die Übertragung dieses Verhaltens auf die Mensch-Roboter Kooperation wird in diesem Teil vertieft.

Die menschliche Intention ist ein verdeckter Zustand. Das heißt, sie ist nicht direkt beobachtbar, da sie lediglich einen mentalen Zustand, gewissermaßen „im Kopf“ des Menschen darstellt. Was allerdings direkt beobachtbar ist, sind die Handlungen und Aktionen des Menschen, von denen man annehmen kann, dass sie durch dessen Intention angetrieben oder verursacht werden. Daher ist es möglich, von diesen Handlungen und Aktionen auf die zugrunde liegende Intention zurückzuschließen.

Um das Rückschließen zu ermöglichen, wird ein Modell benötigt, das die Verbindung von Intentionen zu Aktionen beschreibt. Genauer gesagt muss ein Modell des Menschen, oder ein Modell für menschliches Verhalten, verwendet werden. Ein solches Modell lässt sich durch menschliche Experten sehr gut aus Teilsystemen aufbauen. Aus diesem Grund lässt sich der Ansatz der kaskadierten Systeme hier sehr gut anwenden.

Das hier verwendete Konzept der Intentionserkennung ist eng verwandt mit der Planerkennung und wurde auf der *ICINCO 2005* [97] vorgestellt. Die Intentionserkennung wird hier im Rahmen der Mensch-Roboter-Kooperation behandelt.

## 6.1 Stand der Technik

Die Mensch-Roboter-Kooperation ist ein wichtiges Feld im Bereich der Forschung zu humanoiden Robotern. Dabei wird heute ein starker Fokus auf soziale Aspekte der Interaktion von

Menschen und Robotern gelegt [14, 15]. Laut Watt [105] tendieren Menschen dazu, Maschinen zu anthromorphisieren, d.h. sie zu vermenschlichen. Je größer die Ähnlichkeit der Maschine zu einem Menschen wird, desto mehr menschenähnliches Verhalten wird von der Maschine erwartet [31]. Dies geht Hand in Hand mit einer steigenden Erwartung an die kognitiven Fähigkeiten der Maschine, bzw. des Roboters. Mit anderen Worten, Menschen erwarten, dass sich humanoide Roboter verhalten wie sie aussehen – wie Menschen.

Eine hauptsächliche Herausforderung in der Mensch-Mensch-Kooperation ist die Einschätzung des Kooperationspartners. Nur wenn ein Mensch die Absichten seines Gegenübers optimal einzuschätzen vermag, ist eine gute Kooperation möglich. Durch die Anthromorphisierung von Robotern wird diese Fähigkeit auch in der Mensch-Roboter-Kooperation relevant [23]. Menschen erwarten, dass der Roboter ihre Intentionen erkennt, da sie dieses Verhalten von menschlichen Kooperationspartnern gewöhnt sind. Auf der anderen Seite kann ein Robotersystem vielfältige Informationen über sein Gegenüber erlangen, wenn es dessen Intentionen berücksichtigt. Aus Sicht eines Designers von humanoiden Robotern ermöglicht dieses Wissen eine neue Qualität in der high-level Taskplanung.

Das Problem der Intentionserkennung wird in verschiedenen Gebieten der Mensch-Maschine-Interaktion behandelt. Klassische Ansätze sind in der Literatur zur Planerkennung zu finden [63, 37]. Diese Ansätze sind auch im Bereich der Computerspiele sehr populär [34, 2]. In der Robotik wird die Intentionserkennung in verschiedenen Kontexten betrachtet. Ein Beispiel aus der mobilen Robotik zeigt, dass Intentionserkennung zur Verbesserung der Steuerung eines autonomen Rollstuhls verwendet werden kann, um die Annäherung an Objekte zu ermöglichen, die sonst als Hindernis erkannt worden wären [103]. Im „Programmieren durch Vormachen“-Paradigma [111] wird Intentionserkennung als Segmentierung einer Benutzerdemonstration, sowie der Klassifizierung von Griffen, wie der Mensch sie beabsichtigt hat, verstanden.

Da Menschen es gewöhnt sind mit Unsicherheiten und Ambiguitäten in ihrem täglichen Leben umzugehen, ist es sehr natürlich, probabilistische Ansätze zur Intentionserkennung zu verwenden. Ein bekanntes Beispiel für probabilistische Methoden in der Intentionserkennung ist das Lumière-Projekt [48] von Microsoft Research. Hier werden die Ziele eines Menschen anhand von Aktionen an einem Computer mit Bürosoftware erkannt. Zur Modellierung der Abhängigkeiten von Benutzerzielen und -aktionen werden in diesem System Bayesnetze verwendet. Aus diesem Projekt ist der Assistent in Microsofts Officepaket entstanden.

## 6.2 Vorgehen

Um das Vorgehen bei der Intentionserkennung zu erläutern, muss zunächst der Begriff der Intention geklärt werden. Es handelt sich dabei um ein Konzept, das beispielsweise in der Philosophie und in der Psychologie betrachtet wird. Hier einige Definitionen des Begriffs Intention:

[109][lateinisch] *Absicht, Bestreben, Vorhaben, Plan, klarer Wille, Zielsetzung, Hinwendung auf ein Ziel.*

*Adjektiv: intentional, zweckbestimmt, zielgerichtet, absichtlich.*

[82] *Eine Absicht (lat. intentio) oder Intention ist das bewusste Streben auf ein Ziel und auch das, worauf es bei einer Willenshandlung abgesehen ist, das Ziel selbst.*

Für diese Arbeit soll als Definition folgendes genügen:

**Definition 6.1** *Die Intention ist eine bewusste Absicht bzw. Hinwendung auf ein Ziel.*

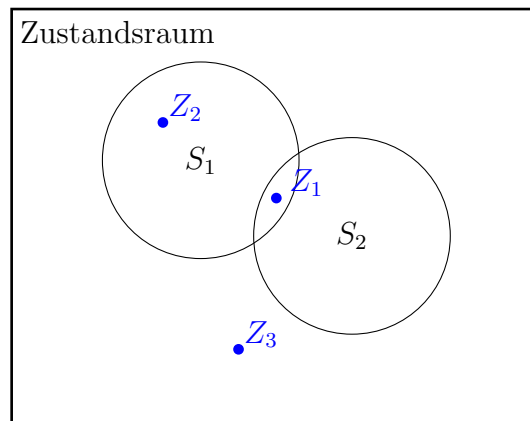
Daher kann die Intentionserkennung auch als Absichtserkennung bezeichnet werden. Über die reine Planerkennung geht die Intentionserkennung hinaus, da sie sowohl Ziele als auch Pläne berücksichtigt. Planerkennung betrachtet nur die Abfolge von Aktionen, wohingegen die Zielerkennung nur das Ziel der menschlichen Handlung berücksichtigt. Die Intentionserkennung vereint diese beiden Ansätze.

Da Intentionen bzw. Absichten sich im Allgemeinen nicht direkt äußern, wird eine direkt beobachtbare Größe benötigt, die mit der Intention im Zusammenhang steht. In der Philosophie wird der Zusammenhang zwischen Intentionen und Handlungen des Menschen untersucht. Hier sind vor allem die Arbeiten von Anscombe [5] und Davidson [24] zu nennen. Anscombe stellt den Zusammenhang zwischen Aktionen und deren Grund her, wohingegen Davidson die Ursache-Wirkungs-Beziehung zwischen Gründen und Handlungen betrachtet. Aus diesen Betrachtungen wird klar, dass die menschliche Handlung in direktem Zusammenhang mit der menschlichen Intention steht, ja sogar durch sie zu erklären ist. Intention ist also Ursache menschlicher Handlung.

Die menschliche Handlung lässt sich durch Folgen von Aktionen beschreiben. Dies sind einzelne Taten, die der Mensch ausführt, um sich dem Ziel seiner Intention zu nähern. Für die Intentionserkennung sollen hier elementare *pick-and-place*-Aufgaben, wie „*Der Mensch nimmt die Tasse*“, als Aktion bezeichnet werden. Nach [5] werden nur intendierte Handlungen betrachtet, also Aktionsfolgen, denen eine Intention zu Grunde liegt. Unintendierte Handlungen sind beispielsweise Unfälle wie „*Der Mensch stolpert*“. Um nun von den Aktionen auf die zugrunde liegende Intention rückschließen zu können, wird ein Modell benötigt. In diesem Fall handelt es sich um ein Modell des menschlichen Verhaltens.

Da der Mensch von seiner Umwelt beeinflusst wird, hat diese natürlich auch eine Wirkung auf dessen Intention. Daher sollte der Kontext des Menschen ebenfalls in dem Modell berücksichtigt werden. Der Kontext beschreibt hier den relevanten Teil der Umwelt, der eine Auswirkung auf die Intention haben kann.

Im Folgenden wird nun eine systematische Definition der Begriffe Situation und Kontext gegeben, wie sie für die Intentionserkennung verwendet werden sollen. Dazu wird das Venndiagramm in Abbildung 6.1 als schematische Hilfe verwendet. Zunächst wird die Umwelt des Menschen



**Abbildung 6.1:** Venndiagramm zur Definition von Situation und Kontext.  $Z_1, Z_2$  und  $Z_3$  bezeichnen Zustände,  $S_1$  und  $S_2$  Kontexte

als Zustandsraum betrachtet. Dieser Zustandsraum beinhaltet sämtliche Konfigurationen von Objekten in der Umwelt sowie die Konfiguration des Menschen und des Roboters. Unter Konfigurationen sind beispielsweise Positionen und Orientierungen von Objekten zu verstehen. Eine mögliche Gesamtkonfiguration wird im Folgenden als Zustand  $Z$  bezeichnet. Eine Situation ist eine Menge von Zuständen, die über eine Bedingung definiert wird ( $S$ ). Beispiele für solche Mengen sind: „*Es stehen vier Teller auf dem Tisch*“ ( $S_1$ ) oder „*Der Mensch steht am Tisch*“ ( $S_2$ ). Bei der Beschreibung von Situationen ist die exakte Position eines Objekts unter Umständen nicht relevant, weshalb eine Menge von Zuständen die geforderte Bedingung erfüllt. Als Beispiel sind in Abbildung 6.1 die Situationen  $S_1$  und  $S_2$  angegeben. Hier sieht man, dass sich die Mengen auch überschneiden können. Ein Kontext wird nun durch den aktuell vorherrschenden Zustand bestimmt. Der Kontext in dem sich der Mensch befindet wird als die Menge der Situationen, in denen der aktuelle Zustand enthalten ist, definiert. Es handelt sich hierbei also um eine Menge von Mengen. Im Beispiel wäre dies die Menge aus „*Es stehen vier Teller auf dem Tisch*“ und „*Der Mensch steht am Tisch*“. In Abbildung 6.1 ist dies beispielsweise der Kontext von  $Z_1$  der durch  $\{S_1, S_2\}$  gegeben ist. Der Kontext von  $Z_2$  ist  $\{S_1\}$  und der Kontext von  $Z_3$  ist  $\{\emptyset\}$ , die leere Menge. Letztere kann als unbekannte Situation betrachtet werden.

Ferner beeinflusst der Mensch mit seinen Aktionen auch die Umwelt selbst. Dies ist insbesondere dann wichtig, wenn die Aktionen selbst schwer zu erkennen sind. In solchen Situationen lässt sich dann zumindest von der Beobachtung der Umwelt ein Rückschluss auf die ausgeführte Aktion ziehen. Dies kann auch als Aktionserkennung bezeichnet werden.

Eine Skizze des Menschmodells ist in Abbildung 6.2 zu sehen. Dieses Modell zeigt die Intention als versteckten Zustand „im Kopf“ des Menschen. Kontext, Aktionen und Umwelt sind beobachtbare Größen.

Gemäß der philosophischen Betrachtung handelt es sich hierbei um ein gerichtetes Modell. Das heißt, es gibt Ursachen und Wirkungen. Diese Modellierung wird durch die Richtung der Pfeile verdeutlicht. Diese Kausalität legt eine systemtheoretische Modellierung, wie in Teil I eingeführt, nahe. Eine Vorwärtsmodellierung ist demnach hier bereits natürlich vorgegeben.



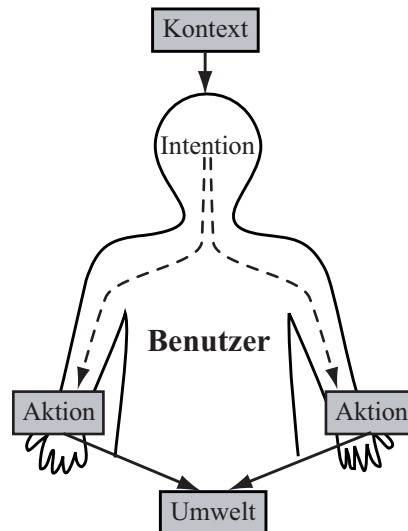


Abbildung 6.2: Modell des Menschen für die Intentionserkennung.

Ein solches Modell beinhaltet allerdings viele Unsicherheiten. So ist der Zusammenhang von Intentionen und Aktionen im Allgemeinen nicht deterministisch und von Mensch zu Mensch unterschiedlich ausgeprägt. Außerdem sind die Aktionen eventuell nicht exakt beobachtbar und der Einfluss des Kontexts auf das Gesamtsystem ist unsicher.

Die Struktur der Skizze in Abbildung 6.2 legt die Verwendung von kaskadierten Systemen bereits nahe, da der Kontext und die Aktionen bei gegebener Intention voneinander unabhängig sind. Abbildung 6.3 zeigt nun ein generisches Modell für die Intentionserkennung in einer Bayesnetzrepräsentierung. Diese Form der Darstellung wird hier zunächst gewählt um die Ähnlichkeit zum skizzierten Modell in Abbildung 6.2 zu verdeutlichen. Die Form der Knoten soll hierbei

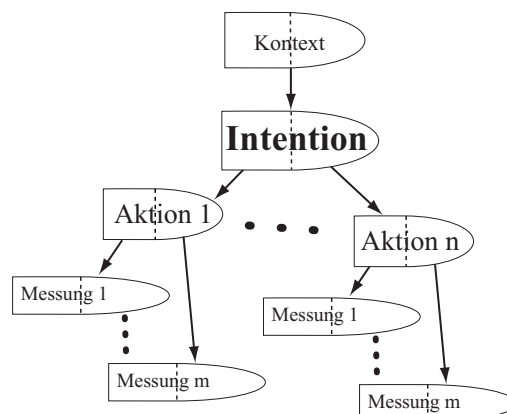


Abbildung 6.3: Generisches Modell für die Intentionserkennung.

verdeutlichen, dass die jeweiligen Zufallsvariablen kontinuierlich (oval) oder diskret (rechteckig) sein können. Häufig werden Intentionen und Aktionen durch symbolische Beschreibung, wie „*öffnet die Spülmaschine*“ angegeben, was durch diskrete Zufallsvariablen ausgedrückt werden kann. Die Sensormessungen, die von solchen Aktionen hervorgerufen werden, sind allerdings häufig kontinuierlich, wie beispielsweise Distanzen vom Menschen zu Objekten. Diese können

durch kontinuierliche Zufallsvariablen beschrieben werden. Es handelt sich hierbei folglich um ein hybrides Modell.

Um nun die Schätzung der Intention vorzunehmen, muss der Einfluss sämtlicher direkt beobachtbarer Größen in das Modell eingespeist und die entsprechenden Bayes'schen Inferenzschritte, wie in Abschnitt 2.2 beschrieben, ausgeführt werden. Als direkt beobachtbar werden hier die Sensormessungen sowie der jeweilige Kontext angenommen. Man kann dies auch als eine Informationsfusion der bekannten Beobachtungen im Sinne von [9] betrachten.

Ein weiterer Aspekt, der im generischen Modell berücksichtigt werden soll, ist die Dynamik. Da zu einer Intention häufig eine Folge von Aktionen gehört und eine Aktion häufig eine Sequenz von Sensormessungen erzeugt, muss dieser zeitliche Aspekt mit berücksichtigt werden. Dazu kann eine dynamische Bayesnetzdarstellung (DBN), wie in Abbildung 6.4 gezeigt, verwendet werden. Da es sich hierbei um eine zeitdiskrete Modellierung handelt, wird das generische Modell

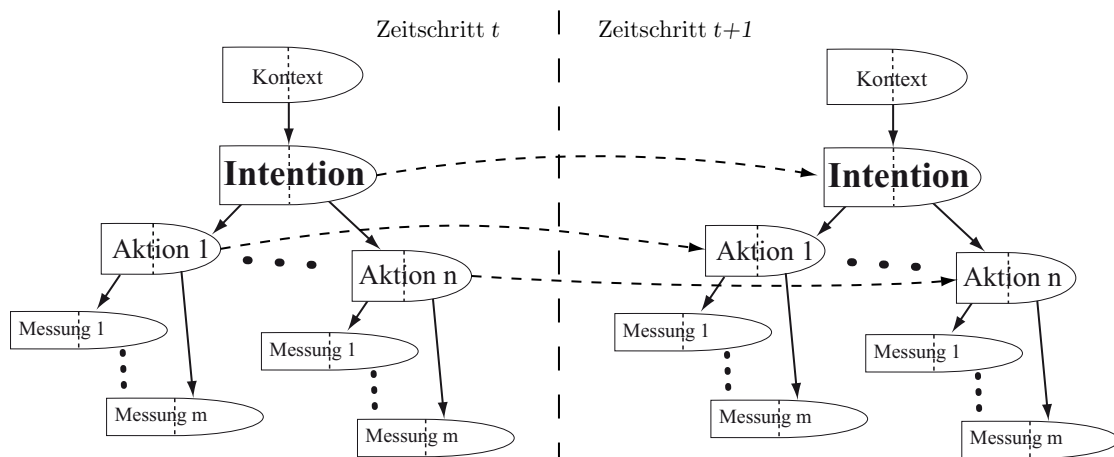


Abbildung 6.4: Dynamisches Bayesnetzmodell für die Intentionserkennung.

aus Abbildung 6.3 für zwei Zeitschritte  $t$  und  $t + 1$  dargestellt. Die zeitlichen Abhängigkeiten sind durch gestrichelte Kanten angegeben. Hierbei wird die Markov-Annahme erster Ordnung verwendet. D. h. jeder Zustand ist nur von seinem unmittelbaren Vorgängerzustand abhängig. Diese Art der Darstellung wird in der Literatur auch als  $2T$ -DBN bezeichnet [68].

Eine Blockdiagrammdarstellung des generischen Modells ist in Abbildung 6.5 zu sehen. Hier

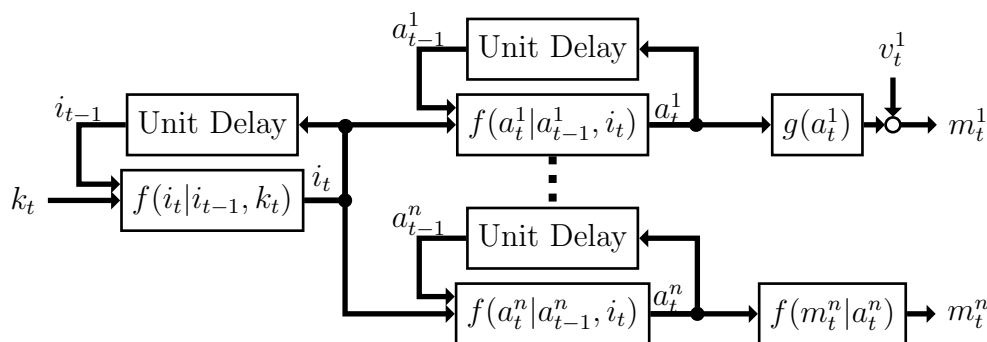


Abbildung 6.5: Blockdiagrammmodell für die Intentionserkennung.

werden die entsprechenden Zufallsvariablen abgekürzt durch  $k$  für Kontext,  $i$  für Intention,  $a$  für Aktion und  $m$  für Messung bezeichnet. Ferner ist der jeweilige Zeitindex durch  $t$  annotiert. Die verschiedenen Aktionen und Messungen sind durch einen oberen Index gekennzeichnet. Der Übersicht wegen wird hier für jede Aktion nur eine Messung angenommen. Dies muss im Allgemeinen nicht so sein. Für die Messung  $m_t^1$  ist ein generatives Modell mit additivem Rauschen  $v_t$  angegeben, für die Messung  $m_t^n$  ein probabilistisches Modell. Dies dient der Verdeutlichung, dass beide Modellierungsarten verwendet werden können. Darüber hinaus ist auch vorstellbar, dass die Messungen über weitere Kaskaden ermittelt werden.

In Abbildung 6.6 ist nun der Schätzer für dieses generische Modell angegeben. Der Schätzer

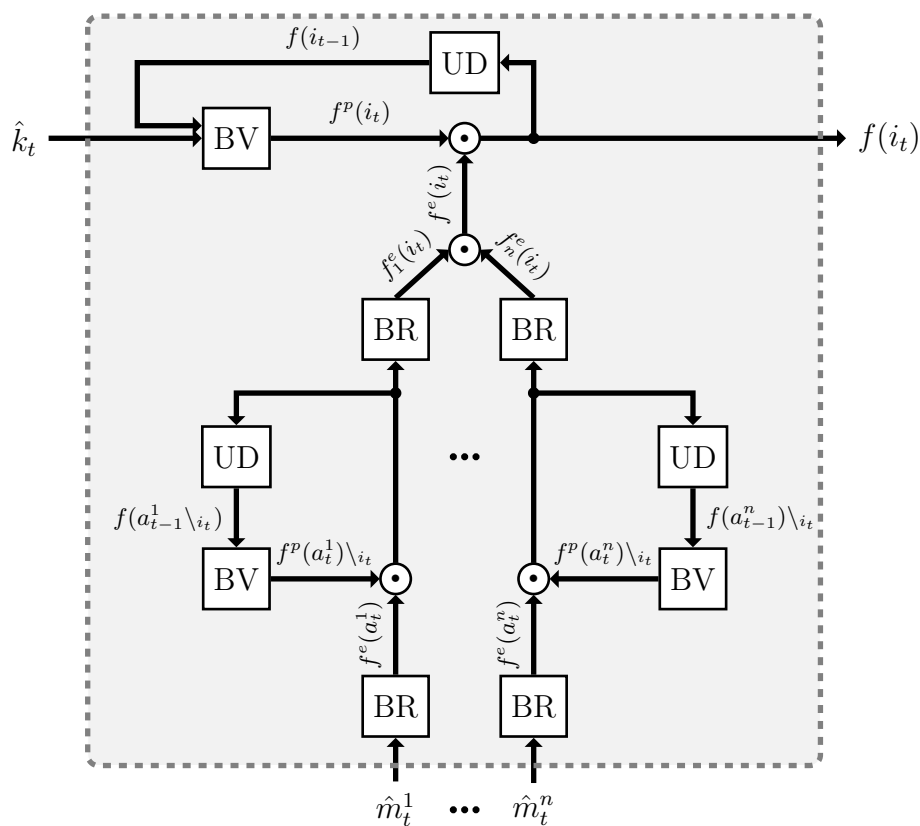


Abbildung 6.6: Schätzer für das generische Intentionserkennungsmodell.

erhält als Eingabe den aktuellen Kontext  $\hat{k}_t$  sowie die Messungen  $\hat{m}_t^j$  für  $j = 1, \dots, n$ . Als Ausgang liefert er die Schätzung der vorliegenden Intention in Form der Wahrscheinlichkeitsdichte  $f(i_t)$ . Die Inferenz basiert auf den in Kapitel 2.2 vorgestellten Berechnungsvorschriften. Die mit  $BV$  und  $BR$  bezeichneten Blöcke stehen für die Bayes'sche Vorwärts- und Rückwärtsinferenz. Die mit  $UD$  bezeichneten Blöcke beschreiben eine Einheitsverzögerung (engl.: Unit Delay). Die Operatoren mit Punkt sind Multiplikationen. Da als einzige Marginaldichte die Dichte über die Intention berechnet wird, kann beim Eingang des Bayes'schen Vorwärtsschritts für die Aktionen die Intention weggelassen werden. Diese Information würde sonst bei der Berechnung von  $f(i_t)$  doppelt berechnet.

Dieser Schätzer kann nun für konkrete Szenarien instanziiert werden. Dazu muss zuerst festgelegt werden, ob die Variablen kontinuierlich oder diskret sind. Danach müssen die entsprechenden probabilistischen oder generativen Modelle festgelegt werden. Dies kann entweder datengetrieben oder durch Experten erfolgen, was beispielhaft in Kapitel 7 gezeigt wird.

### 6.3 Fallstudie

Um herauszufinden wie gut ein Roboter die Intentionen eines Menschen erkennen können sollte, wurde eine Fallstudie durchgeführt, in der zunächst geklärt wurde, wie gut Menschen die Intention anderer Menschen erkennen. Hierzu wurde in einem Experiment untersucht, welche Informationen der Mensch benötigt, um zu erkennen, was das Ziel einer anderen Person ist.

In dem Experiment wurden von verschiedenen Personen unterschiedliche Tätigkeiten in einer Küche ausgeführt und mit einer Videokamera aufgezeichnet. Den Personen wurde dabei jeweils eine Aufgabe gestellt, die sie in der Küche umsetzen mussten. Dazu wurden die folgenden Aufgaben formuliert:

- Sie möchten
- die Spülmaschine beladen
  - das Geschirr abwaschen
  - kochen
  - den Tisch decken

Die Gruppe der Präsentierenden bestand aus vier Personen, die jeweils alle vier Aufgaben in aufeinanderfolgenden Aufzeichnungen ausführten. Ein Bild aus einer exemplarischen Aufzeichnung einer Person, die kochen möchte, ist in Abbildung 6.7 zu sehen.



**Abbildung 6.7:** Videoaufzeichnung einer Person in der Küche. Diese Person möchte kochen.

Ziel des Experimentes war es, eine andere Gruppe von Personen schätzen zu lassen, um welche Intention es sich jeweils handelt. Dazu wurden Interviews mit acht verschiedenen Personen durchgeführt, denen eine Auswahl der aufgezeichneten Videos gezeigt wurde. Das jeweilige Video wurde immer für 10 Sekunden gespielt und dann angehalten. Die interviewte Person wurde anschließend gefragt, welche Intention bei der gezeigten Person ihrer Meinung nach vorliegt. Danach wurde das Video wieder 10 Sekunden weitergespielt, angehalten und die Frage fortgesetzt.

Diese Prozedur wurde bis zum Ende des Videos wiederholt. Die Dauer der Aufzeichnungen lag zwischen 80 und 100 Sekunden. Den befragten Personen wurden keine Vorgaben über mögliche Intentionen gegeben, sie mussten frei antworten.

Das Resultat dieses Experiments war erstaunlich, da alle befragten Personen in der Lage waren, die korrekte Antwort auf die Frage spätestens nach der zweiten Frage zu geben. Also nach 20 Sekunden. Hierbei zeigt sich bereits die hohe Leistung, die Menschen bei der Erkennung der Intention anderer Menschen an den Tag legen.

Um nun die Leistungen des Menschen mit den Leistungen eines technischen Systems vergleichen zu können, wurden die Videos transkribiert. D.h. aus den Videos wurde eine Sequenz klassifizierter Aktionen der präsentierenden Person generiert. Diese Transkription wurde von Hand generiert und führte zu folgenden Aktionen:

- Öffnet Spülmaschine
- Nimmt Gegenstand aus der Spüle
- Schließt Spülmaschine
- Lässt Wasser laufen
- Nimmt Gegenstand aus der Spülmaschine
- Nimmt Spülmittel
- Stellt Gegenstand in die Spülmaschine
- Stellt Topf auf den Herd
- Öffnet Schrank
- Nimmt Essen
- Schließt Schrank
- Gibt Essen in den Topf
- Nimmt Gegenstand aus dem Schrank
- Gibt Gegenstand in den Topf
- Stellt Gegenstand in den Schrank
- Stellt Gegenstand auf den Tisch
- Stellt Gegenstand auf die Arbeitsfläche
- Stellt Essen auf den Tisch
- Nimmt Gegenstand von der Arbeitsfläche
- Benutzt den Mülleimer
- Stellt Gegenstand auf die Spüle
- Öffnet den Kühlschrank
- Nimmt Gegenstand von der Spüle
- Schließt den Kühlschrank
- Stellt Gegenstand in die Spüle

Als Idee hinter dieser Transkription stand die Frage: „Was würde ein Mensch sagen, der diese Aktion des anderen Menschen beschreiben sollte?“ Es stellte sich heraus, dass die Aktionen, die als relevant eingestuft wurden, letztendlich alle als *pick-and-place* Operationen zu beschreiben waren. Die Objekte, die bei diesen Aktionen involviert sind, lassen sich in zwei Klassen einteilen:

- *Gegenstände*: Diese Klasse umfasst alle Arten von Besteck (Messer, Gabel, Löffel, etc.) und Geschirr (Teller, Tassen, Schüsseln, etc).
- *Essen*: Diese Klasse umfasst alle essbaren und trinkbaren Dinge in der Küche.

Einige Gegenstände wie Spülmittel konnten keiner Klasse zugeordnet werden und werden daher separat behandelt.

Die transkribierten Aktionssequenzen wurden, wie zuvor die Videosequenzen, den befragten Personen präsentiert. Um ein Vorwissen über die Präsentation auszuschließen, wurde den Personen eine andere Präsentation vorgelegt als beim ersten Interview. Der zeitliche Abstand zwischen dem ersten und zweiten Interview betrug ungefähr vier Wochen.

Die Aktionen wurden in schriftlicher Form durch die graphische Oberfläche in Abbildung 6.8 präsentiert. Den Personen wurden im Gegensatz zum vorherigen Interview vier mögliche Inten-

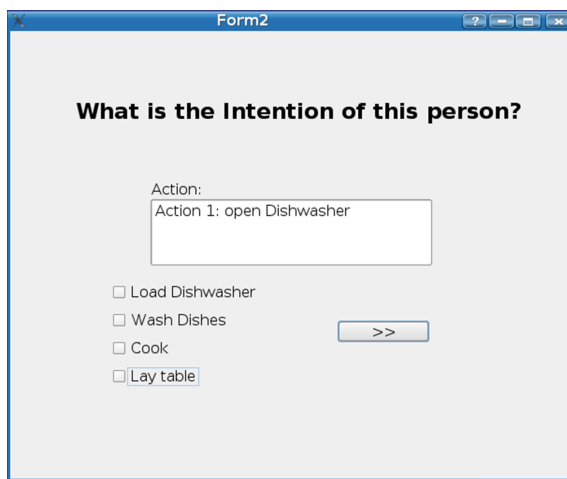


Abbildung 6.8: Graphische Oberfläche zur Präsentation transkribierter Aktionsfolgen.

tionen zur Auswahl gegeben. Es war hier auch eine Mehrfachauswahl möglich, um eine gewisse Unsicherheit ausdrücken zu können. Jeder Person wurden vier Präsentationen gezeigt, wobei alle acht Personen dieselben Präsentationen sahen.

Als Ergebnis der Befragung standen vier mal acht Sequenzen von Einschätzungen der beobachtenden Personen zur Verfügung. Diese Sequenzen wurden über die acht Personen gemittelt und normiert. Das Ergebnis ist in Abbildung 6.9 zu sehen.

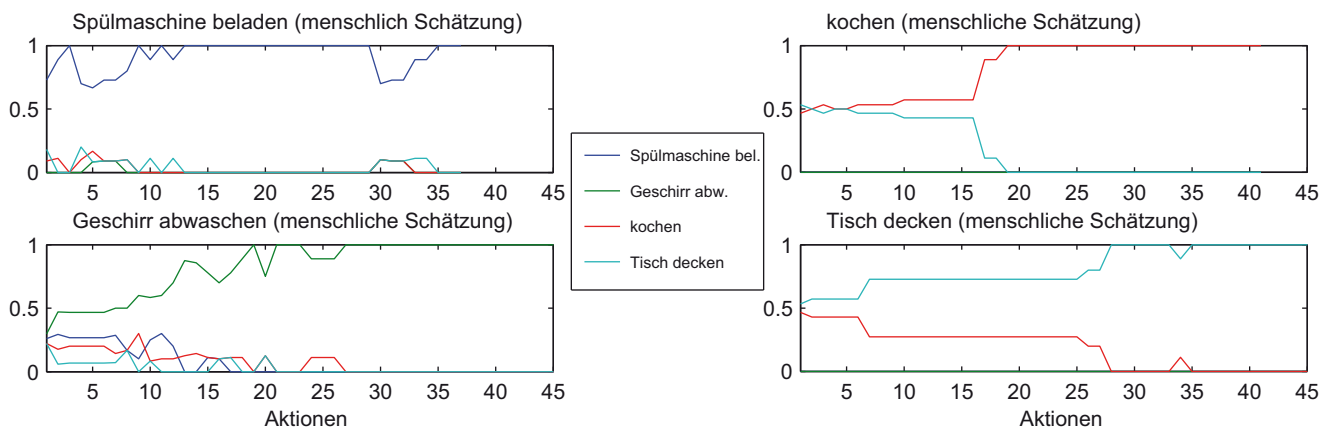


Abbildung 6.9: Menschliche Intentionsschätzung aus transkribierten Aktionsdaten.

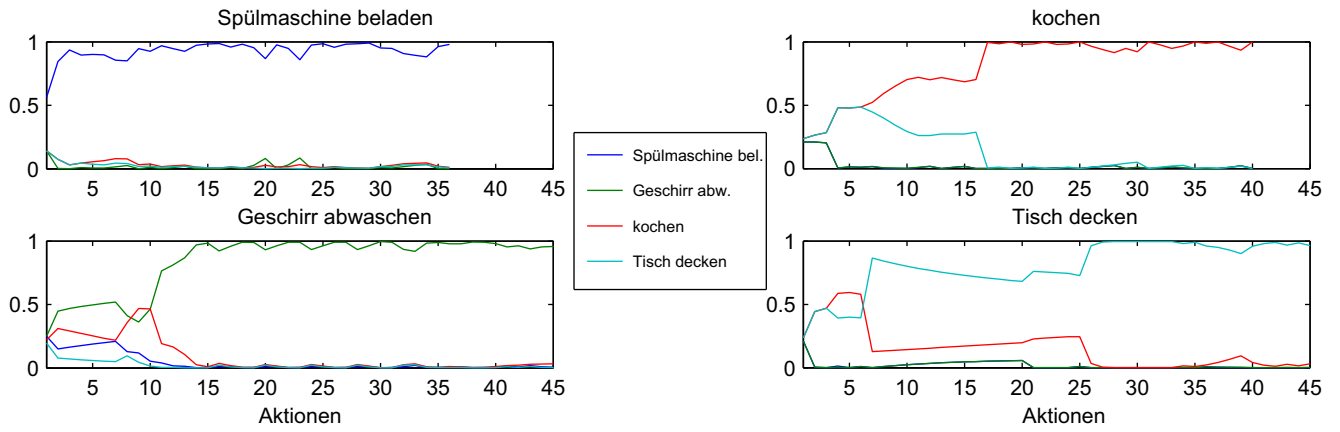
Die Überschriften der Plots zeigen die jeweils vorliegende wahre Intention des präsentierenden Menschen. Jede Grafik zeigt die gemittelte und normierte Einschätzung der acht Personen über die Folge der Aktionen. Für jeden Zeitpunkt (jede Aktion) liegt also eine Schätzung über die vier möglichen Intentionen vor, die durch die Werte der vier Graphen gegeben ist. Die Summe der Graphen ergibt zu jedem Zeitpunkt Eins. Die größte mögliche Unsicherheit wäre also gegeben, wenn alle Graphen den Wert  $\frac{1}{4}$  annehmen würden, die größte Sicherheit, wenn ein Graph den Wert Eins annimmt.

Die Ergebnisse zeigen eine hohe anfängliche Unsicherheit in der Präsentation von „*Geschirr abwaschen*“ (links unten). Diese verschwindet aber über die Zeit gänzlich. Bei der Präsentation von „*Spülmaschine beladen*“ (links oben) hingegen sind sich die Beobachter von Beginn an sehr sicher. Die Präsentationen von „*kochen*“ (rechts oben) und „*Tisch decken*“ (rechts unten) scheinen sehr ähnlich zu sein, da in beiden Fällen anfangs eine große Unsicherheit zwischen diesen beiden Möglichkeiten herrscht.

Um nun die Möglichkeiten der modellbasierten Schätzung von Intentionen auszuloten, wurde ein einfaches diskretes, kaskadiertes System aufgestellt. Dieses System enthält eine Variable für die Intention als verdeckten Eingang sowie eine Variable für jede mögliche Aktion aus der Transkription als Ausgang. Die Gesamtkaskade hat also einen verdeckten Eingang und 25 beobachtbare Ausgänge. Die bedingten Wahrscheinlichkeiten wurden dabei von Hand festgelegt. Dazu wurde jeweils die Frage gestellt: „Gegeben die Intention ist  $x$ , wie wahrscheinlich ist es, dass die Aktion  $y$  ausgeführt wird?“ Da Menschen für das betrachtete Küchenszenario als Experten angesehen werden können, sind sie sehr gut in der Lage ein solches Modell aufzustellen. Außerdem wurde ein dynamisches Modell verwendet, um die Historie der Schätzung zu berücksichtigen. Dazu wurde eine Markovbedingung erster Ordnung für die Intention angenommen. Das heißt, die Intention zum Zeitpunkt  $t + 1$  hängt nur von der Intention zum Zeitpunkt  $t$  ab. Als Modell wurde dafür angenommen, dass der Mensch zu 95% bei seiner Intention bleibt und mit 1% Wahrscheinlichkeit in eine andere Intention wechselt. Um das Modell einfach zu halten, wurden die Aktionen bei gegebener Intention als unabhängig angenommen. Diese Annahme ist im Allgemeinen nicht immer gültig. Deshalb wird im folgenden Kapitel eine Methode vorgestellt, wie auch komplexere Abhängigkeiten betrachtet werden können.

Die Ergebnisse der Schätzung mit dem einfachen Modell sind in Abbildung 6.10 zu sehen. Die Grafiken sind wieder so wie in Abbildung 6.9 zu interpretieren: die Grafiken zeigen den Verlauf der Wahrscheinlichkeiten der einzelnen Intentionen über der Folge von Aktionen. Auch hier sind die Wahrscheinlichkeiten normiert. Es ist bemerkenswert, dass die Ergebnisse des Schätzers der Einschätzung der Befragten Personen sehr ähnlich sind. Das heißt, an den Stellen an denen die Menschen eine große Unsicherheit zeigen, resultiert auch eine große Unsicherheit im Ergebnis des Schätzers. An Stellen die den menschlichen Beobachtern sofort klar waren, entscheidet der Schätzer auch sehr sicher zu Gunsten der korrekten Intention.

Dieses Beispiel zeigt, dass die modellbasierte Intentionsschätzung durchaus in der Lage ist, die Leistung eines Menschen bei der Intentionserkennung zu erreichen. Für komplexere Szenarien sind allerdings komplexere Modelle als das hier betrachtete nötig. Insbesondere, da die Aktionen



**Abbildung 6.10:** Modellbasierte Intentionsschätzung aus transkribierten Aktionsdaten.

selbst auch erkannt werden müssen, wodurch weitere Unsicherheit in die Schätzung mit einfließt. Aus diesem Grund werden im folgenden Kapitel weitere Betrachtungen zu geeigneten Modellen für die Intentionserkennung angestellt.



# Modelle und Evaluationsumgebung

## 7.1 Entkopplung und Komplexitätsreduktion

Bei der Konstruktion von kaskadierten Systemmodellen ist die Annahme der Unabhängigkeit von Teilsystemen essentiell. Allerdings ist eine solche Annahme häufig nicht korrekt, was sich in schlechten Inferenzergebnissen niederschlägt. Ein exaktes Modell, das alle Abhängigkeiten berücksichtigt, besitzt aber im Allgemeinen eine sehr komplexe Struktur, die beispielsweise Zyklen enthalten kann.

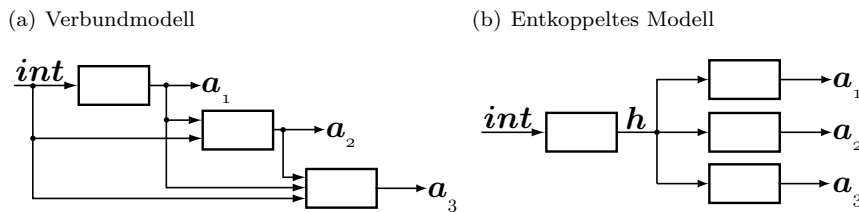
Eine gängige Methode um solche Abhängigkeit aufzulösen ist die Einführung so genannter versteckter Variablen, die die Abhängigkeiten des Verbundraums subsumieren [77]. Ein Nachteil dieser Methode ist, dass der Verbundraum extrem groß werden kann, was auch wieder zu einer schlechten Handhabbarkeit führt. Daher werden Methoden zur Reduktion des Zustandsraums der versteckten Variablen benötigt, wobei die Ergebnisse der Inferenz in diesem kaskadierten System im Idealfall nicht vom unreduzierten Fall abweichen sollten.

In der Literatur existieren Verfahren, um geeignete versteckten Variablen zu finden [33, 13], wenn diese nicht bereits bekannt sind. Andere Verfahren beschreiben, wie die Kardinalität der versteckten Variablen bestimmt werden [32, 66] und deren Parameter aus Daten gelernt werden [87, 59] können. Ein generischer Ansatz zur Reduktion des Zustandsraums von Variablen wird in [108] vorgestellt, wobei dort der Hauptaugenmerk jedoch auf diskrete Zustände, die aus der Quantisierung kontinuierlicher Variablen entstehen, liegt.

Im Weiteren wird der Fall von diskreten Systemen am Beispiel der Intentionserkennung betrachtet. Dies ist interessant, da selbst hier die Bestimmung der Kardinalität der reduzierten Variable, also die Anzahl der Zustände, sowie die entsprechenden bedingten Wahrscheinlichkeiten nicht ohne einen erheblichen Aufwand zu bestimmen sind. Für den kontinuierlichen Fall können Clusterverfahren, wie sie in [101] vorgestellt wurden, verwendet werden. Das hier vorgestellte Verfahren wurde auf der *IROS 2007* [92] publiziert und stellt durch die Konstruktion eines kaskadierten Systems für die Intentionserkennung, basierend auf Expertenwissen, eine alternative Vorgehensweise zu datengetriebenen Verfahren dar. Es werden daher keine Lernverfahren für Struktur oder Parameter des Modells benötigt.

Bei der Behandlung der Intentionserkennung für Menschen kann man im Allgemeinen davon ausgehen, dass menschliche Experten eine gute Vorstellung vom Zusammenhang zwischen Intentionen und Aktionen haben. Man kann dies auch als Modellwissen bezeichnen. Solche Modelle können beispielsweise als eine Abbildung von einer Intention auf eine Aktion angegeben werden. Basierend auf solchen Abbildungen sollen probabilistische Modelle abgeleitet werden, bei denen die Intention durch eine diskrete Zufallsvariable  $int$ , mit so vielen Zuständen wie bekannten Intentionen, beschrieben wird. Aktionen werden durch binäre Zufallsvariablen  $a_i$ , mit  $i = 1, \dots, n$  beschrieben, wobei  $n$  die Anzahl der beobachtbaren Aktionen ist. Die Unsicherheit in diesem Modell wird repräsentiert durch die bedingte Wahrscheinlichkeit  $P(a_1, \dots, a_n | int)$ . Im Folgenden wird von Wahrscheinlichkeiten und bedingten Wahrscheinlichkeiten anstatt Dichten gesprochen um zu verdeutlichen, dass es sich um wertediskrete Systeme handelt.

Im Allgemeinen sind die Aktionen, gegeben die Intention, nicht unabhängig voneinander, weshalb keine unabhängigen Modelle  $P(a_i | int)$  angegeben werden können. Da Abhängigkeiten zwischen den Aktionen bestehen, muss also zunächst der komplette Verbundraum über alle beteiligten Zufallsvariablen betrachtet werden. Ein mögliches kaskadiertes System, das den kompletten Verbund beschreibt, ist in Abbildung 7.1(a) gezeigt. Dieses graphische Modell beschreibt



**Abbildung 7.1:** Kaskadiertes System für Intensions-Aktions Relationen. (a) zeigt das Verbundmodell und (b) das entkoppelte Modell.

nach der Kettenregel das Verbundsystem

$$P(int, a_1, \dots, a_n) = P(int)P(a_1|int)P(a_2|a_1, int) \cdot \dots \cdot P(a_n|int, a_1, \dots, a_{n-1}) .$$

Diese Darstellung hat den großen Nachteil, dass sie (ungerichtete) Zyklen enthält, wodurch das rekursive Berechnungsverfahren aus Abschnitt 2.2 unbrauchbar wird. In der Literatur sind zwar Topologien bekannt, für die eine so genannte *loopy belief propagation* garantiert konvergiert [69], für allgemeine kaskadierte Systeme ist diese Konvergenz allerdings nicht gesichert.

Ein verbreiteter Ansatz um Zyklen zu entfernen, ist das Einführen von so genannten versteckten Variablen wie in Abbildung 7.1(b) gezeigt. Dies führt zu einer Entkopplung des Systems und damit zu einer weniger komplexen Struktur, die sich durch

$$P(int)P(h|int)P(a_1|h) \dots P(a_n|h)$$

beschreiben lässt.

Diese Vereinfachung bekommt man allerdings nicht geschenkt. Der Preis, den man für die naive Anwendung dieses Ansatzes zahlen muss, ist die Komplexität der Zufallsvariable  $h$  in Form der

Anzahl ihrer Zustände. Sie muss den kompletten Verbundraum abdecken, was im gegebenen Beispiel zu  $|int| \cdot 2^n$  Zuständen führt, wobei  $|int|$  die Anzahl der bekannten Intentionen und  $n$  die Anzahl der bekannten Aktionen beschreibt. Dieses Modell wird im Folgenden als das *exakte* Modell bezeichnet.

Der Umgang mit solch großen Zustandsräumen ist für praktische Anwendungen häufig nicht durchführbar. Daher wird nun ein approximativer Ansatz verfolgt, der mit weniger Zuständen für die versteckte Variable  $\mathbf{h}$  auskommt, aber in der Inferenz vergleichbare Ergebnisse zur Verwendung des exakten Modells erzielt. Die Herleitung des approximativen Ansatzes wird von einem Beispiel begleitet. Dieses Beispiel ist bewusst einfach gehalten, um die Nachvollziehbarkeit zu gewährleisten. Außerdem erlaubt das Beispiel mit der exakten Lösung zu beginnen, die dann in die approximative Lösung überführt wird.

### Beispiel 7.1: Entkopplung und Modellreduktion

Es wird der Fall von drei bekannten Intentionen  $int_1, int_2, int_3$  und drei möglichen Aktionen  $a_1, a_2, a_3$ , die ein Mensch ausführen kann, betrachtet. Von einem menschlichen Experten erhält man eine Abbildung, die besagt, welche Aktionen ausgeführt werden, wenn eine bestimmte Intention vorliegt. Für dieses Beispiel wird die Abbildung

$$\begin{aligned} int_1 &\rightarrow a_1, a_2, \\ int_2 &\rightarrow a_2, \\ int_3 &\rightarrow a_3. \end{aligned} \tag{7.1}$$

angenommen.

Es sei darauf hingewiesen, dass die erste Intention zwei Aktionen auslösen kann. Es wird aber keine Annahme getroffen, dass diese Aktionen gleichzeitig auftreten müssen. Außerdem gilt zu beachten, dass die Aktion  $a_2$  sowohl durch die Intention  $int_1$  als auch durch die Intention  $int_2$  ausgelöst werden kann. ■

#### 7.1.1 Exakte Entkopplung

Die exakte Entkopplung durch Einführung der versteckten Variable  $\mathbf{h}$  benötigt, wie bereits erwähnt, einen Zustandsraum mit  $|int| \cdot 2^n$  Zuständen. Jeder Zustand von  $\mathbf{h}$  stellt dabei eine mögliche Kombination aus Intention und dem Auftreten oder Nichtauftreten aller Aktionen dar. Der komplette Verbundraum  $(int, \mathbf{a}_1, \dots, \mathbf{a}_n)$  wird also auf den Zustandsraum von  $\mathbf{h}$  abgebildet.

Die Kombination der Intentionen mit dem Verbundraum der Aktionen  $(\mathbf{a}_1, \dots, \mathbf{a}_n)$  kann allerdings auch mit Hilfe der bedingten Wahrscheinlichkeit  $P(a_1, \dots, a_n, |int)$  modelliert werden. Daher muss  $\mathbf{h}$  lediglich den Verbundraum der Aktionen abbilden, wodurch die Anzahl der Zustände von  $\mathbf{h}$  auf  $2^n$  verringert wird. Es handelt sich hierbei nicht um eine Approximation, da die Abbildung von Intention auf Aktionenverbund von  $\mathbf{h}$  lediglich in die bedingte Wahrscheinlichkeit  $P(h | int)$  verschoben wird, was immer noch das exakte Modell beschreibt. Allerdings

ist  $2^n$  immer noch eine sehr hohe Anzahl von Zuständen bei einer entsprechend hohen Anzahl von Aktionen.

**Beispiel 7.1: Enkopplung und Modellreduktion (Fortsetzung)**

Es wird nun eine Zufallsvariable  $h$  mit  $2^n$  Zuständen eingeführt, wobei  $n = 3$  ist. Jeder Zustand  $h_i$ , für  $i = 0, \dots, 7$  von  $h$  steht für eine Kombination der möglichen Aktionen. Jeder Verbundzustand  $(a_1, a_2, a_3)$  wird im Folgenden durch eine Binärzahl ausgedrückt. Das gleichzeitige Auftreten aller Aktionen beispielsweise durch 111. Somit erhält man folgende Abbildung:

$$\begin{aligned} h_0 &\leftarrow 000 & h_1 &\leftarrow 001 & h_2 &\leftarrow 010 & h_3 &\leftarrow 011 \\ h_4 &\leftarrow 100 & h_5 &\leftarrow 101 & h_6 &\leftarrow 110 & h_7 &\leftarrow 111 \end{aligned}$$



Basierend auf dem vom Experten angegeben Modell, müssen nun die Parameter der bedingten Wahrscheinlichkeit  $P(h | int)$  bestimmt werden. Diese werden im Allgemeinen in einer Wahrscheinlichkeitstabelle angegeben. Um dies tun zu können, muss noch ein Fall betrachtet werden, der im Modell des Experten nicht explizit angegeben ist – Der Mensch kann eine Intention im Kopf haben, führt aber zum aktuellen Zeitpunkt keine Aktion aus. Dies kann beispielsweise dann auftreten, wenn der Mensch sich in einer Transitionsphase befindet, in der er sich von einer Position zu einer anderen bewegt, um dort die relevante Aktion ausführen zu können. Um diesen „idle“-Zustand beachten zu können muss er Teil des Modells  $P(h | int)$  sein.

**Beispiel 7.1: Enkopplung und Modellreduktion (Fortsetzung)**

Die Tabelle der bedingten Wahrscheinlichkeit  $P(h | int)$  kann aus der Abbildung des Experten aus (7.1) gewonnen werden.

Für dieses Beispiel wird als Wahrscheinlichkeit für den „idle“-Zustand  $h_0$  10% angenommen. Daraus erfolgt die bedingte Wahrscheinlichkeit  $P(h_0 | int_i) = 0.1$  für  $i = 1, \dots, 3$ . Die restlichen Wahrscheinlichkeiten müssen sich demzufolge zu 0.9 addieren.

$P(h   int)$	$P(h_0)$	$P(h_1)$	$P(h_2)$	$P(h_3)$	$P(h_4)$	$P(h_5)$	$P(h_6)$	$P(h_7)$
$int_1$	1/10	0	3/10	0	3/10	0	3/10	0
$int_2$	1/10	0	9/10	0	0	0	0	0
$int_3$	1/10	9/10	0	0	0	0	0	0

Für  $int_1$  ist die Wahrscheinlichkeitsmasse über das einzelne und gleichzeitige Auftreten der Aktionen  $a_1$  und  $a_2$  verteilt, da die Regel des Experten keine Angaben darüber macht, welche Aktion wahrscheinlicher ist.



### 7.1.2 Reduktion

Die Modellierung des gesamten Aktionsverbundraumes in  $\mathbf{h}$  ist mit  $2^n$  Zuständen immer noch sehr komplex. Daher wird nun eine Methode hergeleitet, mit deren Hilfe die Anzahl der Zustände weiter reduziert werden kann. Die Grundannahme, die dieser Zustandsreduktion zugrundeliegt, ist die Tatsache, dass die Aktionen im Allgemeinen eher separat als gleichzeitig auftreten. Bei vielen Aktionen ist der Mensch gar nicht in der Lage diese gleichzeitig auszuführen. Diese Einsicht kann als logische ODER-Verknüpfung der Aktionen, gegeben eine Intention, interpretiert werden.

Der Zustandsraum von  $\mathbf{h}$  kann nun reduziert werden, indem alle Zustände, in denen eine Aktion  $a_i$  ausgeführt wird, zusammengefasst werden, was der besagten ODER-Verknüpfung von relevanten Aktionen entspricht. Diese Zusammenfassung lässt sich mit einer Zufallsvariable  $\mathbf{h}'$  ausdrücken die nur noch  $n+1$  Zustände benötigt!

Die bedingte Wahrscheinlichkeit  $P(\mathbf{h}' | \text{int})$  lässt sich aus  $P(\mathbf{h} | \text{int})$  durch einfache Regeln der Wahrscheinlichkeitsrechnung berechnen. Hier wird die Regel, dass für zwei Zufallsvariablen  $\mathbf{a}$  und  $\mathbf{b}$

$$P(a \vee b) = P(a) + P(b) - P(ab) \quad (7.2)$$

gilt, angewendet. Die Anwendung wird im folgenden Beispiel gezeigt.

#### Beispiel 7.1: Entkopplung und Modellreduktion (Fortsetzung)

Es wird nun eine versteckte Zufallsvariable  $\mathbf{h}'$  mit vier Zuständen eingeführt. Deren Zustände repräsentieren die ODER-Verknüpfung der relevanten Zustände von  $\mathbf{h}$  nach

$$\begin{aligned} h'_0 &\leftarrow h_0 \\ h'_1 &\leftarrow h_4 \vee h_5 \vee h_6 \vee h_7 \\ h'_2 &\leftarrow h_2 \vee h_3 \vee h_6 \vee h_7 \\ h'_3 &\leftarrow h_1 \vee h_3 \vee h_5 \vee h_7 \end{aligned} .$$

Der Zustand, in dem keine Aktion ausgeführt wird, bleibt davon unberührt als  $\mathbf{h}' = h'_0$ .

Die bedingte Wahrscheinlichkeit  $P(\mathbf{h}' | \text{int})$  wird nun nach (7.2) aus  $P(\mathbf{h} | \text{int})$  berechnet durch

$$\begin{aligned} P(h'_1 | \text{int}) &= P(h_4 \vee h_5 \vee h_6 \vee h_7 | \text{int}) \\ &= P(h_4 | \text{int}) + P(h_5 | \text{int}) \\ &\quad + P(h_6 | \text{int}) + P(h_7 | \text{int}) . \end{aligned}$$

Hierbei wurde bereits ausgenutzt, dass die Zustände von  $\mathbf{h}$  immer einzeln auftreten, weshalb der letzte Term von (7.2) Null ist. Ähnlich lässt sich so auch die bedingte Wahrscheinlichkeit für  $h'_2$  und  $h'_3$  durch

$$\begin{aligned} P(h'_2 | \text{int}) &= P(h_2 | \text{int}) + P(h_3 | \text{int}) \\ &\quad + P(h_6 | \text{int}) + P(h_7 | \text{int}) \\ P(h'_3 | \text{int}) &= P(h_1 | \text{int}) + P(h_3 | \text{int}) \\ &\quad + P(h_5 | \text{int}) + P(h_7 | \text{int}) \end{aligned}$$

berechnen. Die bedingte Wahrscheinlichkeit  $P(h' | int)$  ist somit bestimmt und muss noch normiert werden. Daraus ergibt sich

$P(h' int)$	$P(h'_0)$	$P(h'_1)$	$P(h'_2)$	$P(h'_3)$
$int_1$	1/10	9/20	9/20	0
$int_2$	1/10	0	9/10	0
$int_3$	1/10	0	0	9/10



Als nächstes müssen die Modelle bestimmt werden, die die Abhängigkeit der Aktionen  $a_i$  von der versteckten Variable  $h$  bzw.  $h'$  beschreiben. Im Falle des exakten Modells ist dies sehr einfach. Die Wahrscheinlichkeitstabellen  $P(a_i | h)$  enthalten nur 1 und 0 Einträge, da die Tabellen eine direkte Abbildung des Verbundraumes der Aktionen auf die einzelnen Aktionen beschreiben. Dies wird im nächsten Beispiel gezeigt.

Um die bedingten Wahrscheinlichkeiten  $P(a_i | h')$  aus  $P(a_i | h)$  berechnen zu können, muss wieder bedacht werden, dass die Zustände von  $h'$  eine ODER-Kombination der Zustände von  $h$  sind. Daher können alle bedingten Wahrscheinlichkeiten über  $a_i$  für die kombinierten  $h_i$  addiert werden. Dies ist im nächsten Beispiel gezeigt.

**Beispiel 7.1: Enkopplung und Modellreduktion (Fortsetzung)**

Die bedingten Wahrscheinlichkeitstabellen  $P(a_i | h)$  für das exakte Modell sind gegeben durch

$P(a_i h)$	$P(\bar{a}_1)$	$P(a_1)$	$P(\bar{a}_2)$	$P(a_2)$	$P(\bar{a}_3)$	$P(a_3)$
$h_0$	1	0	1	0	1	0
$h_1$	1	0	1	0	0	1
$h_2$	1	0	0	1	1	0
$h_3$	1	0	0	1	0	1
$h_4$	0	1	1	0	1	0
$h_5$	0	1	1	0	0	1
$h_6$	0	1	0	1	1	0
$h_7$	0	1	0	1	0	1

Da  $h'_0$  identisch zu  $h_0$  ist, kann dieser Teil direkt übernommen werden. Für die anderen Zustände von  $h'$  müssen die ODER-Kombination berücksichtigt werden. Da

$$h'_1 = h_4 \vee h_5 \vee h_6 \vee h_7$$

ist, lässt dich die bedingte Wahrscheinlichkeit durch

$$P(a_i|h'_1) = P(a_i|h_4) + P(a_i|h_5) + P(a_i|h_6) + P(a_i|h_7)$$

berechnen und normieren. Dies führt zu einer Gleichverteilung für  $a_2$  und  $a_3$ .

Das gleiche gilt für die anderen Zustände von  $h'$ . Hier lassen sich die bedingten Wahrscheinlichkeiten durch

$$P(a_i|h'_2) = P(a_i|h_2) + P(a_i|h_3) + P(a_i|h_6) + P(a_i|h_7)$$

und

$$P(a_i|h'_3) = P(a_i|h_1) + P(a_i|h_3) + P(a_i|h_5) + P(a_i|h_7)$$

berechnen. Die resultierenden bedingten Wahrscheinlichkeitstabellen sind somit durch

$P(a_i h')$	$P(\bar{a}_1)$	$P(a_1)$	$P(\bar{a}_2)$	$P(a_2)$	$P(\bar{a}_3)$	$P(a_3)$
$h'_0$	1	0	1	0	1	0
$h'_1$	0	1	1/2	1/2	1/2	1/2
$h'_2$	1/2	1/2	0	1	1/2	1/2
$h'_3$	1/2	1/2	1/2	1/2	0	1

gegeben. ■

### 7.1.3 Vergleich der Modelle

Es soll nun das reduzierte Modell mit dem exakten Modell, die beide auf der Abbildung von Intentionen auf Aktionen durch einen Experten beruhen, verglichen werden. Dazu werden mit beiden Modellen Schätzungen der Intention bei verschiedenen vorliegenden Aktionen durchgeführt

#### Beispiel 7.1: Entkopplung und Modellreduktion (Fortsetzung)

Es werden zuerst Beobachtungen von Aktionen betrachtet die im Expertenmodell (7.1) angegeben wurden. Die Ergebnisse der Schätzungen der Intention mit dem exakten Modell sind in Tabelle 7.1 gezeigt; die Ergebnisse mit dem reduzierten Modell in Tabelle 7.2.

Tabelle 7.1: Intentionsschätzung mit dem exakten Modell.

$a_1$	$a_2$	$a_3$	$P(int_1)$	$P(int_2)$	$P(int_3)$
0	0	0	$0.\bar{3}$	$0.\bar{3}$	$0.\bar{3}$
1	0	0	1	0	0
0	1	0	0.25	0.75	0
0	0	1	0	0	1
1	1	0	1	0	0

**Tabelle 7.2:** Intentionsschätzung mit dem reduzierten Modell

$a_1$	$a_2$	$a_3$	$P(int_1)$	$P(int_2)$	$P(int_3)$
0	0	0	$0.\bar{3}$	$0.\bar{3}$	$0.\bar{3}$
1	0	0	1	0	0
0	1	0	$0.\bar{3}$	$0.\bar{6}$	0
0	0	1	0	0	1
1	1	0	0.5	0.5	0

Beim Vergleich der Ergebnisse sieht man, dass beim Auftreten keiner Aktion die Schätzungen der Intention identisch sind. Das Gleiche gilt, wenn eine Aktion beobachtet wird, die nur einer Intention zugeordnet ist. Für den Fall, dass eine Aktion mehreren Intentionen zugeordnet ist, wie bei  $a_2$ , ist die Unsicherheit im Ergebnis der Schätzungen mit dem reduzierten Modell leicht erhöht, die Tendenz bleibt jedoch erhalten. Im Falle des gleichzeitigen Auftretens von Aktionen tritt diese Vergrößerung der Unsicherheit deutlicher zu Tage. Dieser Fall widerspricht allerdings auch der Annahme auf der die Reduktion beruht, nämlich dass Aktionen nur einzeln auftreten.

Tabelle 7.3 zeigt die Ergebnisse der Schätzungen mit dem reduzierten Modell basierend auf beobachteten Aktionskombinationen, die nicht Teil des Expertenmodells waren. Diese Kombinationen stehen also gewissermaßen im Widerspruch zum Expertenmodell.

**Tabelle 7.3:** Intentionsschätzung mit dem reduzierten Modell

$a_1$	$a_2$	$a_3$	$P(int_1)$	$P(int_2)$	$P(int_3)$
0	1	1	0.2	0.4	0.4
1	0	1	$0.\bar{3}$	0	$0.\bar{6}$
1	1	1	$0.\bar{3}$	$0.\bar{3}$	$0.\bar{3}$

Man sieht hier deutlich, dass nun größere Unsicherheiten auftreten. Im exakten Modell führen diese Beobachtungen sogar zu ungültigen Schätzungen, da die Wahrscheinlichkeit für alle Intentionen gleich Null ist.



Das Beispiel zeigt, dass das reduzierte Modell zu vergleichbaren Ergebnissen wie das exakte Modell führt. Außerdem erlaubt es, den Umgang mit Beobachtungen von Aktionskombinationen, die im exakten Modell undefiniert waren.

Die Ableitung des reduzierten Modells vom exakten Modell, wie sie hier am Beispiel gezeigt wurde, ist nur für kleine Modelle möglich. Für größere Modelle mit einer höheren Anzahl von Aktionen kann dieser Weg so nicht beschritten werden. Es ist daher nötig, das reduzierte Modell direkt von der Abbildung der Intentionen auf Aktionen des Experten abzuleiten. Wenn man



sich das reduzierte Modell einmal genauer betrachtet, fällt dies auch nicht schwer, da jeder Zustand von  $\mathbf{h}'$  (außer  $h'_0$ ) das Auftreten genau einer Aktion beschreibt. Diese Erkenntnis erlaubt eine einfache Verteilung der Wahrscheinlichkeitsmasse aus der Expertenabbildung. Zuerst muss die Wahrscheinlichkeit  $P(h_0)$  bestimmt werden. Dann wird die Tatsache ausgenutzt, dass jeder Zustand  $h'_i$  mit  $i > 0$  mit genau einer Aktion  $a_i$  verbunden ist. Gegeben eine gewisse Intention  $int_j$  muss nun  $P(h'_i | int_j)$  bestimmt werden. Jeder Aktion  $a_i$ , die vom Experten dieser Intention zugewiesen wurde, wird zunächst der Wert 1 zugewiesen. Im Anschluss werden die Wahrscheinlichkeiten so normiert, dass sie sich zu  $1 - P(h_0)$  aufsummieren. Somit lässt sich auch das approximierte Modell direkt aus der Expertenabbildung gewinnen. Dieser Ansatz wird im folgenden Abschnitt verwendet.

## 7.2 Experimente in der virtuellen Realität mit Techniken der weiträumigen Telepräsenz

Im Folgenden wird eine Evaluationsumgebung für die Intentionserkennung in der Mensch-Roboter-Kooperation, sowie die darin verwendeten Modelle, vorgestellt. Diese Arbeiten sind Teil des Sonderforschungsbereichs SFB 588 *Humanoide Roboter – Lernende und kooperierende multimodale Roboter* [7].

Zur Evaluation der kaskadierten Systemmodelle wurde im Rahmen einer Diplomarbeit [1] eine Umgebung geschaffen, die es erlaubt, Experimente in einem kontrollierten Szenario durchzuführen. Ziel war es dabei, Experimente mit echten Menschen durchzuführen, die, basierend auf ihrer Intention, Aktionen frei ausführen können. Der Aufwand an Sensorik zur Erkennung der Benutzeraktionen sollte allerdings überschaubar gehalten werden, da die Modelle und nicht ein Hardwareaufbau im Vordergrund standen. Somit wurde eine virtuelle Welt geschaffen, in die der Mensch eintauchen und in der er sich frei bewegen kann.

Um eine freie Bewegung in der virtuellen Welt zu ermöglichen, wurden Techniken aus der weiträumigen Telepräsenz verwendet [39]. Diese ermöglichen die Exploration eines virtuellen Szenarios, ähnlich wie die Exploration einer entfernten Welt, durch freies Gehen durch diese Welt. Unter Gehen ist hier wirklich ein Gehen auf zwei Beinen zu verstehen und nicht die Erkundung per Joystick oder ähnlicher Hilfsmittel. Der Mensch erhält somit die Möglichkeit, sich in einen Avatar in der virtuellen Welt hineinzusetzen, genauso, wie er sich bei der traditionellen weiträumigen Telepräsenz in einen Roboter hineinversetzen würde, der sich an einem entfernten Ort befindet. Die Verwendung einer virtuellen Umgebung als entfernte Welt ermöglicht eine sehr hohe Flexibilität bezüglich der zu untersuchenden Szenarien, da sie sehr schnell angepasst und erweitert werden können.

Der für Experimente zur Verfügung stehende Raum ist typischerweise kleiner als der für das virtuelle Szenario benötigte. Daher wurde für die Evaluationsumgebung das Bewegungskompressionsverfahren [72] eingesetzt. Bei diesem Verfahren wird der Pfad, den der Benutzer in der virtuellen Welt zurücklegt, so verkrümmt, dass er in den Raum passt, der ihm in der realen



**Abbildung 7.2:** Head Mounted Display NVisor SX der Firma NVIS.

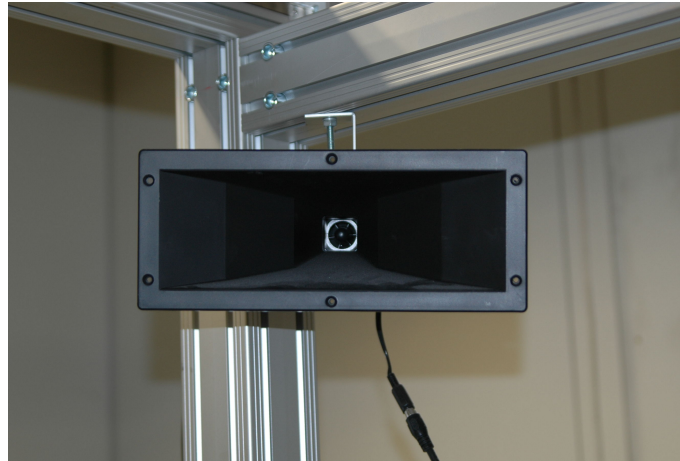
Welt zur Verfügung steht. Durch eine geschickte Benutzerführung wird der Anwender dann auf diesen gekrümmten Pfad gelenkt. Dies geschieht durch einen Versatz in der Darstellung der virtuellen Welt für den Benutzer.

Das verwendete Verfahren ist längentreu und lokal winkeltreu. Das heißt, wenn der Mensch in der virtuellen Welt 10 Meter zurücklegt, so legt er auch 10 Meter in der Realität zurück und wenn er sich um 90 Grad in der virtuellen Welt dreht, so muss er sich auch in der realen Welt um 90 Grad drehen. Diese Technik gibt dem Benutzer somit das Gefühl auch in der virtuellen Welt wirklich vor Ort zu sein.

Das Gesamtsystem zur Durchführung der Experimente besteht aus mehreren Komponenten, die im Folgenden kurz beschrieben werden sollen.

**Head Mounted Display** Um dem Benutzer die virtuelle Welt zu präsentieren, wird ein hochauflösendes *Head Mounted Display* (HMD) der Firma NVIS mit einer Auflösung von  $1280 \times 1024$  Bildpunkten pro Auge verwendet. Das Gerät ist in Abbildung 7.2 gezeigt. Durch Verwendung dieses Displays bekommt der Nutzer einen dreidimensionalen Eindruck der virtuellen Welt, was die Navigationsmöglichkeit deutlich verbessert. Das Gerät ist sehr leicht und zeichnet sich durch einen hohen Tragekomfort aus.

**Trackingsystem** Zur Übertragung der Bewegung des Benutzers in die virtuelle Welt muss dieser in seiner realen Umgebung geortet und verfolgt werden. Dazu wird ein am Lehrstuhl für intelligente Sensor-Aktor-Systeme entwickeltes akustisches Trackingsystem verwendet [8]. Das System basiert auf der Laufzeitmessung von Schallsignalen und ist aufgrund der Verwendung von Signalen im hörbaren Bereich sehr robust gegenüber Verdeckungen. Das System besteht aus sechs Lautsprechern, wie in Abbildung 7.3 zu sehen, die in den Ecken und über der  $4,5 \text{ Meter} \times 4,5 \text{ Meter}$  großen Benutzerumgebung angebracht sind. An jedem in der Umgebung zu verfolgenden Objekt sind vier Mikrophone angebracht. Basierend auf den Laufzeiten der Signale von den Lautsprechern zu den Mikrophenen wird die Position und die Orientierung des



**Abbildung 7.3:** Lautsprecher des akustischen Trackingsystems.

Objekts berechnet. Zur Glättung der Positionsdaten wird diese Lage noch mit den Daten eines angebrachten 3D-Drehratensensors (Gyroskop) fusioniert.

Um die Bewegung des Menschen in die virtuelle Welt zu übertragen wird der Kopf und eine Hand des Menschen verfolgt. Dazu wird ein Kopftracker am HMD, wie in Abbildung 7.4

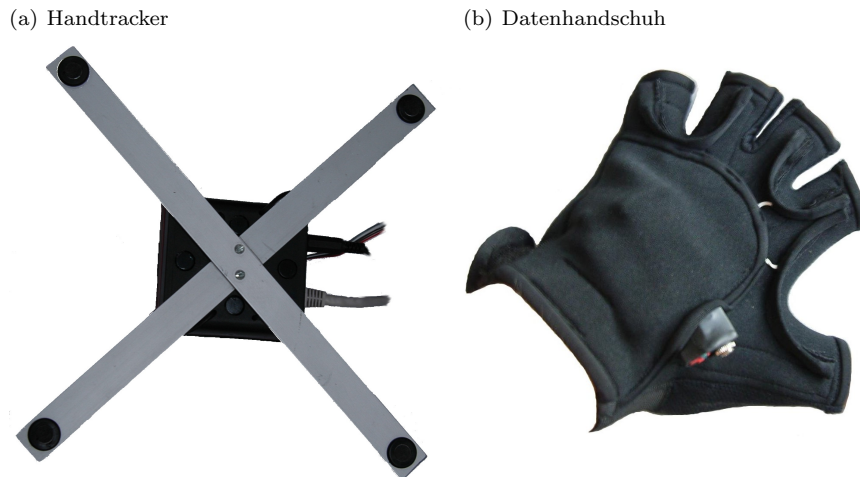


**Abbildung 7.4:** HMD mit angebrachtem Kopftracker.

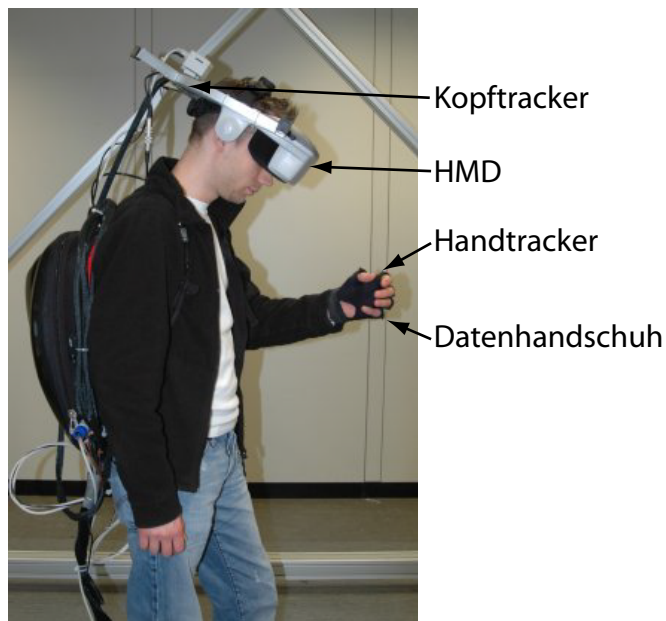
gezeigt, verwendet. Zur Verfolgung der Position und Orientierung der Hand wird ein weiteres Trackingmodul (Abbildung 7.5(a)) eingesetzt.

Ferner soll die Bewegung der Finger in der virtuellen Welt dargestellt werden, um dem Menschen das Greifen von Gegenständen zu ermöglichen. Dazu wird ein drahtloser Datenhandschuh (Abbildung 7.5(b)) mit 8 Freiheitsgraden verwendet. Der Datenhandschuh wurde am Forschungszentrum Karlsruhe (FZK) entwickelt und kommuniziert über eine Bluetooth Schnittstelle mit der Rechnerinfrastruktur.

Der Gesamtaufbau des Trackingsystems am Menschen ist in Abbildung 7.6 zu sehen.



**Abbildung 7.5:** (a) Ein Handtrackingmodul. (b) Drahtloser Datenhandschuh vom Forschungszentrum Karlsruhe (FZK).



**Abbildung 7.6:** Gesamtaufbau des Trackingsystems.

Die Signalverarbeitung für das Trackingsystem wird auf Signalprozessorkarten durchgeführt, die in dem Rucksack, den der Benutzer in Abbildung 7.6 trägt, untergebracht sind. Dort befindet sich auch ein Mini-PC-System, auf dem die dreidimensionale Stereodarstellung der virtuellen Welt für den Benutzer berechnet wird.

**Virtuelle Umgebung** Zur Evaluation der Modelle für die Intentionserkennung wurde in der virtuellen Welt ein Küchenszenario implementiert, das der Demonstrationsumgebung des Sonderforschungsbereichs 588 „Humanoide Roboter – Lernende und kooperierende multimodale

Roboter“ [7] an der Universität Karlsruhe (TH) entspricht. Der Blick des Benutzers in die virtuelle Küche ist in Abbildung 7.7 gezeigt.



**Abbildung 7.7:** Virtuelle Küche in der sich der Nutzer frei gehend umher bewegen kann. Der Mensch trägt hier gerade einen Apfel.

In dieser Küche kann sich der Benutzer aufgrund der verwendeten Techniken der weiträumigen Telepräsenz frei gehend bewegen. Außerdem hat der Benutzer die Möglichkeit mit der Umwelt zu interagieren. Er kann Objekte greifen, sie umher tragen und sie wieder ablegen. In Abbildung 7.7 trägt der Mensch gerade einen Apfel. Durch die Verwendung einer Physiksimulation ist diese Manipulation sehr naturgetreu. Die Objekte können beispielsweise auch auf den Boden fallen. Abgesehen davon kann der Mensch Schränke öffnen und Objekte aus ihnen entnehmen oder hineinstellen. Der in Abbildung 7.7 zu sehende simulierte Roboter kann ebenfalls Objekte greifen, tragen und abstellen.

**Schätzer** Als Kernkomponente für die Intentionserkennung wurde ein Schätzermodul entwickelt. Dieses Modul erhält die Informationen vom Trackingsystem und über die Objekte in der virtuellen Welt und liefert als Ausgabe eine Schätzung über die aktuell vorliegende Intention. Die Visualisierung der Intentionsschätzung ist in Abbildung 7.8 gezeigt.

Herzstück des Schätzers ist eine selbstgeschriebene Bibliothek für hybride dynamische kaskadierte Systeme. Diese Bibliothek ist aus Performancegründen in C++ implementiert. Sie enthält Methoden zum Anlegen beliebiger Topologien kaskadierter Systeme und zur Eingabe der Parameter. Zur Berechnung der Zustandsschätzung enthält die Bibliothek eine Implementierung des Verfahrens aus Abschnitt 2.2.

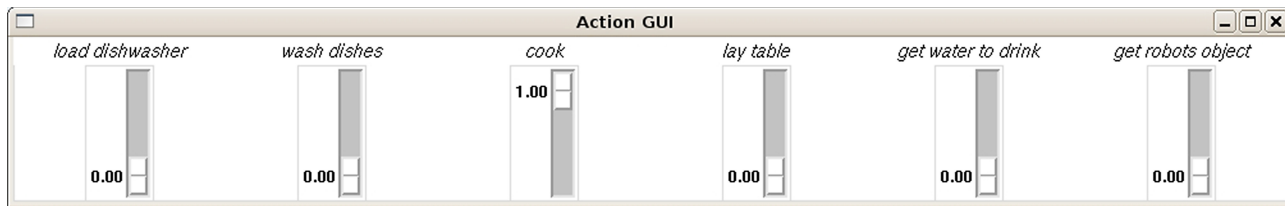


Abbildung 7.8: Visualisierung der Intentionsschätzung.

Um das Anlegen der Modelle und das Ändern der Parameter flexibel zu halten, wurde ein Pythonwrapper für die Bibliothek implementiert. Da Python eine sehr performante Skriptsprache ist, entfällt das Kompilieren der Anwendung bei Änderung der Modellparameter. Dadurch steht ein Prototypingsystem zur Verfügung, mit Hilfe dessen die Modelle sehr schnell angepasst werden können.

**Infrastruktur** Das Gesamtsystem ist über mehrere Rechner verteilt. Es besteht zum einen aus dem bereits erwähnten Rucksackrechner für die Signalverarbeitung des Trackingsystems und die Berechnung der Visualisierung der virtuellen Welt für den Nutzer. Ein anderer Rechner ist für die Berechnung der Bewegungskompression zuständig und auf einem dritten Rechner wird die Schätzung berechnet und ausgegeben.

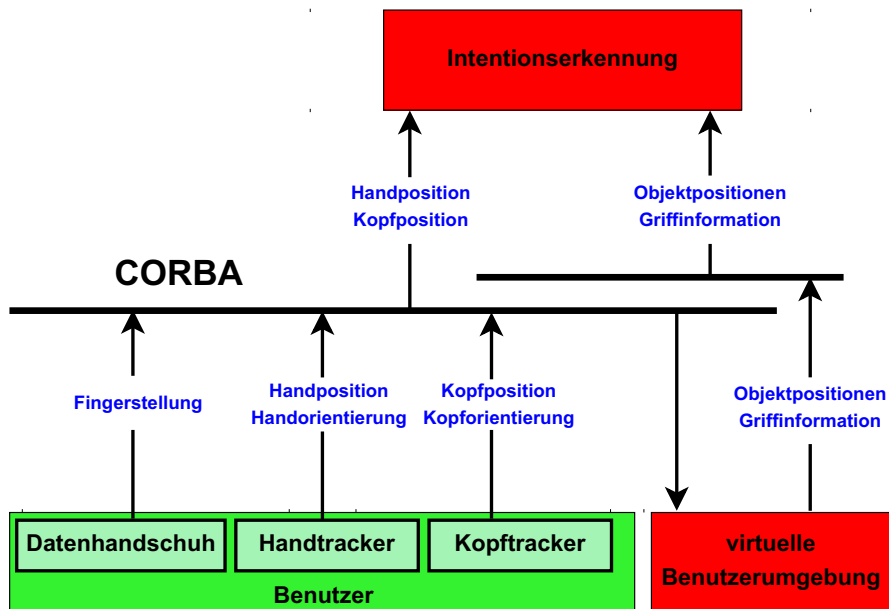


Abbildung 7.9: Kommunikationsstruktur des Evaluationssystems.

Diese Teilsysteme müssen miteinander kommunizieren. Da die einzelnen Teilsysteme mit unterschiedlichen Sprachen implementiert wurden, aber alle objektorientiert sind, wurde als Kommunikationsbus die *Common Object Request Broker Architecture* (CORBA) verwendet. Das System nutzt zwei CORBA-Instanzen, die als Bus in Abbildung 7.9 gezeigt sind. Der grüne

Block zeigt die Systeme, die direkt am Benutzer sind und die Eingabe für das Gesamtsystem darstellen. Die Informationen, die über den Softwarebus laufen, sind jeweils annotiert. Als Eingang für die virtuelle Benutzerumgebung gelten alle Informationen, die an diesem Bus anliegen.

### 7.2.1 Intentionserkennungsmodell mit integrierten Aktionserkennern

Basierend auf den Möglichkeiten in der virtuellen Welt, wurde ein Modell zur Intentionserkennung implementiert. Es handelt sich dabei um ein komplettes kaskadiertes System für Intentionen und Aktionen und stellt eine Realisierung des generischen Modells aus Abschnitt 6.2 dar.

Das implementierte Szenario umfasst sechs zu erkennende Intentionen:

- $int_1$ : Der Mensch möchte die Spülmaschine beladen.
- $int_2$ : Der Mensch möchte abwaschen.
- $int_3$ : Der Mensch möchte kochen.
- $int_4$ : Der Mensch möchte den Tisch decken.
- $int_5$ : Der Mensch möchte etwas zu trinken vom Roboter.
- $int_6$ : Der Mensch möchte das Objekt haben, das der Roboter hält.

Dies wird durch eine diskrete Zufallsvariable  $int$  mit sechs Zuständen modelliert.

Als Kontext wird betrachtet, was der Roboter in der Hand hält. Hierbei wird unterschieden, ob der Roboter ein Gefäß mit einem Getränk, ein anderes Objekt oder kein Objekt in der Hand hält. Dies wird durch eine diskrete Zufallsvariable  $rob$  mit drei Zuständen modellieren. Das Aufstellen der bedingten Wahrscheinlichkeit  $P(int | rob)$  ist sehr einfach durch einen Experten möglich.

Zur Erkennung der Intention stehen im System Erkenner für 21 Aktionen zur Verfügung. Diese Erkenner sind als Teilsysteme des kaskadierten Gesamtsystems integriert. Die Aktionen lassen sich in drei Klassen unterteilen. Die jeweilige Klasse ist in den folgenden drei Tabellen als Überschrift angegeben.

---

#### *Halten eines Gegenstands*

---

Öffnen/Schließen des Küchenschrank	Öffnen/Schließen der Spülmaschine
Öffnen/Schließen des Kühlschranks	Tragen von Lebensmitteln
Halten eines Tablett	Halten eines Bechers

---

*Annähern eines Gegenstands an einen Ort*

Gegenstand in den Küchenschrank legen	Gegenstand in die Spülmaschine legen
Gegenstand in das Spülbecken legen	Gegenstand auf die Spülablage legen
Gegenstand auf den Küchentisch legen	Lebensmittel auf den Küchentisch ablegen
Topf auf Herdplatte abstellen	Lebensmittel in Topf legen
Gegenstand an Topf annähern	Gegenstand auf eine Arbeitsfläche legen

*Entfernen eines Gegenstands von einem Ort*

Gegenstand aus dem Küchenschrank nehmen	Gegenstand aus der Spülmaschine nehmen
Gegenstand aus dem Spülbecken nehmen	Gegenstand von der Spülablage nehmen
Gegenstand von einer Arbeitsfläche nehmen	

Die Einteilung der möglichen Aktionen in Klassen erlaubt die Verwendung von generischen Aktionserkennern, die nur noch mit den jeweiligen Gegenständen und Orten instanziiert werden müssen. Wie bereits in der Fallstudie in Kapitel 6 festgestellt, handelt es sich hierbei vornehmlich um *pick-and-place* Aktionen. Die Modelle zur Erkennung dieser Aktionen werden weiter unten vorgestellt.

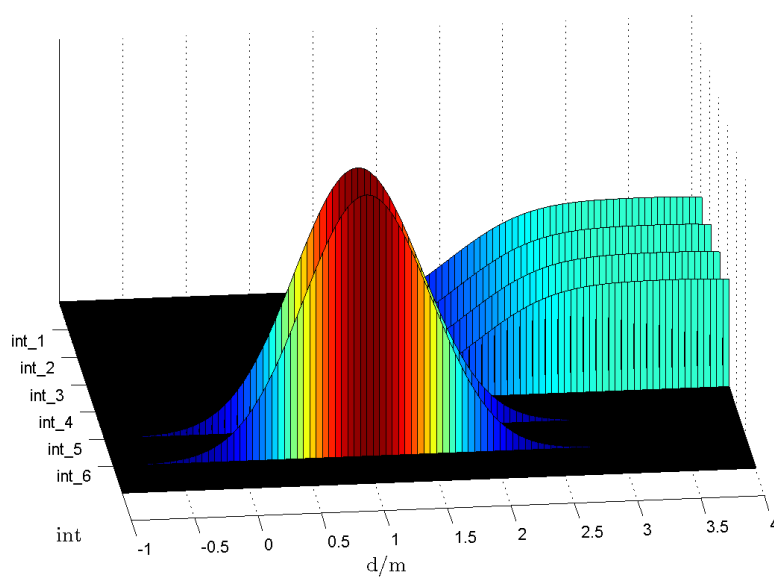
Basierend auf den gegebenen Intentionen und Aktionen wurde von einem Experten ein Modell für deren Zusammenhang aufgestellt. Diese Zuordnung ist in Tabelle 7.4 dargestellt und wird zum Aufstellen des stochastischen Modells verwendet. Zur Entkopplung der Aktionen wird auch hier eine Zufallsvariable  $\mathbf{h}$  nach dem in Abschnitt 7.1 beschriebenen Schema eingeführt.

Zusätzlich zu diesen Aktionen wird noch die Distanz des Menschen vom Roboter betrachtet. Dies ist sinnvoll, da die letzten zwei Intentionen ( $int_5$  und  $int_6$ ) eine direkte Kooperation mit dem Roboter erfordern, im Gegensatz zu den ersten vier Intentionen ( $int_1 - int_4$ ). Das heißt, dass sich die Distanz zwischen Mensch und Roboter wohl verringern wird, wenn der Mensch beispielsweise ein Objekt vom Menschen bekommen möchte. Da die Distanz eine kontinuierliche Größe darstellt, wird hier der Ansatz für hybride Systeme aus Kapitel 3 verwendet. Die verwendete hybride bedingte Dichte  $f(d|int)$  ist in Abbildung 7.10 zu sehen.

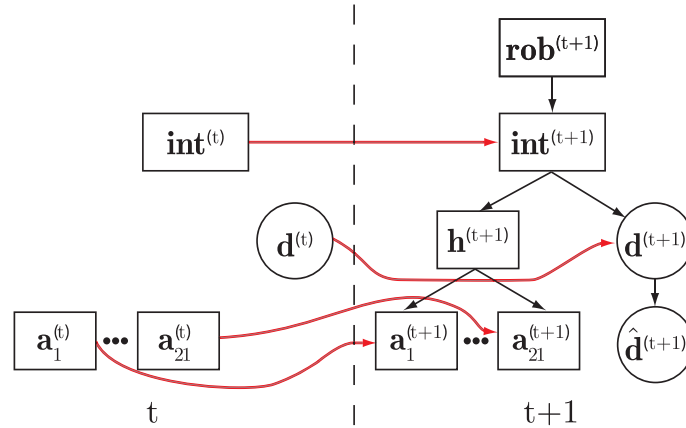


	<i>int</i> <sub>1</sub>	<i>int</i> <sub>2</sub>	<i>int</i> <sub>3</sub>	<i>int</i> <sub>4</sub>	<i>int</i> <sub>5</sub>	<i>int</i> <sub>6</sub>
Gegenstand in den Küchenschrank legen	x	x	x			
Gegenstand aus dem Küchenschrank nehmen			x	x	x	
Gegenstand in die Spülmaschine legen	x		x			
Gegenstand aus der Spülmaschine nehmen	x			x		
Gegenstand in das Spülbecken legen		x	x			
Gegenstand aus dem Spülbecken nehmen	x	x	x			
Gegenstand auf die Spülablage legen	x	x	x			
Gegenstand von der Spülablage nehmen	x	x	x	x	x	
Gegenstand auf den Küchentisch legen			x	x		
Lebensmittel auf den Küchentisch ablegen			x	x		
Gegenstand auf eine Arbeitsfläche legen	x	x	x	x		
Gegenstand von einer Arbeitsfläche nehmen	x	x	x	x	x	
Lebensmittel in Topf legen			x			
Gegenstand an Topf annähern			x			
Topf auf Herdplatte abstellen			x			
Öffnen/Schließen eines Küchenschrank	x	x	x	x	x	
Öffnen/Schließen der Spülmaschine	x			x		
Öffnen/Schließen des Kühlschranks			x	x		
Tragen von Lebensmitteln			x	x		
Halten eines Tablett				x		x
Halten eines Bechers	x	x	x	x	x	
Annäherung an den Roboter					x	x

Tabelle 7.4: Zuordnung der Aktionen zu den Intentionen.


 Abbildung 7.10: Die verwendete hybride bedingte Dichte  $f(d^{(t)} | int^{(t)})$  beschreibt, dass die Distanz zum Roboter bei den Intentionen  $int_5$  und  $int_6$  im Allgemeinen geringer ist als bei den anderen Intentionen.

Für das hier vorgestellte Szenario wird ein dynamisches Modell verwendet, das die Entwicklung des kaskadierten Gesamtsystems über die Zeit berücksichtigt. Der obere Teil des Modells (hier noch ohne die Aktionserkenner) ist in Abbildung 7.11 in einer dynamischen Bayesnetzdarstellung zu sehen. Diese Art der Darstellung ist hier aufgrund ihrer Kompaktheit gewählt und weil direkt zu erkennen ist, welche Variablen kontinuierlich und welche diskret sind. Die grauen



**Abbildung 7.11:** DBN Struktur des oberen kaskadierten Teilsystems. Rote Kanten stehen für zeitliche Abhängigkeiten

Kanten zeigen die Abhängigkeiten über die Zeit. Hier sind auch die Zeitindizes der Variablen in Klammern angegeben.

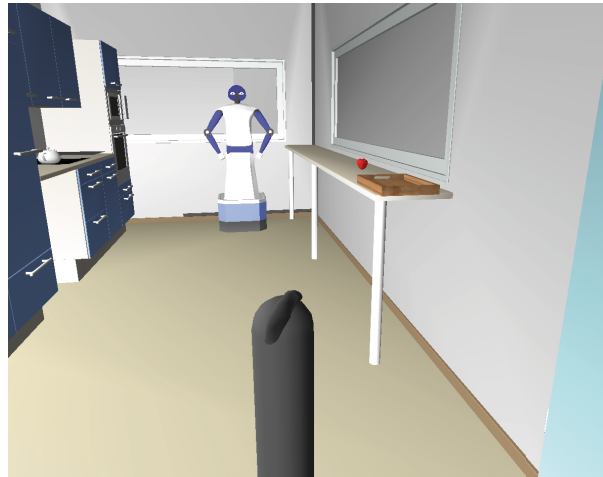
Für die Intensionsvariable wird davon ausgegangen, dass sich die Intention des Menschen im Allgemeinen selten ändert. Daher wird für das dynamische Modell  $P(int^{(t+1)} | int^{(t)})$  die bedingte Wahrscheinlichkeitstabelle

$P(int^{(t+1)}   int^{(t)})$	$P(int_1^{(t+1)})$	$P(int_2^{(t+1)})$	$P(int_3^{(t+1)})$	$P(int_4^{(t+1)})$	$P(int_5^{(t+1)})$	$P(int_6^{(t+1)})$
$int_1^{(t)}$	0.9	0.02	0.02	0.02	0.02	0.02
$int_2^{(t)}$	0.02	0.9	0.02	0.02	0.02	0.02
$int_3^{(t)}$	0.02	0.02	0.9	0.02	0.02	0.02
$int_4^{(t)}$	0.02	0.02	0.02	0.9	0.02	0.02
$int_5^{(t)}$	0.02	0.02	0.02	0.02	0.9	0.02
$int_6^{(t)}$	0.02	0.02	0.02	0.02	0.02	0.9

verwendet. Dieses Modell sagt aus, dass sich mit hoher Wahrscheinlichkeit (0.9) die Intention des Menschen nicht ändern wird. Mit einer kleinen Wahrscheinlichkeit (0.02) kann der Mensch sich aber umentscheiden. Dieses Modell hat eine stark glättende Wirkung auf die Schätzung und wird auch in angepasster Form für die dynamischen Modelle der Aktionen  $P(a_i^{(t+1)} | a_i^{(t)})$  für  $i = 1, \dots, 21$  verwendet.

Der kontinuierliche Part dieses Teilmodells beschreibt, wie bereits erwähnt, die Distanz zwischen Mensch und Roboter. Diese Distanz muss über die Zeit verfolgt werden. Da die Distanz aus Informationen des Kopftrackers und aus der virtuellen Welt berechnet werden muss, unterliegt sie, wie das Trackingsystem selbst, einem Rauschen. Um diesem zu begegnen, wird ein Modell

aus der Koppelnavigation in das Gesamtmodell integriert. Es wird zum einen ein dynamisches Bewegungsmodell verwendet, um die mögliche Distanz im nächsten Zeitschritt vorherzusagen. Zum anderen wird ein Messmodell verwendet, das das Rauschverhalten des Trackers bei der Schätzung berücksichtigt. Als Bewegungsmodell wird eine Zufallsbewegung (engl. *random walk*) angenommen. Dies dient vornehmlich der Glättung der Schätzung über die Zeit. Für das Messmodell der Distanz wird ein additives Rauschen angenommen, das mit Hilfe des in Kapitel 5 vorgestellten Verfahrens identifiziert werden kann. Dazu wurde wie in Abbildung 7.12 zu sehen



**Abbildung 7.12:** Aufnahme der Handposition zur Identifikation des Messrauschens. Der graue Pfosten dient zur Visualisierung des Kalibrierpunktes (0,0,1)

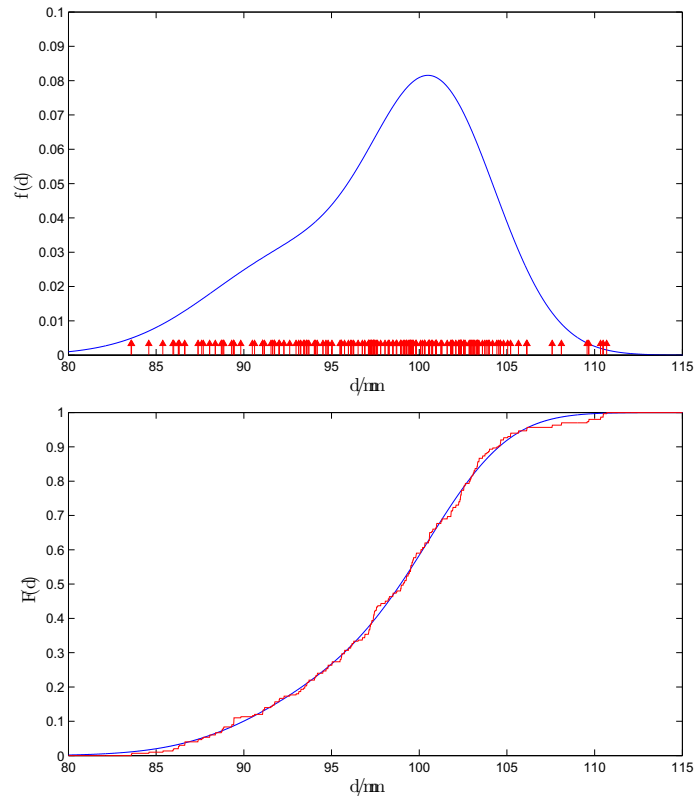
die Hand mittels Tracker auf einem festen Punkt in der virtuellen Realität gehalten. Bei dieser Position wurden 300 Messungen der Position durchgeführt, was bei einer Aktualisierungsrate des Trackers von 50 Hz, 6 Sekunden entspricht. Die gemessenen Werte sind in Abbildung 7.13 in rot als Dichte und als Verteilung zu sehen. Die identifizierte Dichte bzw. Verteilung ist in blau gezeichnet und besteht aus zwei Gausskomponenten.

### Aktionserkennung

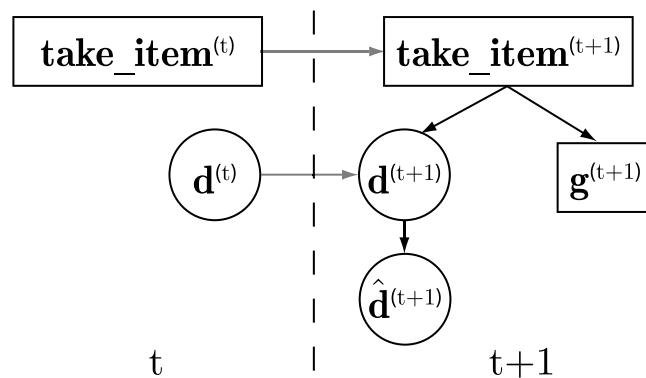
Zur Erkennung der Aktionen aus den gegebenen Informationen stehen drei Typen von Erkennern zur Verfügung, die wieder als dynamische kaskadierte Systeme modelliert sind. Durch Instanziierung eines Erkenners für eine bestimmte Aktion wird das Gesamtmodell um das entsprechende Erkennersmodell erweitert. Die vorhandenen Erkenner sind:

- Halten eines Gegenstands
- Annähern eines Gegenstands an einen Ort
- Entfernen eines Gegenstands von einem Ort

Als Beispiel wird hier das Modell für die Erkennung von **Halten eines Gegenstands** in Abbildung 7.14 gezeigt.



**Abbildung 7.13:** Rauschidentifikation des Trackingsystems. In rot sind die gemessenen Daten in Form eine Dirac-Mixture-Dichte (oben) bzw. als deren Verteilung (unten) gezeigt. In blau sind die geschätzte Dichte (oben) und Verteilung (unten) bestehend aus jeweils zwei Gauß-Komponenten gegeben.



**Abbildung 7.14:** Modell für die Erkennung der Aktion Halten eines Gegenstands.

Ein gehaltenes Objekt bedingt einen Abstand  $d$ , wobei hier der Abstand zwischen der Hand des Menschen und dem Objekt selbst gemeint ist. Zur Verfolgung der Distanz wird wieder das Modell aus der Koppelnavigation verwendet. Darüber hinaus muss ein Griff vorliegen, was über die binäre Zufallsvariable  $g$  ausgedrückt wird. Diese Information liefert der Datenhandschuh.

Beim diesem Modell zur Aktionserkennung handelt es sich um ein hybrides Modell, da die Distanz kontinuierlich und der Griff sowie die Aktion selbst diskret sind. Die Parameter dieses Modells sind ähnlich dem oben beschrieben und werden daher nicht noch einmal separat aufgeführt.

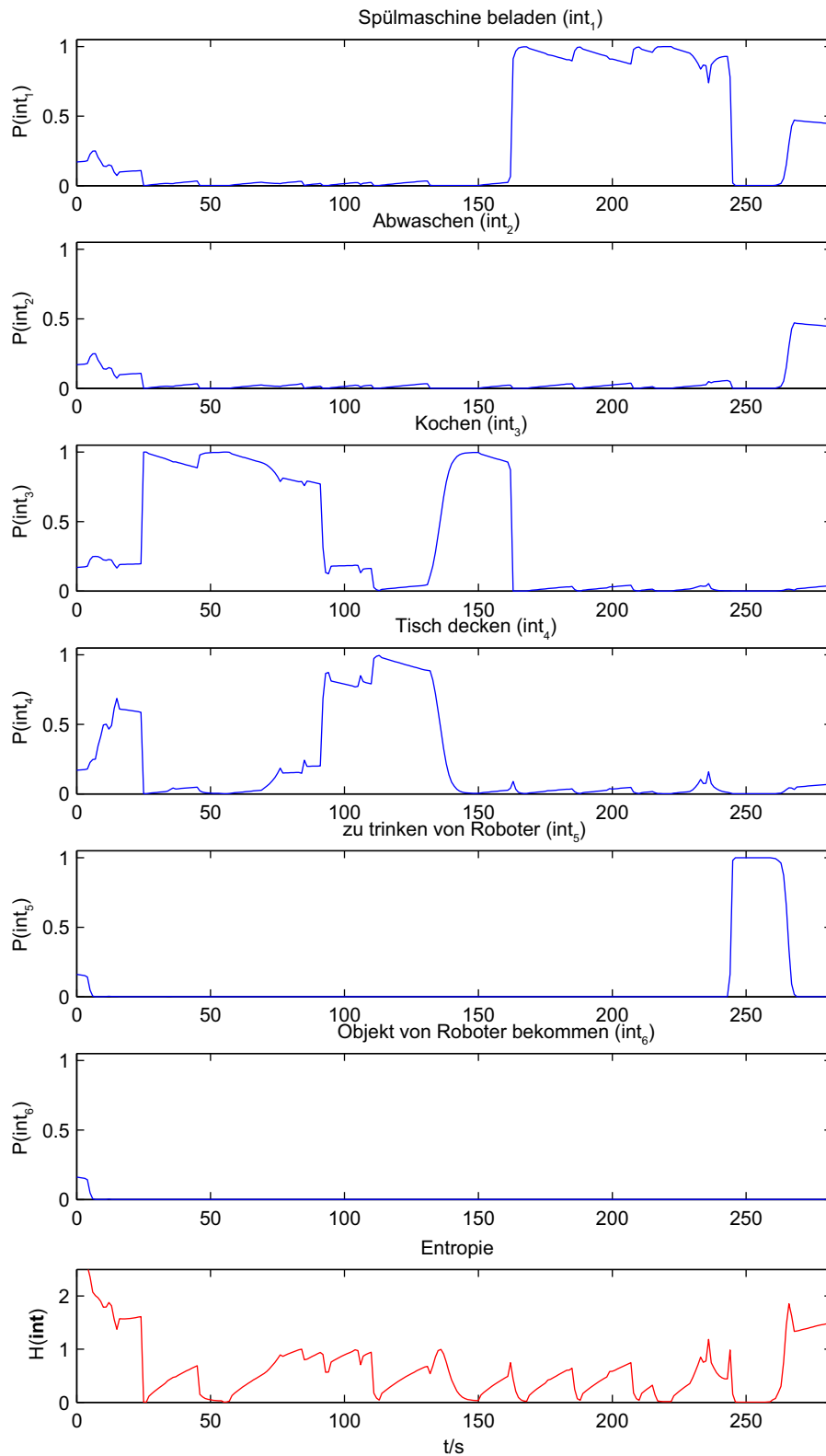
Für die weiteren Aktionserkenner (**Annähern/Entfernen eines Gegenstands an/von einen/einem Ort**) dient die Aktion **Halten eines Gegenstands** als ein Teilmodell, da das Halten eines Objekts hier eine Grundvoraussetzung ist. Als weitere Größe wird in diesen Modellen noch die Geschwindigkeit der Handbewegung in Richtung des Ortes betrachtet. Hierbei handelt es sich auch wieder um eine kontinuierliche Größe die mit dem Koppelnavigationsmodell verfolgt wird.

In der virtuellen Küche sind zahlreiche Objekte, wie ein Topf, Gläser, Tassen, etc. und verschiedene Orte wie Arbeitsflächen, ein Herd, eine Spüle und ein Tisch modelliert. Das Gesamtmodell für die Intentionserkennung inklusive der Aktionserkennungsmodelle besteht daher aus über 300 Zufallsvariablen. Durch das Hinzufügen neuer Objekte oder Orte können jederzeit weitere Variablen ergänzt werden, ohne dass das Gesamtmodell verändert werden muss. Hier zeigt sich ganz besonders die Flexibilität des verwendeten Ansatzes.

### Ergebnisse

Zur Evaluation des Modells wurden durch einen Menschen verschiedene Aktionen in der virtuellen Umgebung durchgeführt, die zu vorgegebenen Intentionen passen sollten. Es handelt sich dabei um folgendes Szenario: Zunächst beginnt der Mensch zu kochen. Während das Essen auf dem Herd ist, deckt er den Küchentisch. Anschließend nimmt er das Essen vom Herd und bringt es ebenfalls zum Tisch. Dann räumt er das Geschirr vom Tisch in die Spülmaschine. Schließlich verlangt er etwas zu trinken vom Roboter.

Die Ergebnisse des Schätzers unter Verwendung des beschriebenen Modells sind in Abbildung 7.15 zu sehen. Jeder einzelne Plot zeigt die Wahrscheinlichkeit der einzelnen Intentionen über die Zeit. Jeder Zeitschritt beträgt hier eine Sekunde. Der unterste Plot zeigt in rot mit der Entropie ein Maß für die Unsicherheit der Schätzung. Man sieht deutlich, wie sich die Tendenzen des Schätzers mit der Vorführung durch den Menschen decken. Die gelegentlichen Abweichungen, die einen Anstieg in der Unsicherheit verursachen, sind zum einen durch das Rauschen der Beobachtungen, zum anderen durch Aktionen, die nicht im Modell berücksichtigt sind, zu erklären. Nichtsdestotrotz liefert der Schätzer ein robustes Ergebnis.



**Abbildung 7.15:** Ergebnisse der Intentionsschätzung in der virtuellen Welt mit einem handelnden menschlichen Probanden. Der unterste Plot zeigt die Entropie der Schätzung als Maß für die Unsicherheit.

# Zusammenfassung und Ausblick

In dieser Arbeit werden theoretische und praktische Aspekte der stochastischen Unsicherheitsbetrachtung in kaskadierten dynamischen Systemen behandelt. Die dabei verfolgte Idee besteht darin, Systeme *bottom-up* aus sehr einfachen Teilsystemen aufzubauen, indem der Ausgang eines Systems als Eingang eines folgenden Systems verwendet wird. Hierbei entstehen verdeckte Zustände die gleichzeitig den Ausgang des einen und den Eingang eines anderen Teilsystems beschreiben und dadurch im Allgemeinen nicht direkt beobachtet werden können.

Das Ziel ist es, für solche Systemkaskaden eine Zustandsschätzung durchzuführen und basierend auf den Beobachtungen von Ein- und Ausgängen der Gesamtkaskade auf die verdeckten Zustände zu schließen. Dabei kann, wie in dieser Arbeit gezeigt wird, die Struktur der Kaskade verwendet werden, um die Zustandsschätzung lokal in den Teilsystemen durchzuführen und die resultierende Schätzung an die benachbarten Systeme weiterzugeben. Dadurch entsteht eine durchgehende Unsicherheitsbetrachtung über die eine Zustandsschätzung für das Gesamtsystem ermöglicht wird.

In dieser Arbeit wird davon ausgegangen, dass offene Kaskaden ohne Zyklen betrachtet werden. Dies vermeidet ein Zirkulieren des in Abschnitt 2.2 vorgestellten Verfahrens. Würde diese Einschränkung nicht getroffen, so wäre eine zusätzliche Betrachtung der Konvergenz des Verfahrens notwendig. Nichtsdestotrotz ergeben sich selbst bei dieser Einschränkung verschiedene Herausforderungen für die Zustandsschätzung, die im Rahmen dieser Arbeit angegangen werden. Die Ein- und Ausgänge der Teilsysteme, die hier betrachtet werden, können kontinuierlich oder diskret sein. Daraus ergeben sich hybride Systeme für deren Behandlung eine einheitliche Dichterepräsentation vorgestellt wird. Einen weiteren Schwerpunkt bildet die Behandlung kontinuierlicher Teilsysteme deren Systemfunktion eine nichtlineare Abbildung von Eingang auf Ausgang beschreibt. Für solche Teilsysteme ist eine exakte Berechnung der Zustandsschätzung häufig nicht möglich, da sich der Typ der Zustandsdichte im Allgemeinen ändert und die Komplexität so anwächst, dass eine Weiterbehandlung in der Kaskade unmöglich ist.

Es werden zwei approximative Ansätze vorgestellt, mit deren Hilfe das Schätzproblem für nichtlineare kontinuierliche Teilsysteme gelöst werden kann. Der erste Ansatz beruht auf der Approximation der Systemfunktion durch Mixture-Dichten mit achsenausgerichteten Komponenten.

Diese Methode ermöglicht es, hybride bedingte Dichten mit Hilfe von Dirac- und Gaussian-Mixture-Komponenten zu modellieren. Ein Vorteil dieses neuartigen Ansatzes besteht zunächst darin, dass die Vorwärts- und Rückwärtsinferenz aufgrund der Systemapproximation geschlossen lösbar sind. Daraus folgt auch unmittelbar, dass die Anzahl der Komponenten die zur Darstellung der posterioren Dichte benötigt werden durch die Anzahl der Komponenten in der Systemapproximation begrenzt ist. Darüber hinaus lässt sich das Verfahren einfach ins Mehrdimensionale übertragen. Ein Nachteil des Verfahrens ist, dass eine geeignete Approximation häufig schwierig zu finden ist und unter Umständen viele Komponenten benötigt werden, um das System gut genug zu approximieren. Daher hängt auch das Ergebnis der Schätzung stark von der Güte der Modellapproximation ab. Allerdings wurde gezeigt, dass Ansätze zur Bestimmung geeigneter Modellapproximationen bestehen [49]. Wenn eine solche Approximation gegeben ist, liefert das Verfahren bessere Ergebnisse als Standardverfahren, wie in Kapitel 3 an Beispiel 3.1 gezeigt wurde.

Das zweite vorgestellte Verfahren basiert auf der Approximation der Dichtefunktion selbst, durch eine parametrische Dichtebeschreibung in Form einer Dirac-Mixture-Dichte. Die Approximation wird als Optimierung durchgeführt, indem ein Gütemaß zwischen der wahren Dichte und der Approximation minimiert wird. Die Vorteile dieses Ansatzes bestehen darin, dass auch hier die Inferenz analytisch geschlossen gelöst werden kann und die Anzahl der Komponenten der posterioren Dichte durch die Anzahl der Komponenten der approximierten Dichte begrenzt wird. Aufgrund der Approximation der Dichte lässt sich die Inferenz auch direkt mit dem gegebenen Modell ausführen. Diese Vorgehensweise ist verwandt zum Ansatz der Partikelfilter. Der große Vorteil des hier vorgestellten Verfahrens besteht allerdings darin, dass die Approximation der Dichte deterministisch ist, d. h. die Approximation liefert immer die selbe (optimale) Repräsentierung, im Gegensatz zu Monte-Carlo-Methoden, wo die Approximation durch zufälliges Ziehen aus der Dichte entsteht. Daraus erwachsen zwei weitere Vorteile: Zum einen ist, wie in Abschnitt 4.6.1 gezeigt, die Konvergenzrate der systematischen Dirac-Mixture-Approximation wesentlich höher, zum anderen lässt sich dadurch die optimale Anzahl der benötigten Komponenten bestimmen (Algorithmus 4.2). Dieses Verfahren birgt jedoch auch ein paar Nachteile. So ist der Berechnungsaufwand im Vergleich zum zufälligen Ziehen deutlich höher ( $O(n^3)$  gegenüber  $O(n)$ ). Dies lässt sich zwar durch geschickte Implementierung abmildern, muss aber für jeden Inferenzschritt wiederholt werden. Hier muss ganz klar zwischen Aufwand und Genauigkeit abgewogen werden.

Wann welches der beiden Verfahren angewendet werden sollte, hängt stark von den gegebenen Teilsystemen ab. Wenn keine Systemapproximation gefunden werden kann, lässt sich der zweite Ansatz immer anwenden, wenn auch mit einer höheren Laufzeitanforderung. Eine Systemapproximation kann auf der anderen Seite offline berechnet werden und bei der Zustandsschätzung in  $O(n)$  ausgewertet werden. In beiden Fällen gilt, dass Genauigkeit und Rechenaufwand durch die Wahl der Anzahl der Approximationskomponenten gegeneinander abgewogen werden können. Beide Verfahren können in jedem Fall innerhalb der selben Kaskade angewendet werden. Dazu wurde gezeigt, dass die Dichterepräsentationen der beiden Verfahren kompatibel



---

sind und jeweils vom einen zum anderen Verfahren weitergereicht werden können, wodurch eine durchgehende Unsicherheitsbetrachtung ermöglicht wird.

Wenn keine Systemmodelle in probabilistischer oder generativer Form vorliegen, so kann ein entsprechendes Modell evtl. aus Daten geschätzt werden. Ein neuer Ansatz dazu wird in Kapitel 5 gezeigt. Dieses Verfahren stellt eine Alternative zum Maximum-Likelihood-Verfahren dar, da es für kleine Mengen von Beobachtungen weniger Probleme mit lokalen Minima hat.

Als eine mögliche Anwendung von kaskadierten Systemen wird in Teil II die Intentionserkennung in der Mensch-Roboter-Kooperation betrachtet. Diese Arbeit enthält nach aktuellem Kenntnisstand die erste systemtheoretische Definition der Intentionserkennung. Hierzu wird ein generisches Modell eingeführt, das auf der Idee der kaskadierten Systeme beruht. Dieses Modell kann für konkrete Anwendungen leicht instanziiert werden. Dazu wird ein Verfahren vorgestellt, welches zeigt, wie hierfür Expertenwissen eingesetzt werden kann. Die Methode wurde in einer Evaluationsumgebung mit menschlichen Probanden erfolgreich getestet.

In Zukunft gilt es, die vorgestellten Verfahren zu erweitern und zu verbessern. Die größte Herausforderung im Bereich der kaskadierten Systeme ist die Erweiterung der Verfahren auf mehrdimensionale Probleme. Für den Systemapproximationsansatz ist dies prinzipiell bereits möglich. Das Problem liegt hier hauptsächlich in der exponentiell wachsenden Anzahl der benötigten Komponenten bei Beschränkung auf die Achsenausrichtung. Für die Dichteapproximation sind bereits Erweiterungen für den mehrdimensionalen Fall in der Entwicklung. Hier zeigt sich, dass für eine globale Optimierung ähnliche Komplexitätsprobleme entstehen. Aus diesem Grund werden suboptimale Verfahren entwickelt, die diesen Problemen entgegenwirken.

Für die Intentionserkennung werden in Zukunft bessere Modelle für die Erkennung von Aktionen benötigt. Hierbei ist besonders die Bewegung des Menschen von großem Interesse. Darüber hinaus sollten die Modellparameter online gelernt und verbessert werden können. Eine weitere Verbesserung ist von der Verwendung von Planwissen über mögliche Aktionsabfolgen zu erwarten.



# Literaturverzeichnis

- [1] ALBRECHT, DAVID: *Aufbau einer virtuellen Simulationsumgebung für die Mensch-Roboter-Kooperation zur Erkennung von Benutzeraktionen*. Diplomarbeit, Universität Karlsruhe (TH), Karlsruhe, April 2007.
- [2] ALBRECHT, DAVID W., INGRID ZUKERMAN und ANNE E. NICHOLSON: *Bayesian Models for Keyhole Plan Recognition in an Adventure Game*. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.
- [3] ALDRICH, JOHN: *R. A. Fisher and the Making of Maximum Likelihood 1912-1922*. *Statistical Science*, 12(3):162–176, August 1997.
- [4] ALSPACH, DANIEL L. und HAROLD W. SORENSON: *Nonlinear Bayesian Estimation using Gaussian Sum Approximations*. *IEEE Transactions on Automatic Control*, 17(4):439–448, August 1972.
- [5] ANSCOMBE, GERTRUDE E. M.: *Intention*. Blackwell, 2. ed., repr. Auflage, 1976.
- [6] ARULAMPALAM, S., S. MASKELL, N. GORDON und T. CLAPP: *A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking*. *IEEE Transactions of Signal Processing*, 50(2):174–188, 2002.
- [7] BECHER, R., P. STEINHAUS und R. DILLMANN: *The Collaborative Research Center 588: Humanoid Robots – Learning and Cooperating Multimodal Robots*. In: *Proceedings of Humanoids 2003, Karlsruhe, Germany*, 2003.
- [8] BEUTLER, FREDERIK und UWE D. HANEBECK: *Closed-Form Range-Based Posture Estimation Based on Decoupling Translation and Orientation*. In: *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, Band 4, Seiten 989–992, Philadelphia, USA, März 2005.
- [9] BEYERER, J.: *Fusion heterogener Informationen: Essay*. *visIT*, (2005/2):4–5, 2005.
- [10] BEYERER, J., J. SANDER und S. WERLING: *Bayes'sche Methodik zur lokalen Fusion heterogener Informationsquellen*. *tm Technisches Messen*, 74(3):103–111, 2007.
- [11] BØLVIKEN, E., P. J. ACKLAM, N. CHRISTOPHERSEN und J.-M. STØRDAL: *Monte Carlo Filters for Non-Linear State Estimation*. *Automatica*, 37:177–183, 2001.

- [12] BOOS, D. D.: *Minimum Distance Estimators for Location and Goodness of Fit*. Journal of the American Statistical association, 76(375):663–670, 1981.
- [13] BOYEN, XAVIER, NIR FRIEDMAN und DAPHNE KOLLER: *Discovering the Hidden Structure of Complex Dynamic Systems*. In: *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence —UAI1999*, Seiten 91–100. San Francisco: Morgan Kaufmann, 1999.
- [14] BREAZEAL, CYNTHIA: *Robots in Society: Friend or Appliance?* In: *Agents99 Workshop on Emotion-based Agent Architecture*, Seiten 18–26, Seattle, WA, 1999.
- [15] BREAZEAL, CYNTHIA und BRIAN SCASSELLATI: *How to build robots that make friends and influence people*. In: *Proceedings of the IROS*, 1999.
- [16] BRIECHLE, KAI und UWE D. HANEBECK: *Simultane Lokalisierung und Kartenaufbau für einen mobilen Serviceroboter*. In: G. SCHMIDT, U. D. HANEBECK, F. FREYBERGER (Herausgeber): *Autonome Mobile Systeme 1999 (AMS'99)*, 15. Fachgespräch, München, *Informatik aktuell*, Seiten 200–210. Springer, Munich, Germany, November 1999.
- [17] BRIECHLE, KAI und UWE D. HANEBECK: *Localization of a Mobile Robot Using Relative Bearing Measurements*. IEEE Transactions on Robotics and Automation, 20:36–44, Februar 2004.
- [18] BRIGO, DAMIANO, BERNARD HANZON und FRANCOIS LEGLAND: *A Differential Geometric Approach to Nonlinear Filtering: The Projection Filter*. IEEE Transactions on Automatic Control, 42(2):247–252, 1998.
- [19] BRONSTEIN, I.N. und K.A. SEMENDJAJEW: *Taschenbuch der Mathematik*, 25. 1991.
- [20] CHALLA, SUBHASH, YAAKOV BAR-SHALOM und VIKRAM KRISHNAMURTHY: *Nonlinear filtering via generalized Edgeworth series and Gauss-Hermite quadrature*. Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on], 48(6):1816–1820, Juni 2000.
- [21] CHARNIAK, EUGENE: *Bayesian Networks without Tears*. AI Magazine, Seiten 50–63, 1991.
- [22] CODY, W. J.: *Rational Chebyshev Approximations for the Error Function*. Mathematics of Computation, 23(107):631–637, 1969.
- [23] DAUTENHAHN, K.: *Robots as Social Actors: AURORA and The Case of Autism*. In: *Proceedings of the Third International Cognitive Technology Conference, CT99, San Francisco*, August 1999.
- [24] DAVIDSON, DONALD: *Actions, Reasons, and Causes*. The Journal of Philosophy, 60(23):685–700, November 1963.

- 
- [25] DEMPSTER, A. P., LAIRD, N. M. und RUBIN, D. B.: *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.
- [26] DORMAND, J. R. und P. J. PRINCE: *A family of embedded Runge-Kutta formulae*. Journal of Computational and Applied Mathematics, 6:19–26, 1980.
- [27] DOUCET, A., N. DE FREITAS und N. GORDON: *Sequential Monte Carlo Methods in Practice*. Statistics for engineering and Information Science. Springer Verlag, 2000.
- [28] DOUCET, A., S. GODSILL und C. ANDRIEU: *On Sequential Monte Carlo Sampling Methods for Bayesian Filtering*. Statistics and Computing, 10(3):197–208, 2000.
- [29] DRIVER, ERIC und DARRYL MORRELL: *Implementation of continuous Bayesian networks using sums of weighted Gaussians*. In: BESNARD und HANKS (Herausgeber): *Proc. 11th Conf. Uncertainty in Artificial Intelligence*, 1995.
- [30] EBERLING, JULIA: *Approximation bedingter Wahrscheinlichkeiten für Bayes-Netze durch Gaußmischverteilungen*. Studienarbeit, Universität Karlsruhe (TH), Karlsruhe, Juni 2006.
- [31] EDDY, T.J., G.G. GALLUP JR. und D.J. POVINELLI: *Attribution of cognitive states to animals: Anthropomorphism in comparative perspective*. Journal of Social Issues, 49:87–101, 1993.
- [32] ELIDAN, G. und N. FRIEDMAN: *Learning the dimensionality of hidden variables*. In: *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference*, Seiten 144–151, 2001.
- [33] ELIDAN, GAL, NOAM LOTNER, NIR FRIEDMAN und DAPHNE KOLLER: *Discovering Hidden Variables: A Structure-Based Approach*. In: *NIPS*, Seiten 479–485, 2000.
- [34] FAGAN, MICHAEL und PADRAIG CUNNINGHAM: *Case-Based Reasoning Research and Development*, Band 2689/2003 der Reihe *Lecture Notes in Computer Science*, Kapitel Case-Based Plan Recognition in Computer Games, Seiten 161–170. Springer Berlin / Heidelberg, 2003.
- [35] FREEDMAN, DAVID und PERSI DIACONIS: *On the Histogram as a Density Estimator: L2 Theory*. Probability Theory and Related Fields, V57(4):453–476, Dezember 1981.
- [36] GEWEKE, J.: *Bayesian Inference in Econometric Models using Monte Carlo Integration*. Econometrica, 24:1317–1399, 1989.
- [37] GOLDMAN, ROBERT, CHRISTOPHER GEIB und CHRISTOPHER MILLER: *A New Model of Plan Recognition*. In: *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Seiten 245–25, San Francisco, CA, 1999. Morgan Kaufmann.
-

- [38] GORDON, N.: *Bayesian Methods for Tracking*. Doktorarbeit, University of London, 1993.
- [39] GROENDA, HENNING, FABIAN NOWAK, PATRICK RÖSSLER und UWE D. HANEBECK: *Telepresence Techniques for Controlling Avatar Motion in First Person Games*. In: *Intelligent Technologies for Interactive Entertainment, First International Conference, INTETAIN 2005, Madonna di Campiglio, Italy*, Seiten 44–53, 2005.
- [40] HAIRER, E., S. P. NORSETT und G. WANNER: *Solving Ordinary Differential Equations I*. Springer Verlag, 2000.
- [41] HANEBECK, UWE D.: *Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Measurement Step*. In: *Proceedings of the 2003 IEEE Conference on Multi-sensor Fusion and Integration for Intelligent Systems (MFI 2003)*, Seiten 173–178, Tokyo, Japan, Juli 2003.
- [42] HANEBECK, UWE D., KAI BRIECHLE und ANDREAS RAUH: *Progressive Bayes: A New Framework for Nonlinear State Estimation*. In: *Proceedings of SPIE*, Band 5099, Orlando, Florida, Mai 2003. AeroSense Symposium.
- [43] HANEBECK, UWE D. und OLGA FEIERMANN: *Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Filter Step for Scalar Measurements and Multidimensional States*. In: *Proceedings of the 2003 IEEE Conference on Decision and Control (CDC 2003)*, Seiten 5366–5371, Maui, Hawaii, Dezember 2003.
- [44] HANEBECK, UWE D. und OLIVER C. SCHREMPF: *Skriptum zur Vorlesung Stochastische Informationsverarbeitung*. Lehrstuhl für Intelligente Sensor-Aktor-Systeme, Universität Karlsruhe (TH), 2007.
- [45] HANSELMANN, ANNE, OLIVER C. SCHREMPF und UWE D. HANEBECK: *Optimal Parametric Density Estimation by Minimizing an Analytic Distance Measure*. In: *The 10th International Conference on Information Fusion 2007*, Quebec, Canada, Juli 2007.
- [46] HANSON, KENNETH M.: *Introduction to Bayesian Image Analysis*. In: LOEW, MURRAY H. (Herausgeber): *Proceedings of SPIE*, Band 1898 der Reihe *Medical Imaging*, Seiten 761–731, 1993.
- [47] HENRIKSEN, ROLF: *The truncated second-order nonlinear filter revisited*. *IEEE Transactions on Automatic Control*, 27(1):247– 251, Februar 1982.
- [48] HORVITZ, ERIC, JACK BREESE, DAVID HECKERMAN, DAVID HOVEL und KOOS ROMMELSE: *The Lumière Project: Bayesian User Modelling for Inferring the Goals and Needs of Software Users*. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Seiten 256–265. AUA I, Morgan Kaufman, jul 1998.
- [49] HUBER, MARCO, DIETRICH BRUNN und UWE D. HANEBECK: *Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density*. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Seiten 98–103, Heidelberg, Germany, September 2006.

- 
- [50] HUBER, MARCO F. und UWE D. HANEBECK: *Hybrid Transition Density Approximation for Efficient Recursive Prediction of Nonlinear Dynamic Systems*. In: *International Conference on Information Processing in Sensor Networks (IPSN 2007)*, Seiten 283–292, Cambridge, Massachusetts, USA, April 2007.
- [51] HUBER, MARCO F. und UWE D. HANEBECK: *The Hybrid Density Filter for Nonlinear Estimation based on Hybrid Conditional Density Approximation*. In: *10th International Conference on Information Fusion (Fusion 2007)*, Quebec, Canada, Juli 2007.
- [52] IZENMAN, A. J.: *Recent developments in nonparametric density estimation*. *Journal of the American Statistical Association*, 86(413):205–224, März 1991.
- [53] JOHNSON, D. H. und S. SINANOVIC: *Symmetrizing the Kullback–Leibler Distance*. Submitted to *IEEE Signal Processing Letters*, 2002.
- [54] JULIER, S. und J. UHLMANN: *A New Extension of the Kalman Filter to Nonlinear Systems*. In: *Proceedings of SPIE AeroSense, 11th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL*, 1997.
- [55] JULIER, SIMON J. und JEFFREY K. UHLMANN: *Unscented Filtering and Nonlinear Estimation*. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [56] KOCIS, L. und W. J. WHITEN: *Computational Investigations of Low-Discrepancy Sequences*. *ACM Trans. Math. Softw.*, 23(2):266–294, 1997.
- [57] KSCHISCHANG, FRANK R., BRENDAN J. FREY und HANS-ANDREA LOELIGER: *Factor Graphs and the Sum-Product Algorithm*. *IEEE Transactions on Information Theory*, 47(2):498–519, Februar 2001.
- [58] KULLBACK, S. und R. A. LEIBLER: *On Information and Sufficiency*. *Annals of Mathematical Statistics*, 22(2):79–86, 1951.
- [59] KWOH, C.-K. und D. GILLES: *Using Hidden Nodes in Bayesian Networks*. *Artificial Intelligence*, 88:1–38, 1996.
- [60] LAURITZEN, S. und D. SPIEGELHALTER: *Local computations with probabilities on graphical structures and their application to expert systems*. *Journal of the Royal Statistical Society*, 50(19):157–224, 1988.
- [61] LAURITZEN, STEFFEN L.: *Propagation of Probabilities, Means and Variances in Mixed Graphical Association Models*. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.
- [62] LERNER, URI, ERAN SEGAL und DAPHNE KOLLER: *Exact Inference in Networks with Discrete Children of Continuous Parents*. In: *Proceedings of the 17<sup>th</sup> Conference on Uncertainty in Artificial Intelligence*, Seiten 319–238, Seattle, Washington, USA, 2001.
-

- [63] LESH, N., C. RICH und C. SIDNER: *Using Plan Recognition in Human-Computer Collaboration*. In: *Proceedings of the Seventh International Conference on User Modeling, Banff, Canada*, Seiten 23–32, 1999.
- [64] LLOYD, S. P.: *Least Squares Quantization in PCM*. IEEE Transactions on Information Theory, IT-28(2):129–137, Mar. 1982.
- [65] LOELIGER, H.-A.: *An introduction to factor graphs*. Signal Processing Magazine, IEEE, 21(1):28–41, 2004.
- [66] MARTIN, J. und K. VANLEHN: *Discrete Factor Analysis: Learning Hidden Variables in Bayesian networks*. Technischer Bericht, Department of Computer Science, University of Pittsburgh, 1995.
- [67] MOROKOFF, WILLIAM J. und RUSSEL E. CAFLISCH: *Quasi-Random Sequences and Their Discrepancies*. SIAM Journal on Scientific Computing, 15(6):1251–1279, 1994.
- [68] MURPHY, KEVIN P.: *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD Thesis, UC Berkeley, Computer Science Division, Juli 2002.
- [69] MURPHY, KEVIN P., YAIR WEISS und MICHAEL I. JORDAN: *Loopy Belief Propagation for Approximate Inference: An Empirical Study*. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden*, Seiten 467–475, 1999.
- [70] MUSSO, C., N. OUDJANE und F. LEGLAND: *Improving Regularized Particle Filters*. Technischer Bericht CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.
- [71] NIEDERREITER, G.: *Quasi-Monte Carlo methods and pseudo-random numbers*. Bulletin of the American Mathematical Society, 84(6):957–1041, 1978.
- [72] NITZSCHE, NORBERT, UWE D. HANEBECK und GÜNTHER SCHMIDT: *Motion Compression for Telepresent Walking in Large Target Environments*. Presence: Teleoperators & Virtual Environments, 13(1):44–60, 2004.
- [73] OLESEN, KRISTIAN G.: *Causal Probabilistic Networks with Both Discrete and Continuous Variables*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(3):275–279, März 1993.
- [74] OUDJANE, N. und S. RUBENTHALER: *Stability and Uniform Particle Approximation of Nonlinear Filters in Case of Non-Ergodic Signal*. Technischer Bericht PMA786, Prépublication, Laboratoire de Probabilités et Modèles Aléatoires, Université de Paris VI, Januar 2003.
- [75] PAPOULIS, ATHANASIOS und S. UNNIKRIISHNA PILLAI: *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 4 Auflage, 2002.



- [76] PARZEN, EMANUEL: *On Estimation of a Probability Density Function and Mode*. The Annals of Mathematical Statistics, 33(3):1065–1076, Sep. 1962.
- [77] PEARL, J.: *Fusion, propagation, and structuring in belief networks*. Artificial Intelligence, 29:241–288, 1986.
- [78] PEARL, JUDEA: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [79] PEARL, JUDEA und STUART RUSSELL: *Bayesian Networks*. Technischer Bericht R-277, Department of Computerscience, University of California, Berkeley, November 2000.
- [80] PEARSON, KARL: *On the Systematic Fitting of Curves to Observations and Measurements*. Biometrika, 1(3):265–303, Apr. 1902.
- [81] PETERKA, V.: *Bayesian System Identification*. Automatica, 17(1):41–53, Jan. 1981.
- [82] PHILLEX.DE: *Philosophenlexikon*. <http://www.phillex.de>.
- [83] PHILOMIN, V., R. DURAI SWAMI und L. DAVIS: *Quasi-Random Sampling for Condensation*, Band 1843/2000 der Reihe *Lecture Notes in Computer Science*, Kapitel Computer Vision - ECCV 2000, Seiten 134–149. Springer Berlin / Heidelberg, 2000.
- [84] PRESS, W. H., B. P. FLANNERY, S. A. TEUKOLSKY und W. T. VETTERLING: *Kolmogorov–Smirnov Test*. In: *Numerical Recipes in C: The Art of Scientific Computing*, Seiten 623–628. Cambridge University Press, 1992.
- [85] Q. YAO und H. TONG: *On Prediction and Chaos in Stochastic Systems*. Philosophical Transactions: Physical Sciences and Engineering, 348(1688):357–369, sep 1994.
- [86] RABINER, L.: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, 77(2), 1989.
- [87] RAMACHANDRAN, SOWMYA und RAYMOND J. MOONEY: *Theory Refinement for Bayesian Networks with Hidden Variables*. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.
- [88] RAMONI, MARCO und PAOLA SEBASTIANI: *Parameter Estimation in Bayesian Networks from Incomplete Databases*. Intell. Data Anal., 2(1-4):139–160, 1998.
- [89] RUSSELL, STEWARD J. und PETER NORVIG: *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
- [90] RÖSSLER, PATRICK, OLIVER C. SCHREMPF und UWE D. HANEBECK: *Stochastic Prediction of Waypoints for Extended-Range Telepresence Applications*. In: *2nd International Workshop on Human Centered Robotic Systems (HCRS 2006)*, Seiten 85–89, Munich, Germany, Oktober 2006.

- [91] SANDER, J. und J. BEYERER: *Fusion agents - realizing Bayesian fusion via a local approach*. In: *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI06)*, Heidelberg, Seiten 249–254, 2006.
- [92] SCHREMPF, OLIVER C., DAVID ALBRECHT und UWE D. HANEBECK: *Tractable Probabilistic Models for Intention Recognition Based on Expert Knowledge*. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2007.
- [93] SCHREMPF, OLIVER C., DIETRICH BRUNN und UWE D. HANEBECK: *Density Approximation Based on Dirac Mixtures with Regard to Nonlinear Estimation and Filtering*. In: *IEEE Conference on Decision and Control (CDC 2006)*, San Diego, USA, Dezember 2006.
- [94] SCHREMPF, OLIVER C., DIETRICH BRUNN und UWE D. HANEBECK: *Dirac Mixture Density Approximation Based on Minimization of the Weighted Cramér-von Mises Distance*. In: *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Seiten 512–517, Heidelberg, Germany, September 2006.
- [95] SCHREMPF, OLIVER C., OLGA FEIERMANN und UWE D. HANEBECK: *Optimal Mixture Reduction of the Product of Mixtures*. In: *8th International Conference on Information Fusion (Fusion 2005)*, Band 1, Seiten 85–92, Philadelphia, USA, Juli 2005.
- [96] SCHREMPF, OLIVER C. und UWE D. HANEBECK: *A New Approach for Hybrid Bayesian Networks Using Full Densities*. In: *Proceedings of the 6th International Workshop on Computer Science and Information Technologies (CSIT 2004)*, Band 1, Seiten 32–37, Budapest, Hungary, Oktober 2004.
- [97] SCHREMPF, OLIVER C. und UWE D. HANEBECK: *A Generic Model for Estimating User Intentions in Human-Robot Cooperation*. In: *2nd Intl. Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, Band 3, Seiten 251–256, Barcelona, Spain, September 2005.
- [98] SCHREMPF, OLIVER C. und UWE D. HANEBECK: *Evaluation of Hybrid Bayesian Networks using Analytical Density Representations*. In: *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, Juli 2005. IFAC.
- [99] SCHREMPF, OLIVER C. und UWE D. HANEBECK: *A State Estimator for Nonlinear Stochastic Systems Based on Dirac Mixture Approximations*. In: *4th Intl. Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, Band SPSMC, Seiten 54–61, Angers, France, Mai 2007.
- [100] SCHREMPF, OLIVER C. und UWE D. HANEBECK: *Recursive Prediction of Stochastic Nonlinear Systems Based on Dirac Mixture Approximations*. In: *Proceedings of the American Control Conference (ACC 2007)*, New York City, USA, Juli 2007.
- [101] SCHREMPF, OLIVER C., ANNE HANSELMANN und UWE D. HANEBECK: *Efficient Representation and Fusion of Hybrid Joint Densities for Clusters in Nonlinear Hybrid Bayesian*

- Networks*. In: *The 9th International Conference on Information Fusion (Fusion 2006)*, Florence, Italy, Juli 2006.
- [102] SIMON, DAN: *Optimal State Estimation*. John Wiley & Sons, 2006.
- [103] TAHBOUB, KARIM A.: *Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition*. *Journal of Intelligent Robotics Systems*, 45(1):31–52, 2006.
- [104] TITTERINGTON, D. M., A. F. M. SMITH und U. E. MAKOV: *Statistical Analysis of Finite Mixture Distributions*. *Wiley Series in Probability and Mathematical Statistics*. John Wiley & Sons, 1985.
- [105] WATT, S.: *A Brief Naive Psychology Manifesto*. *Informatica*, 19:495–500, 1995.
- [106] WEBER, M. D., L. M. LEEMIS und R. K. KINCAID: *Minimum Kolmogorov-Smirnov Test Statistic Parameter Estimates*. *Journal of Statistical Computation and Simulation*, 76(3):196–206, 2006.
- [107] WEISSTEIN, ERIC W.: *Newton's Method*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/NewtonsMethod.html>.
- [108] WELLMAN, MICHAEL P. und CHAO-LIN LIU: *State-Space Abstraction for Anytime Evaluation of Probabilistic Networks*. In: *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, Seattle, Washington, USA*, Seiten 567–574, 1994.
- [109] WISSEN.DE: *wissen.de-Lexikon*. <http://www.wissen.de>, 2007.
- [110] WONHAM, W. M.: *Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering*. *SIAM Journal on Control and Optimization*, 2(3):347 – 369, 1965.
- [111] ZÖLLNER, R., O. ROGALLA, R. DILLMANN und M. ZÖLLNER: *Understanding User's Intention: Programming Fine Manipulation Tasks by Demonstration*. In: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland*, 2002.

In dieser Arbeit werden theoretische und praktische Aspekte der stochastischen Unsicherheitsbetrachtung in kaskadierten dynamischen Systemen behandelt. Die dabei verfolgte Idee besteht darin, Systeme bottom-up aus sehr einfachen Teilsystemen aufzubauen, indem der Ausgang eines Systems als Eingang eines folgenden Systems verwendet wird. Das Ziel ist es, für solche Systemkaskaden eine stochastische Zustandsschätzung durchzuführen. Dabei kann, wie in dieser Arbeit gezeigt wird, die Struktur der Kaskade verwendet werden, um die Zustandsschätzung lokal in den Teilsystemen durchzuführen. Die resultierende Schätzung können dann an die benachbarten Systeme weitergegeben werden.

Als eine mögliche Anwendung von kaskadierten Systemen wird die Intentionserkennung in der Mensch-Roboter-Kooperation betrachtet. Diese Arbeit enthält nach aktuellem Kenntnisstand die erste systemtheoretische Definition der Intentionserkennung. Hierzu wird ein generisches Modell eingeführt, das auf der Idee der kaskadierten Systeme beruht. Dieses Modell kann für konkrete Anwendungen leicht instanziiert werden. Dazu wird ein Verfahren vorgestellt, welches zeigt, wie hierfür Expertenwissen eingesetzt werden kann.