FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Neue Algorithmen der automatischen Spracherkennung

unter besonderer Berücksichtigung der speziellen Situation und der Bedürfnisse von Menschen mit Behinderungen

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

eingereicht von

Helmut Hickersberger
Matrikelnummer 9125166

an der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Zagler

Diese Dissertation haben begutachtet:

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Zagler	Ao.Univ.Prof. i.R. Dipl.-Ing. Dr.sc.med. Dr.techn. Dr.rer.nat. Frank Rattay

Wien, 05.04.2013

Helmut Hickersberger

Erklärung zur Verfassung der Arbeit

Helmut Hickersberger
1200 Wien, Brigittaplatz 1-2/10/14

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen – die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 05.04.2013 _____ Helmut Hickersberger

Gleichbehandlung. Aus Gründen der leichteren Lesbarkeit wird im Rahmen des vorliegenden Textes auf eine geschlechtsspezifische Differenzierung, wie z.B. „Benutzer/innen“, verzichtet. Sämtliche entsprechenden Begriffe gelten jedoch im Sinne des Bundesgleichbehandlungsgesetzes (B-GBG) (BGBl. Nr.100/1993 i.d.g.F.) für beide Geschlechter gleichermaßen.

Sprache. Sprechen. Im Rahmen des vorliegenden Textes werden die Begriffe „Spracherkenner“, „Sprachsteuerung“, etc. verwendet. Die Begriffsbildung ist genaugenommen fragwürdig, weil der Begriff „Sprache“ unabhängig von den Sprechmöglichkeiten der Benutzer definiert ist (siehe zum Beispiel Abschnitt 2.1.2 auf Seite 103 und Abschnitt 2.1.3 auf Seite 111) und dabei nicht gemeint ist. Die mit „Sprach…“ beginnenden Begriffe werden im Rahmen dieses Textes jedoch verwendet, da sie in der Literatur etabliert sind, obwohl es sich freilich um eine Erkennung des „gesprochenen Wortes“ handelt und die Begriffe „Sprech-Erkenner“, „Sprech-Steuerung“, etc. genaugenommen eher zutreffender wären. Insbesondere soll dies keine Diskriminierung von Benutzern mit Sprachstörungen beziehungsweise Sprechstörungen bedeuten.

Danksagung

Danken möchte ich all jenen Personen, die mich auf dem Weg zum Rigorosum moralisch sowie mit Rat und Tat unterstützt haben. Meinem Betreuer, Herrn Univ. Prof. Dr. Wolfgang Zagler, gilt besonderer Dank. Als Leiter der „Forschungsgruppe Rehabilitationstechnik“ gab er mir Gelegenheit, den Fokus der Arbeit auf die besondere Berücksichtigung der Bedürfnisse der Zielgruppe der Menschen mit Behinderungen zu legen. An dieser Stelle möchte ich auch Herrn Mag. Wolfgang Tschirk, dem Software-Architekten der Siemens SICARE-Sprachsteuerungen danken, dessen inspirierende Diskussionen über Algorithmen am Gebiet der automatischen Spracherkennung mir äußerst wertvoll waren. Mein Dank gilt natürlich auch meinen Freunden und Verwandten, insbesondere meinen Eltern, welche durch großzügige moralische und finanzielle Unterstützung mein Wirken erst ermöglichten.

Abstract

In this work, a robust and simple voice command system is developed, implemented and tested. The algorithms are particularly suitable for speaker-independent and speaker-dependent isolated-word recognition, for example for the purpose of controlling the environment.

i) State of the art. Inside voice command systems, command words are usually represented via command word models. Each of the command word models consists of a sequence of “sub-word unit models”, for example “phoneme models”. The Viterbi dynamic programming algorithm is currently the de facto standard in order to determine the naturally varying durations of the “sub-word units”, for example linguistic “phones”, that are considered to be contained in an observed speech signal in the course of evaluating a command word model. The points in time, at which a command word model switches from one sub-word unit model to the next, are determined optimally, in some sense. The Viterbi dynamic programming algorithm is an essential integral part of the hidden Markov model speech recognizers.

ii) Main thesis. It is evaluated whether, corresponding to the main thesis, a particular modification of the Viterbi dynamic programming algorithm called "run-length limited dynamic programming algorithm" causes a statistically significant improvement of the confusion error rate: Using this modification, the points in time at which the sub-word unit models are switched, are searched within a restricted set of solutions, containing only those solutions, for which additional constraints apply for the run-lengths of the sub-word-unit models.

iii) Side theses. It is further evaluated whether, corresponding to the side theses, the following measures cause a statistically significant improvement of the confusion error rate: Linear transformation of the feature vectors according to a linear discriminant analysis; usage of alternative window functions for the short-time Fourier transformation; improvement of the modeling of the command words by additional prediction models that consist of “hidden control neural networks”.

iv) Methods. The measures to be evaluated are activated in all possible combinations and are tested using two speech databases (German command words, English command words). Each of the measures is tested for statistical significance by means of a Wilcoxon signed-rank test with Bonferroni correction.

v) Results. The measure of the main thesis is the only one that leads to a statistically significant reduction of the confusion error rate. In this case, the improved modeling of the command words using “hidden control neural networks” is not necessary and simple “centroid sub-word unit models” can be used.

vi) Application. The voice command system developed in this work enables the AUTONOMY system [Zagler1997] [Panek2002] [Loidolt1995] to be controlled by voice commands. The AUTONOMY system consists of an environmental control system combined with an alternative and augmentative communication system. Based on the needs of people with disabilities, requirements for the system design of the voice command system are stated. In the course of this work the voice command system is implemented and prepared for practical operation. It is noteworthy that the voice command system is designed to be used by people with speech disorders: The respective command words can be chosen arbitrarily and individually, as long as the implemented “automatic quality check module” qualifies the command word recordings to be sufficient to provide a good classification quality. In preparation for future research, special logging dialogs are configured, which occasionally ask the user, whether the system did understand correctly.

Kurzfassung

Im Rahmen dieser Arbeit wird eine robuste und einfache Sprachsteuerung entwickelt, implementiert und getestet. Die eingesetzten Algorithmen eignen sich besonders zur sprecherunabhängigen und sprecherabhängigen Einzelworterkennung, beispielsweise zum Zwecke der Steuerung der Umgebung.

i) State-of-the-Art. In Sprachsteuerungen werden Steuerbefehle in der Regel mittels Steuerwortmodellen repräsentiert. Jedes der Steuerwortmodelle besteht aus einer Folge von „Sub-Word-Unit-Modellen“, beispielsweise Phonem-Modellen. Der „Viterbi-Dynamic-Programming-Algorithmus“ gilt derzeit als De-facto-Standard, um bei der Bewertung eines Steuerwortmodells in Bezug auf ein beobachtetes Sprachsignal, die auf natürliche Weise schwankenden zeitlichen Längen der im Sprachsignal als vorhanden angenommenen „Sub-Word-Units“ (beispielsweise Phone) zu bestimmen. Es werden dabei die, in bestimmter Hinsicht, optimalen Umschaltzeitpunkte ermittelt, zu denen von einem Sub-Word-Unit-Modell zum nächsten umgeschaltet wird. Der „Viterbi-Dynamic-Programming-Algorithmus“ ist unter anderem ein wesentlicher Bestandteil der Hidden-Markov-Modell-Spracherkenner.

ii) Hauptthese. Es wird evaluiert, ob eine bestimmte Modifikation am „Viterbi-Dynamic-Programming-Algorithmus“ zu einer statistisch signifikanten Verbesserung der Verwechslungsfehlerrate führt: Diese Modifikation „Run-Length-Limited-Dynamic-Programming-Algorithmus“ besteht darin, die Umschaltzeitpunkte zwischen den Sub-Word-Unit-Modellen mittels dynamischer Programmierung nur unter jenen Möglichkeiten zu suchen, bei denen zusätzliche „Lauflängenbedingungen“ für die Sub-Word-Unit-Modelle eingehalten werden.

iii) Nebenthesen. Es wird weiters evaluiert, ob folgende Maßnahmen zu einer statistisch signifikanten Verbesserung der Verwechslungsfehlerrate führen: Lineare Transformation der Merkmalsvektoren gemäß linearer Diskriminanzanalyse, Verwendung alternativer Fensterfunktionen bei der Kurzzeit-Fourier-Transformation, Verbesserung der Wortmodellierung mittels zusätzlicher „Prediction Models“ bestehend aus „Hidden Control Neural Networks".

iv) Methodik. Die zu evaluierenden Verbesserungsmaßnahmen werden in allen möglichen Kombinationen aktiviert und anhand zweier Sprachdatenbanken (deutsche Steuerbefehle, englische Steuerbefehle) getestet. Für jede einzelne der zu evaluierenden Verbesserungsmaßnahmen wird mittels eines Wilcoxon-Vorzeichen-Rangtests unter Berücksichtigung der Bonferroni-Korrektur die statistische Signifikanz geprüft.

v) Resultat. Lediglich die der Hauptthese zugrunde liegende Verbesserungsmaßnahme führt zu einer statistisch signifikanten Reduktion der Verwechslungsfehlerrate. In diesem Falle ist die verbesserte Wortmodellierung mittels „Hidden Control Neural Networks“ gar nicht notwendig, und es können die deutlich einfacheren „Centroid-Sub-Word-Unit-Modelle“ verwendet werden.

vi) Anwendung. Die im Rahmen dieser Arbeit entwickelte Sprachsteuerung ermöglicht die Steuerung des AUTONOM-Systems [Zagler1997] [Panek2002] [Loidolt1995] mittels Einzelwort-Sprachkommandos. Beim AUTONOM-System handelt es sich um eine Umgebungssteuerung kombiniert mit einem alternativen und augmentativen Kommunikationssystem. Auf Basis der Bedürfnisse von Menschen mit Behinderungen werden Folgerungen für das Systemdesign der Sprachsteuerung abgeleitet. Die Sprachsteuerung wird implementiert und der praktische Einsatz vorbereitet. Bemerkenswert daran ist, dass eine Benutzung durch Menschen mit Sprechstörungen vorbereitet ist: Die jeweiligen Steuerbefehle können individuell und frei gewählt werden, soweit das implementierte automatische Qualitätssicherungsanalyse-Modul die gewählten Steuerbefehle als hinreichend klassifizierbar befindet. Spezielle Dialoge mit Protokollierung, welche Rückfragen beinhalten, ob das System richtig verstanden hat, werden – in Vorbereitung zukünftiger Forschungsvorhaben – vorkonfiguriert.

Inhaltsverzeichnis

Formelzeichen

Formelzeichen	Benennung	Beschreibung
I	Merkmalsvektoranzahl	Anzahl aufeinander folgender Merkmalsvektoren, die ein zu analysierendes Schallmuster bilden.
i	Zeitpunkt	Zeitpunkt im Schallmuster. Natürliche Zahl im geschlossenen Intervall von eins bis I.
$\mathbf{x}_i$	Merkmalsvektor	Beobachteter Merkmalsvektor zum Zeitpunkt i.
$\mathbf{X}, (\mathbf{x}_i), (\mathbf{x}'_i)$	Schallmuster	Folge von I beobachteten Merkmalsvektoren $\mathbf{x}_i$.
K	Merkmalsanzahl	Anzahl der Komponenten des Merkmalsvektors.
M	Wortmodellanzahl	Anzahl der Wortmodelle.
m	Wortmodellnummer	Nummer eines Wortmodells. Natürliche Zahl im geschlossenen Intervall von eins bis M.
M_{Keyw}	Keyword-Modellanzahl	Anzahl der Keyword-Modelle zur Modellierung der Steuerbefehle.
$M_{\text{Non-Keyw}}$	Non-Keyword-Modellanzahl	Anzahl der Non-Keyword-Modelle zur Modellierung von Hintergrundgeräuschen und Äußerungen, die keine Steuerbefehle sind.
$M_{\text{Act-Keyw}}$	Active-Keyword-Modellanzahl	Anzahl der Keyword-Modelle, die in der momentanen Dialogsituation aktiviert sind.
S_m	Zustandsanzahl	Anzahl der inneren Zustände des Wortmodells m.
$s_{i,m}$	Zustandsnummer	Zustandsnummer des Wortmodells m zum Zeitpunkt i. Natürliche Zahl im geschlossenen Intervall von eins bis S_m.
$\mathbf{c}_{i,m}$	Zustandsnummern-Codevektor	Zustandsnummer $s_{i,m}$ in codierter Form zur Verarbeitung mittels eines „Hidden Control Neural Network“. Üblicherweise findet der „1-aus-S_m-Code“ oder der „Thermometercode“ Verwendung.
$\mathbf{s}_m, (s_i)_m$	Zustandsnummernfolge	Folge der I Zustandsnummern für das Wortmodell m. Die Indizierung beginnt bei eins.
$\mathbf{s}_m^{\text{optimum}}$	Optimal-Zustandsnummernfolge	Optimale Zustandsnummernfolge, sodass die Pfadkosten das globale Minimum annehmen.
b_m	Wortmodelldistanz	Distanz des Wortmodells m zum beobachteten Schallmuster.
$L_{s,m}^{\min}$	Minimallauflänge	Minimale zulässige Lauflänge des Zustands s einer Zustandsnummernfolge beim Wortmodell m.
$L_{s,m}^{\max}$	Maximallauflänge	Maximale zulässige Lauflänge des Zustands s einer Zustandsnummernfolge beim Wortmodell m.
$e_{i,s,m}$	Modellabweichung	Abweichung der vom Wortmodell m im Zustand s zum Zeitpunkt i berechneten Merkmalsvektor-Hypothese vom tatsächlich beobachteten Merkmalsvektor zum Zeitpunkt i.
$\widehat{e}_{i,s,m}$	Teilpfadkosten	Kosten des Teilpfades zum Zustand s zum Zeitpunkt i beim Wortmodell m.
$\widehat{s}_{i,s,m}$	Wegweiserinformation	Optimaler Vorgängerzustand des Zustands s zum Zeitpunkt i beim Wortmodell m.
$d_{i,\,s',\,s,m}$	Übergangskosten	Kosten des Zustandswechsels vom Zustand s' zum Zustand s des Zeitpunkts i beim Wortmodell m.

$\hat{e}_{i,\,s',\,s,m}$	Zweigkosten	Summe aus den Übergangskosten $d_{i,s',s,m}$ und der Modellabweichung $e_{i,s,m}$.
$\mathbf{E}_m, \left(e_{i,s}\right)_m$	Abweichungsmatrix	Matrix der Modellabweichungen des Wortmodells m .
$\mathbf{w}_m$	Wortmodellparameter	Die freien Parameter des Wortmodells m , welche mittels Wortmodelltraining festgelegt werden.
N	Schallmusteranzahl	Anzahl der Schallmuster zum Zwecke des Tests oder des Trainings.
$N_{\text{Non-Keyw}}$	Non-Keyword-Schallmusteranzahl	Anzahl der Non-Keyword-Schallmuster zum Zwecke des Tests oder des Trainings.
N_{Keyw}	Keyword-Schallmusteranzahl	Anzahl der Keyword-Schallmuster zum Zwecke des Tests oder des Trainings.
μ	Sprachdatenbankkennzahl	Verhältnis der Anzahl der Non-Keyword-Schallmuster zur Anzahl der Keyword-Schallmuster einer Sprachdatenbank beziehungsweise bestimmter Einsatzumstände.
N_{OKR}	Korrektrückweisungsanzahl	Anzahl der korrekterweise zurückgewiesenen Non-Keyword-Schallmuster.
N_{OKA}	Korrektakzeptanzanzahl	Anzahl der korrekterweise akzeptierten Keyword-Schallmuster.
N_{FA}	Falschakzeptanzanzahl	Anzahl der fälschlicherweise akzeptierten Non-Keyword-Schallmuster.
N_{CE}	Verwechslungsfehleranzahl	Anzahl der falsch klassifizierten, akzeptierten Keyword-Schallmuster.
N'_{CE}	Nichtrückweisungs-Verwechslungsfehleranzahl	Anzahl der falsch klassifizierten Keyword-Schallmuster, wenn systembedingt keine Schallmuster zurückgewiesen werden.
α	Sprachdatenbank-Spracherkenner-Kennzahl	Verhältnis der Anzahl der falsch klassifizierten, akzeptierten Keyword-Schallmuster zur Anzahl der falsch klassifizierten Keyword-Schallmuster, wenn systembedingt keine Schallmuster zurückgewiesen werden.
N_{FR}	Falschrückweisungsanzahl	Anzahl der fälschlicherweise zurückgewiesenen Keyword-Schallmuster.
$N_{\text{FR,richtig}}$	Richtigklassifikations-Falschrückweisungsanzahl	Anzahl der fälschlicherweise zurückgewiesenen Keyword-Schallmuster, die richtig klassifiziert worden wären.
$N_{\text{FR,falsch}}$	Falschklassifikations-Falschrückweisungsanzahl	Anzahl der zurückgewiesenen Keyword-Schallmuster, die falsch klassifiziert worden wären.
γ	Sprachdatenbank-Spracherkenner-Kennzahl	Verhältnis der Anzahl der fälschlicherweise zurückgewiesenen Keyword-Schallmuster, die richtig klassifiziert worden wären zur Anzahl der zurückgewiesenen Keyword-Schallmuster, die falsch klassifiziert worden wären.
ρ	Sprachdatenbank-Spracherkenner-Kennzahl	Verhältnis der Anzahl der zurückgewiesenen richtig klassifizierten Keyword-Schallmuster zur Anzahl der richtig klassifizierten Keyword-Schallmuster.
CER, CER_{M}	Verwechslungsfehlerrate, „Confusion Error Rate“	Anteil falsch klassifizierter Keyword-Schallmuster an der Schallmusteranzahl.
CER', CER'_{M}	Nichtrückweisungs-Verwechslungsfehlerrate	Anteil falsch klassifizierter Keyword-Schallmuster an der Schallmusteranzahl, wenn systembedingt keine Schallmuster zurückgewiesen werden.
FAR, FAR_{M}	Falschakzeptanzrate, „False Acceptance Rate“, „Acceptance Error Rate“	Anteil fälschlicherweise akzeptierter Non-Keyword-Schallmuster an der Non-Keyword-Schallmusteranzahl.

FRR, FRR_M	Falschrückweisungsrate, „False Rejection Rate“, „Rejection Error Rate“	Anteil fälschlicherweise zurückgewiesener Keyword-Schallmuster an Keyword-Schallmusteranzahl.
U	Unzufriedenheit	Vermutete Unzufriedenheit der Benutzer mit dem Spracherkenner.
$\mathbf{A}_m$, $\mathbf{A}_{m,s}$	Hidden-Layer-Matrix	Netzgewichtsmatrix im „Hidden Layer“ eines Multi-Layer-Perceptron-Netzes mit einem einzigen „Hidden Layer“.
$\mathbf{a}_m$, $\mathbf{a}_{m,s}$	Offsetvektor	Offsetvektor eines Multi-Layer-Perceptron-Netzes.
$\mathbf{B}_m$, $\mathbf{B}_{m,s}$	Output-Layer-Matrix	Netzgewichtsmatrix im „Output Layer“ eines Multi-Layer-Perceptron-Netzes.
LDA	LDA-Transformationsmatrix	Transformationsmatrix der linearen Diskriminanzanalyse zur optimalen Separierung von Klassen.
$\hat{\alpha}, \hat{\beta}, \hat{\gamma}$	Linearkombinationsgewichte	Linearkombinationsgewichte der „Gaussian Mixture“.
$\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$	Gaussian-Mixture-Matrizen	Matrizen der „Gaussian Mixture“.
$\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$	Gaussian-Mixture-Zentren	Zentren der „Gaussian Mixture“.
$P_{m,m',m''}$	Gewinnwahrscheinlichkeit	Wahrscheinlichkeit, dass das Wortmodell m „besser als“ das Wortmodell m' abschneidet, wenn Schallmuster der Klasse m'' präsentiert werden.
$P^{\text{Rank-List, M}}_{m,r,m''}$	Rank-List-Positionswahrscheinlichkeit	Wahrscheinlichkeit, dass bei M Modellen das Wortmodell m an einer bestimmten Stelle r der „Rank List“ zu liegen kommt, wenn Schallmuster der Klasse m'' präsentiert werden.
$P^{\text{Rank-List, M}}_{r}$	Rank-List-Positionswahrscheinlichkeit	Diskrete Wahrscheinlichkeitsverteilung der Position r des richtigen Wortmodells in der „Rank List“.
R_M	Urnenmodell-Kugelanzahl	Anzahl der weißen Kugeln eines Urnenmodells.
F_M	Urnenmodell-Kugelanzahl	Anzahl der nicht-weißen Kugeln eines Urnenmodells.
ε_M	Korrekturfaktor	Korrekturfaktor zum Vergleich eines Urnenmodells mit einem bestimmten alternativen Modell.
$\mathbf{U}_M$	„Two-Winners Confusion Matrix“	Jede der M Zeilen der quadratischen Matrix ist umkehrbar eindeutig einer Schallmusterklasse zugeordnet. Jeder Spalte ist umkehrbar eindeutig einem Wortmodell zugeordnet. Jedes Element der Matrix gibt an, wie oft jedes einzelne Wortmodell an erster oder zweiter Stelle der „Rank List“ zu liegen kommt, wobei alle Schallmuster jeder einzelnen Schallmusterklasse präsentiert werden.
$\mathbf{V}_M$	Verwechslungsmatrix, „Confusion Matrix“	Jede der M Zeilen der quadratischen Matrix ist umkehrbar eindeutig einer Schallmusterklasse zugeordnet. Jeder Spalte ist umkehrbar eindeutig einem Wortmodell zugeordnet. Jedes Element der „Confusion Matrix“ gibt an, wie oft jedes einzelne Wortmodell an erster Stelle der „Rank List“ zu liegen kommt, wobei alle Schallmuster jeder einzelnen Schallmusterklasse präsentiert werden.
$\tilde{\alpha}$	Modifikationsfaktor	Bei der Verringerung des Wortschatzes um eine Schallmusterklasse fallen in der geschätzten Verwechslungsmatrix jene Schallmuster weg, die fälschlicherweise auf die weggelassene Schallmusterklasse abgebildet wurden. Der Modifikationsfaktor dient zur Beschreibung der Aufteilung auf die verbleibenden Klassen.
M_A	Diagonalelement	Diagonalelement der geschätzten Verwechslungsmatrix bei der Verringerung des Wortschatzes um eine Schallmusterklasse.
M_B	Nichtdiagonalelement	Nichtdiagonalelement der entstehenden geschätzten Verwechslungsmatrix bei der Verringerung des Wortschatzes um eine Schallmusterklasse.

Tabelle 1: Zusammenstellung verwendeter Formelzeichen.

1. Einführung

Einleitung
In diesem Abschnitt findet sich eine grundlegende Einführung in das Gebiet der automatischen Spracherkennung. Es wird auf den allgemeinen Systemaufbau von Sprachsteuerungen sowie auf die Vorverarbeitung von Sprachsignalen eingegangen.

Viterbi Dynamic Programming Speech Recognizers
Es werden in diesem Abschnitt Algorithmen für Sprachsteuerungen in einem ganzheitlichen Kontext dargestellt, in den sich auch übliche Hidden-Markov-Modell-Sprachsteuerungen einordnen lassen. Bezüglich älterer Verfahren, die keine Sprechtempokompensation beinhalten, wird lediglich auf Literatur verwiesen, um den Rahmen nicht zu sprengen, auch für Verfahren, die bei relativ geringen Anforderungen aufgrund der geringen Komplexität heute noch eingesetzt werden.

Überlegungen bezüglich Benutzerzufriedenheit
In diesem Abschnitt werden Fehlerraten definiert, die für die Zufriedenheit des Benutzers mit der Sprachsteuerung wesentlich sind. Es werden, unter bestimmten Annahmen, Formeln zur Abschätzung der Verwechslungsfehlerrate in Abhängigkeit der Vokabulargröße hergeleitet. Weiters werden die Auswirkungen verschiedener konstruktiver Maßnahmen auf die Fehlerraten diskutiert.

Neue Paradigmen der automatischen Spracherkennung
In diesem Abschnitt wird auf aktuelle Paradigmenwechsel auf dem Gebiet der automatischen Spracherkennung eingegangen.

1.1 Einleitung

1.1.1 Allgemeiner Systemaufbau einer Sprachsteuerung

Vom funktionellen Standpunkt aus betrachtet, lässt sich das Systemdesign einer typischen Sprachsteuerung nach Abbildung 1-1 aufteilen. Im Blockschaltbild links ist das Mikrofon angedeutet. Die Informationsverarbeitung besteht in der Vorverarbeitung des Schallsignals im Pattern-Builder-Funktionsblock, der Klassifikation des Schallmusters im Classifier-Funktionsblock sowie der Weiterverarbeitung zu einer eventuellen Systemreaktion im Dialog-State-Machine-Funktionsblock.

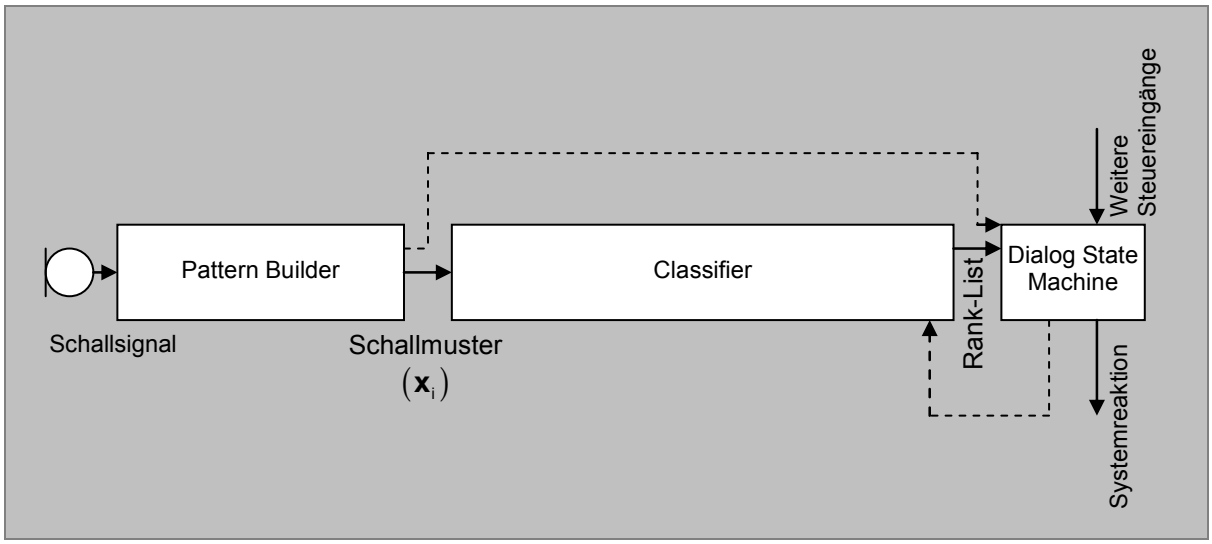

Abbildung 1-1: Allgemeines Systemdesign einer Sprachsteuerung.

Abbildung 1-2 auf Seite 17 zeigt eine übliche Detaillierung des Pattern-Builder-Funktionsblocks und des Classifier-Funktionsblocks, die auch speziell für die im Rahmen dieser Arbeit implementierte Sprachsteuerung zutrifft.

Dialog-State-Machine-Funktionsblock. Mittels des Dialog-State-Machine-Funktionsblocks wird aufgrund der „Rank List“ unter Berücksichtigung des aktuellen Dialogzustandes, also insbesondere unter Berücksichtigung des aktiven Vokabulars berechnet, ob eine Systemreaktion erfolgen soll, beziehungsweise welche Systemreaktion erfolgen soll. Der Dialog-State-Machine-Funktionsblock kann von weiteren Steuereingängen beeinflusst werden: Dabei kann es sich beispielsweise um einen Betreuertaster handeln, der dazu dient, den Dialog zu pausieren.

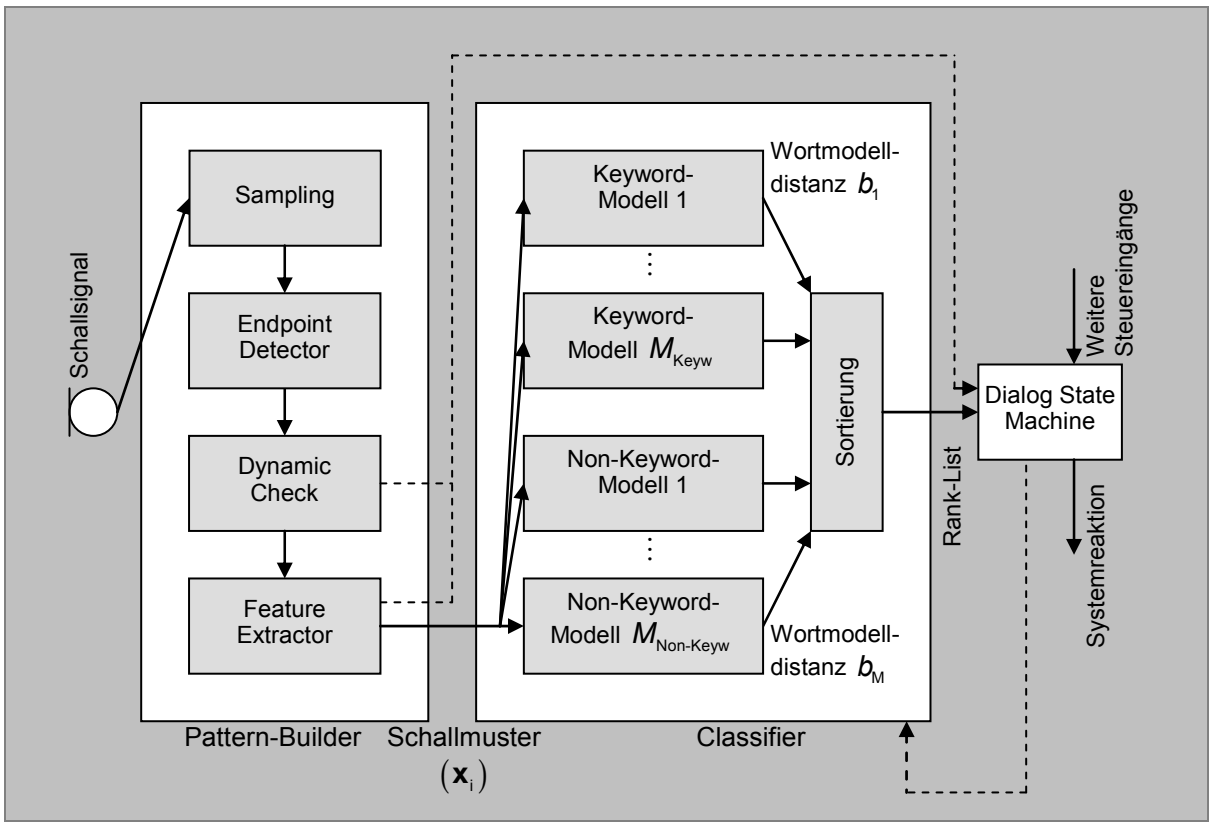

Abbildung 1-2: Übliche Detaillierung der Funktionsblöcke, die auch speziell für die im Rahmen dieser Arbeit implementierte Sprachsteuerung zutrifft.

Pattern-Builder-Funktionsblock. Im Pattern-Builder-Funktionsblock erfolgen zunächst die Abtastung und die Analog-Digitalwandlung. Der Endpoint-Detector-Funktionsblock hat die Aufgabe, die laufende Sample-Folge zu analysieren und einen Abschnitt aufeinander folgender Samples weiterzuleiten, wenn es sich bei deren Inhalt um einen Steuerbefehl handeln könnte. Die Entscheidung basiert dabei üblicherweise auf Lautstärkeanalysen und auf der Überprüfung von Bedingungen bezüglich der Länge der Pausen, das heißt der leisen Passagen vor beziehungsweise nach dem potenziellen Steuerbefehl. Es hat sich als günstig herausgestellt, zusätzlich eine Dynamiküberprüfung durchzuführen: Der Dynamic-Check-Funktionsblock verhindert, dass Passagen, die sich zu wenig von den Hintergrundgeräuschen abheben, zu einem Schallmuster weiterverarbeitet werden.[1] Im Feature-Extractor-Funktionsblock erfolgt im Allgemeinen eine Gruppierung der Samples in überlappende „Frames“. Charakteristische Merkmale des Sprachsignals werden quantitativ analysiert und für jeden „Frame“ zu einem Merkmalsvektor zusammengestellt. Die endliche Folge von Merkmalsvektoren des analysierten Abschnitts bildet das Schallmuster.

[1] Mit dieser Maßnahme wird sichergestellt, dass die Benutzer relativ laut über den Hintergrundgeräuschen sprechen, da sie wissen, dass sonst keine Systemreaktion erfolgt. Der Vorteil dabei besteht darin, dass die Fehlerraten von Sprachsteuerungen bei höherem Signal-Rauschabstand in der Regel niedriger sind.

Classifier-Funktionsblock. Das Schallmuster wird mehreren Wortmodellen übergeben. Dabei stellt es sich als günstig heraus, neben den Keyword-Modellen auch eine Reihe von Non-Keyword-Modellen zu verwenden, welche Umgebungsgeräusche und Wörter modellieren, die nicht Steuerbefehle sind. Siehe Abschnitt 1.3.3 auf Seite 76. Denn neben der Zielsetzung, die Verwechslungsfehlerrate zu optimieren, ist es ebenso wichtig, die Falschakzeptanzrate und die Falschrückweisungsrate nicht außer Acht zu lassen. Um eine Balance zwischen den beiden Letztgenannten zu finden, ist es notwendig, Mechanismen einzusetzen, die Schallereignisse, welche nicht zu den Steuerbefehlklassen gehören, zurückzuweisen. Allerdings sollte dies nicht dazu führen, dass Schallereignisse, die zu einer Steuerbefehlklasse gehören, fälschlicherweise zurückgewiesen werden. Eine Möglichkeit, einen derartigen Mechanismus zu realisieren besteht eben darin, wie in Abbildung 1-2 auf Seite 17 dargestellt, neben den Keyword-Modellen auch Non-Keyword-Modelle zu verwenden. Die Anzahl der Keyword-Modelle sollte in einem balancierten Verhältnis zur Anzahl der Non-Keyword-Modelle stehen. Beim implementierten System liefert jedes Wortmodell eine Wortmodelldistanz. Bei anderen Systemen liefern die Wortmodelle gegebenenfalls Wahrscheinlichkeiten, sogenannte „Likelihoods“. Jede „Likelihood“ gibt an, wie wahrscheinlich das beobachtete Schallereignis vom jeweiligen Wortmodell erzeugt worden wäre. Gegebenenfalls werden die „Likelihoods“ noch mit den A-priori-Wahrscheinlichkeiten der jeweiligen Wortmodelle multipliziert, um die Maximum-Liklihood-Schätzung zu einer Maximum-a-posteriori-Schätzung zu verfeinern (siehe auch Abschnitt 1.3.6 auf Seite 81) [Rabiner1989a]. Die Wortmodelldistanzen beziehungsweise Wahrscheinlichkeiten werden zu einer „Rank List“ sortiert, die die Reihenfolge der Wortmodelle angibt. Die Sortierung erfolgt gemäß steigenden Wortmodelldistanzen beziehungsweise fallenden Wahrscheinlichkeiten.

Direkte Wirkung des Pattern-Builder-Funktionsblocks auf den Dialog-State-Machine-Funktionsblock. Die Entscheidung, ob überhaupt eine Systemreaktion erfolgen soll, kann im allgemeinen Fall auch auf speziellen Analysen des Schallmusters basieren, welche nicht der Klassifikation dienen. Dies sei durch die obere gestrichelte Linie in Abbildung 1-1 auf Seite 16 und Abbildung 1-2 auf Seite 17 angedeutet. Beispielsweise kann die Entscheidung, ob es sich überhaupt um einen Steuerbefehl handelt, von Analysen bezüglich der Stimmhaftigkeit beeinflusst werden. Dazu käme beispielsweise eine Analyse infrage, die auf einem „Voice Activity Detector“ beruht [Beaufays2003]. Auch der Dynamic-Check-Funktionsblock kann eine Wirkung auf den Dialog-State-Machine-Funktionsblock ausüben: Beispielsweise wird beim implementierten System der Benutzer unter Umständen darauf hingewiesen, dass er doch im Vergleich zu den momentanen Umgebungsgeräuschen nach Möglichkeit lauter sprechen möge.

Rückwirkung des Dialog-State-Machine-Funktionsblocks auf den Classifier-Funktionsblock. Prinzipiell sind auch Rückwirkungen des Dialog-State-Machine-Funktionsblocks auf den Classifier-Funktionsblock möglich und sinnvoll. Diese Rückwirkungen sind ebenfalls durch eine gestrichelte Linie in Abbildung 1-1 auf Seite 16 und Abbildung 1-2 auf Seite 17 angedeutet. Der Classifier-Funktionsblock kann beispielsweise

Vorteile aus der Kenntnis des in der aktuellen Dialogsituation aktiven Vokabulars ziehen, als unter Umständen nicht alle Keyword-Modelle bewertet werden müssen,[2] was die Reaktionszeit des Gesamtsystems verkürzt.

1.1.2 Vergleich zum Systemaufbau eines Diktiersystems

Auf Diktiersysteme wird im Rahmen der vorliegenden Arbeit nur am Rande eingegangen, da der Rahmen sehr bald wesentlich gesprengt wäre. An dieser Stelle soll der Übersicht halber lediglich erwähnt sein, dass typische Diktiersysteme als Continuous-Speech-Systeme ausgeführt sind, das heißt, dass der Benutzer keine unnatürlich langen Sprechpausen zwischen den Wörtern einhalten muss [Tubach1991] [Cox2000]. Demnach entfällt im Allgemeinen der Endpoint-Detector-Funktionsblock und der Dynamic-Check-Funktionsblock nach Abbildung 1-3 auf Seite 20 und es wird die laufende Folge von Merkmalsvektoren direkt der weiteren Analyse zugeführt. Die Wortgrenzen ergeben sich dabei im Allgemeinen erst als Nebenprodukt der Analyse einer Wortfolge. Exemplarisch seien zwei interessante Literaturstellen angeführt [Jelinek1976] [Wachsmuth1997].

1.1.3 Extraktion von Merkmalen aus Sprachsignalen

Die Merkmalsextraktion, also die Gewinnung geeigneter Merkmale aus dem Sprachsignal, ist eines der wichtigsten Teilprobleme der automatischen Spracherkennung. Eine gute Übersicht der üblichen Merkmale findet sich in der Literatur [Picone1993]. Merkmale, die bezüglich der Klassifikation „relevant“ sind, erleichtern den Entwurf des nachfolgenden Classifier-Funktionsblocks nach Abbildung 1-1 auf Seite 16 [Ruske1988].

Im Folgenden wird zunächst auf die Einbettung des Feature-Extractor-Funktionsblocks in den Pattern-Builder-Funktionsblock einer Sprachsteuerung nach Abbildung 1-1 auf Seite 16 eingegangen und Bezug auf die spezielle Merkmalsextraktion der im Rahmen dieser Arbeit entwickelten Sprachsteuerung genommen. Darauf folgend werden weitere allgemein übliche Methoden der Merkmalsextraktion behandelt.

---

[2] Andererseits können jedoch inaktive Keyword-Modelle als zusätzliche „Täuschkörper“ verwendet werden, um bei Non-Keyword-Schallmustern die Falschakzeptanzrate zu senken. Siehe Abschnitt 1.3.5 auf Seite 80. Der Kompromiss zwischen Falschakzeptanzrate und Falschrückweisungsrate kann beispielsweise dialogsituationsabhängig gestaltet werden.

1.1.3.1 Merkmalsextraktion und Endpunktdetektion

Zu klassifizierende Schallmuster werden im Pattern-Builder-Funktionsblock aus dem Schallsignal erzeugt. Zum Zwecke der Mustererkennung sollen Signalmerkmale bestimmt werden, die für die Klassifizierung möglichst hilfreich sind. Man spricht dabei von „Feature Extraction“. Im Gegensatz zur Sprechererkennung wird bei der Spracherkennung darauf hingearbeitet, Merkmale zu bestimmen, welche unabhängig von Sprecher und Prosodie sind [Atal1974]. Dies gilt im besonderen Maße für jene Systeme, die als „sprecherunabhängig“ bezeichnet werden.

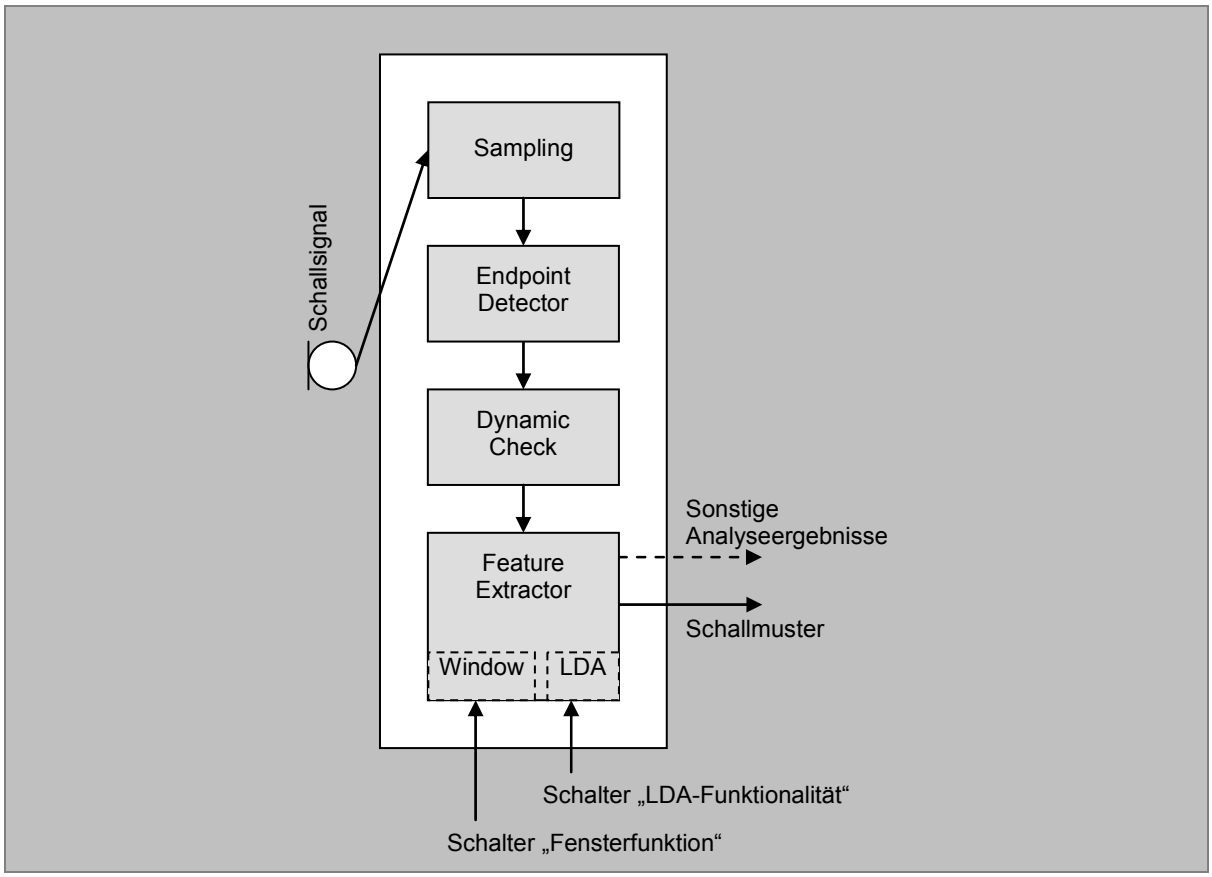

Abbildung 1-3: Detaillierung des Pattern-Builder-Funktionsblocks.

Abbildung 1-3 zeigt eine übliche Detaillierung des Pattern-Builder-Funktionsblocks. Speziell bezüglich der implementierten Variante eingezeichnet sind der Schalter „Fensterfunktion“, welcher die Umschaltung der Fensterfunktion der implementierten Kurzzeit-Fourier-Transformation ermöglicht und der Schalter „LDA-Funktionalität“, welcher die Aktivierung einer abschließenden LDA-Transformation ermöglicht (siehe Abschnitt 4.1.2 auf Seite 163). Bezüglich „Fensterfunktion“: Siehe auch Anhang II auf Seite 196. Bezüglich „Linearer Diskriminanzanalyse“: Siehe auch Anhang III auf Seite 201.

Sampling-Funktionsblock. Die Abtastung erfolgt beispielsweise mit 16 kHz, wobei sich bei der späteren Gruppierung in 50%-überlappende Fenster zu je 512 Samples für jedes Fenster eine übliche Zeitspanne von 32 ms ergibt, die es repräsentiert. Während dieser Zeitspanne kann das Sprachsignal gerade noch als stationär angesehen werden, was für eine Reihe weitergehender Analysen von Vorteil ist. Kleinere Fensterdauern sind durchaus üblich, allerdings müssen dann bei einer eventuellen diskreten Kurzzeit-Fourier-Transformation Abstriche bezüglich der Frequenzauflösung in Kauf genommen werden, da sich diese als Kehrwert der Fensterdauer ergibt.

Endpoint-Detector-Funktionsblock. Die Aufgabe des Endpoint-Detector-Funktionsblocks besteht darin, aus der laufenden Sample-Folge eine endliche Folge, die einen Steuerbefehl enthalten könnte, auszuschneiden und zur Analyse weiterzuleiten [Li2001] [Gu2003]. Fehlfunktionen dieses Blocks wirken sich direkt auf die Falschrückweisungsrate aus. Die Falschakzeptanzrate kann ebenfalls negativ beeinflusst werden, wenn gleichzeitig andere Mechanismen zur Falschakzeptanzvermeidung versagen.

Bestandteil des Endpoint-Detector-Funktionsblocks ist ein sogenannter „Voice Activity Detector“ [Kwon2003] [Wei2003] [Ramirez2004] [Shafran2003]. Derartige Detektoren werden auch im Zusammenhang mit Sprachsignalcodierung verwendet, beispielsweise beim Codec nach ITU-T G.729, wobei die Detektoren zur Datenreduktion dienen, indem Gesprächspausen nicht vollständig übertragen werden. Fehlfunktionen bei einem derartigen Einsatz sind jedoch zweifelsohne wesentlich unproblematischer, als beim Einsatz im Endpoint-Detector-Funktionsblock, da der „Voice Activity Detector“ dabei so eingestellt werden kann, dass er eher eine Pause als Sprache klassifiziert als umgekehrt, was schlimmstenfalls zu einer größeren zu übertragenden Datenmenge führt. Fehlfunktionen im Endpoint-Detector-Funktionsblock wirken sich jedoch im Allgemeinen direkt und ungünstig auf die Fehlerraten der Sprachsteuerung aus.

Die Funktionsweise des Endpoint-Detector-Funktionsblocks stellt sich beispielsweise beim im Rahmen dieser Arbeit implementierten System folgendermaßen dar: Die Samples werden, wie auch später im Feature-Extractor-Funktionsblock, in überlappende Fenster gruppiert. Ist ein Fenster komplett, so wird der Pegel der Signalleistung in einem bestimmten Frequenzband für dieses Fenster berechnet und zwei verschiedenen digitalen Filtern zugeführt. Eines der beiden Filter ist so konzipiert, dass es seine Ausgangsgröße relativ rasch der Eingangsgröße nachregelt, wo hingegen das andere so konzipiert ist, dass es seine Ausgangsgröße langsam nachregelt. Bei den Eingangsgrößen und Ausgangsgrößen der Filter handelt es sich um Pegelmaße. Wenn der Ausgangspegel des schnell nachregelnden Filters den Ausgangspegel des langsam nachregelnden Filters um einen gewissen Sockelbetrag, beispielsweise um 6 dB, überschreitet, so wird das aktuelle Fenster als „laut“ markiert. Andernfalls wird es als „leise“ markiert. Blickt man nun vom aktuellen Zeitpunkt in die Vergangenheit zurück, so müssen eine Reihe von Bedingungen erfüllt sein, damit ein Block von Fenstern zu einem Schallmuster weiterverarbeitet wird: Eine bestimmte Anzahl an „leisen Frames“ gefolgt von „lauten Frames“, deren Anzahl in einem Intervall liegen muss, gefolgt von einer bestimmten Anzahl an „leisen Frames“

müssen beispielsweise aufgetreten sein. Ein gewisser Prozentsatz an Ausnahmen sollte jedoch zulässig sein. Sind alle Bedingungen erfüllt, so werden die Samples des für einen Steuerbefehl infrage kommenden Zeitintervalls zur weiteren Verarbeitung weitergeleitet.

Dynamic-Check-Funktionsblock. Es wird sodann eine Dynamiküberprüfung durchgeführt, das heißt, der Benutzer muss mit einer gewissen Lautstärke über den Hintergrundgeräuschen reden, damit überhaupt ein Schallmuster erzeugt wird.

Feature-Extractor-Funktionsblock. Es erfolgt üblicherweise eine Gruppierung der Samples in überlappende „Frames“. Sodann werden die Merkmalsvektoren dieser „Frames“ berechnet. Dazu eignet sich beispielsweise eine Filterbank, deren Filter entlang der Bark-Skala äquidistante Frequenzbereiche abdecken. Der Feature-Extractor-Funktionsblock enthält noch eine Reihe weiterer Verfahren, wie beispielsweise Normierungen. Für die Fehlerraten des Systems stellt der Feature-Extractor-Funktionsblock eine „sehr empfindliche Schraube“ dar.

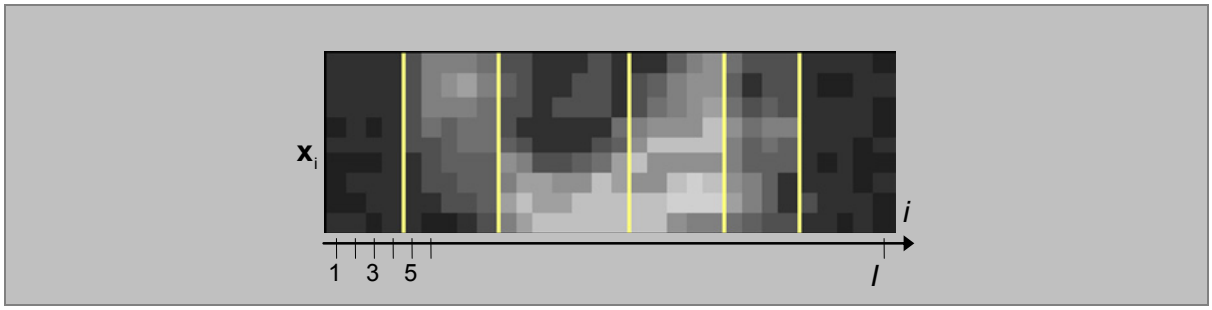

Abbildung 1-4: Darstellung eines Schallmusters des Wortes „Sohle“ am Ausgang des Endpoint-Detector-Funktionsblocks. Für jeden Zeitpunkt ist ein Merkmalsvektor mittels Graustufen-Quantisierung dargestellt. Eingezeichnet sind weiters Segmentgrenzen als Ergebnis einer Sprechtempokompensation.

Ein Schallmuster $\mathbf{X} = (\mathbf{x}_i)$ des Wortes „Sohle“ ist in Abbildung 1-4 beispielhaft dargestellt. Der Index i der Merkmalsvektoren kennzeichnet äquidistante Zeitpunkte.

Lineare Diskriminanzanalyse. Im Rahmen der vorliegenden Arbeit werden verschiedene Varianten des Pattern-Builder-Funktionsblocks implementiert, die sich aber lediglich durch unterschiedliche Schalterstellungen in Abbildung 1-3 auf Seite 20 unterscheiden. Mittels eines „Schalters“ können die Merkmalsvektoren einer abschließenden Multiplikation mit einer LDA-Matrix nach Gleichung 1 unterzogen werden. Es handelt sich somit bei der LDA-Transformation somit um eine lineare Transformation.

$$(\mathbf{x}_i') = (\mathbf{A}^T \cdot \mathbf{x}_i) \qquad \textbf{Gleichung 1}$$

Die optimale Transformationsmatrix wird beim im Rahmen dieser Arbeit implementierten Algorithmus mittels „Linear Discriminant Analysis“ nach Fisher [Dahmen2001] bestimmt. Die „Linear Discriminant Analysis“ erfolgt dabei über die Berechnung der „Between-Class Scatter Matrix“ und der „Within-Class Scatter Matrix“. Siehe dazu Seite 36 im Zusammenhang mit Anhang III auf Seite 201.

Gute Einführungen zur Anwendung der LDA-Transformation finden sich an verschiedenen Stellen in der Literatur [Martinez2001a] [Dahmen2001]. Im Rahmen der LDA-Transformation findet üblicherweise eine deutliche Dimensionalitätsreduktion der Merkmalsvektoren statt. Über die Vorteile der Anwendung der LDA-Transformation zum Zwecke der sprecherunabhängigen Spracherkennung mit großem Vokabular wird berichtet [Aubert1993] [HaebUmbach1992] [HaebUmbach1993]. Eine interessante Idee stellt die Kombination der LDA-Transformation mit dem sogenannten „Kernel Trick“ zur nichtlinearen Trennung der Klassen dar [Schölkopf1999] [Mika1999].

1.1.3.2 Weitere übliche Methoden der Merkmalsextraktion

Hauptkomponentenanalyse. Die Grundidee der sogenannten „PCA-Transformation“ beziehungsweise der „Principle Component Analysis“ besteht darin, orthogonale Hauptachsenvektoren zu berechnen, aus deren Linearkombination die Merkmalsvektoren mit minimalem quadratischen Fehler rekonstruiert werden können, wobei die Anzahl der berechneten Hauptachsenvektoren kleiner ist, als die Dimension der Merkmalsvektoren [Ruske1988] [Schölkopf1999]. Denn in der Praxis wird mit der Anwendung der PCA-Transformation im Allgemeinen auch der Zweck verfolgt, die Dimensionalität der Merkmalsvektoren zu reduzieren, wobei bei dieser Projektion der Punktwolke der Merkmalsvektoren wesentliche Nachbarschaftsbeziehungen erhalten bleiben.

Kohonen Layer. Eine weitere Möglichkeit zur Dimensionalitätsreduktion der Merkmalsvektoren im Feature-Extractor-Funktionsblock stellt der neurophysiologisch motivierte sogenannte „Kohonen Layer“ dar [Kohonen1990]. Es handelt sich dabei um eine Schicht aus Neuronen, wobei von jedem Neuron zu jedem anderen Neuron der Schicht eine Entfernung definiert ist. Zum Zwecke des Trainings werden die Inputvektoren wiederholt präsentiert. Jeweils wird jenes Neuron bestimmt, welches den Inputvektor am besten „repräsentiert“. Dieses wird weiter in Richtung des Inputvektors adaptiert und entsprechend der jeweiligen Entfernung werden auch die anderen Neuronen adaptiert, die näheren stärker, die weiter entfernten schwächer. Die Ausgänge der Neuronen werden bei der Verwendung des „Kohonen Layer“ zum Outputvektor zusammengestellt. Die dadurch realisierte Abbildung hat die Eigenschaft, Nachbarschaftsverhältnisse der Inputvektoren möglichst umfassend auf die Outputvektoren zu übertragen.

Verbindung verschiedener Merkmalsvektoren. Oftmals wird auch versucht, verschiedenartige Merkmalsvektoren zu verbinden, um die Robustheit bei Hintergrundgeräuschen zu steigern. Normalerweise werden die Vektoren gewichtet und aneinandergehängt [Huang2003]. Merkmalsselektionsstrategien anhand der Trainingsdatenbank werden ebenfalls angewendet. LDA-Transformation und PCA-Transformation bieten sich ebenfalls zur Verbindung von Merkmalsvektoren an.

Merkmalsselektion. Die Merkmalsselektion stellt eine eigenständige Aufgabe dar. Hierunter versteht man, dass aus den extrahierten Merkmalen eine Teilmenge ausgewählt und zu einem neuen Merkmalsvektor zusammengestellt wird [Ruske1988]. Es stellt sich dabei die Aufgabe, die relevantesten Merkmale zu bestimmen. Beispielsweise können mehrere genormte Merkmalsvektoren zu einem neuen Merkmalsvektor verbunden werden. Zur Lösung der Aufgabe kommen Suchverfahren zum Einsatz, wobei das Optimalitätsmaß eine entscheidende Rolle spielt. Als Optimalitätsmaß bietet sich beispielsweise der Verwechslungsfehler nach Abschnitt 1.4.2.3 an.

Im Folgenden wird auf die verschiedenen Arten von Merkmalsvektoren eingegangen. Dabei kann jedoch keineswegs der Anspruch auf Vollständigkeit erhoben werden, denn die Fülle an Varianten und Verfahren, über die in der Literatur berichtet wird, ist enorm.

Analyse der Formanten. Resonanzphänomene des Vokaltrakts im Rahmen der Artikulation werden bei der Merkmalsextraktion mittels Kurzzeit-Fourier-Transformation als sogenannte „Formanten“ in den Spektren sichtbar. Die Resonanzfrequenzen treten als Maxima der spektralen Hüllkurven auf und ändern sich im Allgemeinen während einer Äußerung in charakteristischer Art und Weise. Daher würden sie sie sich grundsätzlich gut als Merkmale zur Klassifikation eignen. Leider ist die robuste Extraktion solcher Merkmale mit verschiedenen Schwierigkeiten behaftet. Eine gute Übersicht über die verwendeten Verfahren findet sich in der Literatur [Schmid1990].

Verwendung eines Voice-Activity-Merkmals. Ein „Voice Activity Detector“ soll bei menschlicher Sprache ansprechen, nicht jedoch bei Pausen zwischen den Wörtern oder Hintergrundgeräuschen. Die Entscheidung wird beispielsweise mittels eines Neuronalen Netzes aus verschiedenen Merkmalen wie etwa Stimmhaftigkeit, Spektralschieflage und Lautstärke berechnet. Einige Verfahren sind im Zusammenhang mit Verfahren zur Sprachsignalcodierung genormt. Von der Verringerung der Verwechslungsfehlerrate beim Hinzufügen eines Voice-Activity-Merkmals zum Merkmalsvektor wird berichtet [Beaufays2003].

Verwendung von Phaseninformation. Bei fast allen Verfahren der Merkmalsextraktion, die auf der Kurzzeit-Fourier-Transformation beruhen, wird die Phaseninformation verworfen und lediglich die Amplitudeninformation weiter verarbeitet. Gelegentlich wird jedoch die Phaseninformation verwendet, um Phasenmerkmale, sogenannte „Phase Features“, zu berechnen, welche mit dem MFCC-Merkmalsvektor (siehe Seite 26) kombiniert zu einer Erhöhung der Klassifikationsleistung führen können [Schlüter2001].

„Feature Normalization“. Die akustischen Gegebenheiten ändern sich im Allgemeinen von Aufnahme zu Aufnahme bezüglich der Hintergrundgeräusche, der Übertragungsfunktionen sowie der Artikulation des Sprechers. Zur Erhöhung der Robustheit kommen eine Reihe von Merkmalsvektor-Transformationen infrage, wie beispielsweise: „Mean Variance Normalization“, „Histogram Normalization“, „Feature Space Rotation“, „Vocal Tract Length Normalization“ [Dharanipragada2002]. Eine interessante Übersicht findet sich in der Literatur [Molau2003]. Die Erkennung der Sprache von Kindern ist eine besondere Herausforderung. Über den Einsatz von „Vocal Tract Length Normalization“ in diesem Zusammenhang berichten verschiedene Autoren [Narayanan2002] [Giuliani2003].

Fast-Hartley-Transformation. Viele Berechnungsverfahren für Merkmalsvektoren beruhen auf der Kurzzeit-Fourier-Transformation, wobei üblicherweise die Fast-Fourier-Transformation angewendet wird. In vielen Fällen wird die Phaseninformation verworfen und durch Quadrieren der Amplitudeninformation das Leistungsdichtespektrum berechnet. Es existiert ein direkter Weg über die Fast-Hartley-Transformation [ONeill1988], welcher weniger rechenintensiv ist.

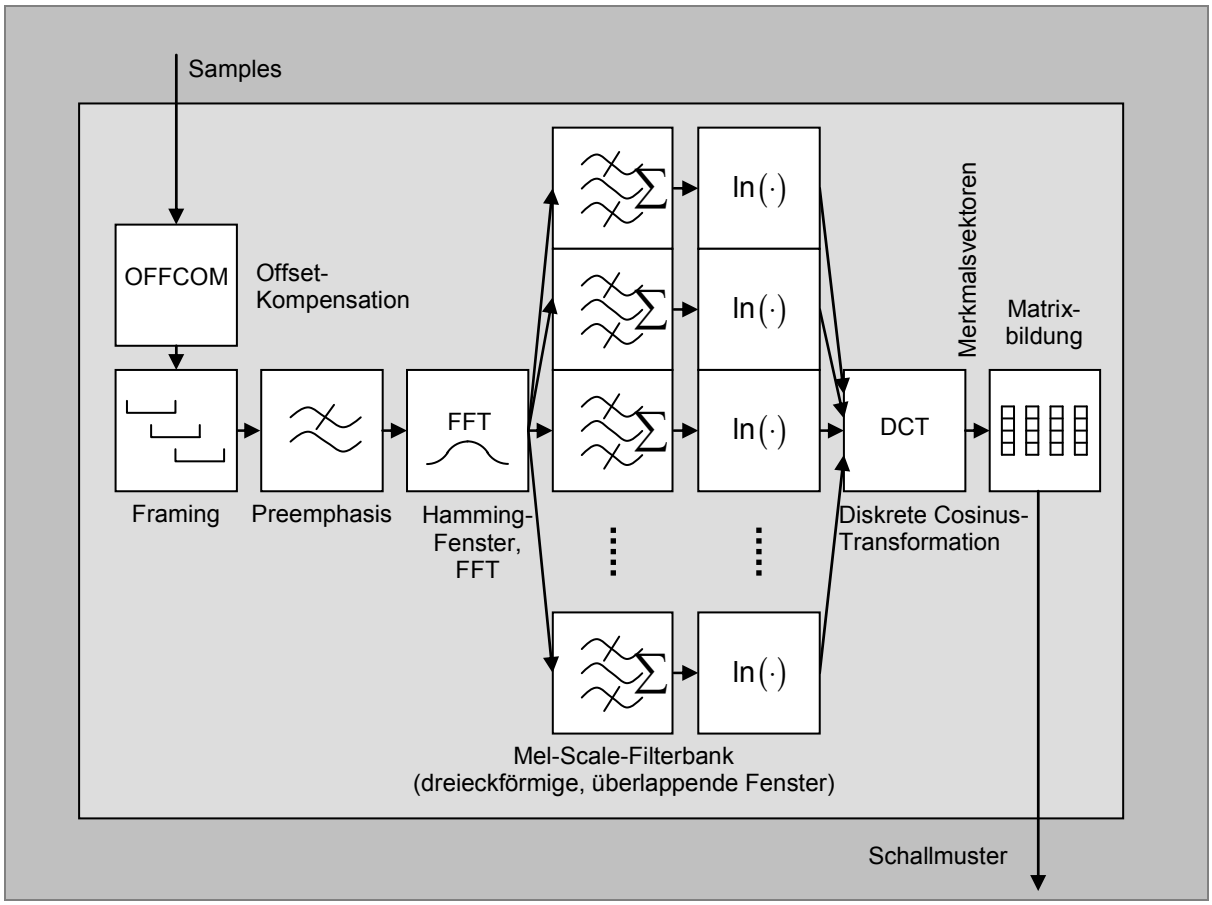

Abbildung 1-5: Detaillierte Darstellung des Feature-Extractor-Funktionsblocks bei der Berechnung von „Mel Scale Frequency Cepstral Coefficients“. Die Berechnung ist nach ETSI-ES-201-108 standardisiert.

MFCC-Merkmalsvektoren. Bei den sogenannten „Mel Scale Frequency Cepstral Coefficients“ handelt es sich um Filterbank-Merkmalsvektoren. MFCC-Merkmalsvektoren sind auf dem Gebiet der Spracherkennung weit verbreitet und die Berechnung ist durch eine Norm des „European Telecommunication Standardization Institute“ (ETSI) standardisiert. Abbildung 1-5 zeigt das prinzipielle Blockschaltbild zur Berechnung der MFCC-Merkmalsvektoren im Feature-Extractor-Funktionsblock nach ETSI-ES-201-108. Dabei wird die Filterbank nach der Kurzzeit-Fourier-Transformation im Frequenzbereich realisiert. Mittels Logarithmierung und diskreter Kosinustransformation werden die endgültigen Merkmalsvektoren berechnet. Vielfach werden auch andere Berechnungsverfahren abseits der exakten Implementierung von ETSI-ES-201-108 als „Mel Scale Frequency Cepstral Coefficients“ bezeichnet. Alternative Berechnungsverfahren sind häufig anzutreffen [Zheng2001]. MFCC-Merkmalsvektoren werden zur Klassifikation von Schallereignissen afrikanischer Elefanten erfolgreich eingesetzt [Clemins2003]. Es ist möglich, aus den MFCC-Merkmalsvektoren das ursprüngliche Sprachsignal zu rekonstruieren, wenn als zusätzliche Information der zeitliche Verlauf der Stimmhaftigkeit beziehungsweise der Stimmbandgrundfrequenz vorliegt [Shao2003].

HFCC-Merkmalsvektoren. Die Filterkurven der Filterbankfilter der „Mel Scale Frequency Cepstral Coefficients“ sind vielfach Gegenstand von Untersuchungen zum Zwecke der Verbesserung. Beim Verfahren der Berechnung der „Human Factor Cepstral Coefficients“ handelt es sich um eine Modifikation des Verfahrens zur Berechnung der MFCC-Merkmalsvektoren, wobei die Bandbreiten der Filterbankfilter in anderer Weise gewählt werden. Durch diese Maßnahme können unter Umständen verbesserte Klassifikationsergebnisse im Vergleich mit den MFCC-Merkmalsvektoren erzielt werden [Skowronski2003].

AFCC-Merkmalsvektoren. Beim Verfahren der Berechnung der „Auditory-based Feature Cepstral Coefficients“ handelt es sich ebenfalls um eine Modifikation des Verfahrens zur Berechnung der MFCC-Merkmalsvektoren, wobei die Filterkurven der Filterbankfilter während einer Trainingsphase bestimmt werden. Im Vergleich mit den MFCC-Merkmalsvektoren können dadurch unter Umständen bessere Klassifikationsergebnisse erzielt werden [Mak2004].

PEMO-Merkmalsvektoren. Ein vielversprechendes Paradigma, nämlich die Modellierung des menschlichen Gehörs, wird beispielsweise mittels der sogenannten „PEMO-Merkmalsvektoren“ des Oldenburger Perzeptionsmodells [Kleinschmidt2000] umgesetzt. Dabei handelt es sich um eine Filterbankberechnung im Zeitbereich. Für nähere Details sei auf den Abschnitt 1.4.1.2 auf Seite 90 verwiesen.

LPC-Merkmalsvektoren. Gelegentlich stößt man auf die Verwendung von „Linear Predictive Coding“ zur Merkmalsextraktion. Das Verfahren erhält seine Motivation aus dem Quelle-Filter-Modell der menschlichen Spracherzeugung [Eppinger1993]. Dabei handelt es sich um einen relativ früh bekannten Ansatz aus den Sechzigerjahren des vorigen Jahrhunderts [Atal1974]. Derzeit geht der Trend eher in Richtung der Modellierung der Eigenschaften des Gehörs. Siehe dazu Abschnitt 1.4.1. auf Seite 84. Die LPC-Merk-

malsvektoren haben dennoch bis heute eine gewisse Bedeutung behalten [Juang1992a] [Grassi1998].

PLP-Merkmalsvektoren. Dabei handelt es sich um eine Weiterentwicklung der LPC-Merkmalsvektoren, wobei psychoakustische Eigenschaften des Gehörs berücksichtigt werden. Bei der „Perceptual-Linear-Prediction-Merkmalsextraktion“ werden als psychoakustisches Modell die nichtlinare Bark-Skala sowie eine dem menschlichen Gehör nachempfundene Filterkurve und eine Amplitudentransformation auf empfundene Lautstärken verwendet [Hermansky1990]. Vom Einsatz der PLP-Merkmalsvektoren wird auch in neueren Publikationen berichtet [Jaitly2012].

PARCOR-Merkmalsvektoren. Dabei handelt es sich ebenfalls um Merkmalsvektoren, die einen Bezug zur menschlichen Spracherzeugung herstellen. Diese, mittels „Partial Correlation“ erzeugten Merkmalsvektoren sind eng mit den LPC-Merkmalsvektoren verwandt [Eppinger1993].

ZCPA-Merkmalsvektoren. Bei der Zero-Crossings-Peak-Amplitudes-Merkmalsextraktion wird das Signal zunächst mittels einer Filterbank in Teilbandsignale zerlegt. Die Teilbandsignale werden bezüglich steigender Nulldurchgänge untersucht. Jeweils zwischen zwei steigenden Nulldurchgängen wird der maximale Signalwert bestimmt. Weiters werden die Zeitdifferenzen zwischen den Zeitpunkten der steigenden Nulldurchgänge bestimmt. Sodann wird ein Histogramm der invertierten Zeitdifferenzen aller Teilbandsignale kumuliert. Allerdings werden die Klassenzähler des Histogramms jeweils nicht um eins, sondern um den Logarithmus des jeweiligen maximalen Signalwertes, erhöht. Schließlich wird zu Dekorrelationszwecken eine diskrete Cosinus-Transformation durchgeführt [Lee1997] [Gajic2003].

EMG-Merkmalsvektoren. Dabei handelt es sich derzeit eher um eine Kuriosität. Elektromyogrammsignale werden am Kehlkopf und unterhalb des Kiefers aufgenommen und mittels Wavelet-Transformation aufbereitet [Jorgensen2003]. Dabei ergeben sich neue Möglichkeiten der „Silent Speech Recognition“. Ein derartiger Ansatz ist eventuell auch geeignet, Spracherkennung in sehr lauter Umgebung, beispielsweise in einem Flugzeugcockpit, zu ermöglichen.

TDNN-Merkmalsvektoren. Ein sehr pragmatischer Ansatz besteht darin, Verfahren, die ursprünglich zur Klassifikation entwickelt wurden, als Phonemklassifikatoren einzusetzen und vom Classifier-Funktionsblock in den Feature-Extractor-Funktionsblock in Abbildung 1-2 auf Seite 17 zu verschieben. Die Phonemmodelldistanzen werden dann als Komponenten des Merkmalsvektors verwendet. Man spricht dabei von „Phoneme Spotting“. Infrage kommen dafür beispielsweise sogenannte „Time Delay Neural Networks“ [Waibel1989] [Bottou1989] [Miyatake1990]. Eine gute Einführung bezüglich „Time Delay Neural Networks“ findet sich in [Prante2001].

SSC-Merkmalsvektoren. Bei der Berechnung von Merkmalsvektoren nach dem Verfahren „Spectral Subband Centroids“ werden zunächst mittels Kurzzeit-Fourier-Transformation und Quadrierung Leistungsdichtespektren berechnet. Aus jedem

Leistungsdichtespektrum werden durch Multiplikation mit verschiedenen Filterkurven mehrere Filterbankkanal-Leistungsdichtespektren erzeugt. Für jedes Filterbankkanal-Leistungsdichtespektrum ergibt sich ein Merkmal als Quotient des Moments erster Ordnung und des Moments nullter Ordnung [Chen2004].

DPF-Merkmalsvektoren. Bei der Berechnung dieser Merkmalsvektoren werden zunächst sogenannte „Local Features“ als Zwischenstufe extrahiert. An dieser Stelle kommen verschiedene Arten von Merkmalsvektoren, beispielsweise auch MFCC-Merkmalsvektoren, infrage. Sodann werden Neuronale Netze verwendet, um die Merkmalsvektoren auf sogenannte „Distinctive Phonetic Features“ abzubilden. Diese repräsentieren das Auftreten bestimmter phonetischer Merkmale. Zum Zwecke japanischer Spracherkennung werden beispielsweise die folgenden elf phonetischen Merkmale vom Neuronalen Netz extrahiert: „Vocalic-Merkmal“, „Consonantal-Merkmal“, „High-Merkmal“, „Back-Merkmal“, „Low-Merkmal“, „Anterior-Merkmal“, „Coronal-Merkmal“, „Obstruent-Merkmal“, „Voiced-Merkmal“, „Continuant-Merkmal“, „Nasal-Merkmal“ [Fukuda2003].

NPC-Merkmalsvektoren. Beim Verfahren „Neural Predictive Coding“ werden Neuronale Netze zur nichtlinearen Vorhersage im Zeitbereich trainiert. Eine Möglichkeit besteht darin, im Sinne eines „Phoneme Spotting“, Phonemmodelldistanzen als Komponenten des Merkmalsvektors zu verwenden, oder eine symbolische Folge von Phonemen zu erzeugen, wobei ein mehrstufiges Verfahren verwendet werden kann: Beispielsweise sei ein Neuronales Netz auf die Klassifikation innerhalb der Lautgruppe [p] [t] [k] und ein anderes auf die Klassifikation innerhalb der Lautgruppe [b] [d] [g] spezialisiert. Es erfolgt zunächst die Klassifikation der Lautgruppe und dann die Klassifikation innerhalb der Lautgruppe [Chetouani2002] [Chetouani2002a] [Chetouani2003]. Die erhaltene symbolische Phonemfolge wird nun als Merkmalsvektor codiert.

1.2 Viterbi Dynamic Programming Speech Recognizers

In diesem Abschnitt wird die Klasse der „Viterbi Dynamic Programming Speech Recognizer“ beschrieben, zu der auch die Varianten des „Hidden Markov Model Speech Recognizer“ gehören. Der Abschnitt verfolgt die Zielsetzung, heutige moderne Verfahren der Spracherkennung und klassische Verfahren der Spracherkennung in einem übergeordneten Kontext darzustellen.

Welche Spracherkenner gehören dazu? Als „Viterbi Dynamic Programming Speech Recognizer“ zählen im Rahmen dieser Arbeit alle Spracherkenner, deren Wortmodelle durch das Blockschaltbild nach Abbildung 1-7 auf Seite 31 beschrieben werden können. Im Rahmen dieser Arbeit werden eine ganze Reihe bekannter Verfahren dieser Klasse behandelt: Beispielsweise der „Hidden Control Neural Network Speech Recognizer“, übliche Varianten des „Hidden Markov Model Speech Recognizer“, insbesondere auch die neu entwickelten „Hybrid-Systeme“ [Do2011] [Dahl2012] [Jaitly2012], der „Predictive Neural Network Speech Recognizer“ sowie verschiedene Varianten des patentierten Klangfolgenerkenners [Hickersberger2004]. Letzterer wird im Rahmen dieser Arbeit als „Centroid Sequence Speech Recognizer“ bezeichnet. Für alle Spracherkenner aus der Klasse der „Viterbi Dynamic Programming Speech Recognizer“ eignet sich der in Abschnitt 1.2.2 beschriebene Run-Length-Limited-Sprechtempokompensator und stellt eine Verbesserung gegenüber dem, in Abschnitt 1.2.1 beschriebenen, üblichen Viterbi-Dynamic-Programming-Sprechtempokompensator dar.

Welche Spracherkenner gehören nicht dazu? Es gehören insbesondere keine Verfahren dazu, welche ohne Sprechtempokompensation arbeiten. Derartige Verfahren waren vor dem Siegeszug der „Hidden Markov Model Speech Recognizer“ üblich. Erwähnt seien beispielsweise der klassische „Neural Network Classifier“ und der klassische „Nearest Neighbor Pattern Matcher“. Der klassische „Dynamic Time Warping Pattern Matcher“ wird im Rahmen dieser Arbeit nicht beschrieben, obwohl bei diesem populären Verfahren ein ähnlicher Dynamic-Programming-Algorithmus zur Sprechtempokompensation angewendet wird. Das dem „Dynamic Time Warping Pattern Matcher“ [Yaniv2003] zugrunde liegende Prinzip „Pattern Matching“ besteht allerdings im Vergleich zweier Schallmuster untereinander und nicht – wie bei den beschriebenen Verfahren – im Vergleich eines Schallmusters mit einem Wortmodell, welches die Eigenschaften einer Schallmusterklasse repräsentiert. Es soll nicht der Eindruck erweckt werden, dass die nicht behandelten Verfahren nicht relevant wären oder heutzutage nicht angewendet würden. Im Gegenteil, es sind für Sprachsteuerungen derzeit verschiedenste „Dynamic Time Warping Pattern Matcher“ [Gu2005] [Yaniv2003] sowie „Neural Network Classifier“ [Tschirk1999] im Einsatz. Es wird auch in diesen Richtungen weiterentwickelt [Chang1992] [Yaniv2003]. „Dynamic Time Warping Pattern Matcher“ werden gerne als Benchmark-Implementierungen verwendet, an denen komplexere Verfahren gemessen

werden [Bottou1989]. Auch Varianten des Pattern-Builder-Funktionsblocks nach Abbildung 1-1 auf Seite 16 werden damit evaluiert [Barbier1991].

„Hidden Markov Model Speech Recognizer". Als Definition des Begriffs „Hidden Markov Model Speech Recognizer" gelte die bekannte Literaturstelle [Rabiner1989a]. Der Viterbi-Algorithmus basiert auf dem Viterbi-Dynamic-Programming-Algorithmus und erweitert ihn um eine wahrscheinlichkeitstheoretische Interpretation der Daten [Forney1973]. Spracherkenner, die den Viterbi-Dynamic-Programming-Sprechtempokompensator verwenden, können zu Recht als „Hidden Markov Model Speech Recognizer" bezeichnet werden, wenn die wahrscheinlichkeitstheoretische Interpretation der Daten überzeugend argumentiert werden kann.

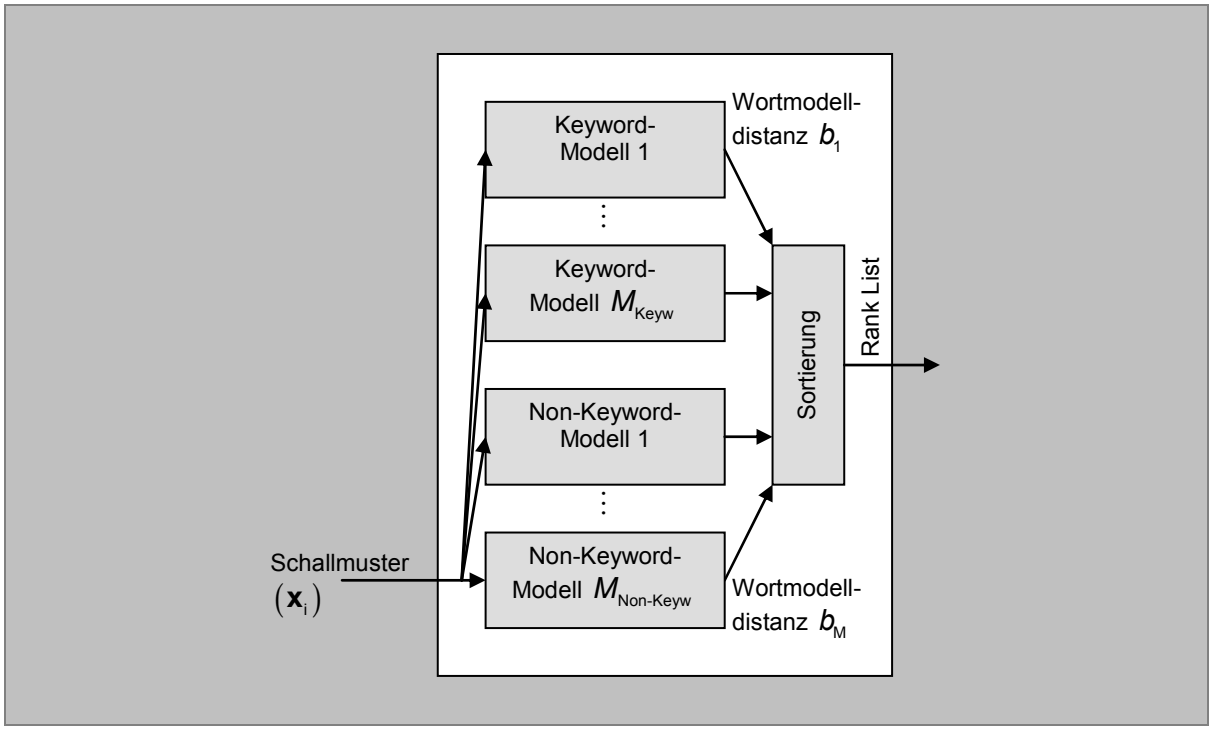

Abbildung 1-6: Der Classifier-Funktionsblock im Detail: Aus dem Schallmuster wird die „Rank List" berechnet. Im Allgemeinen besteht die Möglichkeit, dass Non-Keyword-Modelle zur Verringerung der Anzahl der Falschakzeptanzfehler zum Einsatz kommen.

Classifier-Funktionsblock. Die Wortmodelldistanz b_m ist eine reelle Zahl, welche ein Maß für die Übereinstimmung des Wortmodells m mit dem beobachteten Schallmuster $(\mathbf{x}_i)$ darstellt[3]. Die „Rank List" wird verwendet, um zu bestimmen, ob eine Systemreaktion erfolgen soll und gegebenenfalls welche Systemreaktion erfolgen soll.

[3] Bei der Berechnung der Wortmodelldistanzen bietet sich grundsätzlich Parallelverarbeitung an.

Wird nur das Wortmodell an der ersten Position der „Rank List“ berücksichtigt, so handelt es sich um eine einfache Minimum-Distance-Auswahl[4].

Wortschatzerweiterung. Der Wortschatz wird bei der beschriebenen Klasse der „Viterbi Dynamic Programming Speech Recognizer“ durch mehrere Wortmodelle im Classifier-Funktionsblock nach Abbildung 1-6 auf Seite 30 gebildet. Wenn kein diskriminatives Training gemäß Abschnitt 1.4.2.1 auf Seite 97 durchgeführt wird und auch keine LDA-Transformation gemäß Seite 36 durchgeführt wird, so können neue Wortmodelle einfach hinzugefügt werden, ohne dass die bestehenden Wortmodelle neu trainiert werden müssen.

Wortmodellfunktionsblöcke. Abbildung 1-7 auf Seite 31 zeigt die Funktionsweise eines Viterbi-Dynamic-Programming-Wortmodells. Es handelt sich um die detaillierte Darstellung eines der Wortmodelle aus Abbildung 1-6. Das Schallmuster wird zunächst in einem Schallmusterspeicher abgelegt. Mit dem Zeitpunkteingang kann jeder einzelne Merkmalsvektor, samt einer gewissen Anzahl an zeitlichen Nachfolgervektoren und Vorgängervektoren, wieder ausgelesen werden.

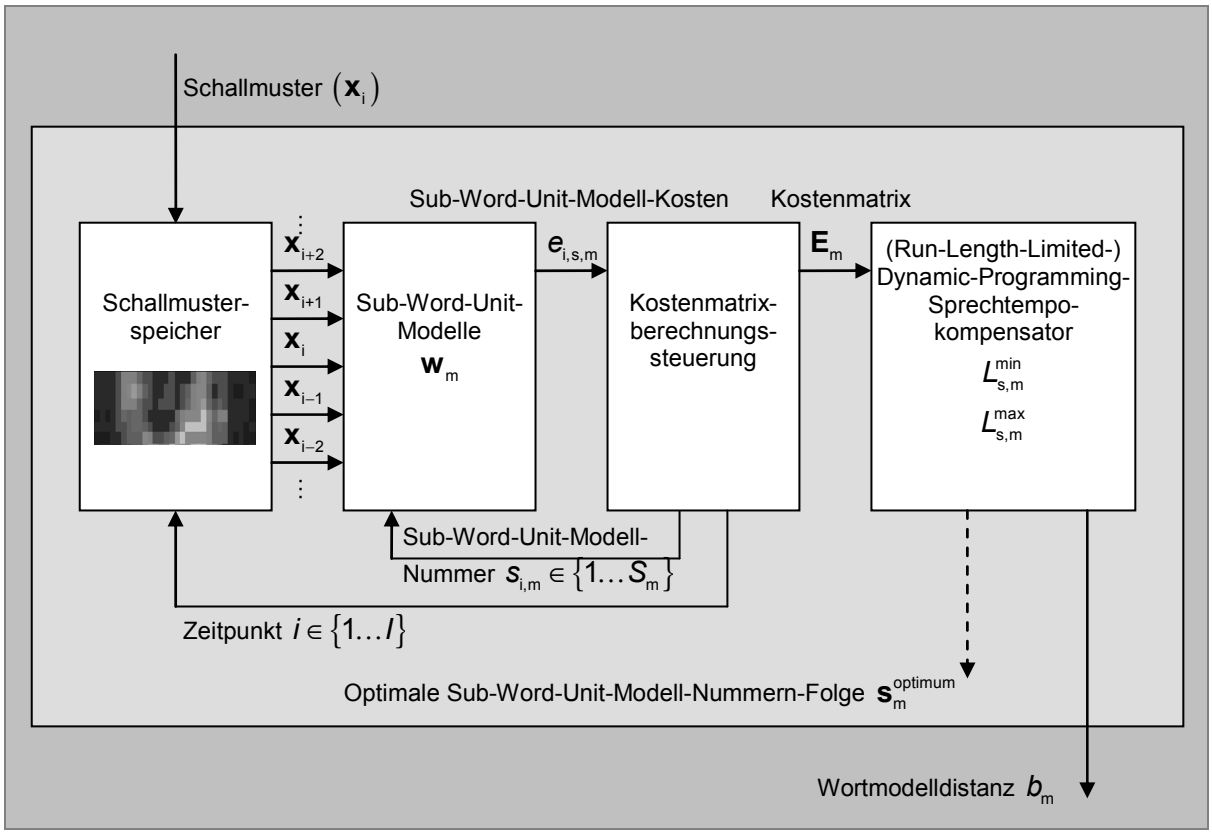

Abbildung 1-7: Blockschaltbild eines Viterbi-Dynamic-Programming-Wortmodells.

[4] Es sind durchaus andere Möglichkeiten denkbar: Die Verarbeitung der Wortmodelldistanzen zum Klassifikationsergebnis kann etwa mittels eines Neuronalen Netzes durchgeführt werden. Im Allgemeinen kann das Klassifikationsergebnis auch durch syntaktische und semantische Regeln beeinflusst werden.

An den Eingängen des Sub-Word-Unit-Modell-Funktionsblocks werden die Merkmalsvektoren der Umgebung eines Zeitpunktes bereitgestellt, sowie die Sub-Word-Unit-Modellnummer, die angibt, welches Sub-Word-Unit-Modell im Block aktiviert wird. Die Anzahl der Sub-Word-Unit-Modelle im Funktionsblock wird im Folgenden mit S bezeichnet. Die Sub-Word-Unit-Modellnummern $s_{i,m}$ nehmen nur Werte der Menge $\{1,2,3,\ldots S_m\}$ an, wobei jedes Element dieser Menge umkehrbar eindeutig je einem Sub-Word-Unit-Modell im Funktionsblock zugeordnet ist.

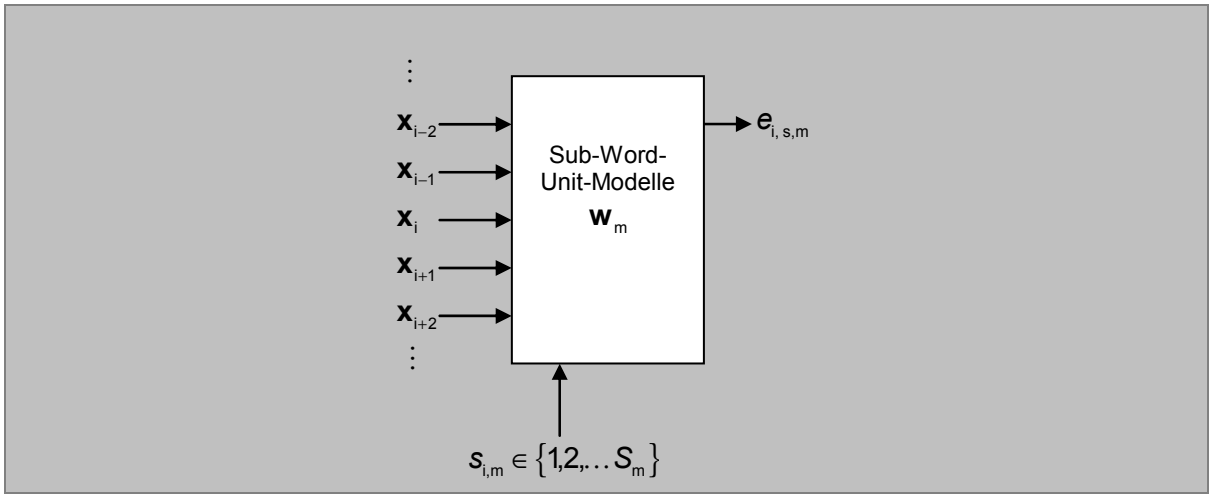

Abbildung 1-8: Die Schnittstellen des Sub-Word-Unit-Modell-Funktionsblocks. Die Zustandsnummer zu einem bestimmten Zeitpunkt aktiviert im Funktionsblock ein bestimmtes Sub-Word-Unit-Modell. Ein Fehlermaß, die Sub-Word-Unit-Modell-Distanz, wird ausgegeben.

Abbildung 1-8 zeigt die Schnittstellen des Sub-Word-Unit-Modell-Funktionsblocks. Die verschiedenen Spracherkenner der Klasse der „Viterbi Dynamic Programming Speech Recognizer“ unterscheiden sich durch unterschiedliches Innenleben dieses Funktionsblocks. Unterschiedliche Sub-Word-Unit-Modell-Funktionsblöcke weisen auch unterschiedliche freie Parameter auf, die das „Wissen“ des Funktionsblocks über die zu repräsentierende Klasse verkörpern. Der Parametersatz eines Sub-Word-Unit-Modell-Funktionsblocks wird im Folgenden pauschal mit $\mathbf{w}_m$ bezeichnet.

Die Kostenmatrix-Berechnungssteuerung beginnt, alle Elemente der Kostenmatrix zu indizieren, indem sie die Sub-Word-Unit-Modell und den Zeitpunkt durchzählt, wobei die Reihenfolge im Prinzip beliebig ist. Für jeden beobachteten Merkmalsvektor wird bezüglich jedes Sub-Word-Unit-Modells die Abweichung der Modellvorstellung vom tatsächlich beobachteten Merkmalsvektor berechnet. Wichtig dabei ist nicht nur, dass die Sub-Word-Unit-Modell-Distanzen bei Schallereignissen, die das Wortmodell repräsentieren soll, gering sind, sondern auch, dass die Sub-Word-Unit-Modell-Distanzen bei Schallereignissen, die das Wortmodell nicht repräsentieren soll, weil sie etwa durch andere Wortmodelle repräsentiert werden, oder weil es sich um Hintergrundgeräusche handelt, hoch sein sollen. Siehe dazu Abschnitt 1.4.2 auf Seite 96.

Sprechtempokompensation. Die Kostenmatrix wird üblicherweise einem Viterbi-Dynamic-Programming-Sprechtempokompensator übergeben, wobei im Rahmen dieser Arbeit als allgemeine Verbesserung der Run-Length-Limited-Sprechtempokompensator vorschlagen wird. Zum Zwecke des Trainings wird die optimale Folge von Sub-Word-Unit-Modellnummern ermittelt und zum Zwecke der Erkennung wird die Wortmodelldistanz ausgegeben. Die Bestimmung der Zustandsnummernfolge $\mathbf{s}_m = (s_i)_m$ für das Wortmodell m ist gleichbedeutend mit der Unterteilung des Schallmusters $(\mathbf{x}_i)$ in Zeitabschnitte. Die Zustandsnummer gibt für jeden Zeitpunkt i an, welches Sub-Word-Unit-Modell des Wortmodells aktiv ist. Beim Sprechtempoausgleich mittels des Viterbi-Dynamic-Programming-Algorithmus wird nicht nur die optimale Zustandsfolge $\mathbf{s}_m^{optimum} = (s_i)_m^{optimum}$ des Wortmodells m bestimmt, sondern, quasi als Nebenprodukt, auch die zugehörige Wortmodelldistanz b_m. Die Wortmodelldistanz b_m erfasst quantitativ, wie gut ein gegebenes Wortmodell m auf ein gegebenes Schallmuster $(\mathbf{x}_i)$ passt.

Wortmodelldistanzberechnung im Wortmodellfunktionsblock. Im Folgenden bedeutet die Wortmodelldistanz $b_m = 0$ ideale Übereinstimmung der Eigenschaften des Schallmusters mit den Klasseneigenschaften, welche durch das Wortmodell repräsentiert werden. Je kleiner der Betrag der Wortmodelldistanz b_m ist, desto höher sei die Übereinstimmung der Eigenschaften.

$$b_m \geq 0 \qquad \textbf{Gleichung 2}$$

Es gelte weiters Gleichung 2. Die Bedingung nach Gleichung 2 soll keine Einschränkung der Allgemeinheit bedeuten, da sich entartete Distanzfunktionen, welche die Bedingung nicht erfüllen üblicherweise mittels einfacher Abbildungen[5] derart transformiert werden können, dass die Bedingung eingehalten wird.

Die Folge der Sub-Word-Unit-Modell-Distanzen eines Wortmodells wird im Folgenden mit $(e_{i,s})_m$ bezeichnet. Die Summe der Vorhersagefehler des Wortmodells bei gegebenem Schallmuster $(\mathbf{x}_i)$, gegebener Folge von Segmentnummern $(s_i)_m$ und gegebenem Parametersatz $\mathbf{w}_m$ wird im Folgenden als Pfadkosten v_m bezeichnet:

$$v_m = \sum_{i=1}^{I} e_{i,s_{i,m},m} \qquad \textbf{Gleichung 3}$$

Die Parameter eines Wortmodells $\mathbf{w}_m$ führen also mit einer Zustandsfolge $\mathbf{s}_m = (s_i)_m$ bezüglich eines Schallmusters $\mathbf{X} = (\mathbf{x}_i)$ zu bestimmten Pfadkosten v_m.

---

[5] Die zusätzliche Abbildung wird dann freilich Teil der neuen Distanzfunktion und ist im Rahmen einer physikalischen beziehungsweise stochastischen Interpretation der Distanzfunktion mit zu berücksichtigen.

$$\textit{Pfadkostenfunktion}: \left(\mathbf{X}, \mathbf{w}_m, \mathbf{s}_m\right) \mapsto v_m \qquad \textbf{Gleichung 4}$$

Die zunächst unbekannte Zustandsnummernfolge, die sogenannte „Hidden Control“, wird stets in optimaler Weise gewählt. Optimal bedeutet, dass $\mathbf{s}_m^{\text{optimum}}$ aus der Menge der Möglichkeiten derart gewählt wird, dass die Pfadkostensumme v_m das globale Minimum, die sogenannte „Wortmodelldistanz“ b_m, annimmt.

$$\mathbf{s}_m^{\text{optimum}} = \underset{\text{alle zulässigen } \mathbf{s}_m}{\operatorname{argmin}} \textit{Pfadkostenfunktion}\left(\mathbf{X}, \mathbf{w}_m, \mathbf{s}_m\right) \qquad \textbf{Gleichung 5}$$

Beim Viterbi-Dynamic-Programming-Algorithmus [Levin1993] [Forney1973] [Bellman1957] wird neben $\mathbf{s}_m^{\text{optimum}}$ auch gleich die Wortmodelldistanz b_m mitbestimmt:

$$\begin{aligned} b_m = v_m^{\text{optimum}} &= \min_{\text{alle zulässigen } \mathbf{s}_m} \textit{Pfadkostenfunktion}\left(\mathbf{X}, \mathbf{w}_m, \mathbf{s}_m\right) = \\ &= \textit{Pfadkostenfunktion}\left(\mathbf{X}, \mathbf{w}_m, \mathbf{s}_m^{\text{optimum}}\right) \end{aligned} \qquad \textbf{Gleichung 6}$$

Die Wortmodelldistanz b_m ist somit lediglich eine Funktion des Schallmusters $(\mathbf{x}_i)$ und des Parametersatzes $\mathbf{w}_m$. Dabei kommt üblicherweise der Viterbi-Dynamic-Programming-Sprechtempokompensator nach Abschnitt 1.2.1 oder als Verbesserung bezüglich der Klassifikationsleistung des Wortmodells der Run-Length-Limited-Sprechtempokompensator nach Abschnitt 1.2.2, ein Search-Space-Reduction-Verfahren, zur Anwendung.

$$b_m = \textit{Wortmodelldistanz}\left(\mathbf{X}, \mathbf{w_m}\right) \qquad \textbf{Gleichung 7}$$

Das Training des Wortmodellfunktionsblocks. Das Training eines Wortmodells erfolgt, indem die Wortmodellparameter $\mathbf{w}_m$ derart bestimmt werden, dass das Wortmodell bei der Berechnung der Wortmodelldistanz eines Schallmusters der Schallmusterklasse, für die es trainiert ist, möglichst gut passt. Dies soll insbesondere nicht nur für Trainingsschallmuster sondern vor allem für Schallmuster während der späteren Benutzungsphase gelten. Wichtig ist daher die Fähigkeit der Wortmodelle, zu verallgemeinern.

Das Training erfolgt durch abwechselnde Neuschätzung der Wortmodellparameter und Bestimmung der zum Wortmodell gehörenden Segmentierung mittels des Viterbi-Dynamic-Programming-Algorithmus [Levin1993] [Forney1973] [Bellman1957] beziehungsweise des Run-Length-Limited-Dynamic-Programming-Algorithmus [Hickersberger2004]. Die Abbruchbedingung ist erfüllt, wenn sich die Segmentierung nicht mehr wesentlich ändert.

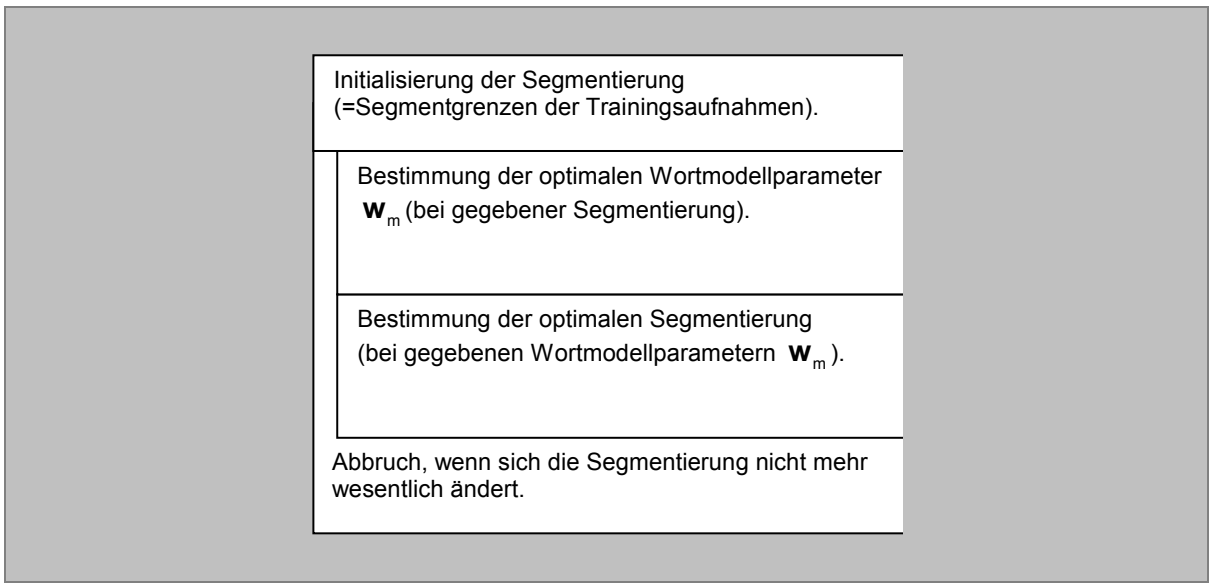

Abbildung 1-9: Programmablauf des Trainings eines Wortmodellfunktionsblocks. Wechselspiel der Bestimmung der für die gegebene Segmentierung optimalen Wortmodellparameter und der Sprechtempokompensation mittels des Viterbi-Dynamic-Programming-Algorithmus.

Konvergenz lässt sich einfach zeigen: Die Summe der Wortmodelldistanzen der Trainingsschallmuster wird abwechselnd durch Neuschätzung der Segmentierung und Aktualisierung der Wortmodellparameter verringert. Die Folge konvergiert, da jede einzelne Wortmodelldistanz größer null ist und bei jedem der beiden Schritte verringert wird [Levin1993]. Dass jede einzelne Wortmodelldistanz größer null ist, also die Bedingung nach Gleichung 2 auf Seite 33 erfüllt ist, muss durch die Wahl eines geeigneten Verfahrens zur Wortmodelldistanzberechnung sichergestellt werden.

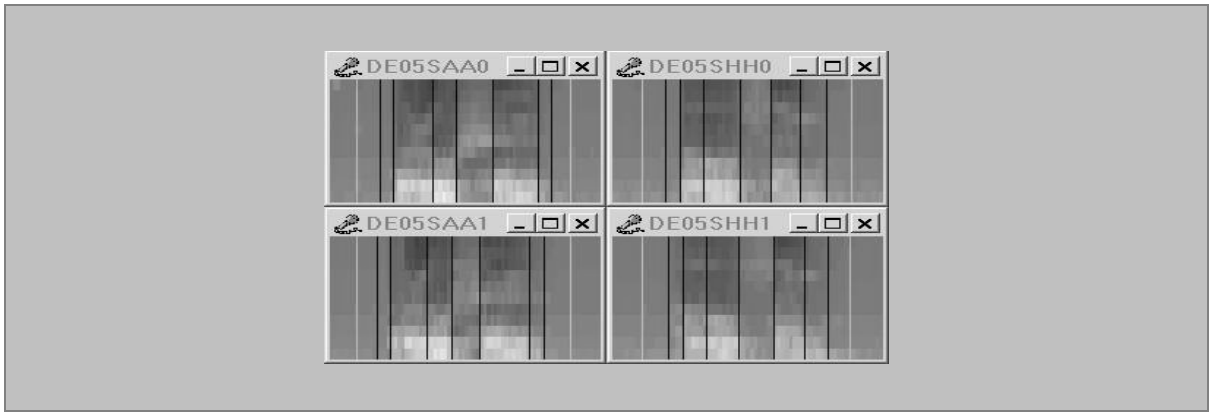

Abbildung 1-10: Trainingsschallmuster des Wortes „rufen“ gesprochen von verschiedenen Sprechern. Die Sprechtempokompensation ändert sich schon nach wenigen Trainingsiterationen nicht mehr. Es ergeben sich äquivalent klingende Zeitabschnitte. Die jeweils äußeren beiden Linien stellen die vorab geschätzten Wortgrenzen dar, welche zur Initialisierung der Segmentierung verwendet werden.

Lineare Diskriminanzanalyse. Ist der Schalter „LDA-Funktionalität“ in Abbildung 1-3 auf Seite 20 aktiviert, so wird eine lineare Diskriminanzanalyse durchgeführt: Nachdem initialen Training aller Wortmodelle, werden die Merkmalsvektoren anhand der Segmentierung gemäß Abbildung 1-11, verschiedenen Punktwolken (○,△,▱,♥,◎,✚,◖) zugeordnet:

Abbildung 1-11: Zuordnung der Merkmalsvektoren zu Punktwolken für die lineare Diskriminanzanalyse. Die Zuordnung erfolgt anhand der Segmentierung, welche ein Nebenprodukt eines vorangegangenen (gegebenenfalls initialen) Wortmodelltrainings ist.

Das erste Segment und das letzte Segment repräsentieren die Stille vor beziehungsweise nach dem Schallereignis und werden der selben Punktwolke zugeordnet. Mittels Linearer Diskriminanzanalyse (siehe Anhang III auf Seite 201) wird eine gemeinsame Transformationmatrix für die Merkmalsvektoren aller Wortmodelle bestimmt, die sogenannte „LDA-Matrix“. Das Training der Wortmodelle wird sodann mit transformierten Merkmalsvektoren wiederholt.

State-Insertion, State-Deletion. Es hat sich bewährt, das gesamte Training – gegebenenfalls inklusive der Durchläufe für die Linearer Diskriminanzanlyse – mehrfach durchzuführen, wobei die Initialsegmentierung vom vorangegangenen Trainingsdurchgang übernommen wird, jedoch systematisch Segmentgrenzen eingefügt beziehungsweise entfernt werden: Für jede Möglichkeit[6], zwischen zwei Grenzen eine Grenze einzufügen, wird das Wortmodelltraining durchgeführt. Entsprechend der beim Wortmodelltraining optimierten Größe[7], werden die Wortmodellparameter und die Segmentierung der besten Möglichkeit gespeichert. Die anderen Möglichkeiten werden verworfen. Sodann wird für jede Möglichkeit[8], eine Grenze zu entfernen, das Wortmodelltraing durchgeführt und die Wortmodellparameter sowie die Segmentierung der dabei besten Möglichkeit gespeichert, und so weiter.

---

[6] Es gibt gleich viele Möglichkeiten wie Segmente.

[7] Beim implementierten Algorithmus ist das der gesamte Modellfehler.

[8] Es gibt eine Möglichkeit weniger als Segmente.

1.2.1 Viterbi-Dynamic-Programming-Sprechtempokompensator

Bereits in den 1950er-Jahren wurde die Klasse der Dynamic-Programming-Algorithmen [Bellman1957] allgemein beschrieben. Die Algorithmen dieser Klasse dienen zur Lösung von Optimierungsproblemen[9]. Die Anwendungsbereiche sind weit gestreut und reichen bis in die Wirtschaftswissenschaften. Auch auf dem Gebiet der Sprach-erkennung gibt es vielfältige Einsatzmöglichkeiten, deren Beschreibung den Rahmen dieser Arbeit bald wesentlich sprengen würde: Erwähnt sei beispielhaft der bekannte und verbreitete „Dynamic Time Warping Speech Recognizer“ [Yaniv2003], bei dem ein Schall-muster bezüglich des Sprechtempos an ein anderes Schallmuster zum Zwecke des Vergleichs angepasst wird. Der mittels eines Dynamic-Programming-Algorithmus gefundene Pfad wird dabei „Warping-Funktion“[10] genannt.

Kostenmatrix und Trellis-Diagramm. Für jedes Sub-Word-Unit-Modell werden ja mittels der Kostenmatrixberechnungssteuerung, wie in Abbildung 1-7 auf Seite 31 ersichtlich, bezüglich jedes Zeitpunkts bestimmte Sub-Word-Unit-Modell-Distanzen oder Kosten berechnet, die in der Kostenmatrix zusammengefasst werden.

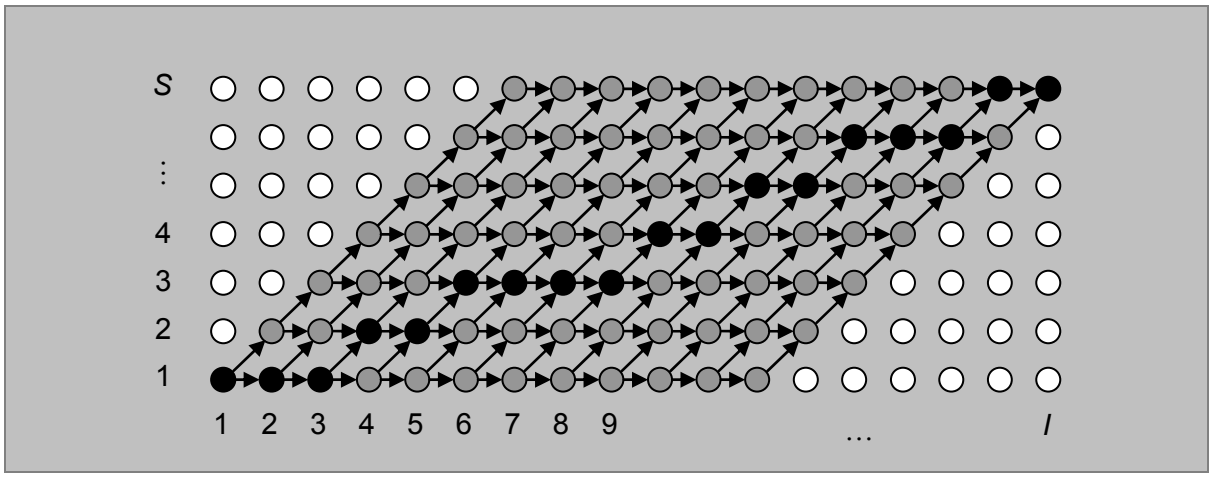

Abbildung 1-12: Der Viterbi-Dynamic-Programming-Algorithmus dient zur Bestimmung des optimalen Pfades durch das Trellis-Diagramm.

---

[9] Ein bekanntes Optimierungsproblem, dass mittels eines „Dynamic-Programming-Algorithmus“ gelöst wird, ist das sogenannte „Rucksack-Problem“: Es gilt aus einer Menge von Objekten, die jeweils ein Gewicht und einen Nutzwert haben, eine Teilmenge auszuwählen, deren Gesamtgewicht eine vorgegebene Gewichtsschranke nicht überschreitet, wobei der Gesamtnutzwert der ausgewählten Objekte maximal ist.

[10] Im Allgemeinen werden die erlaubten Übergänge eingeschränkt, sodass die Warping-Funktion eine bestimmte „Fläche“ nicht verlassen darf, deren „Rand“ beispielsweise der halben beziehungsweise doppelten Sprechgeschwindigkeit bestimmt wird. Ein Übergang „auf gleicher Höhe“ nach rechts entspricht der Sprechgeschwindigkeit Null und ein Übergang nach rechts und zweimal nach oben entspricht der doppelten Sprechgeschwindigkeit des einen Schallmusters bezüglich des anderen Schallmusters.

Jeder Kreis in Abbildung 1-12 symbolisiert ein Element der Kostenmatrix. Jedem Kreis ist ein Zustand s zugeordnet, das heißt ein aktiviertes Sub-Word-Unit-Modell, zu einem bestimmten Zeitpunkt. Der Index i kennzeichnet die Zeitpunkte im Schallereignis.

Left-to-Right-Zustandsmodell. Abbildung 1-12 zeigt das sogenannte „Trellis-Diagramm“ eines herkömmlichen Left-to-Right-Zustandsmodells, denn die Übergangspfeile sind derart eingetragen, dass folgende Bedingungen erfüllt sind. Erstens: Zum Zeitpunkt $i = 1$ ist $s = 1$, das heißt, der Kreis links unten markiert den Ausgangspunkt. Zweitens: Zum Zeitpunkt $i = I$ ist $s = S_m$, das heißt, der Kreis rechts oben markiert den Zielpunkt. Drittens: Die Zustandsfolge ist monoton steigend. Zunächst unbekannt sind die genauen Zeitpunkte der Übergänge. Die Pfeile kennzeichnen jedenfalls die erlaubten Übergänge[11], wobei jedoch die optimalen Übergänge zunächst noch unbekannt sind. Das in Abbildung 1-12 dargestellte Trellis-Diagramm entspricht dem Zustandsmodell nach Abbildung 1-13.

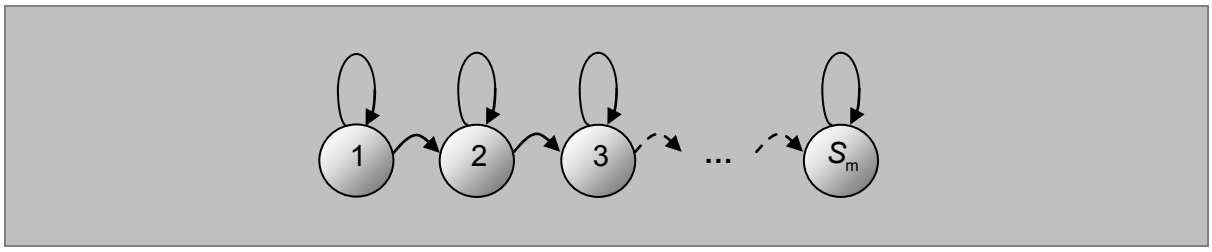

Abbildung 1-13: Ein herkömmliches Left-to-Right-Zustandsmodell.

Abbildung 1-13 zeigt ein herkömmliches Left-to-Right-Zustandsmodell. Eingetragen sind die Zustandsnummern.

Kinderspielplatz-Analogie. Anschaulich gesprochen stellt sich die Aufgabenstellung folgendermaßen dar: Man stelle sich die reellen Zahlen in Abbildung 1-12 auf Seite 37 als Linien unterschiedlicher Länge vor, die aus der Zeichenebene herausragen. Zum Beispiel könnten die Längen jenen von Baumstümpfen auf einem Kinderspielplatz entsprechen: Eine Person startet am Baumstumpf unten links und möchte zum Baumstumpf oben, rechts. Die Person darf Schritte, den Pfeilen entsprechend, tätigen. Gesucht sei nun der Weg, bei dem die Summe der Höhen der berührten Baumstümpfe minimal ist, sowie eben diese Summe.

Bestimmung des globalen Minimums. Der herkömmliche Viterbi-Dynamic-Programming-Sprechtempokompensator bestimmt nun jenen Pfad, entlang dessen die Summe der Kosten minimal ist. Dazu geht er die Zeitpunkte von links nach rechts durch und bestimmt für jeden Zustand zu jedem Zeitpunkt die minimal auflaufenden Kosten und den zu diesen Kosten gehörenden Vorgängerzustand. Vom letzten Zustand ausgehend,

[11] Im Allgemeinen können durchaus andere Übergänge erlaubt sein, welche aber im Trellis-Diagramm stets eine Spalte nach rechts führen müssen. Das heißt, die Folge der Zeitpunkte muss streng monoton steigen. Die erlaubten Übergänge ergeben sich aus dem zugehörigen Zustandsmodell.

werden sodann die Vorgängerzustände zurückverfolgt und der optimale Pfad $\mathbf{s}_m = (s_i)_m$ mitprotokolliert.

Abbildung 1-12 auf Seite 37 zeigt horizontal diskrete Zeitpunkte vertikal diskrete Zustandsnummern. Zu jedem Zeitpunkt sind nur bestimmte Zustände möglich. Der Zustand zum Zeitpunkt $i = 1$ ist $s_{1,m} = 1$. Der Zustand zum Zeitpunkt $i = I$ ist $s_{I,m} = S_m$. Die Folge der Zustände $(s_i)_m$ ist außerdem monoton steigend. Es wird also ein Zustandsmodell nach Abbildung 1-13 auf Seite 38 vorausgesetzt. Aus diesen Bedingungen ergibt sich, dass gewisse Zustände zu gewissen Zeitpunkten nicht erreichbar sind. Sie sind Abbildung 1-12 auf Seite 37 weiß unterlegt. Die Folge $(s_i)_m$ muss den folgenden Bedingungen nach Gleichung 8 genügen:

$$\begin{aligned} s_{1,m} &= 1 \\ s_{I,m} &= S_m \\ (s_{i,m} = s_{i-1,m}) \text{ oder } (s_{i,m} &= s_{i-1,m} + 1) \quad \forall i \in \{2 \ldots I\} \end{aligned}$$

Gleichung 8

Als „Zweig“ bezeichnet man im Trellis-Diagramm ein Wegstück, welches vom Zustand des Zeitpunktes i zum Zustand des Zeitpunkts $i+1$ führt.[12] Als „Pfad“ bezeichnet man eine Aneinanderreihung sämtlicher $I-1$ Zweige. Er beginnt bei der Position $(1,1)$ und endet bei der Position (I, S_m).[13] Als „Teilpfad“ wird im Folgenden ein unvollständiger Pfad bezeichnen, welcher bei der Position $(1,1)$ beginnt und bei einer möglichen Position (i,s) endet. Unter den „Zweigkosten“ versteht man jene reelle Zahl, die dem Zweig zugeordnet ist. „Pfadkosten“ beziehungsweise „Teilpfadkosten“ sind entsprechend als Summe der betreffenden Zweigkosten definiert.

Die Zweigkosten stellen den Datensatz dar, den der Viterbi-Dynamic-Programming-Algorithmus verarbeitet. Im allgemeinen Fall setzen sich die Zweigkosten aus „Zustandskosten“ und „Übergangskosten“ zusammen. Jeder möglichen Position (i,s) sind bestimmte Zustandskosten $e_{i,s,m}$ zugeordnet. Jedem möglichen Übergang, der von einer Position $(i-1,s)$ zur infrage stehenden Position (i,s) führt, sind Übergangskosten $d_{i,s',s,m}$ zugeordnet.

$$\hat{e}_{i,s',s,m} = d_{i,s',s,m} + e_{i,s,m}$$

Gleichung 9

Es ist üblich, für die im Abschnitt 1.2 beschriebenen Spracherkenner $d_{i,s',s,m} = 0$ zu setzen, sodass die Zweigkosten allein aus den Zustandskosten bestehen. Jedem möglichen Pfad $\mathbf{s}_m = (s_i)_m$ von (1,1) nach (I, S_m) sind Pfadkosten ν_m zugeordnet.

---

[12] Jeder mögliche Zweig ist in Abbildung 1-12 auf Seite 37 mit einem Pfeil versehen.

[13] Ein möglicher Pfad, das heißt eine mögliche Folge von Zuständen (s_i), ist in Abbildung 1-12 auf Seite 37 schwarz unterlegt.

$$v_m = \sum_{i=1}^{I} e_{i,\, s_{i,m},m} \qquad \textbf{Gleichung 10}$$

Jeder möglichen Position (i,s) ist eine Menge aller Teilpfade zugeordnet, welche dorthin führen. Für jede Position (i,s) zeichnet sich ein bestimmter Teilpfad aus, der sogenannte „Survivor-Teilpfad“. Das ist jener, welchem die kleinsten Teilpfadkosten zugeordnet sind. Weisen mehrere Teilpfade die gleichen Teilpfadkosten auf, so wird ein beliebiger als „Survivor-Teilpfad“ festgelegt.

Gesucht sind die minimalen Pfadkosten sowie ein Pfad, der die minimalen Pfadkosten aufweist. Wie wird diese Aufgabe also gelöst? Alle möglichen Pfade durchzuprobieren, ist glücklicherweise nicht notwendig.

$$b_m = \min_{\text{alle zulässigen } \mathbf{s}_m} v_m \qquad \textbf{Gleichung 11}$$

$$b_m = \min_{\text{alle zulässigen } \mathbf{s}_m} \sum_{i=1}^{I} e_{i,\, s_{i,m},m} \qquad \textbf{Gleichung 12}$$

Die meisten Teilpfade, nämlich alle außer den Survivor-Teilpfaden, können vorzeitig verworfen werden, da Teilpfade von Survivor-Teilpfaden stets wieder Survivor-Teilpfade sind. Das erste Glied der Zustandsfolge ist bekannt $s_1 = 1$. Es wird nun für jede mögliche Position in Abbildung 1-12 auf Seite 37 ein Survivor-Teilpfad berechnet und dessen Teilpfadkosten $\hat{e}_{i,s,m}$ gespeichert. Es erweist es sich nicht als notwendig, für jede Position den gesamten Survivor-Teilpfad zu speichern. Es genügt, festzuhalten, welche Position der Survivor-Teilpfad einen Zeitpunkt zuvor einnahm. Der dort anschließende Teilpfad ist ja wieder ein Survivor-Teilpfad und auch ihm zugehörend ist die Vorgängerposition gespeichert. Mittels dieser Wegweiserinformation ist der gesamte Survivor-Pfad rekonstruierbar. Die Berechnung der Wegweiserinformation des Survivor-Pfades zu einer bestimmten Position geschieht nun folgendermaßen: Die Zweigkosten, welcher ein Übergang insgesamt verursacht, also $d_{i,s',s,m} + e_{i,s,m}$, werden für jeden möglichen Vorgängerzustand zu dessen aufgelaufenen Teilpfadkosten addiert. Daraus ergeben sich verschiedene mögliche neue Teilpfadkosten zur Position (i,s). Diese werden nun verglichen und die geringsten möglichen Teilpfadkosten $\hat{e}_{i,s,m}$ werden zur Position (i,s) gespeichert. Weiters bestimmen diese geringsten Teilpfadkosten die Wegweiserinformation $\hat{s}_{i,s,m}$ der Position (i,s), das heißt also, den besten Vorgängerzustand. Diesen Vorgang beschreiben Gleichung 13 und Gleichung 14.

$$\hat{e}_{i,s,m} = e_{i,s,m} + \min_{\text{alle zulässigen } s'} \left(\hat{e}_{i-1,s,m} + d_{i,s',s,m} \right)$$ **Gleichung 13**

$$\hat{s}_{i,s,m} = \operatorname*{argmin}_{\text{alle zulässigen } s'} \left(\hat{e}_{i-1,s,m} + d_{i,s',s,m} \right)$$ **Gleichung 14**

Für das Left-to-Right-Zustandsmodell nach Abbildung 1-5 auf Seite 25, das heißt für die erlaubten Übergänge nach Abbildung 1-9 auf Seite 35 und den Spezialfall verschwindender Übergangskosten[14] $d_{i,s',s,m} = 0$ ergibt sich Gleichung 15.

$$\hat{e}_{i,s,m} = e_{i,s,m} + \min_{s' \in \{s, s-1\}} \left(\hat{e}_{i-1,s,m} \right)$$ **Gleichung 15**

Ist für jede infrage kommende Position ein Survivor-Teilpfad berechnet, so kennt man schließlich auch den Survivor-Pfad für (I, S_m), welcher der optimale Pfad $\mathbf{s}_m^{\text{optimum}}$ im Sinne der Pfadkosten ist.[15] Man erhält weiters, gewissermaßen als Nebenprodukt, die Wortmodelldistanz b_m nach Gleichung 16.

$$\begin{aligned} b_m &= \textit{Wortmodelldistanz}\left(\mathbf{X}, \mathbf{w}_m \right) = \\ &= \textit{Pfadkostenfunktion}\left(\mathbf{X}, \mathbf{w}_m, \mathbf{s}_m^{\text{optimum}} \right) = \hat{e}_{I,S,m} \end{aligned}$$ **Gleichung 16**

[14] Wenn die Übergangskosten bezüglich jedem möglichen Vorgängerzustand definitionsgemäß gleich sind, so können diese ebenfalls null gesetzt werden und die Kosten dafür den Zustandskosten zugeschlagen werden.

[15] Da zu jeder Position der Survivor-Pfad nur durch die Vorgängerposition gespeichert wird, muss der Survivor-Pfad letztendlich noch vom Endzustand zum Endzeitpunkt ausgehend zurückverfolgt werden.

1.2.1.1 Hidden Control Neural Network Speech Recognizer

Hidden-Control-Neural-Network-Sub-Word-Unit-Modell. Die Funktionalität des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 wird beim „Hidden Control Neural Network Speech Recognizer“ [Levin1990] [Sorensen1992a] [Levin1993] [Hickersberger1997] [Hickersberger1998] mittels eines Hidden-Control-Neural-Network-Sub-Word-Unit-Modells, welches aus einem einzigen Multi-Layer-Perceptron-Netz besteht, realisiert.

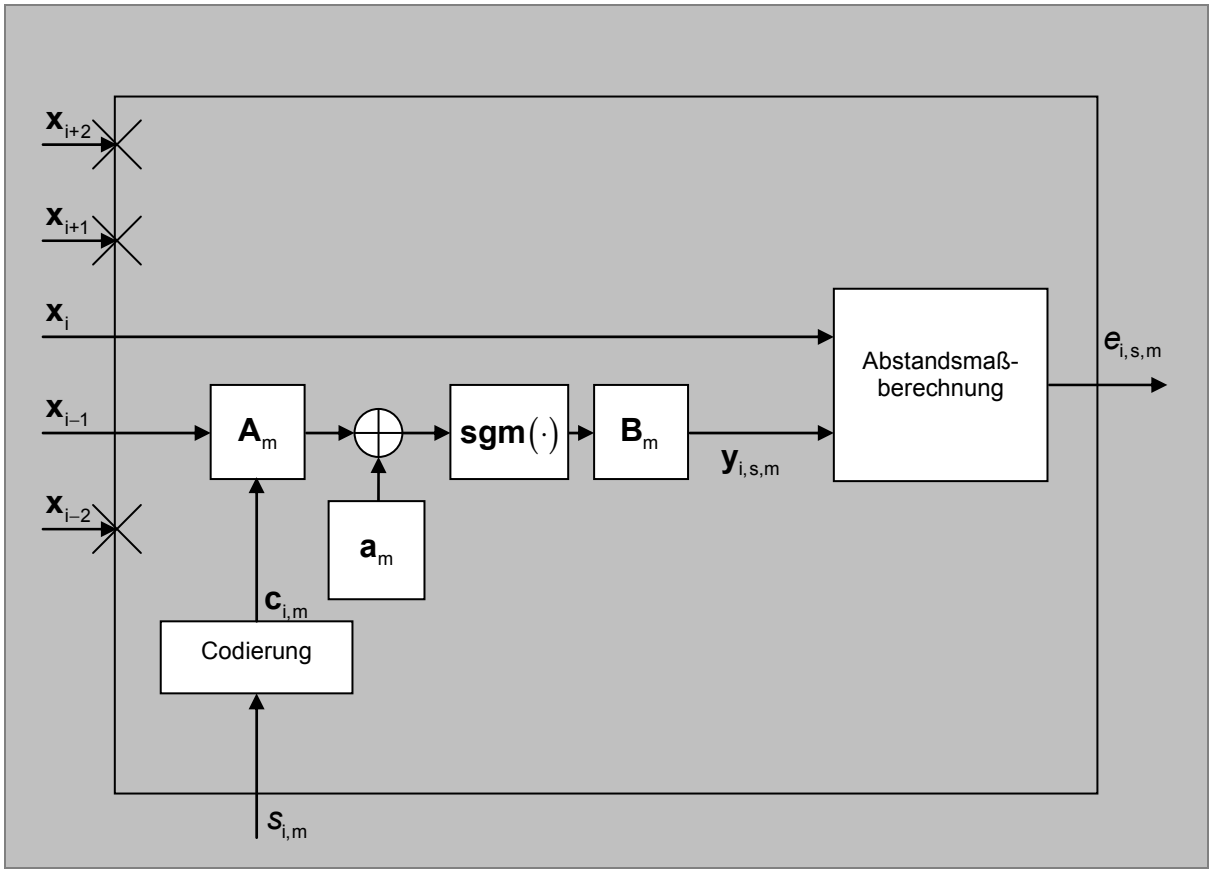

Abbildung 1-14: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Hidden-Control-Neural-Network-Sub-Word-Unit-Modell.

Abbildung 1-14 zeigt ein Hidden-Control-Neural-Network-Sub-Word-Unit-Modell, wie es im Rahmen von [Hickersberger1997] implementiert und evaluiert wurde. Das besondere an diesem Modell ist, das die Zustandsnummer mittels eines 1-aus-S_m-Codes oder eines Thermometercodes codiert wird und dem Eingangsvektor des Neuronalen Netzes zugesetzt wird. Siehe auch Abbildung 1-15 auf Seite 43.

Es wird eine Distanzfunktion zwischen dem Schätzvektor, das heißt dem Ausgangsvektor des Neuronalen Netzes und dem Zielvektor, dem Merkmalsvektor für den Zeitpunkt berechnet. Eine übliche Distanzfunktion ist die quadrierte euklidische Distanz.

Im Spezialfall der Verwendung der quadrierten euklidischen Distanz als Distanzfunktion ergibt sich Gleichung 17:

$$e_{i,s,m} = \left| \mathbf{x}_i - \mathbf{B}_m \cdot \mathbf{sgm}\left(\mathbf{a}_m + \mathbf{A}_m \cdot \begin{pmatrix} \mathbf{x}_{i-1} \\ \mathbf{c}_{i,m} \end{pmatrix} \right) \right|^2 \qquad \textbf{Gleichung 17}$$

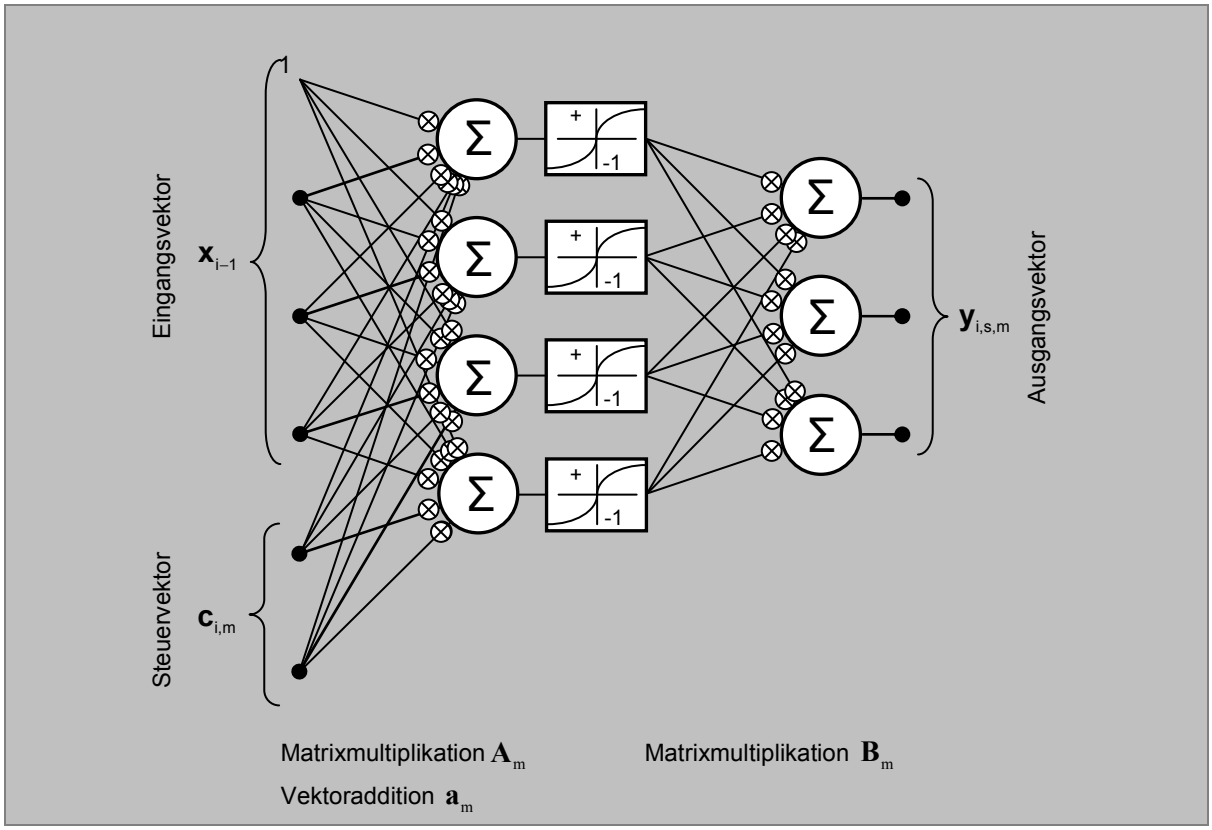

Abbildung 1-15: Kolmogorov-Netz als „Hidden Control Neural Network". Die implementierte Abbildung wird mittels zusätzlicher Steuereingänge, dem Steuervektor, umgeschaltet.

Abbildung 1-15 zeigt den Einsatz eines Kolmogorov-Netzes als „Hidden Control Neural Network" [Levin1989] [Kim1993] [Martinelli1994] [Na1995] [Falaschi1993]. Es erfolgt eine Matrixmultiplikation und eine Vektoraddition. Sodann wird auf jedes einzelne Vektorelement eine Sigmoid-Funktion angewendet und es erfolgt nochmals eine Matrixmultiplikation. Bei dieser Architektur handelt es sich um das einfachst-mögliche Multi-Layer-Perceptron-Netz, welches als „Universeller Approximator" bezeichnet werden kann [Cybenko1989] [Carrol1989]. Gelegentlich stößt man auf die Bezeichnung „Kolmogorov-Netz" [Rojas1996]. Im deutschsprachigen Raum wird diese Architektur eher mit der Literaturstelle [Hornik1989] in Verbindung gebracht. Das Training dieses Neuronalen Netzes erfolgt im Allgemeinen mittels Backpropagation-Optimierung [Widrow1990] [Rojas1996].

„Large Vocabulary Recognition“. Eine Möglichkeit besteht darin, das Hidden-Control-Neural-Network-Sub-Word-Unit-Modell um zusätzliche Steuereingänge zu erweitern, die digital codierte Eigenschaften der repräsentierten Phoneme bereitstellen. Das Modell wird dem „Linked Predictive Neural Network Speech Recognizer“ [Petek1992] [Petek1993] [Petek2000] gegenübergestellt.

„Weighted Hidden Control Neural Network Speech Recognizer“. Die Wortmodelle werden bei diesem Ansatz derart modifiziert, dass die Sub-Word-Unit-Modell-Distanzen mit einem Gewichtsfaktor multipliziert werden, der vom Zustand abhängt [Na1996] [Hickersberger1997] [Hickersberger1998]. Dies ermöglicht dem Spracherkenner das „Hinhorchen“ an bestimmte Stellen im Wort, nämlich an jene, die entscheidungsrelevant sind. Die Verwechslungsfehlerrate wird durch diese Maßnahme gesenkt, weil Hintergrundgeräusche im Allgemeinen zu Merkmalsvektoren führen, die für die Klasse untypisch sind und diese Merkmalsvektoren in den nicht entscheidungsrelevanten Passagen dann schwächer in das Klassifikationsergebnis einfließen. Die Bestimmung der Gewichte der Zustände ist zwar keineswegs trivial, aber die Maßnahme lässt sich sinngemäß auf sämtliche im Abschnitt 1.2 beschriebenen Spracherkenner übertragen und vorteilhaft anwenden.

„Self-Structuring Hidden Control Speech Recognizer“. Dieser Ansatz stellt eine Modifikation der Wortmodelle dahin gehend dar, dass die Neuronen im „Hidden Layer“ Eingänge besitzen, die mit Ausgängen bestimmter anderer Neuronen im „Hidden Layer“ verbunden sind. Das Modell vermeidet Schwierigkeiten bei der Wahl der optimalen Anzahl von Neuronen im „Hidden Layer“ beim „Hidden Control Neural Network“ [Sorensen1992] [Sorensen1992b] [Sorensen1993] [Sorensen1995].

Sprechtempokompensation. Zur Sprechtempokompensation bietet sich beim „Hidden Control Neural Network Speech Recognizer“ neben dem Viterbi-Dynamic-Programming-Algorithmus insbesondere der Run-Length-Limited-Dynamic-Programming-Algorithmus an.

1.2.1.2 Predictive Neural Network Speech Recognizer

Predictive-Neural-Network-Sub-Word-Unit-Modell. Die Realisierung des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 erfolgt beim „Predictive Neural Network Speech Recognizer“ [Lapedes1987] [Iso1990] mittels eines Predictive-Neural-Network-Sub-Word-Unit-Modells bestehend aus umgeschalteten Multi-Layer-Perceptron-Netzen. Eine interessante Einführung in die allgemeine Thematik „Prediction Models“ findet sich in der Literatur [Tishby1990].

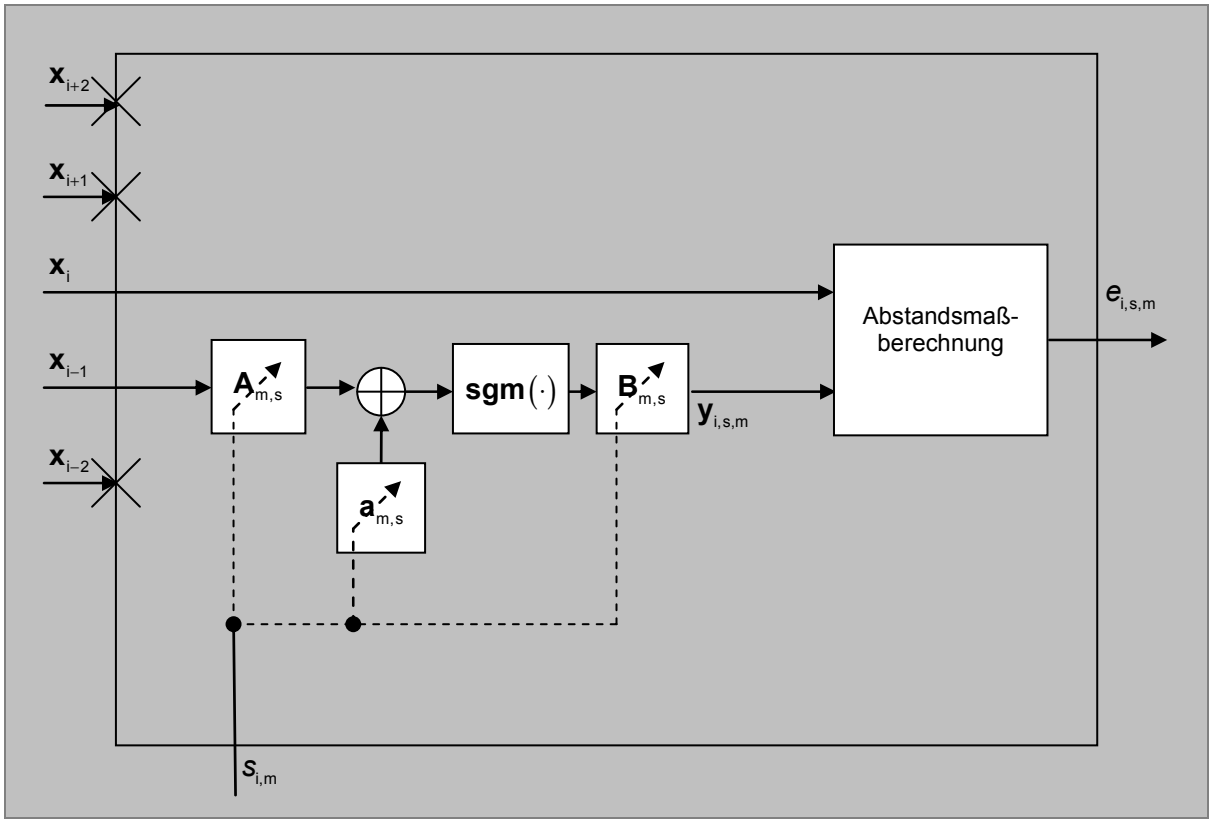

Abbildung 1-16: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Predictive-Neural-Network-Sub-Word-Unit-Modell.

Im Spezialfall der Verwendung der quadrierten euklidischen Distanz als Distanzfunktion ergibt sich Gleichung 18:

$$e_{i,s,m} = \left| \mathbf{x}_i - \mathbf{B}_{m,s} \cdot \mathbf{sgm}\left(\mathbf{a}_{m,s} + \mathbf{A}_{m,s} \cdot \mathbf{x}_{i-1} \right) \right|^2 \qquad \textbf{Gleichung 18}$$

Abbildung 1-16 zeigt das bekannte Predictive-Neural-Network-Sub-Word-Unit-Modell. Es wurde im Rahmen der Diplomarbeit [Hickersberger1997] implementiert und evaluiert. Der Unterschied zum Hidden-Control-Neural-Network-Sub-Word-Unit-Modell besteht darin, dass die Zustandsnummer nicht in codierter Form als Steuervektor zuge-

setzt wird, sondern die Matrizen und Vektoren des Neuronalen Netzes umgeschaltet werden. Das Training erfolgt üblicherweise mittels Backpropagation-Optimierung [Widrow1990] [Rojas1996] und für jede Zustandsnummer unabhängig voneinander.

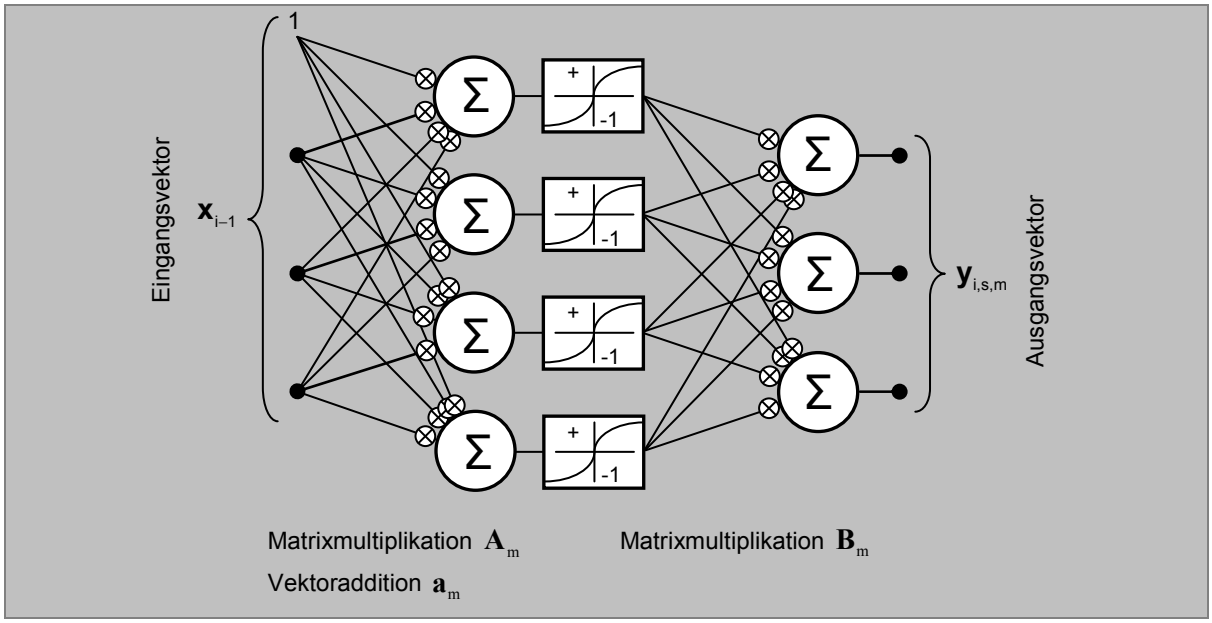

Abbildung 1-17: Kolmogorov-Netz: Das einfachste mögliche Multi-Layer-Perceptron-Netz, welches die Eigenschaft besitzt, bei theoretischer unbegrenzter Anzahl an Hidden-Layer-Neuronen ein sogenannter „Universeller Approximator“ zu sein.

Abbildung 1-17 zeigt ein Kolmogorov-Netz. Dabei handelt es sich um das einfachst-mögliche Multi-Layer-Perceptron-Netz, welches die Eigenschaft besitzt, bei theoretischer unbegrenzter Anzahl an Hidden-Layer-Neuronen ein sogenannter „Universeller Approximator“ zu sein [Hornik1989] [Cybenko1989] [Carrol1989].

„Large Vocabulary Recognition“. Beim sogenannten „Linked Predictive Neural Network Speech Recognizer“ wird mit jedem der vorhandenen Predictive-Neural-Network-Sub-Word-Unit-Modelle ein bestimmter Teil eines bestimmten Phonems modelliert. Vor der eigentlichen Klassifikation werden die Wortmodelle des aktiven Vokabulars gemäß der Lautschrift der Vokabularwörter aus den Predictive-Neural-Network-Sub-Word-Unit-Modellen zusammengesetzt. „Linked Predictive Neural Network Speech Recognizer“ eignen sich für Spracherkenner mit großem Vokabular [Tebelskis1990]. Eine interessante Gegenüberstellung von „Linked Predictive Neural Network Speech Recognizer“ und „Hidden Control Neural Network Speech Recognizer“ findet sich in der Literatur [Petek1992] [Petek2000].

Sprechtempokompensation. Zur Sprechtempokompensation bietet sich beim „Predictive Neural Network Speech Recognizer“ neben dem Viterbi-Dynamic-Programming-Algorithmus insbesondere der Run-Length-Limited-Dynamic-Programming-Algorithmus an.

1.2.1.3 Predictive Wiener Matrix Speech Recognizer

Predictive-Wiener-Matrix-Sub-Word-Unit-Modell. Die Realisierung des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 erfolgt beim „Predictive Wiener Matrix Speech Recognizer“ mittels eines Predictive-Wiener-Matrix-Sub-Word-Unit-Modells bestehend aus umgeschalteten affinen Transformationen. Der Ausgangsvektor entsteht daher aus dem Eingangsvektor mittels einer Matrixmultiplikation und einer Vektoraddition. Die Matrix und der Vektor werden entsprechend der Zustandsnummer umgeschaltet.

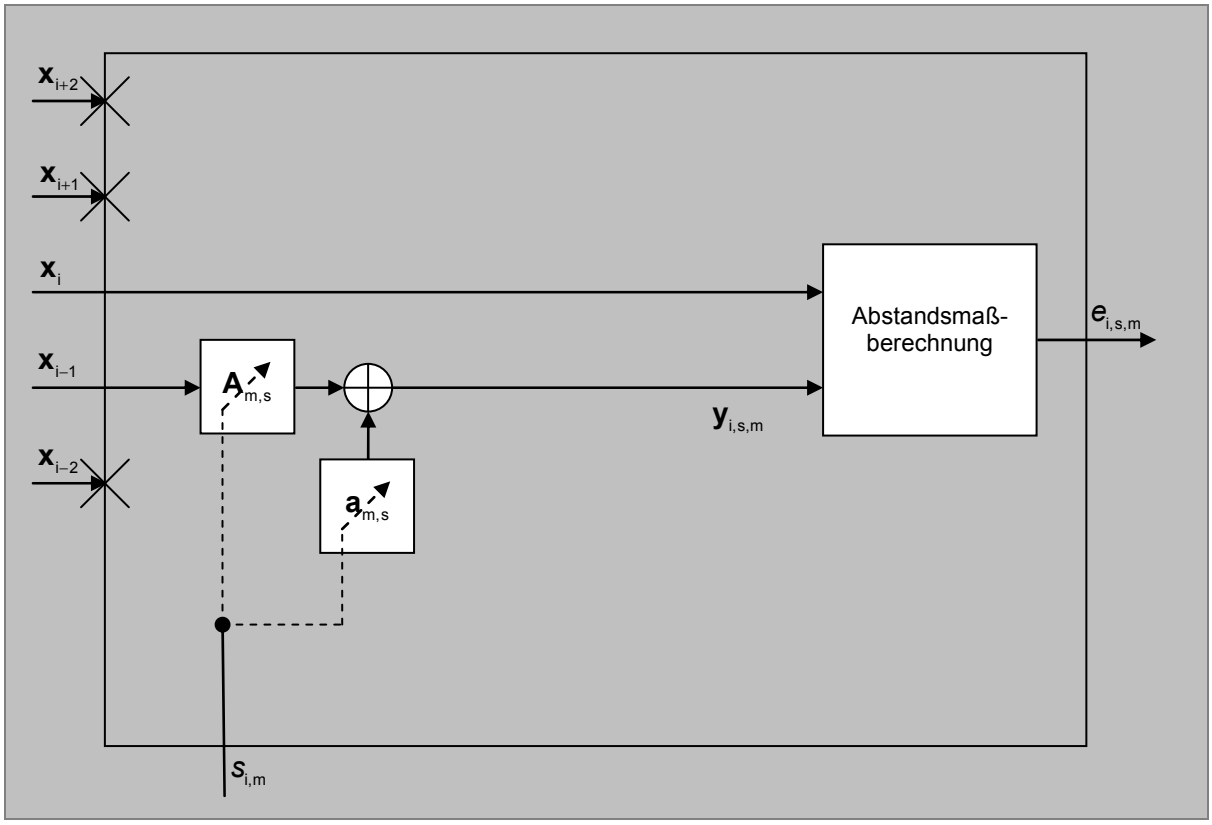

Abbildung 1-18: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Predictive-Wiener-Matrix-Sub-Word-Unit-Modell.

Im Spezialfall der Verwendung der quadrierten euklidischen Distanz als Distanzfunktion ergibt sich Gleichung 19.

$$e_{i,s,m} = \left| \mathbf{x}_i - \left(\mathbf{a}_{m,s} + \mathbf{A}_{m,s} \cdot \mathbf{x}_{i-1} \right) \right|^2 \qquad \textbf{Gleichung 19}$$

Das Sub-Word-Unit-Modell nach Abbildung 1-18, wird im Folgenden als Predictive-Wiener-Matrix-Sub-Word-Unit-Modell bezeichnet.

Einfaches Modelltraining. Es wird kein iteratives Training, beispielsweise mittels Backpropagation-Optimierung benötigt, obwohl dieses natürlich dennoch möglich ist. Ein Vorteil des Verfahrens besteht nämlich darin, dass die optimalen Matrizen und die optimalen Vektoren mittels Pseudoinversionen beziehungsweise Mittelwertbildung nichtiterativ aus den Trainingsdaten berechnet werden können. Die zu addierenden Vektoren ergeben sich beispielsweise als Schätzwerte der Erwartungswerte der $\mathbf{x}_i$. Die Matrizen werden als sogenannte „Wiener Matrizen“ bestimmt, welche die optimalen linearen Transformationen beschreiben. Die „Wiener Matrizen“ lassen sich berechnen, indem man die jeweiligen Eingangsvektoren der Matrixmultiplikation in je einer Eingangsmatrix zusammenstellt, die jeweils gewünschten Ausgangsvektoren der Matrixmultiplikation, welche vor dem Summationszeichen im Idealfall auftreten sollen in je einer Zielmatrix zusammenstellt und jeweils die Pseudoinverse der Eingangsmatrix mit der Zielmatrix multipliziert.

Eine andere Möglichkeit besteht darin, die zu addierenden Vektoren im Zuge der Bestimmung der Multiplikationsmatrizen mitzubestimmen, indem dem Eingansvektor eine Komponente zugesetzt wird, deren Wert gleich eins gesetzt wird.

Sprechtempokompensation. Zur Sprechtempokompensation bietet sich beim „Predictive Wiener Matrix Speech Recognizer“ neben dem Viterbi-Dynamic-Programming-Algorithmus insbesondere der Run-Length-Limited-Dynamic-Programming-Algorithmus an.

1.2.1.4 Centroid Sequence Speech Recognizer

Im Folgenden wird der „Centroid Sequence Speech Recognizer“ beschieben, der im Rahmen dieser Arbeit in Verbindung mit dem Run-Length-Limited-Sprechtempokompensator als „Klangfolgenerkenner“ zur Patentanmeldung gelangte [Hickersberger2004].

Centroid-Sub-Word-Unit-Modell. Die Realisierung des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 erfolgt beim „Centroid Sequence Speech Recognizer“ mittels eines Centroid-Sub-Word-Unit-Modells bestehend aus umgeschalteten Ausgangsvektoren. Es handelt sich um ein einfaches Codebuch. Die Ausgangsvektoren sind die Zentroide der zu repräsentierenden Punktwolken. Der Eingangsvektor wird lediglich für die Abstandsmaßberechnung verwendet.

Das äußerst einfache Sub-Word-Unit-Modell zeigt Abbildung 1-19. Es wurde im Rahmen dieser Arbeit implementiert und evaluiert. Die Zustandsnummer selektiert einen Vektor. Dieser stellt eine Schätzung für den tatsächlich beobachteten Merkmalsvektor dar und eine Distanzfunktion wird berechnet. Im Spezialfall der Verwendung der quadrierten euklidischen Distanz als Distanzfunktion ergibt sich Gleichung 20.

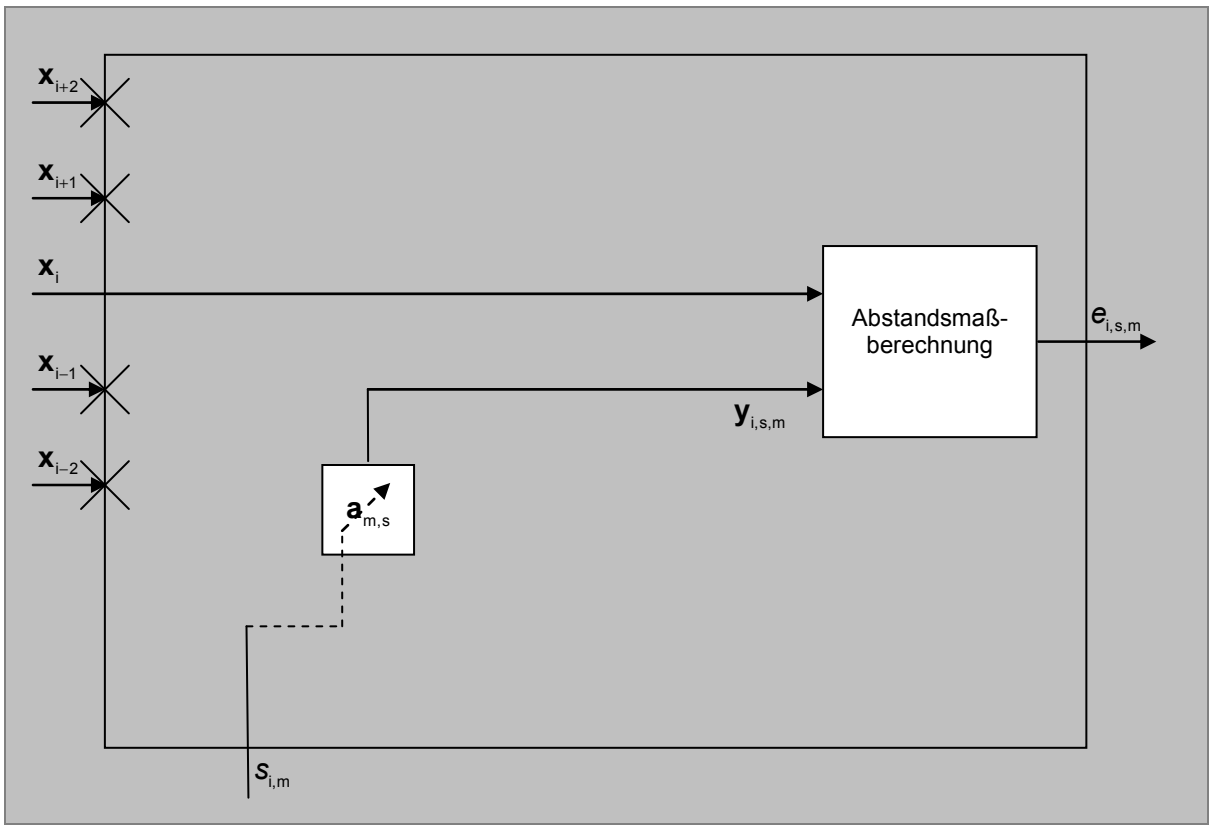

Abbildung 1-19: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Centroid-Sub-Word-Unit-Modell, welches zusammen mit dem Run-Length-Limited-Sprechtempokompensator als Klangfolgenerkenner zur Patentanmeldung gelangte.

$$e_{i,s,m} = \left| \mathbf{x}_i - \mathbf{a}_{m,s} \right|^2 \qquad \textbf{Gleichung 20}$$

Sprechtempokompensation. Die Sprechtempokompensation beim „Centroid Sequence Speech Recognizer“ wird traditionellerweise mittels des relativ einfachen Viterbi-Dynamic-Programming-Sprechtempokompensators bewerkstelligt. Das erfindungsgemäße Verfahren [Hickersberger2004] verwendet hingegen Sprechtempokompensatoren, die die zusätzliche Eigenschaft aufweisen, plausible Zeitdauerintervalle der Zustände zu berücksichtigen. Vorzugsweise wird der Run-Length-Limited-Sprechtempokompensator verwendet.

Minimaler Speicherressourcenbedarf. Im Vergleich zu den anderen beschriebenen Verfahren des Abschnitts 1.2 ist der Speicherressourcenbedarf extrem gering. Jeder Klassenrepräsentant ist in einem zugehörigen Klassenrepräsentantspeicher, vorzugsweise datenkomprimiert abgespeichert. Neben der Anzahl der Zustände (etwa drei bis fünfzehn), wird für jeden der Zustände die Zustandsinformation, bestehend aus einem einzigen Merkmalsvektor, der zulässigen Minimalzeitdauer und der zulässigen Maximalzeitdauer gespeichert.

1.2.1.5 Centroid Sequence Hidden Control Neural Network Speech Recognizer

Centroid-Hidden-Control-Neural-Network-Sub-Word-Unit-Modell. Die Funktionalität des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 wird beim „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ mittels eines „Centroid-Hidden-Control-Neural-Network-Sub-Word-Unit-Modells“ bestehend aus einem Centroid-Sub-Word-Unit-Modell realisiert, wobei ein „Hidden Control Neural Network“ den Restfehler verbessert.

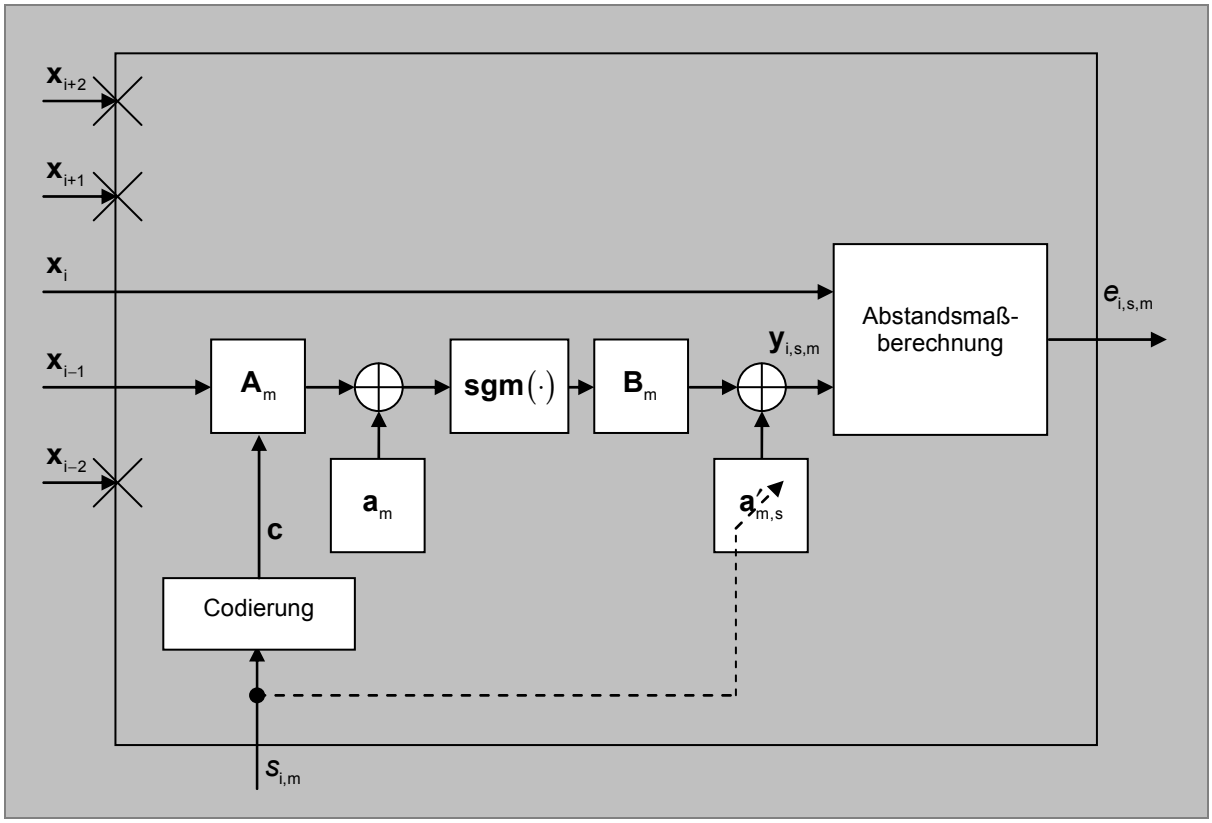

Abbildung 1-20: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Centroid-Hidden-Control-Neural-Network-Sub-Word-Unit-Modell. Das „Hidden Control Neural Network“ minimiert den verbleibenden Restfehler des Centroid-Sub-Word-Unit-Modells.

Abbildung 1-20 zeigt ein weiteres, im Rahmen dieser Arbeit implementiertes und evaluiertes Sub-Word-Unit-Modell. Es handelt sich um ein „Hidden Control Neural Network“. Zum Ausgangsvektor wird im Vergleich zu Abbildung 1-14 auf Seite 42 noch ein zusätzlicher Vektor addiert. Im Spezialfall der Verwendung der quadrierten euklidischen Distanz als Distanzfunktion ergibt sich Gleichung 21.

$$e_{i,s,m} = \left| \mathbf{x}_i - \left(\mathbf{a}'_{m,s} + \mathbf{B}_m \cdot \mathbf{sgm}\left(\mathbf{a}_m + \mathbf{A}_m \cdot \begin{pmatrix} \mathbf{x}_{i-1} \\ \mathbf{c}_{i,m} \end{pmatrix} \right) \right) \right|^2 \qquad \textbf{Gleichung 21}$$

Es ergibt sich eine praktikable Möglichkeit des Trainings: Zuerst kann ein Centroid-Sub-Word-Unit-Modell nach Abschnitt 1.2.1.4 trainiert werden und dann kann das Kolmogorov-Netz zur Schätzung des Restfehlers trainiert werden.

Sprechtempokompensation. Zur Sprechtempokompensation bietet sich beim „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ neben dem Viterbi-Dynamic-Programming-Algorithmus insbesondere der Run-Length-Limited-Dynamic-Programming-Algorithmus an.

1.2.1.6 Hidden Markov Model Speech Recognizer

Gaussian-Mixture-Sub-Word-Unit-Modell. Die Funktionalität des Sub-Word-Unit-Modell-Funktionsblocks nach Abbildung 1-8 auf Seite 32 wird beim typischen „Hidden Markov Model Speech Recognizer“ mittels eines Gaussian-Mixture-Sub-Word-Unit-Modells bestehend aus umgeschalteten „Gaussian Mixtures“ realisiert [Seeger1997] [Deshmukh1999] [Dahmen2001] [Gray2001].

Das Sub-Word-Unit-Modell nach Abbildung 1-21 wurde nicht im Rahmen dieser Arbeit implementiert. Es soll lediglich darauf hinweisen sein, dass sich auch derartige Wortmodelle in den Kontext des Abschnitts 1.2 einordnen lassen.

$$G(\mathbf{x}, \mathbf{z}, \mathbf{Z}) = (2\pi)^{-\frac{K}{2}} \cdot |\mathbf{Z}|^{-\frac{1}{2}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mathbf{z})^T \cdot \mathbf{Z}^{-1} \cdot (\mathbf{x}-\mathbf{z})} \qquad \textbf{Gleichung 22}$$

$$\begin{aligned} GM_{i,s,m} = {} & \hat{\alpha}_{m,s} \cdot G\left(\mathbf{x}_i, \hat{\mathbf{a}}_{m,s}, \hat{\mathbf{A}}_{m,s}\right) + \\ & + \hat{\beta}_{m,s} \cdot G\left(\mathbf{x}_i, \hat{\mathbf{b}}_{m,s}, \hat{\mathbf{B}}_{m,s}\right) + \\ & + \hat{\gamma}_{m,s} \cdot G\left(\mathbf{x}_i, \hat{\mathbf{c}}_{m,s}, \hat{\mathbf{C}}_{m,s}\right) \end{aligned} \qquad \textbf{Gleichung 23}$$

$$e_{i,s,m} = -\ln\left(GM_{i,s,m}\right) \qquad \textbf{Gleichung 24}$$

Es wird eine Linearkombination von Gauss-Verteilungen berechnet und es ist durchaus üblich den negativen Logarithmus auf die Linearkombination anzuwenden, wobei diese Daten als logarithmierte Wahrscheinlichkeiten plus einer Konstante interpretiert werden. Man spricht dabei von „Log-Likelihood-Daten“ und bezeichnet die

Anwendung des Viterbi-Dynamic-Programming-Algorithmus auf die „Log-Likelihood-Daten“ als „Viterbi-Algorithmus“ [Forney1973].

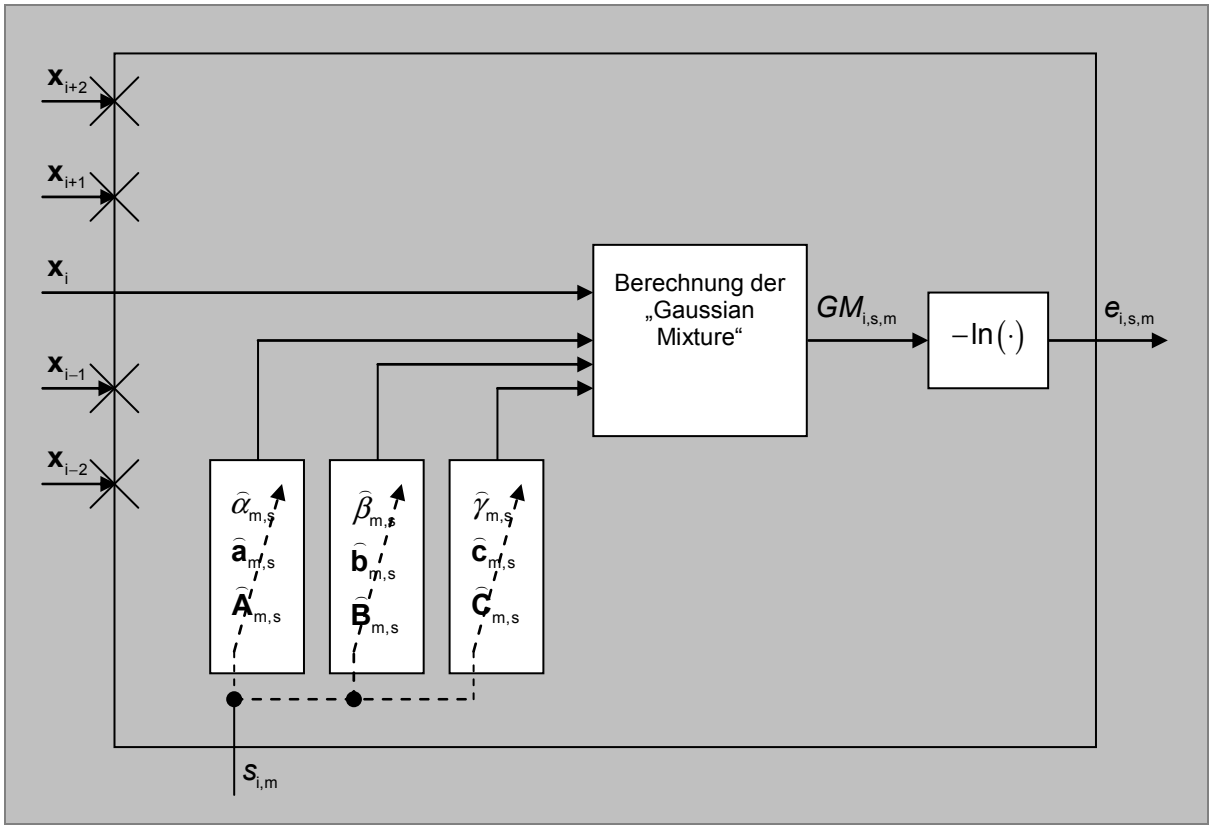

Abbildung 1-21: Der Sub-Word-Unit-Modell-Funktionsblock ausgeführt als Gaussian-Mixture-Sub-Word-Unit-Modell. Diese Ausführungsform ist typisch für „Hidden Markov Model Speech Recognizer“.

Eine gute Einführung bezüglich „Hidden Markov Model Speech Recognizer“ finden sich in der Literatur [Deshmukh1999]. Zu erwähnen sind natürlich die bekannten Literaturstellen [Rabiner1986] [Rabiner1989] [Rabiner1989a]. Hidden-Markov-Modelle werden in Zusammenhang mit Neuronalen Netzen von verschiedenen Autoren beschrieben und eingesetzt [Bourlard1990] [Bridle1992] [Lee2002] [Trentin2003]. Erfolgreiche Neuentwicklungen finden sich in [Do2011] [Do2011a] [Dahl2012] [Jaitly2012]. Auch zur Segmentierung von Musikdaten finden sie Verwendung [Raphael1999].

Sprechtempokompensation. Zur Sprechtempokompensation wird beim „Hidden Markov Model Speech Recognizer“ typischerweise der Viterbi-Algorithmus verwendet. Verschiedene Alternativen zum Viterbi-Algorithmus finden sich in der Literatur [Merhav1991]. Der Run-Length-Limited-Dynamic-Programming-Algorithmus ist vorteilhaft anwendbar.

Viterbi-Sprechtempokompensator. Beim Viterbi-Algorithmus [Forney1973] [Rabiner1989a] handelt es sich um ein statistisches Schätzverfahren[16], nämlich um einen Maximum-a-posteriori-Schätzer für die „Hidden Control“, das heißt für die nach außen hin unbeobachtete Zustandsfolge der Zustände einer Markov-Kette [Eier1994] [Eier1999], welche einem Hidden-Markov-Zufallsprozess zugrunde liegt.

Wir betrachten zwei Zufallsgrößen, die beobachtete endliche Merkmalsvektorfolge, welche als Matrix $\mathbf{X}$ notiert sei und die unbeobachtete Zustandsfolge[17] des Hidden-Markov-Zufallsprozesses nach Abbildung 1-12 auf Seite 37, welche „Hidden Control“ genannt wird, und als Vektor $\mathbf{s}_m$ notiert sei, wobei der Index *m* die Wortmodellnummer bedeutet. Man stelle sich vor, das beobachtete Schallmuster bestünde aus *I* Merkmalsvektoren und demnach bestünde auch die „Hidden Control“ aus *I* Zustandsnummern. Die A-priori-Wahrscheinlichkeitsverteilung der erlaubten Zustandsfolgen sei mit $P(\mathbf{s}_m)$ bezeichnet. Die A-posteriori-Wahrscheinlichkeitsverteilung der erlaubten Zustandsfolgen unter Berücksichtigung der Beobachtung $\mathbf{X}$ sei mit $P(\mathbf{s}_m \mid \mathbf{X})$ bezeichnet. Gleichung 25 notiert den Zusammenhang mit der Verbundwahrscheinlichkeitsverteilung $P(\mathbf{s}_m, \mathbf{X})$ in Form des Bayes-Theorems.

$$P(\mathbf{s}_m, \mathbf{X}) = p(\mathbf{X} \mid \mathbf{s}_m) \cdot P(\mathbf{s}_m) = P(\mathbf{s}_m \mid \mathbf{X}) \cdot p(\mathbf{X}) \qquad \textbf{Gleichung 25}$$

Viterbi-Sprechtempokompensator als Maximum-a-posteriori-Schätzer. Bestimmt wird der optimale Parameter $\mathbf{s}_m^{\text{optimum (MAP)}}$, sodass die Wahrscheinlichkeit der A-posteriori-Wahrscheinlichkeitsverteilung $P(\mathbf{s}_m \mid \mathbf{X})$ maximal ist.

$$\mathbf{s}_m^{\text{optimum (MAP)}} = \arg\max_{\mathbf{s}_m} P(\mathbf{s}_m \mid \mathbf{X}) \qquad \textbf{Gleichung 26}$$

Die Berechnung erfolgt durch Anwendung von Gleichung 25. Der Faktor $p(\mathbf{x})$ ändert das Ergebnis der Optimierung nicht und kann daher weggelassen werden:

$$\mathbf{s}_m^{\text{optimum (MAP)}} = \arg\max_{\mathbf{s}_m} \frac{P(\mathbf{X} \mid \mathbf{s}_m) \cdot P(\mathbf{s}_m)}{p(\mathbf{X})} = \arg\max_{\mathbf{s}_m} \left(P(\mathbf{X} \mid \mathbf{s}_m) \cdot P(\mathbf{s}_m) \right) \qquad \textbf{Gleichung 27}$$

---

[16] Unter „statistischen Schätzverfahren“ versteht man Methoden zur Schätzung der Parameter einer Verteilung einer Zufallsgröße basierend auf der Entnahme von Stichproben. Der Typ der Verteilung kann bei manchen Aufgabenstellungen bereits festgelegt sein, wobei nur noch die freien Parameter der Verteilung geschätzt werden. Andererseits trifft man auch auf Aufgabenstellungen, bei denen der zu schätzende Parameter als Index eine von mehreren infrage kommenden Verteilungen auswählt.

[17] Es handelt sich um die Zustandsfolge eines Hidden-Markov-Zufallsprozesses, welcher Merkmalsvektoren erzeugt.

Der Term rechts in Gleichung 27 entspricht der Verbundwahrscheinlichkeit, notiert in Gleichung 28.

$$\mathbf{s}_{\mathrm{m}}^{\text{optimum (MAP)}} = \arg\max_{\mathbf{s}_{\mathrm{m}}} p\left(\mathbf{X}, \mathbf{s}_{\mathrm{m}}\right) \qquad \textbf{Gleichung 28}$$

Diese Gleichung bildet die im Allgemeinen die Grundlage zur Implementierung des Algorithmus: Die Suche des globalen Minimums erfolgt in optimaler Weise durch den Viterbi-Dynamic-Programming-Algorithmus.

Viterbi-Sprechtempokompensator mit Hidden-Markov-Zufallsprozessen erster Ordnung. Wenn man annimmt, dass die beobachtete Folge von einem Hidden-Markov-Zufallsprozess erster Ordnung erzeugt wird[18], so wird dabei erstens angenommen, dass es sich bei der zugrunde liegenden Zustandsfolge „Hidden Control“ um eine Markov-Kette erster Ordnung handelt. Das heißt, dass die Wahrscheinlichkeit eines bestimmten Zustandes zu einem bestimmten Zeitpunkt nur vom Zustand zum vorhergehenden Zeitpunkt abhängt. Die Elemente der Übergangswahrscheinlichkeitsmatrix können mittels einer Bigrammstatistik geschätzt werden. Die Wahrscheinlichkeit einer bestimmten „Hidden Control“ $\mathbf{s}_{\mathrm{m}}$ kann somit nach Gleichung 29 berechnet werden.

$$P\left(\mathbf{s}_{\mathrm{m}}\right) = \prod_{\mathrm{i}=1}^{\mathrm{I}} P\left(s_{\mathrm{i,m}} \mid s_{\mathrm{i-1,m}}\right) \qquad \textbf{Gleichung 29}$$

Zweitens wird dabei angenommen, dass jedem Zustand eine Emissionswahrscheinlichkeitsdichtefunktion zugeordnet ist, die nicht vom Zeitpunkt abhängt.

$$p\left(\mathbf{X} \mid \mathbf{s}_{\mathrm{m}}\right) = \prod_{\mathrm{i}=1}^{\mathrm{I}} p\left(\mathbf{x}_{\mathrm{i}} \mid s_{\mathrm{i,m}}\right) \qquad \textbf{Gleichung 30}$$

Multipliziert man nun jeweils die Linksterme und die Rechtsterme von Gleichung 29 und Gleichung 30, so erhält man nach Anwendung des Bayes-Theorems:

$$p\left(\mathbf{X}, \mathbf{s}_{\mathrm{m}}\right) = \prod_{\mathrm{i}=1}^{\mathrm{I}} P\left(\mathbf{x}_{\mathrm{i}} \mid s_{\mathrm{i,m}}\right) \cdot P\left(s_{\mathrm{i,m}} \mid s_{\mathrm{i-1,m}}\right) \qquad \textbf{Gleichung 31}$$

[18] Wenn auch diese Annahme in vielen Fällen gerechtfertigt sein mag, so ist dennoch zu bedenken, dass sie gerade bei Sprachsignalen offensichtlich nicht ausreichend zutrifft. Dieser Umstand ist ein wichtiges Argument gegen die klassische Implementierung des „Hidden Markov Model Speech Recognizer“[Rabiner1989a].

Denn es gilt ja:

$$P(\mathbf{X},\mathbf{s}_m) = P(\mathbf{s}_m \mid \mathbf{X}) \cdot p(\mathbf{X}) = p(\mathbf{X} \mid \mathbf{s}_m) \cdot P(\mathbf{s}_m) \qquad \textbf{Gleichung 32}$$

Der Viterbi-Algorithmus dient zur Bestimmung jener Folge von Zuständen, der „Hidden Control“, aus der Menge gültiger Folgen, welche bei gegebener Folge von Beobachtungen die wahrscheinlichste ist.

$$\mathbf{s}_m = \underset{\mathbf{s}_m}{\arg\max} \prod_{i=1}^{I} p(\mathbf{x}_i \mid s_{i,m}) \cdot P(s_{i,m} \mid s_{i-1,m}) \qquad \textbf{Gleichung 33}$$

Zur Suche des globalen Optimums wird im Allgemeinen vorteilhaft der Viterbi-Dynamic-Programming-Algorithmus angewendet, welcher eine Umformung des Rechtsterms in eine Summe voraussetzt:

$$\mathbf{s}_m = \underset{\mathbf{s}_m}{\arg\max} \sum_{i=1}^{I} \log\left(p(\mathbf{x}_i \mid s_{i,m}) \cdot P(s_{i,m} \mid s_{i-1,m})\right) \qquad \textbf{Gleichung 34}$$

Dabei werden jedem Zweig in Abbildung 1-12 auf Seite 37 bestimmte Zweigkosten zugeordnet, die sich im allgemeinen Fall aus Zustandskosten und Übergangskosten[19] zusammensetzen. Die Zweigkosten werden durch Logarithmierung des Produkts der jeweiligen Zustandsübergangswahrscheinlichkeit mit der jeweiligen Emissionswahrscheinlichkeitsdichte des tatsächlich beobachteten Merkmalsvektors bestimmt, da der Viterbi-Dynamic-Programming-Algorithmus ja die Summe der Zweigkosten optimiert.

Viterbi-Sprechtempokompensator mit Hidden-Markov-Prozess zweiter Ordnung. Wenn man annimmt, dass die beobachtete Folge von einem Hidden-Markov-Zufallsprozess zweiter Ordnung erzeugt wird, so wird dabei erstens angenommen, dass es sich bei der zugrunde liegenden Zustandsfolge „Hidden Control“ um eine Markov-Kette zweiter Ordnung handelt. Das heißt, dass die Wahrscheinlichkeit eines bestimmten Zustandes zu einem bestimmten Zeitpunkt nur vom Zustand zum vorhergehenden Zeitpunkt und vom Zustand zum Zeitpunkt vor dem vorhergehenden Zeitpunkt abhängt. Die Übergangswahrscheinlichkeiten können mittels einer Trigrammstatistik[20] geschätzt werden. Die Wahrscheinlichkeit einer bestimmten „Hidden Control“ $\mathbf{s}_m$ kann somit wiederum als Produkt berechnet werden. Zweitens wird angenommen, dass jedem Zustand eine Emissionswahrscheinlichkeitsdichtefunktion zugeordnet ist, die nicht vom Zeitpunkt abhängt.

---

[19] Man trifft durchaus häufig auf Anwendungen, bei denen die Übergangskosten null gesetzt sind.

[20] Eine Verallgemeinerung der Idee, die Ordnung der Markov-Zufallprozesse zu erhöhen ist, die Wahrscheinlichkeiten der jeweiligen Zustände zu den jeweiligen Zeitpunkten mittels einer Funktion zu beschreiben, die neben den Zuständen zu den Vorgängerzeitpunkten auch völlig andere Parameter aufweist.

Zur Berechnung der Zustandsfolge „Hidden Control“, die der gegebenen Folge von Beobachtungen am wahrscheinlichsten zugrunde lag, kann wiederum der Viterbi-Algorithmus verwendet werden, wenn die möglichen Zustände der Zustandsfolge in entsprechend viele Subzustände aufgespaltet werden, deren Aufgabe darin besteht, zu codieren, welcher Zustand zuvor eingetreten ist [Fink2000]. Diese Vorgangsweise lässt sich leicht auf Hidden-Markov-Zufallsprozesse höherer Ordnung verallgemeinern.

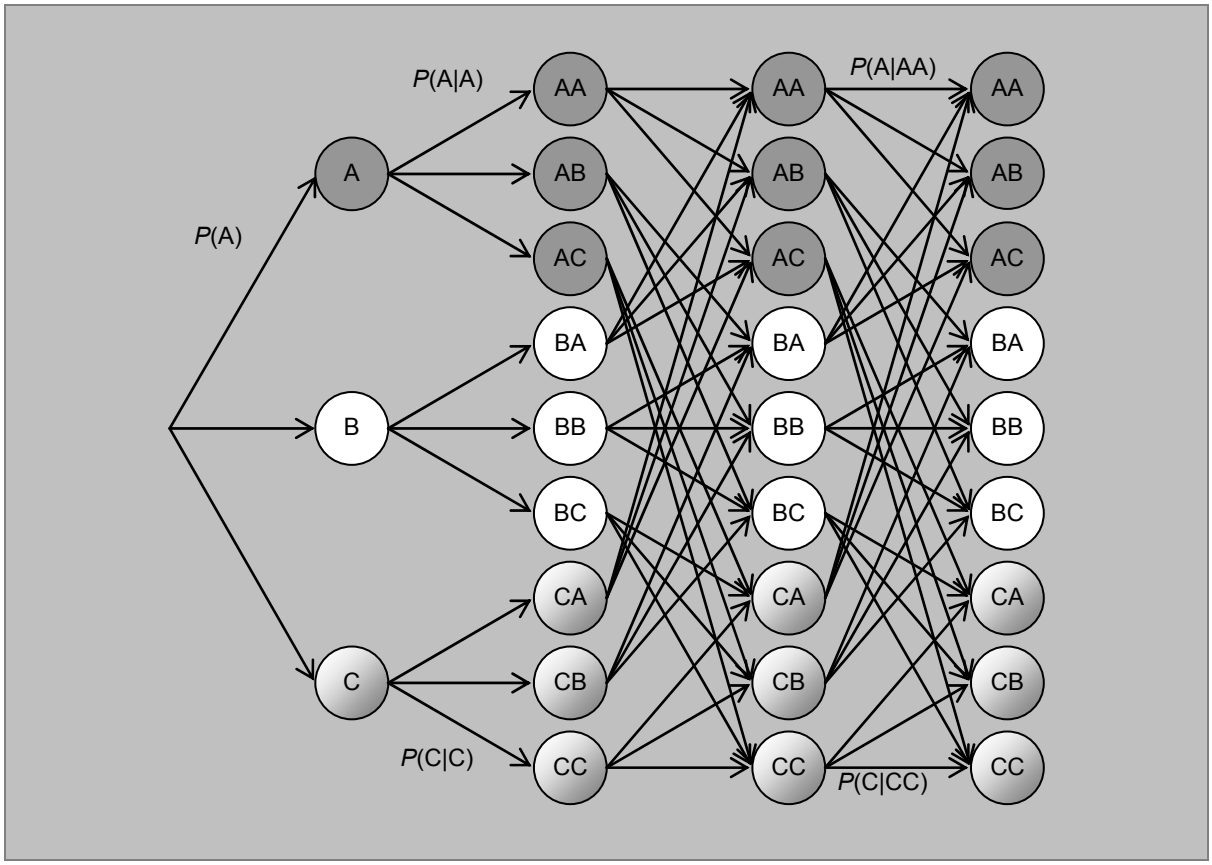

Abbildung 1-22: Berechnung der Wahrscheinlichkeit der Zustandsfolge „Hidden Control“ mit dem Viterbi-Algorithmus unter Annahme eines Hidden-Markov-Zufallsprozesses zweiter Ordnung.

1.2.2 Run-Length-Limited-Sprechtempokompensator

Im Folgenden wird der Run-Length-Limited-Sprechtempokompensator beschrieben, welcher erstmals in der Patentschrift [Hickersberger2004] veröffentlicht wurde. Die Anwendungsmöglichkeiten des Run-Length-Limited-Dynamic-Programming-Algorithmus sind keineswegs auf das Gebiet der Sprechtempokompensation beschränkt.

Search-Space-Reduction-Verfahren. Es handelt sich beim beschriebenen Algorithmus um ein spezielles Search-Space-Reduction-Verfahren. Die Grundidee bei Search-Space-Reduction-Verfahren besteht allgemein darin, nicht alle möglichen Pfade im Trellis-Diagramm nach Abbildung 1-12 auf Seite 37 als Lösungen zuzulassen, das heißt, die Lösung nur in der Teilmenge zulässiger Pfade zu suchen. Üblicherweise wird der Einsatz von Search-Space-Reduction-Verfahren mit dem in diesem Zusammenhang eingesparten Rechenaufwand begründet [Ohno1994] [Schoknecht1999]. Der hier beschriebene Run-Length-Limited-Dynamic-Programming-Algorithmus hat diese Eigenschaft allerdings nicht. Im Gegenteil, die benötigte Rechenleistung ist höher als beim herkömmlichen Viterbi-Dynamic-Programming-Algorithmus.

Lauflängenbedingungen zur Suchraumeinschränkung. Im speziellen unterscheidet sich der Run-Length-Limited-Sprechtempokompensator vom üblichen Viterbi-Dynamic-Programming-Sprechtempokompensator durch die Eigenschaft, den Suchraum der Pfade durch bestimmte zusätzliche Lauflängenbedingungen für die einzelnen Zustände einzuschränken. Pfade, die die Lauflängenbedingungen erfüllen, werden im Folgenden als „plausibel“ bezeichnet.

Verbesserung der Klassifikationsfähigkeit. Der wichtige Vorteil des hier beschriebenen Verfahrens besteht darin, dass sich die Klassifikationsfähigkeit des Classifier-Funktionsblocks in Abbildung 1-1 auf Seite 16 erhöht. Dies liegt daran, dass durch die Beschränkung der Lösungssuche auf „plausible“ Pfade den Wortmodellen Flexibilität bezüglich der Sprechtempokompensation genommen wird, die sie zur Sprechtempokompensation der Schallmuster, die sie repräsentieren sollen, gar nicht benötigen. Klassenfremde Schallmuster erreichen jedoch schlechtere Wortmodelldistanzen, weil bei diesen Schallmustern oftmals „unplausible“ Pfade die beste Wortmodelldistanz liefern würden, die Lösung aber nur unter den „plausiblen“ Pfaden gesucht wird. Dadurch werden die Chancen erhöht, dass sich die korrekten Wortmodelle durchsetzen.

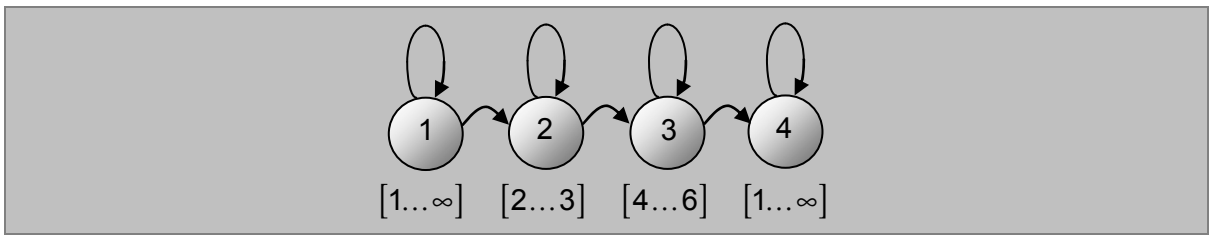

Abbildung 1-23: Ein Left-to-Right-Zustandsmodell mit zusätzlichen Intervallen für die Lauflängenbedingungen.

Bedingungen für einen gültigen Pfad. Das Deltamerkmal zum üblichen Viterbi-Dynamic-Programming-Sprechtempokompensator mit Left-to-Right-Zustandsmodell besteht darin, dass der Run-Length-Limited-Sprechtempokompensator den optimalen Pfad durch das Trellis-Diagramm nach Abbildung 1-12 auf Seite 37 nur unter den Pfaden bestimmt, die in Abbildung 1-23 auf Seite 58 beispielsweise festgelegten Intervallbedingungen für die Zustandslauflängen erfüllen. Es gelten weiters folgende Bedingungen: Der erste Zustand ist dem ersten Zeitpunkt zugeordnet. Der letzte Zustand ist dem letzten Zeitpunkt zugeordnet. Die Zustände folgen aufeinander. Kein Zustand darf ausgelassen werden. Die Lauflängen der Zustände müssen innerhalb der plausiblen Lauflängenintervalle liegen. Unter diesen Bedingungen wählt der Run-Length-Limited-Sprechtempokompensator die Lauflängen der Zustände gemäß dem Optimum der Pfadkostenfunktion. Im praktischen Einsatz werden die Zustandslauflängenbedingungen vorab mit statistischen Methoden geschätzt, sodass sie Eigenschaften der repräsentierten Klasse widerspiegeln.

Run-Length-Limited-Dynamic-Programming-Algorithmus. Wie kann also nun die optimale Lösung unter den zusätzlichen Bedingungen gefunden werden? Der Run-Length-Limited-Sprechtempokompensator berücksichtigt die Lauflängenbedingungen für die einzelnen Zustände, in dem das Zustandsmodell nach einem bestimmten Verfahren in eines ohne Lauflängenbedingungen transformiert wird. Sodann wird der übliche Viterbi-Dynamic-Programming-Algorithmus angewendet und das Resultat auf das ursprüngliche Zustandsmodell zurück übertragen.

Transformation des Zustandsmodells. Die Zustände des transformierten Zustandsmodells werden im Folgenden als Subzustände bezeichnet. Jedem Subzustand ist eindeutig aber im Allgemeinen nicht umkehrbar ein Zustand zugeordnet, wie Abbildung 1-24 auf Seite 59 zeigt. Jeder Zustand wird in genau so viele aufeinander folgende Subzustände aufgeteilt, wie es seiner Maximallauflänge entspricht. Die Minimallauflängenbedingung wird dadurch berücksichtigt, dass ab der Minimallauflänge Zustandsübergänge zum ersten Subzustand des nächsten Zustandes möglich sind. Zunächst erfolgt also eine bestimmte Anzahl von Subzuständen, welche nacheinander durchlaufen werden müssen, ohne dass andere Übergänge erlaubt sind. Diese Anzahl ist gleich der minimalen Lauflänge minus eins. Darauf folgt eine bestimmte Anzahl von Subzuständen, von welchen aus jeweils Übergänge auf den ersten Subzustand des nächsten Zustands möglich sind. Diese Anzahl ist gleich der maximalen Lauflänge minus der minimalen

Lauflänge plus eins. Daher ergibt sich insgesamt als Anzahl der Subzustände die maximale Lauflänge. Beispielsweise müssen bei der Randbedingung für den zweiten Zustand nach Abbildung 1-24 mindestens zwei Subzustände auftreten. Ein dritter Subzustand kann auftreten, aber es kann auch schon vom zweiten Subzustand auf den ersten Subzustand des dritten Zustands gesprungen werden. Analoges gilt für die Randbedingung des dritten Zustands.

Vorteilhafte Wahl der Bedingungen. Vorzugsweise werden Minimallauflängen größer null verwendet, sodass kein Zustand ausgelassen werden kann. Für den ersten und letzten Zustand werden außerdem beliebig lange Lauflängen zugelassen, da diese beiden Zustände hauptsächlich die Hintergrundgeräusche vor und nach dem interessierenden Schallereignis repräsentieren. In diesem Fall wird zur Realisierung der Maximallauflängenbedingung ein einziger Subzustand verwendet, von dem aus Übergänge auf sich selbst möglich sind. Siehe den ersten und vierten Zustand in Abbildung 1-24 auf Seite 59. Die Anzahl der Subzustände für diese Zustände ist gleich der minimalen Lauflänge, nämlich eins.

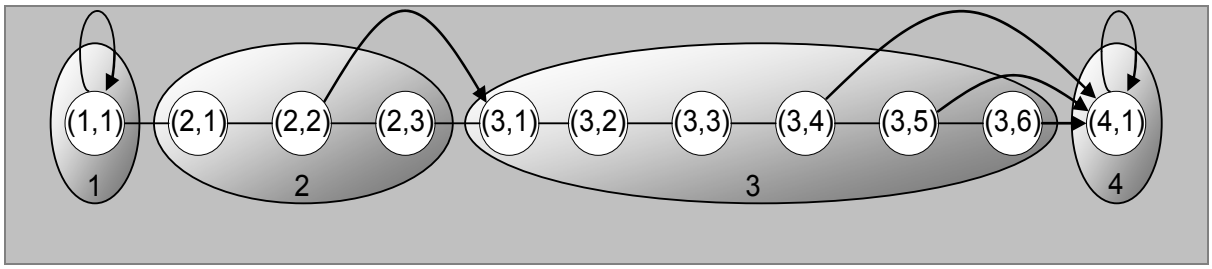

Abbildung 1-24: Transformation des Zustandsmodells beim Run-Length-Limited-Sprechtempokompensator.

Abbildung 1-24 zeigt ein transformiertes Zustandsmodell, wie es beim Run-Length-Limited-Sprechtempokompensator verwendet wird. Aus den abgespeicherten, zulässigen Zeitdauerintervallen der Zustände wird dieses Zustandsmodell gebildet. In Abbildung 1-24 ist ein Zustandsmodell für folgende Lauflängenbedingungen dargestellt: Für den ersten Zustand gilt „1 bis unendlich“. Für den zweiten Zustand gilt „2 bis 3“. Für den dritten Zustand gilt „4 bis 6“. Für den vierten Zustand gilt „1 bis unendlich“.

Darstellung im Trellis-Diagramm. Das neue Trellis-Diagramm, mittels dessen die zusätzlichen Lauflängenbedingungen berücksichtigt werden, zeigt Abbildung 1-25. Die neue Kostenmatrix wird aus der ursprünglichen Kostenmatrix dadurch gebildet, dass die Subzustände die Kosten der Zustände erben, denen sie zugeordnet sind.

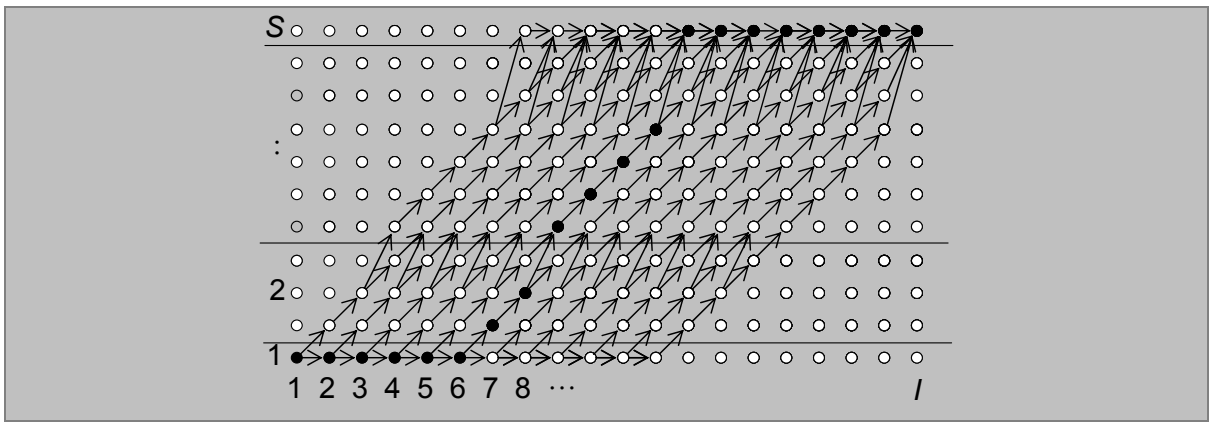

Abbildung 1-25: Trellis-Diagramm eines Left-to-Right-Zustandsmodells mit zusätzlichen Lauflängenbedingungen, deren Einhaltung sichergestellt wird, indem die Zustände in Subzustände aufgespaltet werden.

Bestimmung des optimalen Pfades. Der optimale Pfad durch das transformierte Zustandsmodell wird mittels des üblichen Viterbi-Dynamic-Programming-Algorithmus anhand des neuen Trellis-Diagramms bestimmt. Die Lösung wird schließlich wieder auf das ursprüngliche, einfache Left-to-Right-Zustandsmodell zurück übertragen. Damit erhält man die optimale Lösung unter den Lauflängenbedingungen. Für nähere Details sei auf die Patentschrift [Hickersberger2004] verwiesen.

1.3 Überlegungen bezüglich Benutzerzufriedenheit

In der Fachliteratur wird zum Vergleich unterschiedlicher Sprachsteuerungen fast ausschließlich die Verwechslungsfehlerrate herangezogen. Dabei handelt es sich um den Anteil der Keyword-Schallmuster, die eine falsche Systemreaktion hervorrufen, an der Keyword-Schallmusteranzahl. Benutzer von Sprachsteuerungen ärgern sich jedoch im Allgemeinen über drei unterschiedliche Arten von Fehlern: Verwechslungsfehler „Confusion Errors“, Falschakzeptanzfehler „Acceptance Errors“ sowie Falschrückweisungsfehler „Rejection Errors“. Zur aussagekräftigen Bewertung einer Sprachsteuerung sind demnach neben der Verwechslungsfehlerrate zumindest die beiden anderen Fehlerraten notwendig [Tschirk2001]: Die Falschrückweisungsrate, welche den Anteil der Keyword-Schallmuster, die keine Systemreaktion hervorrufen an der Keyword-Schallmusteranzahl angibt, und die Falschakzeptanzrate, welche den Anteil der Non-Keyword-Schallmuster, die fälschlicherweise eine Systemreaktion hervorrufen an der Non-Keyword-Schallmusteranzahl angibt.

Man könnte, zum Zwecke der Modellbildung, die Benutzerzufriedenheit beziehungsweise Unzufriedenheit als Funktion dieser drei Fehlerraten ansetzen. Siehe hierzu Gleichung 42 auf Seite 63. Allerdings stellen sich die Möglichkeiten der Optimierung einer solchen Funktion mittels Justierung konstanter Betriebsparameter der Sprachsteuerung in der Regel als bescheiden heraus. Es wird üblicherweise versucht, die Verwechslungsfehlerrate so weit wie möglich zu senken und die Falschakzeptanzrate gegen die Falschrückweisungsrate abzuwägen[21].

1.3.1 Definition von Fehlerraten

Während des Betriebs ist eine Sprachsteuerung mit zwei verschiedenen Arten von Schallereignissen konfrontiert, welche zur Analyse gelangen: Keyword-Schallmuster und Non-Keyword-Schallmuster. Unter Non-Keyword-Schallmustern werden im Folgenden sowohl vokabularfremde Wörter als auch aufgeschnappte Umgebungsgeräusche verstanden. Zählt man innerhalb eines gewissen Beobachtungszeitraumes die Systemreaktionen gemäß Tabelle 2, so erhält man die beobachtete Häufigkeitsverteilung, welche natürlich von verschiedensten Testbedingungen und Umgebungsbedingungen abhängt. Um verschiedene Sprachsteuerungen zu vergleichen, werden sie daher mit den gleichen Schallmustern, der sogenannten „Testdatenbank“, konfrontiert und die Systemreaktionen

[21] Einen speziellen Kompromiss stellt die „Equal-Error-Rates-Einstellung“ dar. Unter Umständen kann eine weitere Steigerung der Benutzerzufriedenheit durch dialogsituationsabhängige Veränderung des Kompromisses zwischen der Falschakzeptanzrate und der Falschrückweisungsrate erreicht werden.

werden protokolliert[22]. Die beobachteten Häufigkeitsverteilungen lassen sich durch verschiedene Kennzahlen charakterisieren, welche im Folgenden definiert werden.

Im Folgenden sei zur Vereinfachung der Modellbildung angenommen, dass unterschiedliche Schallmusterklassen stets unterschiedliche Systemreaktionen auslösen. Andernfalls würden bestimmte Verwechslungsfehler dem Benutzer nicht auffallen.

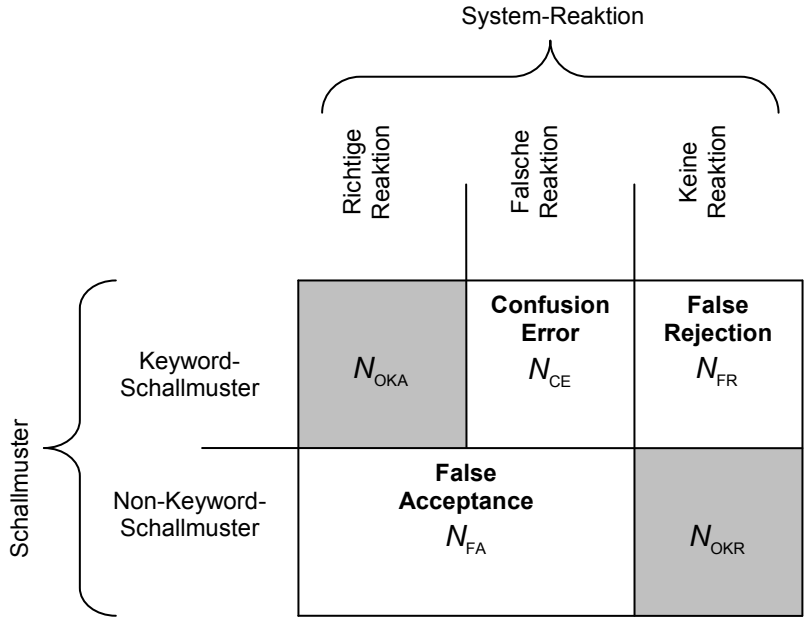

Tabelle 2: Zur Definition üblicher Fehlerraten bei Betrachtung einer Sprachsteuerung als Blackbox.

Die Anzahl der Keyword-Schallmuster ergibt sich als Zeilensumme der ersten Zeile in Tabelle 2. Die Anzahl der Non-Keyword-Schallmuster ergibt sich als Zeilensumme der zweiten Zeile in Tabelle 2.

$$N_{Keyw} = N_{OKA} + N_{CE} + N_{FR} \qquad \textbf{Gleichung 35}$$

$$N_{Non\text{-}Keyw} = N_{FA} + N_{OKR} \qquad \textbf{Gleichung 36}$$

Die gesamte Anzahl der dem System präsentierten Schallmuster ergibt sich aus der Summe der Anzahl der Keyword-Schallmuster und der Anzahl der Non-Keyword-

[22] Ein wichtiger Stolperstein ist der folgende: Wird eine Testdatenbank während der Entwicklung eines Spracherkennungsalgorithmus mehrmals verwendet – was sich aus Mangel an Datenbanken im Allgemeinen kaum vermeiden lässt – tritt oftmals eine unbeabsichtigte Random-Search-Optimierung des zu entwickelnden Algorithmus beziehungsweise dessen Parametersatzes auf. Das heißt, der Algorithmus wird im Laufe der Zeit unbeabsichtigt so entwickelt, dass er speziell bezüglich der verwendeten Testdatenbank gute Fehlerraten liefert, welche sich aber nicht in die Praxis übertragen.

Schallmuster. Im Folgenden wird das Verhältnis dieser beiden Anzahlen definiert. In diesem Verhältnis finden wichtige Umgebungsbedingungen ihren Niederschlag[23].

$$N = N_{\text{Keyw}} + N_{\text{Non-Keyw}}$$ **Gleichung 37**

$$\mu = \frac{N_{\text{Non-Keyw}}}{N_{\text{Keyw}}}$$ **Gleichung 38**

Es werden nun folgende für den Benutzer relevante Fehlerraten definiert [Tschirk2001]: Die Verwechslungsfehlerrate *CER* nach Gleichung 39, die Falschakzeptanzrate *FAR* nach Gleichung 40 sowie die Falschrückweisungsrate *FRR* nach Gleichung 41. Die Verwechslungsfehlerrate ist definiert als Anteil der falsch klassifizierten, emittierten[24] Keyword-Schallmuster an der Anzahl der dem System dargebotenen Keyword-Schallmustern. Die Falschakzeptanzrate ist definiert als Anteil der emittierten Non-Keyword-Schallmuster an der Anzahl der dem System dargebotenen Non-Keyword-Schallmustern[25]. Die Falschrückweisungsrate ist definiert als der Anteil der nicht emittierten Keyword-Schallmuster an der Anzahl der dem System dargebotenen Keyword-Schallmuster.

$$CER = \frac{N_{\text{CE}}}{N_{\text{Keyw}}}$$ **Gleichung 39**

$$FAR = \frac{N_{\text{FA}}}{N_{\text{Non-Keyw}}} = \frac{N_{\text{FA}}}{N_{\text{Keyw}}} \cdot \mu^{-1}$$ **Gleichung 40**

$$FRR = \frac{N_{\text{FR}}}{N_{\text{Keyw}}}$$ **Gleichung 41**

Die Unzufriedenheit des Benutzers lässt sich beispielsweise nach Gleichung 42 modellieren, wobei angenommen wird, dass sich der Benutzer über jeden Fehler des Systems gleichermaßen ärgert.

$$U = \frac{N_{\text{CE}} + N_{\text{FA}} + N_{\text{FR}}}{N} = \frac{CER + FRR + FAR \cdot \mu}{1 + \mu}$$ **Gleichung 42**

---

[23] Auf einer lauten Straße könnten beispielsweise wesentlich mehr Störgeräusche zur Analyse gelangen, als in einem ruhigen Innenraum.

[24] Unter „Emission“ ist hier eine Systemreaktion zu verstehen.

[25] Da die Vielfalt an zu erwartenden lautlichen Ereignissen aufgrund der Einsatzumgebung stark variieren kann, ist die Falschakzeptanzrate stets in Zusammenhang mit einer Einsatzumgebung zu sehen.

Genau genommen wird die Unzufriedenheit des Benutzers in der Praxis natürlich von verschiedensten Faktoren beeinflusst, im speziellen auch von der Dialogsituation: Es ist durchaus denkbar, dass sich der Benutzer, je nachdem, was mit den Steuerbefehlen bewirkt werden soll, über die eine Fehlerart mehr als über die andere ärgert. Ein Ansatz der Unzufriedenheit als Linearkombination der drei Fehlerraten wird herangezogen, um Sprachsteuerungen während des laufenden Betriebs bezüglich der jeweiligen Dialogsituation nachzujustieren [Tschirk2004].

Die in Gleichung 42 definierte Unzufriedenheit U hängt wesentlich vom Verhältnis der Anzahl der Keyword-Schallmuster zur Anzahl der Non-Keyword-Schallmuster ab, das heißt, sie steigt beispielsweise, wenn unter ungünstigen Umgebungsbedingungen viele Umgebungsgeräusche zur Analyse gelangen.

Die Vorgangsweise bei der experimentellen Bestimmung der Fehlerraten stellt sich allgemein wie folgt dar: Die Aufnahmen der Testdatenbank werden der Sprachsteuerung präsentiert. Jede Aufnahme, die zu einem Schallmuster verarbeitet wird, führt zu einem, der in Tabelle 2 gezeigten Zustände. Die Testdatenbank sollte möglichst repräsentativ für die Einsatzbedingungen der Sprachsteuerung sein[26]. Es sollten daher Keyword-Schallmuster und Non-Keyword-Schallmuster, also beispielsweise auch Geräusche, beinhaltet sein. Die gewonnenen Kennzahlen $N_{OKA}, N_{OKR}, N_{CE}, N_{FA}, N_{FR}$ repräsentieren die beobachtete Qualität der Sprachsteuerung für die analysierten Schallmuster[27].

Für theoretische Untersuchungen sowie zur Fehlersuche hat es sich bewährt, eine weitere Größe zu definieren, welche das Verhalten des Classifier-Funktionsblocks in Abbildung 1-1 auf Seite 16 beschreibt. Dieser ist prinzipiell stets in der Lage, eine Reaktion in Form einer Rank-List beziehungsweise eines Gewinnerwortmodells zu liefern, auch wenn die Reaktion nach außen hin nicht sichtbar wird, weil das Schallereignis zum Beispiel für ein Geräusch gehalten wird[28]. Beobachtet man ausschließlich die Reaktion des Classifier-Funktionsblocks, so erhält man die Verwechslungsfehlerrate CER' eines niemals zurückweisenden Systems nach Gleichung 43.

$$CER' = \frac{N'_{CE}}{N_{Keyw}} = \frac{N_{CE} + N_{FR,falsch}}{N_{Keyw}} = \frac{N_{CE} + N_{FR} - N_{FR,richtig}}{N_{Keyw}} \qquad \textbf{Gleichung 43}$$

[26] Schallmuster können beispielsweise zum Aufbau der Testdatenbank während eines tatsächlichen Einsatzes mitprotokolliert werden.

[27] Genau genommen bewirkt das Verhalten des Endpoint-Detector-Funktionsblocks in Abbildung 1-2, dass nicht alle Schallereignisse zur Analyse gelangen (zum Beispiel wenn das Schallereignis zu kurz ist, um ein Keyword-Schallmuster sein zu können), das heißt, die durch die Kennzahlen repräsentierten Schallmuster bilden nur eine Teilmenge der eventuell von einem gut hörenden Benutzer beobachteten Schallereignisse.

[28] Wenn keine Systemreaktion erfolgen soll, wird die konkrete Reaktion des Classifier-Funktionsblocks, je nach Implementierung, unter Umständen gar nicht berechnet.

Die Anzahl der direkt am Ausgang des Classifier-Funktionsblocks beobachteten Verwechslungsfehler N'_{CE} entspricht der Anzahl N_{CE} vermehrt um den Anteil an den Rückweisungsfehlern, die nicht korrekt klassifiziert werden $N_{FR,falsch}$.

Bei genauerer Systemanalyse empfiehlt es sich, nicht nur die Kennzahlen $N_{OKA}, N_{OKR}, N_{CE}, N_{FA}, N_{FR}$ heranzuziehen, sondern den vollständigen Satz $N_{OKA}, N_{OKR}, N_{CE}, N_{FA}, N_{FR}, N'_{CE}$, wobei dann bessere Schlussfolgerungen bezüglich der einzelnen Funktionsblöcke nach Abbildung 1-1 auf Seite 16 möglich sind. Allerdings erhält man diesen vollständigen Satz an Kennzahlen nicht allein durch Beobachtung des Gerätes von außen[29]. Es sei folgende Kennzahl definiert:

$$\gamma = \frac{N_{FR,richtig}}{N_{FR,falsch}} \qquad \textbf{Gleichung 44}$$

Die bei rein äußerlicher Beobachtung des Geräts fehlende Information lässt sich bei bekannter Kennzahl γ nach Gleichung 46 berechnen:

$$N_{FR,falsch} = N_{FR} \cdot (1+\gamma)^{-1} \qquad \textbf{Gleichung 45}$$

$$N_{FR,richtig} = N_{FR} \cdot \left(1+\gamma^{-1}\right)^{-1} \qquad \textbf{Gleichung 46}$$

Die Funktionsweise des Matcher-Funktionsblocks wird durch folgende Größen beschrieben: Der Anteil der emittierten und falsch klassifizierten Keyword-Schallmustern an der Anzahl der falsch klassifizierten Keyword-Schallmuster wird im Folgenden mit α bezeichnet. Der Anteil der nicht emittierten richtig klassifizierten Keyword-Schallmuster an der Anzahl der richtig klassifizierten Keyword-Schallmuster wird im Folgenden mit ρ bezeichnet.

$$\alpha = \frac{N_{CE}}{N'_{CE}} \qquad \textbf{Gleichung 47}$$

$$\rho = \frac{N_{Keyw} - N'_{CE} - N_{OKA}}{N_{Keyw} - N'_{CE}} \qquad \textbf{Gleichung 48}$$

Beim Umformen sind folgende Zusammenhänge nützlich:

[29] Bei selbst entwickelter Software ist der vollständige Satz an Kennzahlen natürlich im Allgemeinen leicht protokollierbar.

$$CER' - CER = (1-\alpha)\cdot CER' = (1-CER')\cdot\frac{\rho}{\gamma}$$ **Gleichung 49**

$$\gamma = \rho\cdot\frac{1-\frac{1}{CER'}}{1-\alpha}$$ **Gleichung 50**

$$FRR = (1+\gamma)\cdot(CER' - CER)$$ **Gleichung 51**

Die von außen beobachtbaren Kennzahlen lassen sich vollständig aus den internen Kennzahlen berechnen:

$$CER = CER'\cdot\alpha$$ **Gleichung 52**

$$FRR = (1-CER')\cdot\rho + (1-\alpha)\cdot CER'$$ **Gleichung 53**

Unter Verwendung von Gamma ergeben sich folgende Gleichungen:

$$CER = CER' - \frac{\rho}{\gamma}\cdot(1-CER')$$ **Gleichung 54**

$$FRR = (1+\gamma^{-1})\cdot(1-CER')\cdot\rho = (1+\gamma)\cdot(1-\alpha)\cdot CER'$$ **Gleichung 55**

1.3.2 Einfluss der Vokabulargröße auf die Verwechslungsfehlerrate

Die Verwechslungsfehlerrate sollte so klein wie möglich sein. Verschiedene Maßnahmen tragen dazu bei: Eventuell ist es zweckmäßig, im Rahmen des Systemdesigns ein sprecherabhängiges System zu entwerfen, also vorzusehen, dass für jeden Benutzer persönliche Steuerbefehle aufgenommen werden. Das Problem dabei ist jedoch wiederum, dass die Sprachsteuerung bei der Verwendung die Information benötigt, wer mit ihr spricht, um geringe Fehlerraten zu liefern. Starken Einfluss hat natürlich die Größe des Vokabulars. Im Folgenden wird eine Näherungsformel, Gleichung 78 auf Seite 73 und Gleichung 86 auf Seite 76 abgeleitet, welche unter jeweils wenigen, plausiblen Annahmen, dazu verwendet werden können, aus Testergebnissen für die Verwechslungsfehlerrate bei bestimmten Vokabulargrößen, die Verwechslungsfehlerrate bei anderen Vokabulargrößen abzuschätzen.

1.3.2.1 Modellbildung mittels Minimum-Distance-Auswahlmodell

Im Folgenden ist unter der „Schallmusterklasse“ mit dem Index m die Menge jener Schallereignisse zu verstehen, für die das Wortmodell mit dem Index m modelliert sei.

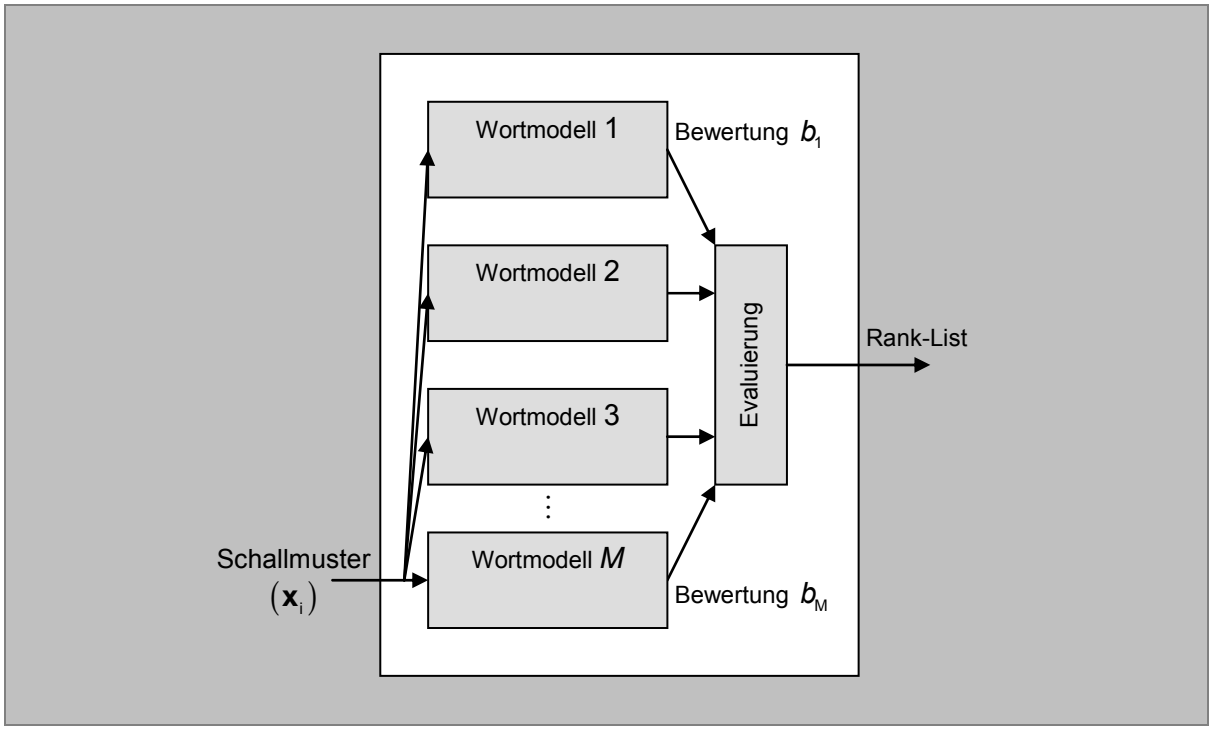

Abbildung 1-26: „Konkurrierende“ Wortmodelle im Classifier-Funktionsblock.

Abbildung 1-26 zeigt die Realisierung des Classifier-Funktionsblocks mittels „konkurrierender“ Wortmodelle. Ein Wortmodell mit der Nummer m liefert in Verbindung mit einem bestimmten Schallmuster eine sogenannte „Wortmodelldistanz“ oder „Bewertung“.

Die Anzahl der Schallmusterklassen sowie die Anzahl der Wortmodelle betragen im Folgenden M. Zu jeder Schallmusterklasse existiert umkehrbar eindeutig ein Wortmodell[30]. Unter einem Wortmodell m soll im Folgenden genau jener Teil des Systems verstanden werden, der zur Detektion der Schallmusterklasse m vorgesehen ist[31].

---

[30] Je nach Konstruktion des Systems kann ein „Wortmodell“ in unterschiedlichster Weise konstruiert sein und durchaus aus verschiedenen (zum Beispiel wiederum konkurrierenden) Modellen bestehen oder aber zum Beispiel aus verschiedenen Repräsentanten der zugehörigen Klasse bestehen oder aber auch aus Sub-Word-Unit-Modellen zusammengesetzt sein. Der Vielfalt an Ausführungsformen sind kaum Grenzen gesetzt.

[31] Ein beliebiges Schallmuster darf dabei nur Element einer einzigen Klasse sein. In diesem Sinne sind die Klassen festzulegen.

Es soll angenommen werden, dass die Wortmodelldistanzen eines zu klassifizierenden Schallmusters zu einer „Rank List“ weiterverarbeitet werden. Dabei soll die einfachste Methode, nämlich das „Sortieren der Wortmodelldistanzen“ Verwendung finden: Dabei legt die „Rank List“ die Reihenfolge der Wortmodelle vom „am besten passenden“ bis zum „am schlechtesten passenden“ fest [32].

Wortmodelldistanzdichtefunktionen. Die Dichtefunktion der kontinuierlichen Zufallsvariablen der Wortmodelldistanzen des Wortmodells m bezüglich Schallmustern der Klasse m'' sei mit $p_{m,m''}(\zeta)$ notiert. Es ergibt sich eine Matrix der Wortmodelldistanzdichtefunktionen:

$$\left(p_{m,m''}(\zeta)\right)=\begin{pmatrix} p_{1,1}(\zeta) & p_{1,2}(\zeta) & \cdots & p_{1,M}(\zeta) \\ p_{2,1}(\zeta) & p_{2,2}(\zeta) & & \\ \vdots & & \ddots & \\ p_{M,1}(\zeta) & & & p_{M,M}(\zeta) \end{pmatrix} \qquad \textbf{Gleichung 56}$$

Aus den Dichtefunktionen $p_{m,m''}(\zeta)$ lassen sich Wahrscheinlichkeiten $P_{m,m',m''}$ berechnen, dass das Wortmodell m „besser als“ das Wortmodell m' abschneidet. Unter folgenden Voraussetzungen lässt sich $P_{m,m',m''}$ nach Gleichung 57 angeben: Die Wortmodelldistanzen sind stets positiv. Die Relation „besser als“ wird ohne Beschränkung der Allgemeinheit mit „kleiner gleich“ definiert. Die Wortmodelldistanzen seien statistisch voneinander unabhängig.

$$P_{m,m',m''}=\int_0^{\infty}\left(p_{m,m''}(\xi)\cdot\int_{\xi}^{\infty}p_{m',m''}(\zeta)\cdot\ \mathrm{d}\zeta\right)\cdot\mathrm{d}\xi \qquad \textbf{Gleichung 57}$$

Anstatt die Wortmodelldistanzdichtefunktionen $p_{m,m''}(\zeta)$ als gegeben anzusehen und daraus die Model-Model-Verwechslungswahrscheinlichkeiten $P_{m,m',m''}$ zu berechnen, können diese auch direkt im praktischen Einsatz des Systems mittels einer verallgemeinerten Verwechslungsfehlermatrix geschätzt werden. Zur Verallgemeinerung der Verwechslungsfehlermatrix siehe Abschnitt 0 auf Seite 73.

Rank-List-Postionswahrscheinlichkeit. Die Wahrscheinlichkeit, dass bei M Modellen das Wortmodell m an einer bestimmten Stelle r der „Rank List“ zu liegen kommt, wenn Schallmuster der Klasse m'' präsentiert werden, soll im Folgenden mit $P_{m,r,m''}^{\text{Rank-List, M}}$ bezeichnet werden. Für die erste Position der „Rank List“ gilt, wenn Schallmuster der zugehörigen Klasse präsentiert werden:

[32] Es sind unter Umständen auch komplexere Verfahren denkbar. Nimmt man bei einfachem „Sortieren der Wortmodelldistanzen“ als Gewinner das Modell an, welches in der „Rank List“ an erster Stelle steht (Minimum-Distance-Auswahl), so geht genau genommen Information verloren: Die Wortmodelldistanzen der anderen Modelle werden nicht berücksichtigt. Man könnte auch die Wortmodelldistanzen der anderen Modelle bei der Erstellung einer „Rank List“ einfließen lassen.

$$P_{m,1,m}^{\text{Rank-List, M}} = \int_0^{\infty} \left(p_{m,m}(\xi) \cdot \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \int_{\xi}^{\infty} p_{m',m}(\zeta) \cdot d\zeta \right) \cdot d\xi \qquad \textbf{Gleichung 58}$$

Für Sprachmuster beliebiger Klassen gilt:

$$P_{m,1,m''}^{\text{Rank-List, M}} = \int_0^{\infty} \left(p_{m,m''}(\xi) \cdot \prod_{\substack{m'=1 \\ m' \neq m}}^{M} \int_{\xi}^{\infty} p_{m',m''}(\zeta) \cdot d\zeta \right) \cdot d\xi \qquad \textbf{Gleichung 59}$$

Um die Wahrscheinlichkeit für die zweite Position in der „Rank List“ zu berechnen, dient die Vorstellung, dass jedes der $M-1$ anderen Modelle die erste Position einnehmen kann, daher:

$$P_{m'',2,m''}^{\text{Rank-List, M}} = \sum_{\substack{k=1 \\ k \neq m''}}^{M} \int_0^{\infty} \left(p_{m'',m''}(\xi) \cdot \left(\int_0^{\xi} p_{k,m''}(\zeta) \cdot d\zeta \right) \cdot \prod_{\substack{m'=1 \\ m' \neq m'' \\ m' \neq k}}^{M} \int_{\xi}^{\infty} p_{m',m''}(\zeta) \cdot d\zeta \right) \cdot d\xi \qquad \textbf{Gleichung 60}$$

Prinzipiell lassen sich Ausdrücke für jede Position in der „Rank List“ anschreiben, wobei die Ausdrücke jedoch mit steigender Positionsnummer immer komplizierter werden. Man kann Gleichung 58 folgendermaßen interpretieren: Für jede Zahl ξ, welche das richtige Wortmodell m liefert, gibt es eine Wahrscheinlichkeit $\hat{P}(\xi)$, dass keines der anderen Wortmodelle gewinnt.

$$\hat{P}(\xi) = \prod_{\substack{m'=1 \\ m' \neq m''}}^{M} \int_{\xi}^{\infty} p_{m',m''}(\zeta) \cdot d\zeta \qquad \textbf{Gleichung 61}$$

Die Wahrscheinlichkeit, dass sich das Modell m' gegen ein anderes Modell durchsetzt, welches die Wortmodelldistanz β geliefert hat, wobei ein Sprachmuster der Klasse m'' analysiert wird, ist gleich $\int_0^{\beta} p_{m',m''}(\zeta) \cdot d\zeta$.

Die Wahrscheinlichkeit, dass sich das Modell m' gegen ein anderes Modell nicht durchsetzt, welches die Wortmodelldistanz β geliefert hat, wobei ein Sprachmuster der Klasse m'' analysiert wird, ist gleich $\int_{\beta}^{\infty} p_{m',m''}(\zeta) \cdot d\zeta$.

Liegt ein Modell in der „Rank List“ an der Stelle r, so sind $M-r$ Modelle schlechter und $r-1$ Modelle besser als das Modell, welches eine bestimmte Wortmodelldistanz β liefert.

Wortmodelldistanzdichtefunktionen. Einerseits sei nun angenommen, dass die Dichtefunktionen der Wortmodelldistanzen für Klassen, für die die jeweiligen Modelle nicht modelliert sind, untereinander gleich wären und gleich der Funktion $p_{\text{falsch}}(\zeta)$ wären. Andererseits sei angenommen, dass die Dichtefunktionen der Wortmodelldistanzen für Klassen, für die die jeweiligen Modelle modelliert sind, ebenfalls untereinander gleich wären und gleich der Funktion $p_{\text{richtig}}(\zeta)$ wären.

Rank-List-Positionswahrscheinlichkeit. So erhält man die Wahrscheinlichkeit für die Position r des richtigen Wortmodells in der „Rank List“ zu:

$$P_{\text{r}}^{\text{Rank-List, M}} = \int_0^{\infty} p_{\text{richtig}}(\beta) \cdot \binom{M-1}{r-1} \cdot \left(\int_0^{\beta} p_{\text{falsch}}(\zeta) \cdot \, \mathrm{d}\zeta \right)^{\text{r}-1} \cdot \left(\int_{\beta}^{\infty} p_{\text{falsch}}(\zeta) \cdot \, \mathrm{d}\zeta \right)^{\text{M}-\text{r}} \cdot \mathrm{d}\beta$$

$$= \binom{M-1}{r-1} \cdot \int_0^{\infty} p_{\text{richtig}}(\beta) \cdot \left(\int_0^{\beta} p_{\text{falsch}}(\zeta) \cdot \, \mathrm{d}\zeta \right)^{\text{r}-1} \cdot \left(\int_{\beta}^{\infty} p_{\text{falsch}}(\zeta) \cdot \, \mathrm{d}\zeta \right)^{\text{M}-\text{r}} \cdot \mathrm{d}\beta \qquad \textbf{Gleichung 62}$$

Die Wahrscheinlichkeit der ersten Position in der „Rank List“ des korrekten Wortmodells ergibt sich zu Gleichung 63:

$$P_1^{\text{Rank-List, M}} = \int_0^{\infty} p_{\text{falsch}}(\beta) \cdot \left(1 - P_{\text{falsch}}(\beta)\right)^{\text{M}-1} \cdot \mathrm{d}\beta \qquad \textbf{Gleichung 63}$$

Kontrolle der Symmetrie. Setzt man in einem Gedankenexperiment $p_{\text{richtig}} = p_{\text{falsch}}$ und $M = 2$, das heißt zwei äquivalente Modelle, so erhält man durch partielle Integration:

$$P_1^{\text{Rank-List, 2}} = \frac{1}{2} \quad \text{qu.e.d.} \qquad \textbf{Gleichung 64}$$

Allgemeiner erhält man für M äquivalente Modelle, wenn man mit $P_{\text{falsch}}(\zeta)$ diejenige Stammfunktion von $p_{\text{falsch}}(\zeta)$ bezeichnet für die $\lim_{\zeta \to \infty} P_{\text{falsch}}(\zeta) = 1$ gilt:

$$\begin{aligned} P_1^{\text{Rank-List, M}} &= \int_0^{\infty} p_{\text{falsch}}(\beta) \cdot \left(1 - P_{\text{falsch}}(\beta)\right)^{\text{M}-1} \cdot \mathrm{d}\beta = \\ &= \int_0^{\infty} p_{\text{falsch}}(\beta) \cdot \sum_{\text{k}=0}^{\text{M}-1} \binom{M-1}{k} \left(-P_{\text{falsch}}(\beta)\right)^{\text{k}} \cdot \mathrm{d}\beta = \\ &= \sum_{\text{k}=0}^{\text{M}-1} \left[\binom{M-1}{k} \cdot (-1)^{\text{k}} \cdot \int_0^{\infty} p_{\text{falsch}}(\beta) \cdot P_{\text{falsch}}^{\text{k}}(\beta) \cdot \mathrm{d}\beta \right] \end{aligned} \qquad \textbf{Gleichung 65}$$

$$\int_0^\infty p_{\text{falsch}}(\beta)\cdot P_{\text{falsch}}^{k}(\beta)\cdot d\beta + \int_0^\infty P_{\text{falsch}}(\beta)\cdot k\cdot P_{\text{falsch}}^{k-1}(\beta)\cdot p_{\text{falsch}}(\beta)\cdot d\beta = \\ = \left[\int p_{\text{falsch}}(\beta)\cdot d\beta\cdot P_{\text{falsch}}^{k}(\beta)\right]_0^\infty = 1 \qquad \textbf{Gleichung 66}$$

Die partielle Integration gemäß Gleichung 66 führt somit auf Gleichung 67.

$$(k+1)\cdot\int_0^\infty p_{\text{falsch}}(\beta)\cdot P_{\text{falsch}}^{k}(\beta)\cdot d\beta = 1 \qquad \textbf{Gleichung 67}$$

Aus Gleichung 68 folgt schließlich Gleichung 69.

$$\sum_{k=0}^{M-1}\left[\binom{M-1}{k}\cdot(-1)^k\cdot\frac{1}{1+k}\right] = \frac{1}{M} \quad \forall \quad M>0 \qquad \textbf{Gleichung 68}$$

$$P_1^{\text{Rank-List, M}} = \frac{1}{M} \quad \text{qu.e.d.} \qquad \textbf{Gleichung 69}$$

Abschätzung der Verwechslungsfehlerrate. Als CER'_M wird die Verwechslungsfehlerrate des niemals zurückweisenden Systems bezeichnet. Die Verwechslungsfehlerrate CER'_M ist somit der Anteil der falsch klassifizierten Keyword-Schallmuster an der Gesamtanzahl der dem System dargebotenen Keyword-Schallmuster bei M Klassen. Die Verwechslungsfehlerrate CER'_M ergibt sich zu Gleichung 71. Man vergleiche mit Gleichung 58 auf Seite 69, sowie Gleichung 78 auf Seite 73 und Gleichung 86 auf Seite 76.

$$CER'_M = 1 - P_1^{\text{Rank-List, M}} \qquad \textbf{Gleichung 70}$$

$$CER'_M = 1 - \int_0^\infty p_{\text{richtig}}(\beta)\cdot\left(\int_\beta^\infty p_{\text{falsch}}(\zeta)\cdot\ d\zeta\right)^{M-1}\cdot d\beta \qquad \textbf{Gleichung 71}$$

1.3.2.2 Modellbildung mittels Urnenmodell

Die Verteilung der Zufallsvariable „Index des Wortmodells, welches gewinnt“, soll im Folgenden mittels „Ziehen färbiger Kugeln aus einer Urne“ modelliert werden, wobei jedem Wortmodellindex eine Farbe zugeordnet wird, und der Index des korrekten Wortmodells zum Beispiel der Farbe Weiß entspricht.

Bezeichnet man bei einem korrekten Wortmodell und $M-1$ nicht korrekten Wortmodellen die Anzahl der weißen Kugeln mit R_{M} und die Anzahl der andersfarbigen Kugeln mit $F_{\mathrm{M}} = F_2 \cdot (M-1)$, so sind die Kugelanzahlen derart zu wählen, dass

$$CER'_{\mathrm{M}} = \frac{F_{\mathrm{M}}}{R_{\mathrm{M}} + F_{\mathrm{M}}}, \quad P_1^{\text{Rank-List, M}} = \frac{R_{\mathrm{M}}}{R_{\mathrm{M}} + F_{\mathrm{M}}} \qquad \textbf{Gleichung 72}$$

Gibt es außer dem korrekten Wortmodell nur ein einziges nicht korrektes Wortmodell, so ergeben sich die Anzahlen der weißen und zum Beispiel schwarzen Kugeln aus der Wahrscheinlichkeit $P_1^{\text{Rank-List, 2}}$. Die Anzahl der schwarzen Kugeln sei zu F_2 gewählt und die Anzahl der weißen Kugeln sei nach Gleichung 73 bestimmt:

$$R_2 = F_2 \cdot \frac{P_1^{\text{Rank-List,2}}}{1 - P_1^{\text{Rank-List,2}}} \qquad \textbf{Gleichung 73}$$

Wird ein zweites, nicht korrektes Wortmodell hinzugefügt, so ist zu modellieren, dass dann, wenn sich ein nicht korrektes Wortmodell durchsetzt, es mit gleicher Wahrscheinlichkeit jedes der nicht korrekten Wortmodelle sein kann. Dies kommt dadurch zum Ausdruck, dass die Anzahl der Kugeln nicht weißer Farben jeweils gleich sind. Eine Möglichkeit, diesen Umstand einfließen zu lassen, besteht darin, das Hinzufügen eines zusätzlichen nicht korrekten Wortmodells dadurch zu modellieren, dass zum Beispiel rote Kugeln hinzugefügt werden. Die Anzahl der hinzugefügten roten Kugeln entspricht der Anzahl der bereits vorhanden gewesenen schwarzen Kugeln.

$$F_{\mathrm{M}} = F_2 \cdot (M-1) \qquad \textbf{Gleichung 74}$$

$$CER'_{\mathrm{M}} = \frac{F_2 \cdot (M-1)}{R_{\mathrm{M}} + F_2 \cdot (M-1)} = \frac{M-1}{\frac{R_{\mathrm{M}}}{F_2} + M - 1} \qquad \textbf{Gleichung 75}$$

$$\Rightarrow \frac{R_{\mathrm{M}}}{F_2} = (M-1) \cdot \left(\frac{1}{CER'_{\mathrm{M}}} - 1 \right) \qquad \textbf{Gleichung 76}$$

Das Entfernen eines nicht korrekten, „roten“ Wortmodells entspricht dem Entfernen der roten Kugeln aus der Urne. Was jedoch passiert mit der Anzahl der weißen Kugeln beim Hinzufügen beziehungsweise Entfernen eines nicht korrekten Wortmodells? Bezüglich der „Variation der Anzahl der weißen Kugeln“ liegt noch Verbesserungspotenzial brach, um das Modell noch besser an beobachtete Daten anzupassen.

Abschätzung der Verwechslungsfehlerrate. Modelliert man, dass die Anzahl der weißen Kugeln gleich leibt, so erhält man ein sehr anschauliches Urnenmodell, welches sich zur Schätzung der Rate CER'_{M+1} aus der Rate CER'_{M} ohne spezielles Wissen über die Funktionsweise der Sprachsteuerung eignet:

$$CER'_{M+1} = \frac{F_{M+1}}{R_{M+1} + F_{M+1}} = \frac{F_2 \cdot M}{R_{M+1} + F_2 \cdot M} \approx \frac{F_2 \cdot M}{R_M + F_2 \cdot M} = \frac{M}{M + \frac{R_M}{F_2}} \qquad \textbf{Gleichung 77}$$

Setzt man nun Gleichung 76 in Gleichung 77 ein, so erhält man Gleichung 78:

$$CER'_{M+1} \approx CER'_{M} \cdot \frac{M}{M - 1 + CER'_{M}} \qquad \textbf{Gleichung 78}$$

1.3.2.3 Modellbildung mittels Confusion-Matrix-Modell

Die in diesem Abschnitt abgeleitete Näherungsformel basiert auf Annahmen über Eigenschaften der „Confusion Matrix“, daher folgt zunächst deren Definition.

Definition der „Confusion Matrix“. Die „Confusion Matrix“ für M Wortmodelle wird im Folgenden mit $\mathbf{V}_M$ bezeichnet. Die Anzahl der Zeilen sowie die Anzahl der Spalten dieser Matrix beträgt M. Jede Zeile der „Confusion Matrix“ ist umkehrbar eindeutig einer Schallmusterklasse zugeordnet. Jeder Spalte der „Confusion Matrix“ ist umkehrbar eindeutig einem Wortmodell zugeordnet. Jedes Element der „Confusion Matrix“ gibt an, wie oft jedes einzelne Wortmodell an erster Stelle der „Rank List“ zu liegen kommt, wobei alle Schallmuster jeder einzelnen Schallmusterklasse präsentiert werden.

Verallgemeinerung der Definition. Gelegentlich ist es zweckmäßig von der Vorstellung einer verallgemeinerten „Confusion Matrix“ auszugehen, deren Elemente Verteilungen darstellen. Die „Two-Winners Confusion Matrix“ sei beispielsweise mit $\mathbf{U}_M$ bezeichnet. Die Anzahl der Zeilen, sowie die Anzahl der Spalten dieser Matrix beträgt M. Jede Zeile der „Two-Winners Confusion Matrix“ ist umkehrbar eindeutig einer Schallmusterklasse zugeordnet. Jeder Spalte der „Two-Winners Confusion Matrix“ ist umkehrbar eindeutig einem Wortmodell zugeordnet. Jedes Element der „Two-Winners Confusion Matrix“ gibt an, wie oft jedes einzelne Wortmodell an erster oder zweiter Stelle der „Rank List“ zu liegen kommt, wobei wiederum alle Schallmuster jeder einzelnen Schallmusterklasse prä-

sentiert werden. Bildet man die Differenzmatrix zur „Confusion Matrix“ $\mathbf{U}_{\mathrm{M}} - \mathbf{V}_{\mathrm{M}}$, so geben die Elemente dieser Matrix an, wie oft jedes einzelne Wortmodell an zweiter Stelle der „Rank List“ zu liegen kommt, wobei alle Schallmuster jeder einzelnen Schallmusterklasse präsentiert werden.

Um die Notation in diesem Abschnitt nicht unnötig zu verkomplizieren, werden gleich viele Schallmuster für jede einzelne Schallmusterklasse angenommen. Der Formalismus ließe sich aber leicht auf unterschiedliche Anzahlen verallgemeinern.

Erste Annahme. Die erste Annahme betrifft die „Confusion Matrix“: Die Diagonalelemente sind untereinander gleich und auch die Nichtdiagonalelemente sind untereinander gleich. Die Anzahl der Schallmusterklassen sei mit M bezeichnet. Die Anzahl aller Schallmuster sei mit N bezeichnet. Mittels der Anzahl der „Confusion Errors“ N_{CE} lässt sich diese „Confusion Matrix“ wie folgt notieren:

$$\mathbf{V}_{\mathrm{M}} \approx \begin{pmatrix} \frac{N-N_{\mathrm{CE}}}{M} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \cdots & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} \\ \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N-N_{\mathrm{CE}}}{M} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \cdots & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} \\ \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N-N_{\mathrm{CE}}}{M} & \cdots & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \frac{N_{\mathrm{CE}}}{M\cdot(M-1)} & \cdots & \frac{N-N_{\mathrm{CE}}}{M} \end{pmatrix} \qquad \textbf{Gleichung 79}$$

Gleichung 80 zeigt die gleiche Matrix unter Verwendung der Verwechslungsfehlerrate des niemals zurückweisenden Systems $CER'_{\mathrm{M}} = N_{\mathrm{CE}}/N$. Darunter ist der Anteil der falsch klassifizierten Schallmuster an der Gesamtanzahl der dem System dargebotenen Schallmuster bei M Schallmusterklassen zu verstehen.

$$\mathbf{V}_{\mathrm{M}} \approx \begin{pmatrix} \frac{1-CER'_{\mathrm{M}}}{M/N} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \cdots & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} \\ \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{1-CER'_{\mathrm{M}}}{M/N} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \cdots & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} \\ \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{1-CER'_{\mathrm{M}}}{M/N} & \cdots & \frac{CER'_{\mathrm{M}}}{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \frac{N\cdot CER'_{\mathrm{M}}}{M\cdot(M-1)} & \cdots & \frac{1-CER'_{\mathrm{M}}}{M/N} \end{pmatrix} \qquad \textbf{Gleichung 80}$$

In einem Gedankenexperiment sollen nun die Schallmuster einer Schallmusterklasse sowie das zugehörige korrekte Wortmodell weggelassen werden. Ohne

Beschränkung der Allgemeinheit wird das Wortmodell M beziehungsweise die Schallmusterklasse M weggelassen.

Zweite Annahme. Wie verändert sich dabei die „Confusion Matrix“? Wie teilen sich die jeweils $N_{CE}/(M\cdot(M-1))$ ursprünglich falsch klassifizierten Schallmuster auf die verbleibenden Elemente der Zeile auf? Es sei angenommen, dass ein bestimmter Anteil $\tilde{\alpha}$ der $N_{CE}/(M\cdot(M-1))$ Schallmuster sodann richtig klassifiziert wird. Die Diagonalelemente der neuen quadratischen „Confusion Matrix“, welche ja $M-1$ Spalten aufweist ergeben sich daher zu:

$$M_A = \frac{N - N_{CE}}{M} + \frac{N_{CE}}{M\cdot(M-1)}\cdot\tilde{\alpha} \qquad \textbf{Gleichung 81}$$

Die restlichen Schallmuster verteilen sich gleichmäßig auf die $M-2$ Nichtdiagonalelemente der Zeile. Für alle Nichtdiagonalelemente der Matrix ergibt sich:

$$M_B = \frac{N_{CE}}{M\cdot(M-1)} + \frac{N_{CE}}{M\cdot(M-1)}\cdot\frac{\tilde{\alpha}}{M-2} \qquad \textbf{Gleichung 82}$$

Die neue „Confusion Matrix“ lässt sich wiederum unter Verwendung der Verwechslungsfehlerrate des niemals zurückweisenden Systems, diesmal CER'_{M-1}, notieren, wobei die Anzahl der verbleibenden Schallmuster $N\cdot(M-1)/M$ beträgt:

$$\mathbf{V}_{M-1} \approx \frac{N\cdot\frac{M-1}{M}}{M-1}\cdot\begin{pmatrix} 1-CER'_{M-1} & \frac{CER'_{M-1}}{M-2} & \frac{CER'_{M-1}}{M-2} & \cdots & \frac{CER'_{M-1}}{M-2} \\ \frac{CER'_{M-1}}{M-2} & 1-CER'_{M-1} & \frac{CER'_{M-1}}{M-2} & \cdots & \frac{CER'_{M-1}}{M-2} \\ \frac{CER'_{M-1}}{M-2} & \frac{CER'_{M-1}}{M-2} & 1-CER'_{M-1} & \cdots & \frac{CER'_{M-1}}{M-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{CER'_{M-1}}{M-2} & \frac{CER'_{M-1}}{M-2} & \frac{CER'_{M-1}}{M-2} & \cdots & 1-CER'_{M-1} \end{pmatrix} \qquad \textbf{Gleichung 83}$$

Durch Gleichsetzen des Diagonalelements aus Gleichung 83 mit dem Diagonalelement aus Gleichung 81 oder aber durch Gleichsetzen des Nichtdiagonalelements aus Gleichung 83 mit dem Nichtdiagonalelement aus Gleichung 82 erhält man nach einigen Umformungsschritten die Gleichung 84.

$$CER'_M \approx CER'_{M-1}\cdot\frac{M-1}{M-1-\tilde{\alpha}} \qquad \textbf{Gleichung 84}$$

Substituiert man M als `$M+1$, so erhält man eine Näherungsformel für die Verwechslungsfehlerrate beim Hinzufügen eines Wortmodells:

$$CER'_{M+1} \approx CER'_{M} \cdot \frac{`M}{`M - \tilde{\alpha}}$$ **Gleichung 85**

Dritte Annahme. Es sei nun ein konkretes $\tilde{\alpha}$ gewählt, beispielsweise $\tilde{\alpha} = 1 - CER'_{M}$, was bei näherer Betrachtung recht plausibel ist.

Abschätzung der Verwechslungsfehlerrate. Man erhält Gleichung 86, welche mit der Näherungsformel des Urnenmodells nach Gleichung 78 auf Seite 73 identisch ist:

$$CER'_{M+1} \approx CER'_{M} \cdot \frac{`M}{`M - 1 + CER'_{M}}$$ **Gleichung 86**

1.3.3 Einfluss des Einsatzes von Non-Keyword-Modellen

Ein Problem bei der Konstruktion von Sprachsteuerungen ist, dass nach heutigem Stand der Technik kein auf der Hand liegendes Verfahren etabliert ist, um zu entscheiden, ob es sich bei einem Schallereignis überhaupt um einen Steuerbefehl handelt. Gemäß dem im letzten Abschnitt betrachteten Formalismus würde jedes lautliche Ereignis klassifiziert, also einem Steuerbefehl zugeordnet werden. Diese Vorgehensweise entspräche den Fehlerraten $FAR'_{M} = 1, FRR'_{M} = 0$. Im Allgemeinen versucht man jedoch die Falschakzeptanzrate gegen die Falschrückweisungsrate abzuwägen und sucht nach konstruktiven Möglichkeiten, die Falschakzeptanzrate zu verringern, wobei die Falschrückweisungsrate als Kehrseite der Medaille dabei im Allgemeinen leider ansteigt:

Eine dieser Möglichkeiten besteht darin, die Anzahl der Klassen M entsprechend zu erhöhen und dabei eine Anzahl $M_{\text{Non-Keyw}}$ an Klassen für Non-Keyword-Modelle vorzusehen. Wenn sich ein solches Non-Keyword-Modell in der „Rank List“ durchsetzt, also an erster Stelle steht, kommt es zu keiner Emission, das heißt die Sprachsteuerung reagiert nicht. Die Idee dahinter ist also, dass ein aufgeschnapptes vokabularfremdes Wort mit gewisser Wahrscheinlichkeit auch auf ein Non-Keyword-Modell abgebildet wird und somit nicht zu einer Emission führt. Es ist zweckmäßig, wenn Non-Keyword-Modelle potenzielle Umgebungsgeräusche modellieren. Die Non-Keyword-Modelle sollen daher an die Benutzungsumstände der Sprachsteuerung angepasst sein[33].

$$M = M_{\text{Keyw}} + M_{\text{Non-Keyw}}$$ **Gleichung 87**

[33] Da Non-Keyword-Modelle oftmals Schallmuster-Klassen modellieren sollen, deren Schallmuster insgesamt untereinander unähnlicher sind, als Schallmuster von Schallmusterklassen von Keyword-Modellen, kann es sinnvoll sein, die eine oder andere Schallmuster-Klasse eines Non-Keyword-Modells aufzuteilen und mittels mehrerer Teilmodelle zu modellieren.

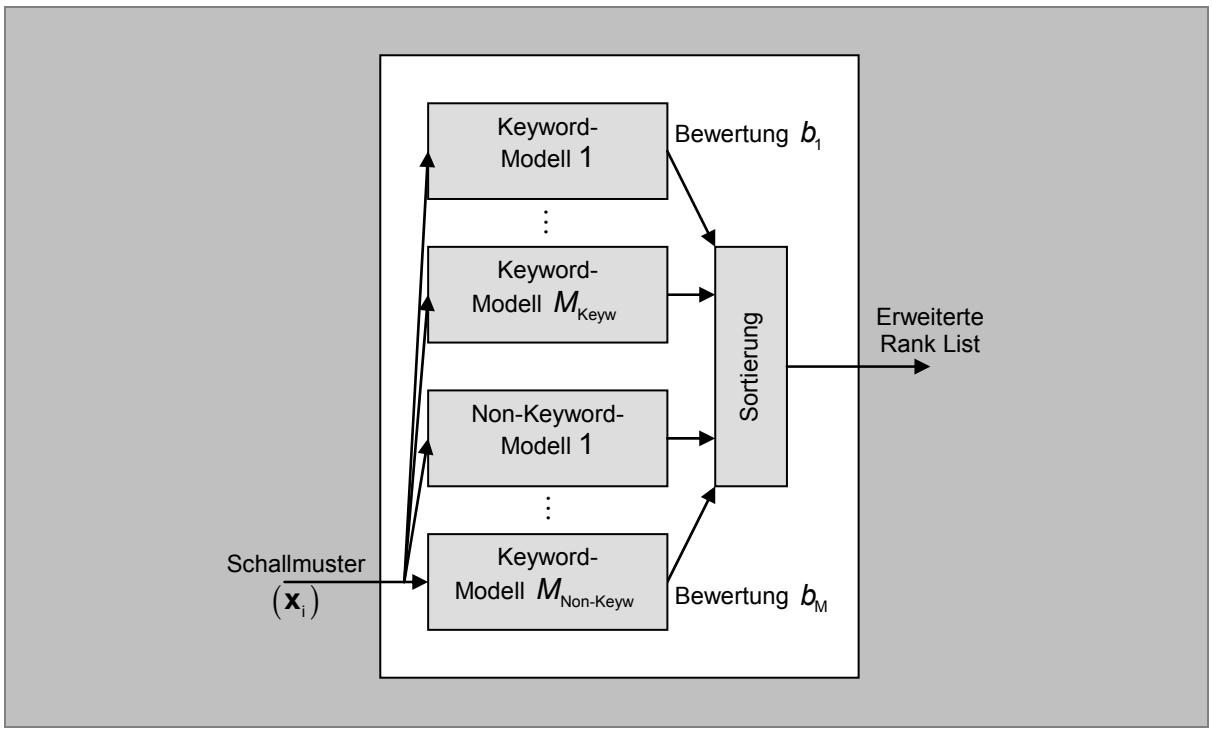

Abbildung 1-27: Maßnahme im Classifier-Funktionsblock: Verringerung der Falschakzeptanzrate durch die Verwendung von Non-Keyword-Modellen.

Setzt man vereinfachend die „Confusion Matrix“ nach Gleichung 88 an und widmet $M_{\text{Non-Keyw}}$ Wortmodelle als Non-Keyword-Modelle um, so ändern die Klassifikationsergebnisse ihre Bedeutung, wie Gleichung 94 auf Seite 78 zu entnehmen ist.

$$\mathbf{V}_1 \approx \begin{pmatrix} \frac{N-N'_{CE}}{M} & \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N'_{CE}}{M\cdot(M-1)} & \cdots & \frac{N'_{CE}}{M\cdot(M-1)} \\ \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N-N'_{CE}}{M} & \frac{N'_{CE}}{M\cdot(M-1)} & \cdots & \frac{N'_{CE}}{M\cdot(M-1)} \\ \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N-N'_{CE}}{M} & \cdots & \frac{N'_{CE}}{M\cdot(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N'_{CE}}{M\cdot(M-1)} & \frac{N'_{CE}}{M\cdot(M-1)} & \cdots & \frac{N-N'_{CE}}{M} \end{pmatrix} \qquad \textbf{Gleichung 88}$$

Gleichung 89 notiert die gleiche Matrix unter Verwendung der Verwechslungsfehlerrate des niemals zurückweisenden Systems CER'_M. Darunter ist der Anteil der falsch klassifizierten Schallmuster an der Gesamtanzahl der dem System dargebotenen Schallmuster bei M Schallmusterklassen zu verstehen.

$$\mathbf{V}_1 \approx \frac{N}{M} \cdot \begin{pmatrix} 1-CER'_M & \frac{CER'_M}{M-1} & \frac{CER'_M}{M-1} & \cdots & \frac{CER'_M}{M-1} \\ \frac{CER'_M}{M-1} & 1-CER'_M & \frac{CER'_M}{M-1} & \cdots & \frac{CER'_M}{M-1} \\ \frac{CER'_M}{M-1} & \frac{CER'_M}{M-1} & 1-CER'_M & \cdots & \frac{CER'_M}{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{CER'_M}{M-1} & \frac{CER'_M}{M-1} & \frac{CER'_M}{M-1} & \cdots & 1-CER'_M \end{pmatrix} \qquad \textbf{Gleichung 89}$$

Zur Vereinfachung der Notation von Gleichung 94 auf Seite 78 seien die Abkürzungen nach Gleichung 90, Gleichung 91 und Gleichung 92 definiert:

$$M_1 = M_{\text{Keyw}}^2 - M_{\text{Keyw}} \qquad \textbf{Gleichung 90}$$

$$M_2 = M_{\text{Non-Keyw}}^2 - M_{\text{Non-Keyw}} \qquad \textbf{Gleichung 91}$$

$$M_0 = M_{\text{Keyw}} \cdot M_{\text{Non-Keyw}} \qquad \textbf{Gleichung 92}$$

Die Anzahl der Schallmuster teilt sich in die Anzahl der Keyword-Schallmuster N_{Keyw} und die Anzahl der Non-Keyword-Schallmuster $N_{\text{Non-Keyw}}$ auf.

$$N_{\text{Keyw}} = N \cdot \frac{M_{\text{Keyw}}}{M}, \quad N_{\text{Non-Keyw}} = N \cdot \frac{M_{\text{Non-Keyw}}}{M} \qquad \textbf{Gleichung 93}$$

$$\mathbf{V}_1 \approx \left(\begin{array}{ccc:ccc} \frac{N_{\text{OKA}}}{M_{\text{Keyw}}} & & \frac{N_{\text{CE}}}{M_1} & \ddots & \vdots & \iddots \\ & \ddots & & \cdots & \frac{N_{\text{FA}}}{M_0} & \cdots \\ \frac{N_{\text{CE}}}{M_1} & & \frac{N_{\text{OKA}}}{M_{\text{Keyw}}} & \iddots & \vdots & \ddots \\ \hdashline \ddots & \vdots & \iddots & \frac{N_{\text{OKR1}}}{M_{\text{Non-Keyw}}} & & \frac{N_{\text{OKR2}}}{M_2} \\ \cdots & \frac{N_{\text{FA}}}{M_0} & \cdots & & \ddots & \\ \iddots & \vdots & \ddots & \frac{N_{\text{OKR2}}}{M_2} & & \frac{N_{\text{OKR1}}}{M_{\text{Non-Keyw}}} \end{array}\right) \qquad \textbf{Gleichung 94}$$

Elementweises Gleichsetzen von Gleichung 89 und Gleichung 94 ergibt die gesuchten Zusammenhänge zur Abschätzung der Verwechslungsfehlerrate, der Falschrückweisungsrate und der Falschakzeptanzrate eines Systems mit Non-Keyword-Modellen aus der Verwechslungsfehlerrate des entsprechenden Systems ohne Non-Keyword-Modellen aber dafür mit entsprechend mehr Keyword-Modellen:

$$CER_{\mathrm{M}} = \frac{N_{\mathrm{CE}}}{N_{\mathrm{Keyw}}} \approx CER'_{\mathrm{M}} \cdot \frac{M_{\mathrm{Keyw}} - 1}{M - 1} \qquad \textbf{Gleichung 95}$$

$$FAR_{\mathrm{M}} = \frac{N_{\mathrm{FA}}}{N_{\mathrm{Non\text{-}Keyw}}} = CER'_{\mathrm{M}} \cdot \frac{M_{\mathrm{Keyw}}}{M - 1} \qquad \textbf{Gleichung 96}$$

$$FRR_{\mathrm{M}} = \frac{N_{\mathrm{FR}}}{N_{\mathrm{Keyw}}} = CER'_{\mathrm{M}} \cdot \frac{M_{\mathrm{Non\text{-}Keyw}}}{M - 1} \qquad \textbf{Gleichung 97}$$

Interessant ist weiters der Spezialfall $M_{\mathrm{Non\text{-}Keyw}} = 1$: Die Falschakzeptanzrate entspricht dann der Verwechslungsfehlerrate des niemals zurückweisenden Systems.

1.3.4 Equal-Error-Rates-Prinzip

Beim Equal-Error-Rates-Prinzip wird eine Balance zwischen Falschakzeptanzrate und Falschrückweisungsrate angestrebt, welche mittels verschiedener Maßnahmen erreicht werden kann.

$$FAR_{\mathrm{M}} = FRR_{\mathrm{M}} \qquad \textbf{Gleichung 98}$$

Setzt man den Ausdruck für die Falschakzeptanzrate aus Gleichung 96 auf Seite 79 und den Ausdruck für die Falschrückweisungsrate aus Gleichung 97 auf Seite 79 gleich, so erhält man jene Anzahl $M_{\mathrm{Non\text{-}Keyw}}$ an Non-Keyword-Modellen, bei der die Equal-Error-Rates-Einstellung erfüllt ist[34], nach Gleichung 99.

$$M_{\mathrm{Non\text{-}Keyw}} = M_{\mathrm{Keyw}} \qquad \textbf{Gleichung 99}$$

[34] Damit die Annahme der vergleichbaren Dichtefunktionen der Wortmodelldistanzen realistisch ist, ist jedoch zu berücksichtigen, dass bezüglich der Non-Keyword-Modelle unter Umständen konstruktive Maßnahmen notwendig werden, da diese im Vergleich zu den Keyword-Modellen mit einer wesentlich größeren Variabilität der Schallmuster umgehen müssen. Die konstruktiven Maßnahmen können beispielsweise in der Zusammensetzung der Non-Keyword-Modelle aus Teilmodellen, beispielsweise jeweils für unterschiedliche Ausprägungen potenzieller Umgebungsgeräusche, bestehen.

1.3.5 Einfluss des Einsatzes inaktiver Keyword-Modelle

Eine weitere Möglichkeit, die Falschakzeptanzrate zu verringern, besteht darin, den Kontext der Dialogsituation zu berücksichtigen. In einer speziellen Dialogsituation sind jeweils nur bestimmte Steuerbefehle zulässig. Bei M_{Keyw} Keyword-Modellen, $M_{\text{Non-Keyw}}$ Non-Keyword-Modellen und einer dialogsituationsabhängigen Perplexität $M_{\text{Act-Keyw}}$, wenn also zur Dialogsituation $M_{\text{Act-Keyw}}$ Keyword-Modelle aktiv sind, das heißt, emittieren können, ergeben sich, je nach Perplexität $0 < M_{\text{Act-Keyw}} \leq M_{\text{Keyw}}$, etwa folgende Fehlerraten:

$$CER_{\text{M}} \approx \frac{M_{\text{Act-Keyw}} - 1}{M_{\text{Act-Keyw}} + M_{\text{Non-Keyw}} - 1} \cdot CER'_{\text{M}} \qquad \textbf{Gleichung 100}$$

$$FAR_{\text{M}} \approx \frac{M_{\text{Act-Keyw}}}{M_{\text{Act-Keyw}} + M_{\text{Non-Keyw}} - 1} \cdot CER'_{\text{M}} \qquad \textbf{Gleichung 101}$$

$$FRR_{\text{M}} \approx \frac{M_{\text{Non-Keyw}}}{M_{\text{Act-Keyw}} + M_{\text{Non-Keyw}} - 1} \cdot CER'_{\text{M}} \qquad \textbf{Gleichung 102}$$

Will man die Equal-Error-Rates-Einstellung in jeder Dialogsituation möglichst exakt durchführen, so besteht eine Möglichkeit in der Variation der Anzahl der verwendeten Non-Keyword-Modelle entsprechend der Dialogsituation nach Gleichung 103. Als Spezialfall folgt wiederum Gleichung 99 auf Seite 79.

$$M_{\text{Non-Keyw}} = M_{\text{Act-Keyw}} \qquad \textbf{Gleichung 103}$$

Zur Variation der Anzahl der verwendeten Non-Keyword-Modelle können gegebenenfalls auch die jeweils nicht aktiven Keyword-Modelle als zusätzliche Non-Keyword-Modelle herangezogen werden.

1.3.6 Einfluss des Einsatzes von A-priori-Wahrscheinlichkeiten

Eine bestimmte Dialogsituation sei durch die zulässigen Steuerbefehle und die bekannten Auftrittshäufigkeiten H_m ihrer Keyword-Schallmuster beschrieben. Sämtliche Non-Keyword-Schallmuster[35] werden dabei im Folgenden der Klasse 0 zugeordnet. $0 < m \leq M$. Über einen längeren Benutzungszeitraum der Sprachsteuerung unter realistischen Benutzungsumständen erfasst, können diese Auftrittshäufigkeiten als Schätzwerte für die Auftrittswahrscheinlichkeiten verwendet werden. Sie können aber auch vorab aus den Dialogerfordernissen geschätzt und nach einer längeren Phase des Betriebs neu festgelegt werden.

$$P_0 + \sum_{m=1}^{M} P_m = 1 \qquad \textbf{Gleichung 104}$$

Wenn, wie es etwa bei Hidden-Markov-Modell-Sprachsteuerungen üblich ist, die Entscheidung für ein Wortmodell auf Grund der Wahrscheinlichkeiten gefällt wird, mit denen die Wortmodelle das beobachtete Schallmuster erzeugen würden, so ermöglicht die Verwendung von A-priori-Wahrscheinlichkeiten den Übergang von einer Maximum-Likelihood-Schätzung zu einer Maximum-a-posteriori-Schätzung. Die Maximum-Likelihood-Schätzung erfolgt gemäß Gleichung 105. Die Maximum-a-posteriori-Schätzung erfolgt gemäß Gleichung 106:

$$m_{ML} = \underset{m \in \{0,1,2...M\}}{\arg\max}\, P\left(\left(\mathbf{x}_i\right) \middle|\, m\right) \qquad \textbf{Gleichung 105}$$

$$m_{MAP} = \underset{m \in \{0,1,2...M\}}{\arg\max}\, P\left(\left(\mathbf{x}_i\right) \middle|\, m\right) \cdot P_m = \underset{m \in \{0,1,2...M\}}{\arg\max}\, P\left(\left(\mathbf{x}_i\right), m\right) = \underset{m \in \{0,1,2...M\}}{\arg\max}\, P\left(m \middle|\, \left(\mathbf{x}_i\right)\right) \qquad \textbf{Gleichung 106}$$

Lassen sich beobachtete Auftrittshäufigkeiten nun zur Verbesserung der Verwechslungsfehlerrate heranziehen? Im Prinzip: „Ja“. Es stellt sich allerdings die Frage, ob diese „Verbesserung“ zu einer Erhöhung der Benutzerzufriedenheit führt: Es sollte darauf geachtet werden, dass sich trotzdem die selteneren Steuerbefehle auch gegen die häufigen durchsetzen können. Am besten ist dies dadurch zu erreichen, dass sich die gleichzeitig möglichen Steuerbefehle in der Aussprache wesentlich unterscheiden[36].

---

[35] Dazu gehören einerseits Non-Keyword-Schallmuster, welche aus den Umgebungsgeräuschen aufgeschnappt werden und andererseits auch vokabularfremde oder in der Dialogsituation nicht erwartete Worte, die fälschlicherweise durch den Benutzer gesprochen werden.

[36] Dies ist bei der Auswahl der Steuerbefehle für die Dialogsituationen zu berücksichtigen.

Man sollte dabei insbesondere bedenken, dass eine Sprachsteuerung nicht nur geringe Fehlerraten aufweisen soll, welche ja arithmetische Mittelwerte sind, sondern auch auf seltene Steuerbefehle korrekt reagieren soll, gerade weil in einem seltenen Ereignis mehr Information steckt[37]. Würde eine Sprachsteuerung als Extrembeispiel das Schallereignis gar nicht analysieren, sondern gleich den häufigsten Steuerbefehl als erkannt auswerfen, so wäre die Verwechslungsfehlerrate gegebenenfalls relativ gering, nämlich gleich eins minus der Auftrittshäufigkeit dieses Steuerbefehls. Eine Steuerung der Umgebung wäre dabei allerdings nicht mehr möglich. Außerdem ist zu bedenken, dass die Benutzer oftmals auf falsch erkannten Steuerbefehlen beharren werden, weil sie ja jeweils eine bestimmte Aktion auslösen möchten[38]. Diese Situation lässt sich mittels einer Markov-Kette [Eier1994] [Eier1999] beschreiben. Die Zustände entsprechen dabei den einzelnen Wünschen, eine bestimmte Aktion auszulösen. Die Lauflängen der Zustände lassen sich annähernd geometrisch verteilt modellieren.

Bei Diktiersystemen ist es üblich, die A-priori-Wahrscheinlichkeiten mittels Bigrammstatistiken beziehungsweise Trigrammstatistiken aus grammatikalisch korrekten Beispielsätzen zu schätzen. Diese Möglichkeit besteht bei Sprachsteuerungen im Allgemeinen nicht, denn selbst wenn die zu einer Dialogsituation gehörenden A-priori-Wahrscheinlichkeiten genaugenommen von den vorangegangenen Dialogzuständen beziehungsweise Benutzereingaben abhängen, ist diese Abhängigkeit höchst anwendungsspezifisch und anwenderspezifisch und in der Praxis schwer zu ermitteln. Es verbleibt jedoch grundsätzlich die Möglichkeit, die A-priori-Wahrscheinlichkeiten aufgrund der Daten anderer Sensoren zu schätzen[39].

[37] Beispielsweise wird ein Steuerbefehl „Hilfe“ relativ selten auftreten. Dennoch sollte er, wenn er dann doch auftritt, mit vergleichbarer Wahrscheinlichkeit richtig klassifiziert werden, wie die anderen Steuerbefehle.

[38] Beispielsweise wird, um Hilfe zu holen, der dafür vorgesehene Steuerbefehl „Hilfe“ immer wieder probiert, solange bis er erkannt wird. Fehlerkennungen werden demgemäß überproportional häufig erlebt.

[39] Wenn beispielsweise mittels Sturzsensor ermittelt wurde, dass der Benutzer gestürzt ist, so kann davon ausgegangen werden, dass ein eventueller Steuerbefehl „Hilfe“ häufiger als normalerweise auftreten wird.

1.4 Neue Paradigmen der automatischen Spracherkennung

Seit den frühen 50er Jahren wird auf dem Gebiet der automatischen Spracherkennung geforscht. Beinahe seit den Anfängen wird der technologische Durchbruch als „kurz bevorstehend“ prophezeit, doch die Schwierigkeit der Mustererkennungsaufgabe wird, angesichts der Leichtigkeit mit der der Mensch sie bewältigt, leicht unterschätzt.

Hohe Erwartungen. Mittlerweile werden hohe Erwartungen an die Qualität von Spracherkennern gestellt und in neuerer Zeit auch erstmals ansatzweise erfüllt. Die hohen Erwartungen werden von mehreren Faktoren genährt: Wissenschaftliche Veröffentlichungen und Vorträge von Systementwicklern haben in der Regel den Zweck die eigene Arbeit im besten Licht darzustellen und betonen daher die Vorteile des eigenen Ansatzes gegenüber den Nachteilen. Wissenschaftliche Buchautoren befürchten unter Umständen, ihre Werke könnten im Lauf der Zeit von der technischen Entwicklung überholt werden, was dazu führt, dass der Stand der Technik meist sehr optimistisch dargestellt wird. Technische Systemdemonstrationen werden, um das System im besten Licht darzustellen, so durchgeführt, dass sie gelingen. Dies kann durch versteckte Laborbedingungen erfolgen, beispielsweise wenn in leiser Umgebung auf Non-Keyword-Schallereignisse verzichtet wird, oder einfach dadurch, dass ein sehr geübter Sprecher eine dem System angepasste Aussprache verwendet und Wörter beziehungsweise Wortfolgen spricht, deren problemlose Erkennung bekannt ist.

Mangelnde Vergleichbarkeit. Ein Problem stellt die unzureichende, evolutionäre Selektion von veröffentlichten Verfahren dar, da sich unterschiedliche Publikationen nur schwer vergleichen lassen. Im Allgemeinen wird lediglich die Verwechslungsfehlerrate veröffentlicht, obwohl der Benutzerzufriedenheit weit mehr Aspekte zugrunde liegen, wie beispielsweise die Falschakzeptanzrate und die Falschrückweisungsrate. Es sind zwar verschiedene Sprachdatenbanken öffentlich verfügbar, welche Vergleiche ermöglichen, allerdings sind diese Ergebnisse mit Vorsicht zu genießen, da die mehrmalige Verwendung derselben Sprachdatenbank, auch wenn sie in eine Testdatenbank und eine Trainingsdatenbank unterteilt wird, zu einer Anpassung des Verfahrens an diese Sprachdatenbank führt. Man spricht von „Overfitting“[40]. Die im Zustand „Overfitting“ erhaltenen exzellenten Fehlerraten werden natürlich gerne veröffentlicht und verglichen.

[40] Die Fähigkeit des Verfahrens, von der zur Verfügung stehenden Sprachdatenbank auf die Schallmuster des späteren Einsatzes zu verallgemeinern, ist dabei mangelhaft.

Neue Paradigmen. Das Paradigma einer „anthropomorphen Merkmalsextraktion“ wäre erstrebenswert, aber Erkenntnisse aus hörtheoretischen Modellen fließen erst sehr langsam, im Lauf der Jahre, in den Stand der Technik ein. Maskierungseffekte sollten Berücksichtigung finden. Hintergrundgeräusche und andere Stimmen sollten unterdrückt werden, das heißt, der sogenannte „Cocktail-Party-Effekt“ des menschlichen Gehörs sollte im Rahmen neuer Methoden der „Quellentrennung“ (siehe Abschnitt 1.4.3 auf Seite 99) nachgebildet werden [Kollmeier2002]. Ein weiteres neues Paradigma stellt das sogenannte „diskriminative Training“ dar (siehe Abschnitt 1.4.2 auf Seite 96).

1.4.1 Anthropomorphe Merkmalsextraktion

Es ist relativ naheliegend, sich auf dem Gebiet der Spracherkennung am besten aller Spracherkenner – dem menschlichen Gehör – zu orientieren. Daher wird zunächst auf den Aufbau und die Funktionsweise des menschlichen Gehörs eingegangen. Daraufhin wird der Einsatz von Perzeptionsmodellen zur Merkmalsextraktion auf dem Gebiet der Spracherkennung behandelt.

Menschliche Spracherkennung. Eine Reihe von Fähigkeiten, die derzeit durch automatische Spracherkenner nur unzureichend nachgeahmt werden können, vereinfachen die Aufgabe der menschlichen Spracherkennung im Vergleich zur automatischen Spracherkennung: Beispielsweise wird die Satzmelodie ausgewertet und es bestehen Erwartungshaltungen über die Stimmen der Sprecher. Dem menschlichen Gehirn stehen, neben Informationen, die es vom Ohr erhält, noch weitere Anhaltspunkte zur Verfügung, die zu Erwartungshaltungen bezüglich der gesprochenen Wörter führen. Es handelt sich beispielsweise um den situativen Kontext, den sprachlichen Kontext, die Redundanz des grammatikalischen Satzbaus sowie die beobachteten Lippenbewegungen.

1.4.1.1 Funktionsweise des menschlichen Gehörs

Außenohr. Mittelohr. Beim Aufbau des Ohres unterscheidet man Außenohr, Mittelohr und Innenohr. Das Außenohr wirkt bezüglich des Schallsignals als richtungsabhängiges Filter. Außenohr und Mittelohr sind mit Luft gefüllt und durch das Trommelfell getrennt. Die Auslenkung des Trommelfells beträgt an der Hörschwelle weniger als die Größenordnung der Atomdurchmesser. Es kann sich nur frei bewegen, wenn auf beiden Seiten die gleichen Druckverhältnisse vorliegen. Dafür sorgt die eustachische Röhre, die vom Mittelohr in den Nasenraum führt. Sie wird zum Druckausgleich geöffnet. Im Mittelohr werden die Schwingungen des Trommelfells durch die drei Gehörknöchelchen Hammer, Amboss und Steigbügel auf die Membran des „ovalen Fensters“ übertragen.

Innenohr. Die beiden Membranen des „ovalen Fensters“ und des „runden Fensters“ bilden die Grenze zwischen Mittelohr und Innenohr. Auf der Seite des Innenohrs liegt die Cochlea (siehe Abbildung 1-28, modifiziert aus Gray’s Anatomy[41] [Gray1918]). Es handelt sich um einen mit Perilymph-Flüssigkeit gefüllten schlauchartigen Hohlraum, welcher vom ovalen Fenster weg und dann daneben wieder zurück zum runden Fenster führt, das sich in unmittelbarer Nähe des ovalen Fensters befindet. Die „Hinleitung“, die sogenannte „Scala vestibuli“, und die „Rückleitung“, die sogenannte „Scala tympani“, sind am Umkehrpunkt, dem sogenannten „Helicotrema“ miteinander verbunden. Der Raum zwischen der „Scala vestibuli“ und der Scala tympani“ wird als „Scala media“ bezeichnet. Die gesamte Cochlea ist schneckenförmig mit etwa 2,75 Windungen aufgerollt.

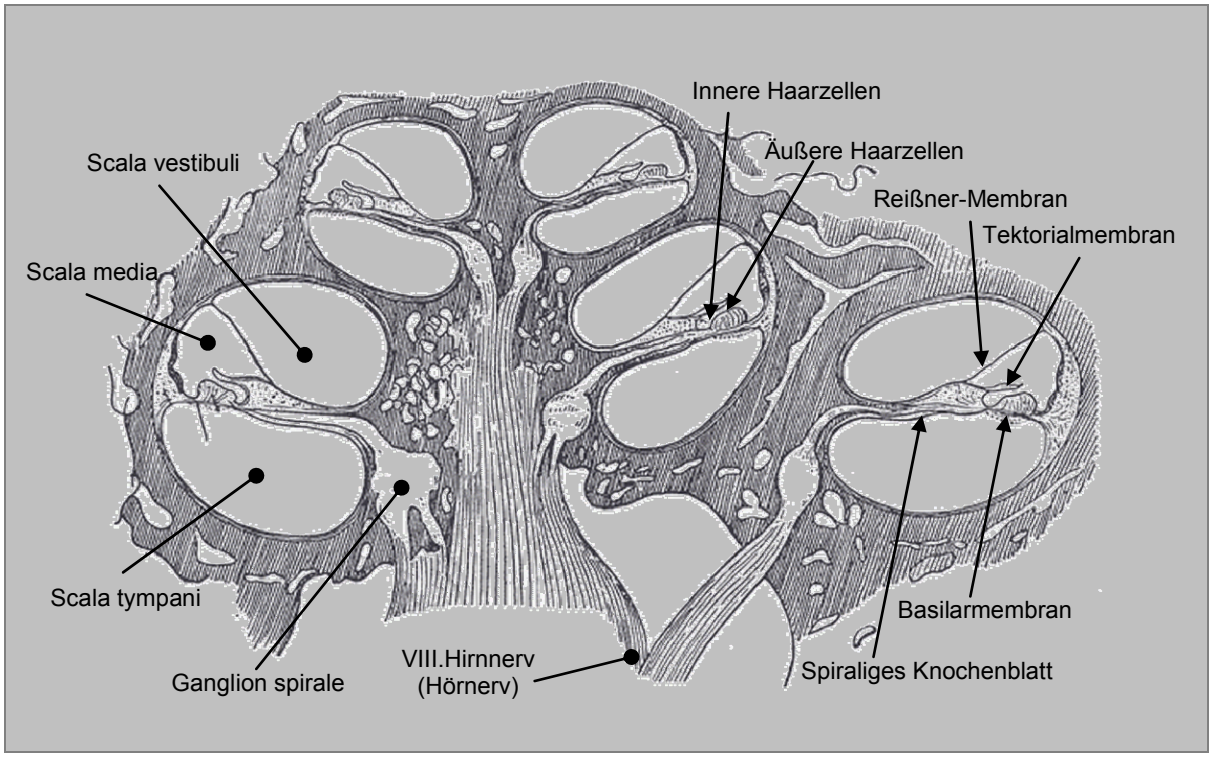

Abbildung 1-28: Querschnitt der Cochlea im Felsenbein.

Während die Perilymph-Flüssigkeit der „Scala vestibuli“ und „Scala tympani“ in der Zusammensetzung dem extrazellulären Milieu ähnelt, weist die Endolymph-Flüssigkeit der „Scala media“ eine hohe Kalium-Konzentration auf. Die Grenzschicht zwischen der „Scala vestibuli“ und der „Scala media“ wird als „Reißner-Membran“ bezeichnet. Die Grenzschicht der „Scala media“ zur „Scala tympani“ ist komplizierter aufgebaut und besteht zur „Scala tympani“ hin aus dem „spiraligen Knochenblatt“ und der „Basilarmembran“. Während der Durchmesser der „Scala media“ zum Helicotrema hin abnimmt, nimmt die Breite der Basilarmembran jedoch zum Helicotrema hin auf Kosten

[41] Public-Domain-Abbildung, Copyright weltweit abgelaufen.

des spiraligen Knochenblattes zu. Die mechanischen Eigenschaften und damit die Schwingungseigenschaften der Basilarmembran ändern sich somit in Abhängigkeit vom Abstand zum ovalen Fenster. In der Nähe des ovalen Fensters ist die Basilarmembran steif und deshalb resonant mit hohen Frequenzen, in der Nähe des Helicotrema ist sie nachgiebig und resonant mit tiefen Frequenzen. Die Endolymph-Flüssigkeit ist für hohe Frequenzen steif und für niedrige zunehmend nachgiebig.

Haarzellen. Auf der Basilarmembran liegt das sogenannte „Corti-Organ“ mit vier Reihen von Haarzellen. Jede Haarzelle besitzt ein Haarbündel sogenannter „Zilien“. Die Haarzellen sind von der sogenannten „Tektorialmembran“ überdeckt, wobei die drei äußeren Reihen, die sogenannten „äußeren Haarzellen“ mit den Spitzen ihrer Zilien in Kontakt mit der Tektorialmembran stehen. Die äußeren Haarzellen sind vorwiegend efferent innerviert, während die inneren Haarzellen vorwiegend afferent innerviert sind, genaugenommen erfolgt die Informationsübertragung jedoch bei beiden Haarzellarten in beiden Richtungen. Die einzelnen Zilien weisen Längen in aufsteigender Reihenfolge auf, deren Spitzen mit sogenannten „Tip-Links“ verbunden sind. Bei mechanischer Auslenkung des Haarbündels zu den längeren Zilien hin, spannen sich die Tip-Links und öffnen Ionenkanäle, was zum Einströmen von Kaliumionen führt, da die Endolymph-Flüssigkeit gegenüber der Perilymph-Flüssigkeit positiv geladen ist. Bei mechanischer Auslenkung in die andere Richtung schließen sich die Ionenkanäle. Ohne Ausrichtung sind einige Ionenkanäle geschlossen, andere hingegen offen. Funktional entspricht dieses Verhalten in etwa einer Halbweggleichrichtung. Das Einströmen von Kalium-Ionen bewirkt das Öffnen von Kalzium-Kanälen und das einströmende Kalzium bewirkt die Ausschüttung von Neurotransmittern. Weiters bewirken die einströmenden Kalium-Ionen das Öffnen weiterer Kalium-Kanäle zur Perilymph-Flüssigkeit hin, wodurch die Kalium-Ionen wieder ausströmen bis der Ausgangszustand erreicht ist. Man spricht im Zusammenhang mit der mechanisch-elektrischen Wandlung vom „Transduktionsprozess der Haarzellen“.

Tonotopische Codierung. Bei Anregung durch sinusförmige Schallsignale am ovalen Fenster findet sich eine Resonanzstelle maximal erhöhter Amplitude der mechanischen Schwingung an einer bestimmten Position entlang der Cochlea, wobei die Position durch die sogenannte „tonotopische Theorie“ beschrieben wird. Eine übliche Vorstellung besteht darin, dass die Cochlea, vereinfacht betrachtet, die Schallschwingung des ovalen Fensters in ihre spektralen Bestandteile zerlegt.

Volley-Codierung. Bei der Anregung durch sinusförmige Schallsignale zeigt sich, dass Neuronen des Hörnervs zu etwa äquidistanten Zeitpunkten elektrische Impulse, sogenannte „Spikes“ erzeugen, welche mit derselben Frequenz wie das Anregungssignal aufeinanderfolgen. Da ein einzelnes Neuron jedoch höchstens bis zu einer Frequenz von einem Kilohertz „feuern“ kann, wird angenommen dass eine gewisse Anzahl von Neuronen zusammenarbeitet, um durch die virtuelle Summe ihrer Aktivitäten ein Signal zu erreichen, in dem die Häufigkeit der Spikes der Frequenz des Anregungssignals entspricht. Man spricht dabei vom „Volley-Prinzip“ [Wever1930]. Weiters zeigt sich, dass es bei ausreichend kurzen transienten Signalen nicht unbedingt zu ausgeprägten

Resonanzstellen auf der Basilarmembran kommt, da das Anschwingen der Resonanzen gewisse Mindestzeitspannen benötigt. Dies ist ein Hinweis darauf, dass die Codierung des Schallsignals nicht in erster Linie beziehungsweise nicht nur über die Resonanzorte erfolgt, sondern vermutlich die relativen Zeitpunkte der Spikes dabei eine wesentliche Rolle spielen [Rattay1997].

Hörbahn. Die Hörbahn beginnt an den Haarzellen, welche über glutamaterge Synapsen Nervenzellen des Hörnervs erregen, deren Zellkörper im „Ganglion spirale“ liegen (siehe Abbildung 1-28 auf Seite 85). Von dort führen Nervenfasern zum „Nucleus cochlearis“ im Stammhirn (siehe Abbildung 1-29). Ein Teil der Nervenfasern führt weiter Richtung ipsilateralem (gleichseitigem) „Colliculus inferior“ im Mittelhirn. Der Großteil der Nervenfasern jedoch wird im kontralateralen „oberen Olivenkernkomplex“ verschaltet und zum kontralateralen „Colliculus inferior“ geführt. Vom „Colliculus inferior“ im Mittelhirn führen die Nervenfasern zum „Corpus geniculatum mediale“ im Zwischenhirn und von dort zum auditiven Cortex im Temporallappen. Nervenfasern kreuzen an verschiedenen Stellen zwischen beiden Seiten. Es existieren auch rückläufige Verbindungen in vielfältiger Art und Weise: Es gibt Verbindungen vom Cortex zurück zum „Corpus geniculatum mediale“ und zurück zum „Colliculus inferior“, vom „Colliculus inferior“ zurück zum „Nucleus cochlearis“ und von einem Kern des „oberen Olivenkomplexes“ zurück zu den äußeren Haarzellen. Neben dieser in Abbildung 1-29 dargestellten „klassischen Hörbahn“ (Abbildung modifiziert nach Adamy [Adamy2003]) wird von einer zusätzlichen „nicht-klassischen Hörbahn“ berichtet [Moller2002].

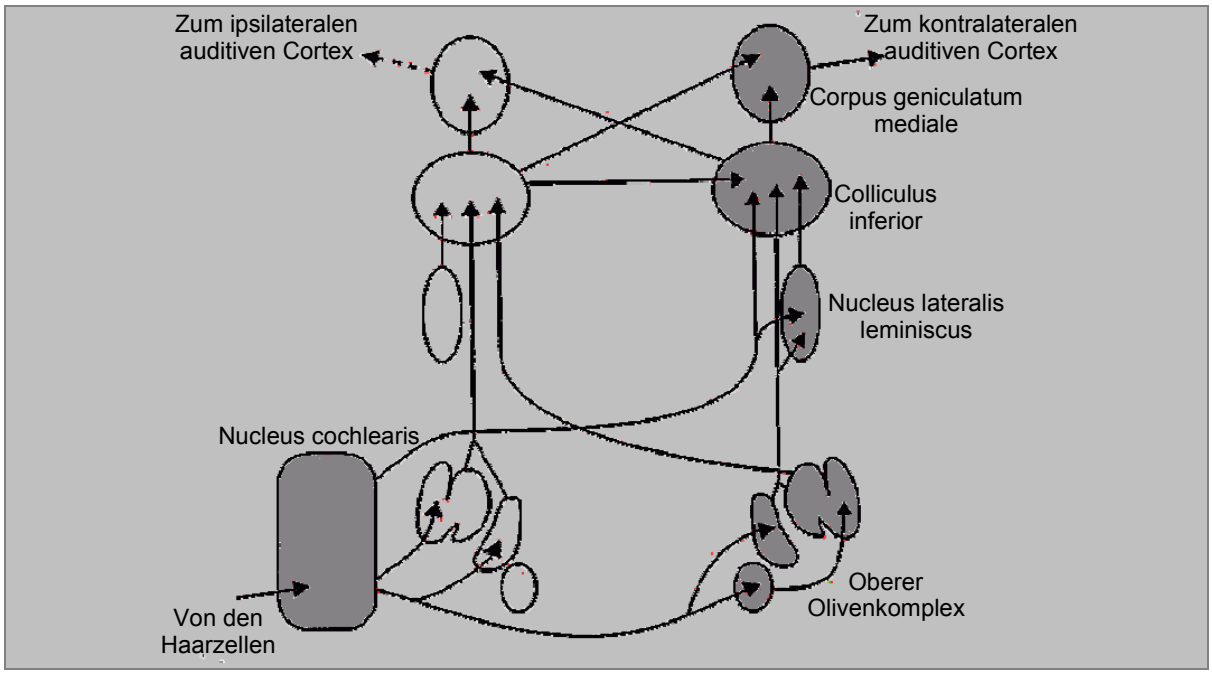

Abbildung 1-29: Klassische Hörbahn.

Zentrales Nervensystem. Als „Nervensysteme“ werden in der Biologie Strukturen bezeichnet, die aus einer Vielzahl hochgradig vernetzter Nervenzellen, den sogenannten „Neuronen“, bestehen. Das menschliche Gehirn besteht, je nach Zählweise, aus ungefähr zehn bis hundert Milliarden Neuronen, welche jeweils mit bis zu zehntausend anderen verbunden sind. Das selbstständige Lernen des Netzes geschieht durch die Auflösung und Neuschaffung von Verbindungen und durch die Veränderung der Eigenschaften der Verbindungen.

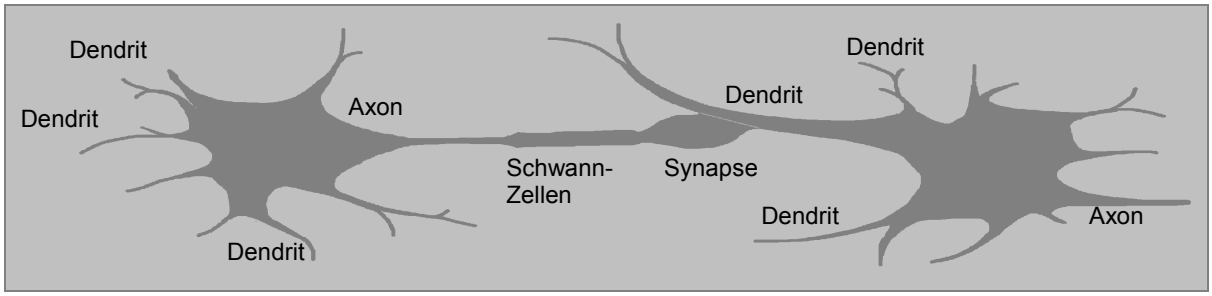

Abbildung 1-30: Verbindung zweier Nervenzellen über Axon, Synapse und Dendrit.

Abbildung 1-30 zeigt zwei verbundene Nervenzellen (Neuronen). Eine Nervenzelle kann von zahlreichen anderen Nervenzellen sowohl anregende als auch hemmende Reize über Synapsen erhalten, die an den baumartigen Dendriten kontaktieren. Im Zellkörper (Soma, Perikaryon), welcher Mitochondrien sowie den Zellkern beinhaltet, werden von den Anregungen gewissermaßen die Hemmungen in Abzug gebracht und im sogenannten „Axonhügel“ ein resultierender Reiz gebildet. Dieser wird über den Zellfortsatz (Axon) und die Synapse an die nachfolgende Nervenzelle weitergeleitet. Die Informationsübertragung erfolgt in der Regel mittels Neurotransmittern über den synaptischen Spalt. Dieser befindet sich zwischen beiden Membranen. Die in den Endköpfchen ankommende Erregung führt beispielsweise zu einer Einströmung von Kalziumionen an der präsynaptischen Membran. Daraus ergibt sich die Verschmelzung der Vesikeln, die sich im Endköpfchen befinden, mit der präsynaptischen Membran, wodurch in ihnen gespeicherte Neurotransmitter in den synaptischen Spalt ausgeschüttet werden. Die Neurotransmitter binden an der postsynaptischen Membran an bestimmte Rezeptoren, erregen die Membran elektrisch und vermitteln so ein Signal. Die hemmende beziehungsweise anregende Wirkung der Neurotransmitter ergibt sich im Zusammenhang mit der Art der involvierten Rezeptoren. Im synaptischen Spalt befindliche Neurotransmitter werden in den Synapsen wieder von der präsynaptischen Membran aufgenommen und in neue Vesikel verpackt, seltener werden die Neurotransmitter von Enzymen gespalten. Neben den Neuronen gehören auch die sogenannten „Schwann-Zellen“ (Bindegewebe) zum Nervensystem. Diese dienen nach heutigem Wissensstand in erster Linie der Umhüllung der Axone. So sind die Zellfortsätze der Nervenzellen von Schwann-Zellen umgeben, die mehrschichtige Isolationshüllen, sogenannte „Myelinscheiden“, ausbilden. Die Lücken am Axon, zwischen den hintereinander angeordneten

Schwann-Zellen, werden „Ranvier-Schnürringe“ genannt. Bei der saltatorischen Erregungsleitung springt die Erregung von Schnürring zu Schnürring, was die Informationsübertragung beschleunigt.

Spezialisierung der Hirnregionen. Aus historischer Sicht gilt das „Wernicke-Areal“ (Carl Wernicke, veröffentlicht 1874) als zuständig für die Sprachrezeption und das „Broca-Areal“ (Paul Broca, veröffentlicht 1861) als zuständig für die Sprachproduktion. Die Unterscheidung in „Wernicke-Areal“ und „Broca-Areal“ gilt jedoch als veraltet: Die Weiterentwicklung bildgebender Verfahren, wie etwa der Positronen-Emissions-Tomographie und der funktionellen Magnetresonanztomografie führt im Laufe der Zeit zu immer weiterer Verfeinerung der Zuordnung zwischen Aufgabenbereichen und Hirnregionen.

Modellbildung mittels künstlicher Neuronaler Netze. Künstliche Neuronale Netze werden in der Informatik im Rahmen der Grafentheorie behandelt. Sie bestehen aus Knoten und Kanten. Die Knoten, die in der Biologie den Nervenzellen entsprechen, verarbeiten die Eingangssignale mittels funktionaler Vorschriften zum Ausgangssignal und die Kanten, welche in der Biologie den Synapsen entsprechen, gewichten die Signale. Das im Neuronalen Netz gespeicherte Wissen ist vollständig durch die Struktur des Netzes und die Gewichte der Kanten festgelegt. Ein weitverbreitetes künstliches Neuronales Netz ist das Multi-Layer-Perceptron-Netz [Widrow1990] [Rojas1996]. Grund für die weite Verbreitung ist wohl die Verfügbarkeit einer relativ einfachen Trainingsmethode, der Backpropagation-Optimierung. Obwohl der physische Aufbau biologischer Neuronen bereits detailliert beschrieben werden kann, kann keine sichere Aussage darüber gemacht werden, inwiefern die gängigen technischen Modelle das Wesentliche am Lernvorgang und an der Wissensspeicherung dem natürlichen Nervensystem beschreiben [Eppinger1993]. Künstliche Neuronale Netze in digitalen Computern abstrahieren die Natur und erreichen oftmals nur begrenzt die Mächtigkeit und Genauigkeit der biologischen Vorbilder: Künstliche Neuronale Netze werden im Gegensatz zur Arbeitsweise natürlicher Nervensysteme in zeitlich diskreten Schritten gerechnet. Die verarbeiteten Signale unterliegen im Allgemeinen mehrfacher Quantisierung, obwohl die übliche Verwendung von Fließkommaarithmetik diesen Umstand mildert. Künstliche Neuronale Netze werden in vielen Fällen auf herkömmlichen Uni-Prozessor-Systemen simuliert, wodurch der Vorteil massiver Parallelität des natürlichen Nervensystems nicht genutzt wird. Parallelrechner, wie etwa Transputersysteme hätten jedoch das Potenzial, die Rechenzeiten der Algorithmen wesentlich zu verkürzen. Im natürlichen Nervensystem finden sich, je nach Aufgabenbereich, die unterschiedlichsten Typen von Neuronen, die den verschiedenen Erfordernissen angepasst sind. Künstliche Neuronale Netze sind dagegen meist nur aus einer Art von Knoten aufgebaut, um das Berechnungsmodel einfach zu halten. Künstliche Neuronale Netze sind fast immer mit statischer Topologie aufgebaut. Sie können also weder wachsen, noch neue Verbindungen knüpfen, noch alte Verbindungen vollständig auflösen, worin eine große Stärke der biologischen Architektur liegt.

1.4.1.2 Perzeptionsmodelle des menschlichen Gehörs

Die Modellierung psychoakustischer Eigenschaften des menschlichen Gehörs ist ein vielversprechendes Paradigma der modernen Sprachsignalverarbeitung. Perzeptionsmodelle bieten Gelegenheit, sich am besten aller Spracherkenner, dem menschlichen Gehör, zu orientieren. Das Gebiet der Merkmalsextraktion zum Zwecke der Spracherkennung entwickelt sich im Laufe der Zeit weg von Modellen, die die Spracherzeugung simulieren, beispielsweise LPC-Merkmalsvektoren oder PARCOR-Merkmalsvektoren, hin zu Perzeptionsmodellen, die das menschliche Gehör simulieren. Im Folgenden werden einige Perzeptionsmodelle angesprochen und auf die Merkmalsvektoren des Oldenburger Perzeptionsmodells, sowie auf die Merkmalsvektoren des Grazer „Anthropomorphic Coding Model“ näher eingegangen.

Boston-Perzeptionsmodell. Dieses Binaural-Perzeptionsmodell wird in Boston entwickelt. Es beschreibt verschiedene Fähigkeiten des Gehörs unter Berücksichtigung beider Ohren mittels expliziter Modellierung von Neuroneneigenschaften [Colburn1996].

Cambrige-Perzeptionsmodell. Dieses in Cambridge entwickelte Perzeptionsmodell setzt Schwerpunkte bei der Tonhöhenerkennung und der Wahrnehmung von Lautübergängen [Patterson1995].

Bochumer Perzeptionsmodell. Dieses Binaural-Perzeptionsmodell wird in Bochum entwickelt. Es legt den Schwerpunkt auf die Modellierung der Informationsverarbeitung des Gehörs, unter Berücksichtigung beider Ohren, mittels nachrichtentechnischer Funktionselemente [Blauert1996] [Bodden1995].

Münchner Perzeptionsmodell. Das in München entwickelte Perzeptionsmodell nach Zwicker beschreibt die Lautheitswahrnehmung in Abhängigkeit der Schallintensität, dem Schallspektrum und der Schalldauer. Eine Erweiterung des Modells beschreibt die Lautheitswahrnehmung bei Schwerhörigkeit. Die Modellierung von Maskierungseffekten nach Zwicker bildet weiters die Grundlage der MP3-Codierung [Zwicker1999].

Oldenburger Perzeptionsmodell. Das Oldenburger Perzeptionsmodell legt einen besonderen Schwerpunkt auf die temporalen Eigenschaften der Signalverarbeitung im Gehör und bildet eine relativ große Zahl von psychoakustischen Effekten quantitativ nach [Dau1996] [Dau1996a].

Grazer Perzeptionsmodell. Dieses in Graz entwickelte Perzeptionsmodell modelliert die Signalverarbeitung des menschlichen Gehörs mittels nachrichtentechnischer Funktionselemente. Das Bemerkenswerte dabei ist, dass der Höreindruck des ursprünglichen Audiosignals aus der Merkmalsvektorfolge wieder rekonstruiert werden kann. Es handelt sich sozusagen um ein Verfahren zur Audiosignalcodierung, das dem menschlichen Gehör nachempfunden ist. In diesem Zusammenhang wird das Grazer Perzeptionsmodell auch als „Anthropomorphic Coding Model“ bezeichnet [Feldbauer2005].

1.4.1.3 Merkmalsextraktion beim Oldenburger Perzeptionsmodell

Das Oldenburger Perzeptionsmodell wird am „Physikalischen Institut“ in Göttingen beziehungsweise ab 1993 an der Universität Oldenburg als Modell für die Signalverarbeitung im menschlichen Gehör entwickelt, um das Antwortverhalten von Versuchspersonen in psychoakustischen Experimenten über temporale und spektrale Maskierungseffekte quantitativ nachzubilden [Dau1996] [Dau1996a] [Dau1997] [Dau1997a]. Das Oldenburger Perzeptionsmodell bildet eine relativ große Zahl von psychoakustischen Effekten des menschlichen Gehörs quantitativ nach. Es basiert auf einer geringen Zahl von Annahmen und Parametern, die in wenigen, „kritischen“ Experimenten festgelegt und für die quantitative Beschreibung anderer Experimente nicht mehr variiert werden.

In weiterentwickelter Form findet es in der Sprachqualitätsmessung[42] [Hansen1996], für Cochlea-Implantate sowie als geeignete Vorverarbeitung zur robusten Spracherkennung [Hartmann2000] [Kasper1997] [Tchorz1996] [Tchorz1997] [Tchorz1999] [Kleinschmidt1998] [Kleinschmidt1999] [Kleinschmidt2000] Verwendung. Spracherkennungsexperimente wurden auch im Zusammenhang mit Richtungsfilterung und „Feature Finding Neural Networks“ durchgeführt [Kleinschmidt1998] [Kleinschmidt2000].

Bei Hörgeräten erschließt sich neben der Vorhersage der Klangqualität noch eine weitere Anwendung[43]: Es wird versucht, das Hörgeräteausgangsschallsignal für Personen mit Hörverluststörungen rechnerisch so zu verändern, dass die modellierte interne Repräsentation möglichst der eines durchschnittlichen Normalhörers bei unmodifiziertem Schallsignal entspricht. Algorithmen zur Richtungsfilterung, motiviert durch den Cocktail-Party-Effekt [Kollmeier2002], werden in diesem Zusammenhang ebenfalls eingesetzt [Wittkop1997] [Wittkop1997a].

---

[42] Unter „Sprachqualitätsmessung“ wird die Messung der Qualität einer Sprachsignalübertragung verstanden. Es existieren hierzu derzeit eine Reihe genormter Verfahren, unter anderen: PESQ und TOSQA.

[43] Quelle: Homepage der Arbeitsgruppe „Medizinische Physik“ an der Universität Oldenburg http://medi.uni-oldenburg.de (Stand: 26. 05. 2004).

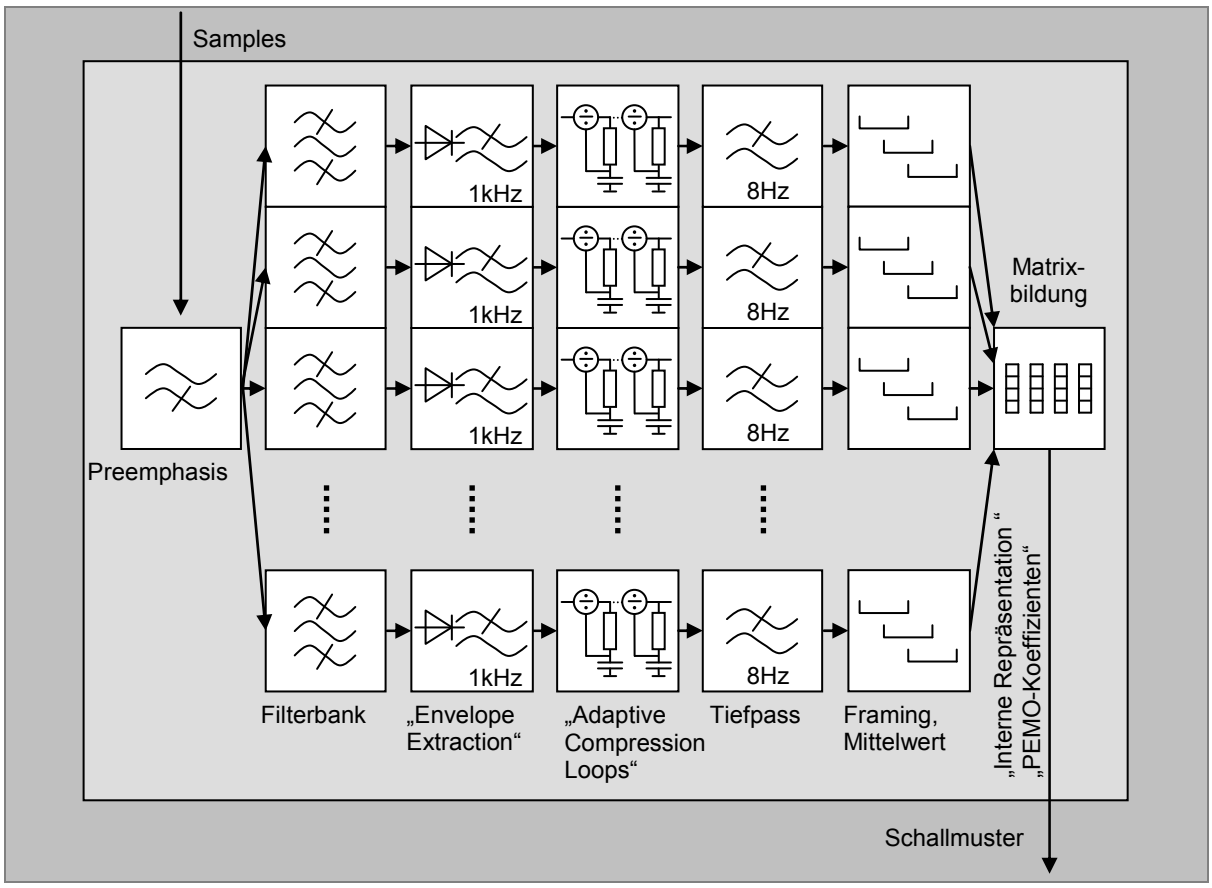

Abbildung 1-31: Verarbeitungsschritte im Feature-Extractor-Funktionsblock bei der Berechnung von Merkmalsvektoren des Oldenburger Perzeptionsmodells.

Beim Oldenburger Perzeptionsmodell im eigentlichen Sinne handelt es sich um eine Verarbeitungskette, welche ein Schallereignis in eine Folge von Merkmalsvektoren umsetzt. Die Merkmalsvektoren werden dabei als „Interne Repräsentation“ bezeichnet. Die Verarbeitungskette besteht wie folgt:

Übertragungseigenschaften des Außenohrs. Es erfolgt eine „Preemphasis“ (Höhenanhebung) zur groben Nachbildung der linearen Übertragungseigenschaften des Außenohrs.

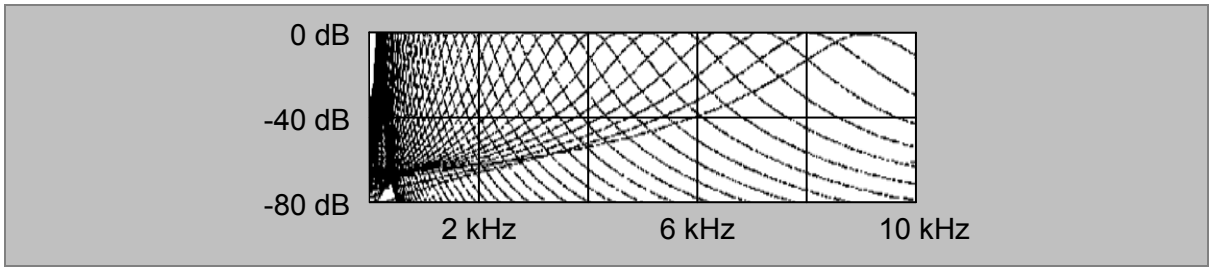

Abbildung 1-32: Amplitudengänge der Filterbank des Oldenburger Perzeptionsmodells. Es handelt sich um stark überlappende Gammatonfilter. Auf der Abszisse ist die Frequenz und auf der Ordinate negative Filterdämpfung aufgetragen.

Tonotopische Theorie des Gehörs. Die sogenannte „Tonotopische Theorie“ beschreibt die Eigenschaft des Gehörs, harmonische Schwingungen gemäß ihrer Frequenz einer bestimmten Stelle der Basilarmembran der Cochlea zuzuordnen. Wie in Abbildung 1-31 auf Seite 92 und Abbildung 1-32 dargestellt, wird dazu das Zeitsignal mittels einer Filterbank in einzelne Filterbankkanäle zerlegt. Dabei kommen sogenannte „Gammatonfilter“ zum Einsatz. Dabei handelt es sich um Filter mit bestimmter „Gammatonimpulsantwort“ [Hohmann2002].

Transduktionsprozess der Haarzellen. Es erfolgt die Extraktion der Hüllkurve durch Einweggleichrichtung mit anschließender Anwendung eines 1kHz-Tiefpasses erster Ordnung. Diese Bildung der Einhüllenden für höhere Frequenzen wird physiologisch motiviert durch den Transduktionsprozess der Haarzellen der Cochlea. Es erfolgt die Nachbildung temporaler Verdeckungseffekte durch adaptive Amplitudenkompression. Die adaptive Amplitudenkompression besteht aus fünf Elementen mit Automatic-Gain-Control-Funktionalität. Jedes Element besteht aus einem Divisions-Funktionsblock, dessen Ausgang auf den zweiten Eingang über einen Tiefpass erster Ordnung rückgekoppelt ist. Siehe dazu Abbildung 1-31 auf Seite 92. Diese Regelkreise haben die Eigenschaft, die Amplitude langsam veränderlicher Eingangssignale annähernd[44] zu logarithmieren. Schnelle Schwankungen werden dagegen nahezu linear übertragen, da sich die rückgekoppelten Signale nur entsprechend der Zeitkonstanten[45] der Tiefpässe langsam ändern können.

Berechnung der Internen Repräsentation. Es folgt ein Tiefpass erster Ordnung mit einer Grenzfrequenz von circa 8 Hz, um höhere Frequenzanteile der Einhüllenden zu dämpfen. Eine weitere Datenreduktion, beispielsweise durch Mittelung der Zeitsignale der einzelnen Kanäle über einen Zeitraum von 10 ms, führt zu den PEMO-Merkmalsvektoren. Diese Merkmalsvektoren werden auch als „Interne Repräsentation“ bezeichnet.

1.4.1.4 Merkmalsextraktion beim Grazer Perzeptionsmodell

Das Grazer Perzeptionsmodell beziehungsweise „Anthropomorphic Coding Model“ [Feldbauer2005] wurde an der Technischen Universität Graz entwickelt. Es handelt sich wie beim Oldenburger Perzeptionsmodell um eine Verarbeitungskette, welche ein Schallereignis in eine Folge von Merkmalsvektoren umsetzt. Dabei wird ebenfalls darauf geachtet, der Signalverarbeitung im menschlichen Gehör so nahe wie möglich zu kommen.

---

[44] $e^x = \lim_{N\to\infty}\left(1+\frac{x}{N}\right)^N ; \quad N = 2^5 = 32; \quad e^x \approx \left(1+\frac{x}{32}\right)^{32} ; \quad x' = e^x \quad \Rightarrow \sqrt[32]{x'} \approx 1+\frac{\ln x'}{32}$

[45] Die fünf Zeitkonstanten werden experimentell bestimmt, sodass sie Maskierungseffekte des menschlichen Gehörs widerspiegeln. Sie liegen etwa im Bereich von 5 ms bis 500 ms.

Rücktransformation. Das besondere am „Anthropomorphic Coding Model“ ist, dass das ursprüngliche Audiosignal wiederum aus der Merkmalsvektorfolge rekonstruiert werden kann.

Der Vorteil beim Einsatz im Feature-Extractor-Funktionsblock in Abbildung 1-2 auf Seite 17 zur Merkmalsextraktion zum Zwecke der Spracherkennung besteht wie folgt: Die Effekte weiterer Verarbeitungsschritte, beispielsweise zum Zwecke der Datenreduktion, können vor der Klassifikation mittels der Rekonstruktion hörbar gemacht werden. Dadurch wird eventuell mittels des menschlichen Gehörs eine Einschätzung möglich, welcher Anteil klassifikationsrelevante Information verloren geht, was die Entwicklung des Feature-Extractor-Funktionsblocks vereinfacht, weil unnötige Testläufe mit Sprachdatenbanken eingespart werden können[46]. Die Verarbeitungskette besteht wie folgt:

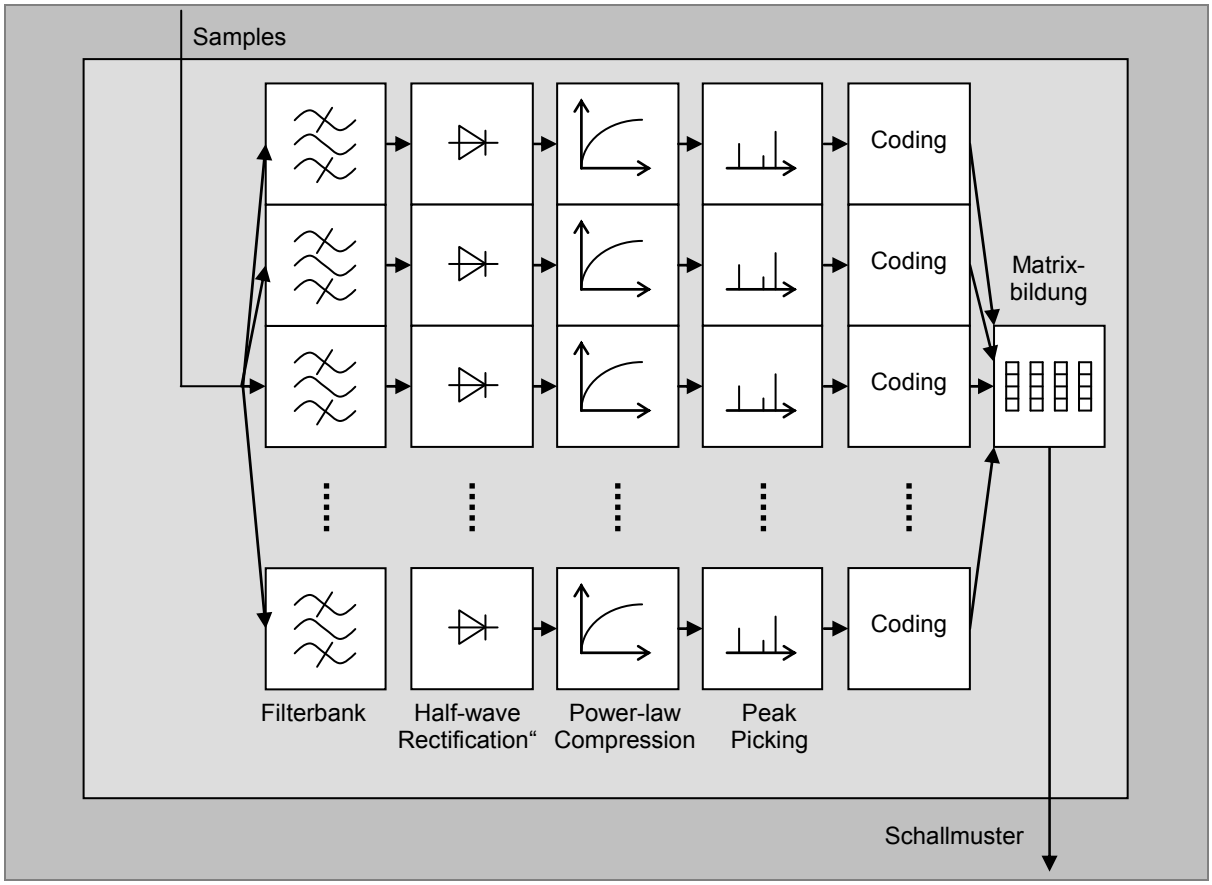

Abbildung 1-33: Verarbeitungsschritte im Feature-Extractor-Funktionsblock bei der Berechnung von Merkmalsvektoren nach dem „Anthropomorphic Coding Model“.

[46] Die Optimierung des Feature-Extractor-Funktionsblocks anhand der Verwechslungsfehlerrate bei Sprachdatenbanktests im Rahmen der Algorithmenentwicklung ist durchaus üblich, wenn nicht auf genormte Algorithmen zur Berechnung der Merkmalsvektoren zurückgegriffen wird.

Übertragungseigenschaften des Außenohrs. Die Übertragungseigenschaften des Außenohrs werden nicht explizit modelliert.

Tonotopische Theorie des Gehörs. Wie beim „Oldenburger Perzeptionsmodell“ wird eine Filterbank in Anlehnung an die „Tonotopische Theorie“ eingesetzt. Die „Tonotopische Theorie“ beschreibt die Eigenschaft des Gehörs, harmonische Schwingungen gemäß ihrer Frequenz einer bestimmten Stelle der Basilarmembran der Cochlea zuzuordnen. Es finden ebenfalls „Gammatonfilter“ und eine nichtlineare Frequenzskala Verwendung.

Transduktionsprozess der Haarzellen. Der Transduktionsprozess der Haarzellen der Cochlea wird zunächst ähnlich dem „Oldenburger Perzeptionsmodell“ mittels Einweggleichrichtung modelliert. Eine nichtlineare Funktion wird jedoch anstelle der Nachregelschleifen des „Oldenburger Perzeptionsmodells“ eingesetzt. Andere Modelle setzen an dieser Stelle Elemente mit Automatic-Gain-Control-Funktionalität ein, um das Verhalten der Synapsen zwischen den Haarzellen und den Nervenfasern zu modellieren [Ivanov2005]. Die einfache Modellierung mittels einer nichtlinearen Funktion wurde gewählt, um die Rücktransformation in einfacher Weise zu ermöglichen. Die Rücktransformation einer Automatic-Gain-Control-Funktionalität würde zu Stabilitätsproblemen führen [Feldbauer2005].

Gruppierung von Neuronen. Man geht davon aus, dass Nervenzellen in ihrem Feuerverhalten der ursprünglich mechanischen Anregung nur bis zu einer gewissen Frequenz folgen können. Daher wird angenommen, dass jeweils eine Gruppe von Nervenzellen für eine Haarzelle zuständig ist. Zur Modellierung des Feuerverhaltens einer solchen Neuronengruppe wird der sogenannte „Peak-Picking-Algorithmus“ angewendet. Dabei werden in jedem Filterbankkanal jene Samples gesucht, die größer als das Vorgängersample und größer als das Nachfolgersample sind. Alle anderen Samples werden null gesetzt.

Berechnung der Merkmalsvektoren. Es folgt die Quantisierung und die Codierung der Vektoren. Zum Zwecke der Klassifikation von Schallereignissen müssen die Vektoren in, für die jeweilige Realisierung des Classifier-Funktionsblock nach Abbildung 1-2 auf Seite 17 geeignete, Merkmalsvektoren transformiert werden, wobei im Allgemeinen eine weitere Datenreduktion erfolgt.

1.4.2 Diskriminatives Training

Klassisches Training. Die klassische Sichtweise der Mustererkennung besteht darin, dass der Classifier-Funktionsblock nach Abbildung 1-1 auf Seite 16 konstruiert ist, um Distanzen des zu klassifizierenden Schallmusters zu den einzelnen Schallmusterklassen zu berechnen[47]. Das Ziel des Trainings des Classifier-Funktionsblocks besteht nun darin: Die freien Parameter des Classifier-Funktionsblocks werden optimiert, indem die Distanzen der Schallmuster zu den jeweilig zugehörenden Klassen optimiert werden. Beispielsweise werden die Quadrate der Summen der Distanzen minimiert. Zur Klassifikation eines Schallmusters wird nun eine „Rank List“ berechnet und die Klasse mit der geringsten Klassendistanz an erster Position der „Rank List“ als „Gewinner“ ausgegeben[48]. Man spricht von „Minimum-Distance-Auswahl“.

Kritik am klassischen Training. Dass das klassische Training nicht optimal sein kann, sieht man schon daran, dass die Anforderung an das Distanzberechnungsmodul des Classifier-Funktionsblocks, für jedes Schallmuster bezüglich der zugehörigen Klasse eine möglichst geringe Klassendistanz zu liefern, alleine nicht ausreicht, um die Klassifikationsaufgabe zu lösen. Die Klassifikation mittels der Minimum-Distance-Auswahl funktioniert nämlich eigentlich nur deshalb, weil das Distanzberechnungsmodul, mehr oder weniger unkontrolliert, für die jeweils klassenfremden Schallereignisse höhere Klassendistanzen liefert, was daran liegt, dass sie einfach nicht zum Training herangezogen werden.

Diskriminatives Training. Beim diskriminativen Training werden zum Training der einzelnen Klassen auch diejenigen Schallmuster, welche nicht zu den jeweiligen Klassen gehören, verwendet. Dabei wird das Distanzberechnungsmodul dahin gehend trainiert, für die jeweils klassenfremden Schallereignisse stets hohe Klassendistanzen zu liefern. Als Optimalitätsmaß bietet sich bei pragmatischer Vorgangsweise der „Minimum Classification Error“ an, obwohl auch andere Optimalitätsmaße denkbar sind und auch gelegentlich angewendet werden. Siehe dazu Abschnitt 1.4.2.3 auf Seite 97.

Nachteile des diskriminativen Trainings. Ein Nachteil des diskriminativen Trainings gegenüber dem klassischen Training besteht darin, dass zum Training jeder einzelnen Klasse auch die Trainingsdaten der anderen Klassen zur Verfügung stehen müssen. Daher ist es prinzipiell nicht mehr möglich, modulare Systeme aufzubauen, bei denen sich Wortmodelle einfach hinzufügen lassen, ohne die bestehenden Wortmodelle neu trainieren zu müssen. Ein weiterer Nachteil besteht darin, dass die Algorithmen im Allgemeinen komplizierter aufgebaut sind, und weil bezüglich jeder Klasse alle Trainings-

---

[47] Die Berechnung kann auf verschiedenste Art und Weise erfolgen: Etwa mit einem oder mehreren Neuronalen Netzen, mit Hidden-Markov-Modellen sowie mit vielen anderen Verfahren. Direkt oder indirekt werden neben einem eventuellen Sprechtempoausgleich im Allgemeinen heuristische oder physiologisch motivierte Distanzfunktionen eingesetzt, wie beispielsweise die quadrierte euklidische Distanz oder die Itakura-Saito-Distanz.

[48] Diese Sichtweise des Klassifikationsproblems hat nach wie vor große Bedeutung und ist sehr verbreitet.

muster berücksichtigt werden müssen, im Allgemeinen auch die Rechenzeit des Trainings wesentlich höher ist.

1.4.2.1 Diskriminatives Classifier-Training

Diskriminatives Training des Classifier-Funktionsblocks wird in der Literatur beispielsweise bezüglich „Hidden Control Neural Network Speech Recognizer“ [Na1995] [Sorensen1995], „Hidden Markov Model Speech Recognizer“ [Bourlard1990], „Dynamic Time Warping Speech Recognizer“ [Chang1992] [Juang1992] sowie bezüglich einer Vielzahl anderer Verfahren [McDermott1994] beschrieben.

1.4.2.2 Diskriminatives Feature-Extractor-Training

Der Gedanke des diskriminativen Trainings lässt sich auch auf den Feature-Extractor-Funktionsblock nach Abbildung 1-2 auf Seite 17 übertragen. Gerade der Feature-Extractor-Funktionsblock bietet enormes Potenzial zur Verbesserung der Klassifikationsfähigkeiten von Sprachsteuerungen. Es zeigt sich, dass die Veränderung von Parametern dieses Funktionsblocks große Auswirkungen auf die Fehlerraten des Gesamtsystems haben. Daher ist der Gedanke verlockend, verschiedene kritische Parameter nicht von vornherein festzulegen, sondern während des Trainings automatisch optimieren zu lassen [Biem2001] [Mak2004]. Diskriminatives Training wurde auch angewendet um neben den freien Parametern des Classifier-Funktionsblocks zusätzlich freie Parameter des Feature-Extractor-Funktionsblocks zu trainieren [Biem2003].

1.4.2.3 Minimum Classification Error Training

Beim Minimum-Classification-Error-Training handelt es sich um diskriminatives Training, wobei eine Funktion der Fehlerraten bezüglich der Trainingsdatenbank als Optimalitätsmaß herangezogen wird[49]. Im Allgemeinen können die freien Parameter des Classifier-Funktionsblocks oder auch die freien Parameter des Feature-Extractor-Funktionsblocks oder beide Arten von Parametern mittels Random-Search-Optimierung, Hill-Climbing-Search-Optimierung oder anderen geeigneten Optimierungsverfahren

[49] Dieser Ansatz entfaltet seine Stärke bei schwierigen Klassifikationsproblemen, also bei Aufgabenstellungen, bei denen nicht selbstverständlich ist, dass die Trainingsschallmuster richtig klassifiziert werden. Als Beispiel sei das Training sprecherunabhängiger Spracherkenner mittels großer Sprachdatenbanken genannt. Der Ansatz ist aber natürlich auch für einfachere Klassifikationsaufgaben einsetzbar. Man hofft im Übrigen, wie beim klassischen Ansatz, auf eine Verallgemeinerung der Klassifikationsfähigkeit auf die Schallmuster der Testdatenbank.

eingestellt werden. Wenn man den Begriff „Parameter“ so allgemein betrachtet, dass ein Parameter auch ganze Algorithmen beziehungsweise Algorithmenteile umschalten kann, so ergibt sich ein sehr allgemeines mächtiges automatisches Verfahren, das bei näherer Überlegung schon der menschlichen Vorgangsweise bei der Evaluierung verschiedener Algorithmen und Einstellungen nahe kommt[50].

„Corrective Minimum Classification Error Training“. Prinzipiell besteht, wenn dem Spracherkenner genügend Rechenleistung und zuverlässige Informationen über die Richtigkeit der Klassifikation während des Betriebs zur Verfügung stehen, auch die Möglichkeit, den Spracherkenner während des Betriebs nachzujustieren oder völlig neu zu trainieren, wobei eine Adaption an die aktuellen Gegebenheiten stattfindet. Diese Vorgangsweise wird als „Corrective Training“ bezeichnet. Die Wahl eines Optimalitätsmaßes als Funktion der beobachteten Fehlerraten ist hierbei im Allgemeinen sinnvoll.

Gradientenabstiegsverfahren. Wenn das Mustererkennungssystem so aufgebaut ist, dass sich zwischen den freien Parametern und der Anzahl der Verwechslungsfehler oder zumindest zwischen den freien Parametern und einem Maß, das stark mit der Anzahl der Verwechslungsfehler korreliert, ein differenzierbarer Zusammenhang herstellen lässt, so können Gradientenabstiegsverfahren angewendet werden [Rojas1996].

[50] Der Unterschied zur menschlichen Evaluierung verschiedener Algorithmen und Einstellungen besteht darin, dass der Mensch im Allgemeinen einen relativ abstrakten Plan verfolgt und verschiedene Algorithmenkombinationen von vornherein ausschließt. Weiters hat der Mensch die Möglichkeit, neue Algorithmen hinzuzufügen.

1.4.3 Quellentrennung und Geräuschunterdrückung

„Noise Reduction“. Störgeräusche sollen bei der Berechnung des Schallmusters mittels des Pattern-Builder-Funktionsblocks nach Abbildung 1-1 auf Seite 16 so weit wie möglich unterdrückt werden. Man spricht dabei von „Noise Reduction“. Eine Übersicht über die gängigsten Verfahren findet sich in der Literatur [Vaseghi1997]. Geräuschreduktionsverfahren sind teilweise neurophysiologisch motiviert und beruhen auf der Signalgeräuschabstandsbestimmung in einzelnen Frequenzbändern [Tchorz2001]. Ein weitverbreitetes Verfahren zur „Noise Reduction“ wird als „Spectral Subtraction“ bezeichnet [Sovka1996]. Analysen der Grundfrequenz beziehungsweise der Harmonischen der Stimmbandschwingung, beispielsweise mittels Autokorrelation, können im Zusammenhang mit „Spectral Subtraction“ vorteilhaft angewendet werden [Beh2003]. Geräuschunterdrückungsalgorithmen werden insbesondere in Hörgeräten eingesetzt [Marzinzik2000]. Die Kombination von Geräuschreduktionsverfahren und psychoakustisch motivierten Merkmalsvektoren des „Oldenburger Perzeptionsmodells“ zum Zwecke der automatischen Spracherkennung erweist sich als vorteilhaft [Kleinschmidt1999]. Ein interessanter Ansatz besteht darin, Neuronale Netze einzusetzen, um Merkmalsvektoren so zu transformieren, dass sie so weit wie möglich unbeeinträchtigten Merkmalsvektoren entsprechen [Barbier1991] [Sorensen1992a].

„Noise Cancellation“. Wird das Störgeräusch mittels eines zweiten Mikrofons aufgenommen, so kann dieses Signal verwendet werden, um das gewünschte Signal zu rekonstruieren. Dabei wird im Allgemeinen eine adaptive Schätzung des Unterschiedes der Übertragungsfunktionen des Schallfeldes von der Störquelle zu den beiden Mikrofonpositionen durchgeführt. Man spricht von „Noise Cancellation“ [Martinez2001].

„Source Separation“. Ein nach derzeitigem Stand der Technik hochgestecktes Ziel besteht darin, das Signal der zu analysierenden Schallquelle aus dem gesamten Schallsignal, das heißt aus der Überlagerung der Signale mehrerer Schallquellen, zu extrahieren. Derzeit werden zwei grundlegende Ansätze zur Lösung dieses Problems vertreten: Der eine basiert auf möglichst wenigen Annahmen über die Eigenschaften der zu trennenden Signale und wird als „blinde Quellentrennung“ beziehungsweise „Blind Source Separation“ bezeichnet [Graupe1992]. Notwendig ist dazu allerdings meist die Verwendung mehrerer Mikrofone an unterschiedlichen Positionen [Anemüller2001] [Anemüller2002]. Die Anordnung der Mikrofone als „Microphone Array“ kann sich als vorteilhaft erweisen [Yamada2002] [Moore2003]. Methoden der Richtungsfilterung werden ebenfalls angewendet [Kleinschmidt1998] [Wittkop1997] [Wittkop1997a]. Der zweite Ansatz, welcher auch als „auditorische Quellentrennung“ [Eggink2001] bezeichnet wird, beruht auf einer Dekomposition des zu verarbeitenden Signals in seine spektralen Komponenten, die anschließend zu verschiedenen Schallereignissen, das heißt zu einzelnen Tönen, Stimmen oder sonstigen Schallereignissen gruppiert werden. Dabei wird

oftmals mit nur einem einzigen Mikrofon das Auslangen gefunden [Eggink2001]. Beispielsweise kann der Umstand ausgenutzt werden, dass stimmhafte Laute als quasiperiodische Signale spektral aus der Stimmbandgrundschwingung und den Harmonischen zusammengesetzt und somit vom Rest des Spektrums trennbar sind [Parsons1976]. Ein interessanter Ansatz zur „auditorischen Quellentrennung“ mit nur einem Mikrofon besteht darin, mit den verwendeten Signalverarbeitungsalgorithmen so weit wie möglich die Signalverarbeitung im menschlichen Gehör nachzuahmen [Rouat2005].

2. Anforderungsanalyse

Systematik der Behinderungen
Die speziellen Bedürfnisse der angestrebten Zielgruppe, nämlich der Menschen mit Behinderungen, werden in diesem Abschnitt im Hinblick auf das Systemdesign einer Sprachsteuerung systematisiert.

Mögliche Hilfestellungen bei der Rehabilitation
In diesem Abschnitt wird der mögliche Nutzen der automatischen Sprach-erkennung in der Rehabilitationstechnik herausgearbeitet. Verschiedene Studien auf diesem Gebiet sowie etablierte Anwendungen werden behandelt.

Abgeleitete Anforderungen an die Sprachsteuerung
Die speziellen Bedürfnisse der Zielgruppe führen zusammen mit der Systematik der möglichen Hilfestellungen in der Rehabilitationstechnik zu bestimmten Entscheidungen bezüglich des Systemdesigns, welche in diesem Abschnitt festgehalten werden. Es wird eine Sprachsteuerung zur Steuerung der Umgebung im Sinne des Design-for-all-Prinzips realisiert.

2.1 Systematik der Behinderungen

Im Folgenden werden verschiedene Beeinträchtigungen beziehungsweise Behinderungen systematisch angeführt. Allgemein ist anzuführen, dass soziale beziehungsweise berufliche Integrationsprozesse so weit wie möglich gefördert werden sollten und dass im Sinne eines konstruktiven gemeinschaftlichen Umgangs, der Fokus weg von der direkten Ansprache der Beeinträchtigung beziehungsweise Behinderung hin zur Ansprache der verbleibenden oder sogar überdurchschnittlich kompensierenden Fähigkeiten verlagert werden sollte.

2.1.1 Motorische Störungen

Motorische Störungen und Behinderungen können von leichten Funktionseinschränkungen bis zur absoluten Unbeweglichkeit reichen und umfassen die gesamte Palette der Schädigungen des Bewegungsapparats. Die Ursachen und Auswirkungen sind vielfältig.

Eine grobe ursächliche Einteilung umfasst Gliedmaßenschädigungen, Gelenkschädigungen, Nervensystemschädigungen sowie Muskelschädigungen. Zu den Gliedmaßenschädigungen gehören beispielsweise angeborene Dysmelien, Amputationen sowie erworbene Skelettschädigungen. Zu den Gelenksschädigungen gehören unter anderen: Arthritis, Arthrose sowie Morbus Bechterew. Zu Schädigungen des Nervensystems gehören angeborene oder erworbene Schädigungen des Gehirns und des Rückenmarks. Dazu zählen beispielsweise: Multiple Sklerose, Gehirnschädigungen durch Schädel-Hirn-Traumata oder Schlaganfälle. Zu den Muskelschädigungen gehören zum Beispiel: Muskeldystrophien, Myositis und Myasthenie.

Systemdesign von Sprachsteuerungen. Meist empfiehlt sich der Einsatz einer Sprachsteuerung in Kombination mit herkömmlichen Sensoren, wie etwa Tastern, Saug-Blas-Schaltern, Neigungssensoren am Brillengestell oder Ähnlichem, da die Benutzung einer Sprachsteuerung bei leichten Bewegungseinschränkungen mühsamer als die Betätigung eines an den Benutzer angepassten Sensors sein kann. Im Falle schwerer Bewegungseinschränkung dürfen keinerlei motorische Eingaben notwendig sein. Allerdings sollten motorische Eingabemöglichkeiten zusätzlich vorhanden sein, um einer Betreuungsperson die Interaktion mit dem System zu ermöglichen, vor allem, wenn es sich um eine sprecherabhängige Sprachsteuerung handelt, die an die Stimme des Benutzers angepasst betrieben wird.

2.1.2 Stimmstörungen und Sprechstörungen

Bei einer Sprechstörung beziehungsweise Stimmstörung handelt es sich um eine Beeinträchtigung bei der korrekten und flüssigen Artikulation der Laute der Sprache. Es handelt sich um eine Beeinträchtigung in der Verwirklichung lautlicher Sprechnormen. Im Gegensatz zur Sprachstörung, das heißt, einer Beeinträchtigung der gedanklichen Erzeugung oder Analyse von Sprache, sind hier nur die motorischen beziehungsweise artikulatorischen Fertigkeiten beeinträchtigt. Das Sprachvermögen an sich kann dabei intakt sein. Sprachstörungen und Sprechstörungen können jedoch auch gemeinsam auftreten. In den folgenden Abschnitten wird zunächst auf die unbeeinträchtigte Lautbildung und sodann auf konkrete Stimmstörungen und Sprechstörungen eingegangen. Zur detaillierten Information sei auf Spezialliteratur, verwiesen [Böhme1997] [Hildebrandt1998]. Störungsbilder betreffend Stimmstörungen und Sprechstörungen werden auch im Sinne einer ursachenbezogenen Einteilung eigens bezeichnet. Beispiele für ursachenbezogene Bezeichnungen sind beispielsweise „Dysarthrie“ beziehungsweise „Dysarthrophonie“. Bei einer Dysarthrie können Sprechmotorik, Phonation und Sprechatmung beeinträchtigt sein. Infolge Schädigung von Hirnnerven oder motorischen Hirnarealen, wie dem motorischen Cortex, kommt es zu Lähmung, Schwäche oder Koordinationsstörungen der am Sprechen beteiligten Sprechmuskulatur. Ursachen sind beispielsweise: Schädel-Hirn-Traumata, Tumore, entzündliche Erkrankungen, Durchblutungsstörungen des Gehirns bezirhungsweise Hypoxien. Folgende Symptome können auftreten: Undeutliche Artikulation, beeinträchtigte Sprechmelodie, beeinträchtigte Stimmqualität, Veränderung der Lautstärke, Veränderung des Sprechtempos. Zur detaillierten symptomatischen Analyse der Sprechstörung eignet sich im Falle von Dysarthrie beispielsweise das sogenannte „Münchner Verständlichkeitsprofil“ [Ahrndt1992] [Ziegler1992]. Eine Beeinträchtigung der Sprechmelodie wird als „Dysprosodie“ bezeichnet. Bei beeinträchtigter Stimmqualität kann die Stimme rau, gepresst und gelegentlich nasal klingen. Die Veränderung des Sprechtempos kann in zu schneller oder zu langsamer Aussprache resultieren. Bei zusätzlicher Beeinträchtigung der Atmung spricht man von einer „Dysarthrophonie“.

Systemdesign von Sprachsteuerungen. Es ist im Allgemeinen eine höhere Benutzerzufriedenheit zu erwarten, wenn die verwendeten Steuerbefehle an den Benutzer angepasst werden. Die Systemdesignerleichterung, dabei einen sprecherabhängigen Spracherkenner einsetzen zu können, wird jedoch oftmals kompensiert durch die Systemdesignerschwernis, dass die Aussprache der Steuerbefehle bei Sprechstörungen teilweise stark variieren kann. Jedenfalls erweist es sich als sinnvoll, die Qualität der Aufnahmen der Steuerbefehle vor der Benutzung dahin gehend zu überprüfen, ob die lautlichen Äußerungen des Benutzers eine genügend fehlerfreie Klassifikation erlauben und auch nicht mit Hintergrundgeräuschen verwechselt werden. Dadurch können den Benutzern in vielen Fällen von vornherein die Enttäuschungen bezüglich übermäßiger Fehlfunktionen der Sprachsteuerung erspart bleiben. In vielen Fällen ist es möglich, durch Sprechtraining

des Benutzers oder einfach durch Variation der Steuerbefehle zu einem Satz von Aufnahmen der Steuerbefehle zu gelangen, der eine genügend fehlerfreie Funktionsweise der Sprachsteuerung erlaubt.

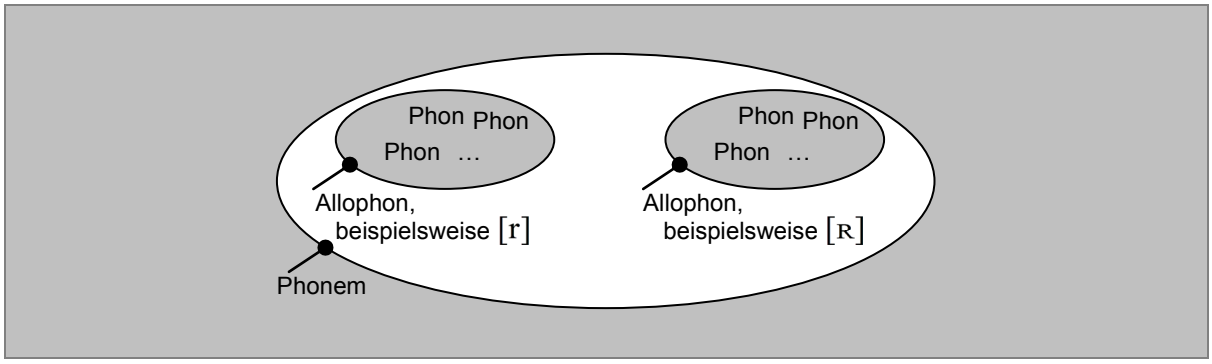

Abbildung 2-1: Mengenmodell zur Veranschaulichung der Begriffsdefinitionen „Phon“, „Allophon“ und „Phonem“.

Standardlautung des Deutschen. Es wird zunächst auf die unbeeinträchtigte Lautbildung eingegangen. Eine gute Einführung in die Standardlautung des Deutschen findet sich im Duden-Aussprachewörterbuch [Mangold2005]. Der Begriff „Phonetik“ bezeichnet die Lehre von der Aktivierung der Sprechwerkzeuge. „Phone“ (Laute) sind die kleinsten Bausteine der gesprochenen Sprache. Wenn sich die Bedeutung des Wortes bei der Modifikation eines Lautes ändert, so gehören der alte und der neue Laut zu unterschiedlichen „Phonemen“. Wenn sich die Wortbedeutung nicht ändert, die Laute sich jedoch unterschiedlich anhören, so gehören der alte und der neue Laut lediglich zu unterschiedlichen „Allophonen“ des gleichen „Phonems“. Beispielsweise sind der Konsonant [r], welcher mit vibrierender Zunge artikuliert wird, und der Konsonant [ʀ], welcher mit vibrierendem Zäpfchen artikuliert wird, im Deutschen Allophone.

Lautschrift. Neben der SAMPA-Lautschrift und der X-SAMPA-Lautschrift, welche ausschließlich ASCII-Zeichen verwenden, und damit Vorteile bezüglich der automatisierten Verarbeitung aufweisen, hat vor allem das „International Phonetic Alphabet“ IPA der 1886 in Paris gegründeten Organisation „Association Phonétique Internationale“[51] weite Verbreitung gefunden. Dieses Lautschriftsystem wird nach den neuesten Erkenntnissen und Erfahrungen ständig verbessert. Da diese Lautschrift für möglichst sämtliche Sprachen, sowie für nicht normgerechte Lautbildung, geeignet sein soll, ist ihr Zeichensatz sehr groß. Eine Übersichtstafel zur Definition der Lautschrift „International Phonetic Alphabet“ sowie Klangbeispiele der einzelnen Laute finden sich im Internet[52].

[51] Die Organisation trägt weiters den englischen Namen „International Phonetic Association“.

[52] Quelle: Homepage der „International Phonetic Association“. http://www.langsci.ucl.ac.uk/ipa/ipachart.html (Stand: 25. 03. 2013).

Artikulation der Vokale. Vokale (Selbstlaute) sind stimmhafte Öffnungslaute. Die Veränderung von einem Vokal zu einem anderen Vokal ergibt sich durch die unterschiedliche Mundöffnung und durch eine Veränderung des Resonanzraumes.

IPA-Symbol	Beispiel		IPA-Symbol	Beispiel
[a]	hat [hat]		[oː]	Boot [boːt]
[aː]	Bahn [baːn]		[o̯]	loyal [lo̯aˈjaːl]
[ɐ]	Ober [ˈoːbɐ]		[õ]	Fondue [fõˈdyː]
[ɐ̯]	Uhr [uːɐ̯]		[õː]	Fond [fõː]
[ã]	Pensee [pãˈseː]		[ɔ]	Post [pɔst]
[ãː]	Gourmand [gʊrˈmãː]		[ø]	Ökonom [økoˈnoːm]
[a͡i]	weit [va͡it]		[øː]	Öl [øːl]
[a͡u]	Haut [ha͡ut]		[œ]	göttlich [ˈgœtlɪç]
[e]	Methan [meˈtaːn]		[œ̃]	Lundist [lœ̃ˈdɪst]
[eː]	Beet [beːt]		[œ̃ː]	Parfum [parˈfœ̃ː]
[ɛ]	hätte [ˈhɛtə]		[ɔ͡y]	Heu [hɔ͡y]
[ɛː]	wähle [ˈvɛːlə]		[u]	kulant [kuˈlant]
[ɛ̃]	timbrieren [tɛ̃’briːrən]		[uː]	Hut [huːt]
[ɛ̃ː]	Timbre [ˈtɛ̃ːbrə]		[u̯]	aktuell [akˈtu̯ɛl]
[ə]	halte [ˈhaltə]		[ʊ]	Pult [pʊlt]
[i]	vital [viˈtal]		[u͡i]	pfui! [pfu͡i]
[iː]	viel [fiːl]		[y]	Mykene [myˈkeːnə]
[i̯]	Studie [ˈʃtuːdi̯ə]		[yː]	Rübe [ˈryːbə]
[ɪ]	bist [bɪst]		[ỹ]	Tuilerien [tỹiləˈriːən]
[o]	Moral [moˈraːl]		[ʏ]	füllt [fʏlt]

Tabelle 3: Die wichtigsten Vokale des Deutschen in IPA-Notation.[53] Sogenannte „diakritische“ Zusatzmarkierungen kennzeichnen verschiedene Modifikationen.

Tabelle 3 zeigt die wichtigsten Vokale des Deutschen mit eingetragenen diakritischen Zusatzmarkierungen nach dem „International Phonetic Alphabet“. Interessant ist ein Vergleich mit der Übersichtstafel zur Definition der Lautschrift „International Phonetic Alphabet“ bezüglich des Artikulationsortes und der Mundöffnung. Der Artikulationsort wird bei Vokalen üblicherweise mit den Attributen „vorne“ beziehungsweise „hinten“ bezeichnet. Die Mundöffnung wird üblicherweise mit den Attributen „offen“ beziehungsweise „geschlossen“ bezeichnet.

Artikulation der Konsonanten. Konsonanten werden dadurch gebildet, dass dem Luftstrom Hindernisse dynamisch entgegengestellt werden, die er überwinden oder umgehen muss. Die Einteilung der Konsonanten erfolgt oftmals nach dem Artikulationsort, das heißt nach jenem Ort, an dem das Hindernis dem Luftstrom entgegengestellt wird.

[53] Quelle: Online Duden – Aussprache
http://www.duden.de/hilfe/aussprache (Stand: 18. 09. 2012).

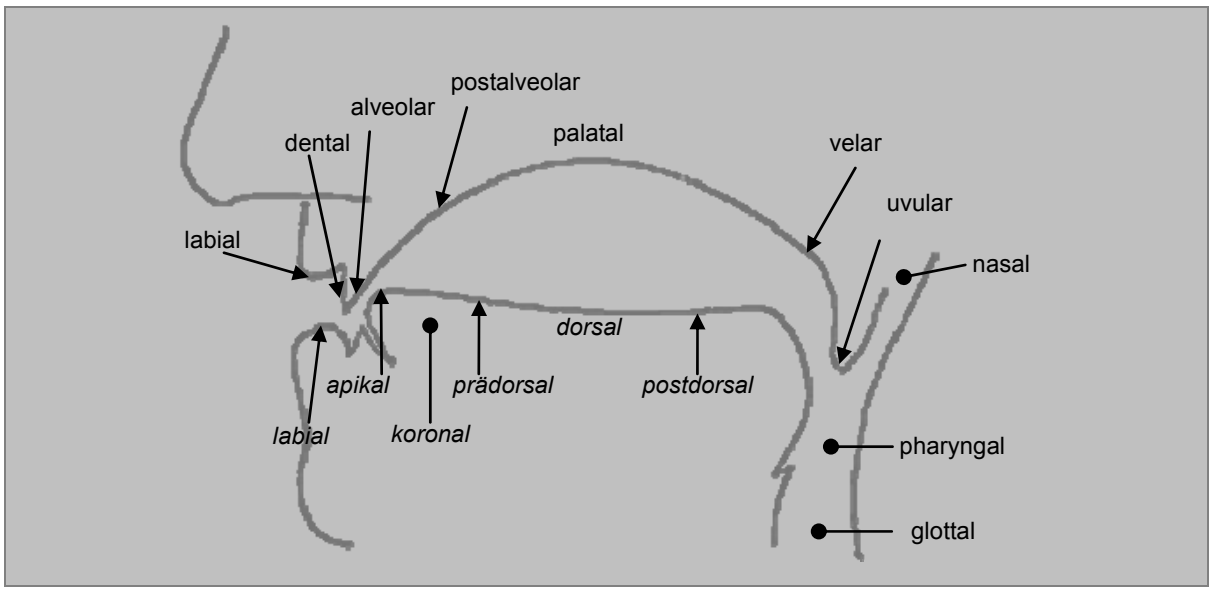

Abbildung 2-2: Attribute zur Beschreibung der Artikulation im Hinblick auf Artikulationsort und Artikulationsorgan.[54] Attribute der Artikulationsorgane sind kursiv dargestellt.

Artikulationsort. Abbildung 2-2 zeigt schematisch den Artikulationstrakt mit eingetragenen Bezeichnungen der Artikulationsorte: Oberlippe (Labiae), Oberzähne (Dentes), Zahndamm (Alveolen), harter Gaumen (Palatum), weicher Gaumen (Gaumensegel, Velum), Zäpfchen (Uvula), Rachenhöhle (Pharynx), Kehlkopf (Larynx) beziehungsweise Stimmritze (Glottis).

Artikulationsorgan. Weiters teilt man die Konsonanten nach dem Artikulationsorgan ein. In Abbildung 2-2 sind die Bezeichnungen kursiv eingetragen. Zu erwähnen sind Zungenspitze (Apex), Zungenkranz (Corona, Koronalenge) und Zungenrücken (Dorsum, Dorsalenge).

Artikulationsausführung. Bezüglich der Ausführung der Artikulation bezeichnet man Laute als: Plosive (Verschlusslaute), Frikative (Reibelaute), Nasale (Nasenhöhlen-Luftführung), Laterale (seitliche Luftführung) sowie Vibranten (Schwinglaute, Tremulanten).

[54] Quelle: Multimediale Referenz der deutschen Sprachlaute http://www.media-enterprise.de/dt/MM_REFERENZ/overview.htm (Stand: 10. 11. 2004).

IPA-Symbol	Beispiel	Ausführung	Artikulationsort	Zungenstellung
[b]	Ball [bal]	Stimmhafter Plosiv	Labiae	-
[ç]	ich [ɪç]	Stimmloser Frikativ	Palatum	Dorsal-Enge
[d]	dann [dan]	Stimmhafter Plosiv	Alveolen	Koronal-Enge
[f]	Fass [fas]	Stimmloser Frikativ	Labiae, Dentes	-
[g]	Gast [gast]	Stimmhafter Plosiv	Velum	Postdorsal-Enge
[h]	hat [hat]	Stimmloser Frikativ	Glottis	-
[j]	ja [jaː]	Stimmhafter Frikativ	Palatum	Dorsal-Enge
[k]	kalt [kalt]	Stimmloser Plosiv	Velum	Postdorsal-Enge
[l]	Last [last]	Stimmhafter Lateral	Alveolen	Koronal-Enge
[m]	Mast [mast]	Stimmhafter Nasal	Labiae	-
[n]	Naht [naːt]	Stimmhafter Nasal	Alveolen	Koronal-Enge
[ŋ]	lang [laŋ]	Stimmhafter Nasal	Velum	Postdorsal-Enge
[p]	Pakt [pakt]	Stimmloser Plosiv	Labiae	-
[r]	Rast [rast]	Stimmhafter Vibrant	Alveolen	Koronal-Enge
[R]	Rast [ʁast]	Stimmhafter Vibrant	Uvula	Postdorsal-Enge
[s]	Hast [hast]	Stimmloser Frikativ	Alveolen	Prädorsal-Enge
[ʃ]	Schal [ʃaːl]	Stimmloser Frikativ	Palatum	Koronal-Enge
[t]	Tal [taːl]	Stimmloser Plosiv	Alveolen	Koronal-Enge
[v]	was [vas]	Stimmhafter Frikativ	Labiae, Dentes	-
[x]	Bach [bax]	Stimmloser Frikativ	Velum	Postdorsal-Enge
[z]	Hase [ˈhaːzə]	Stimmhafter Frikativ	Alveolen	Prädorsal-Enge
[ʒ]	Genie [ʒeˈniː]	Stimmhafter Frikativ	Alveolen, Palatum	Prädorsal-Enge

Tabelle 4: Die wichtigsten Konsonanten des Deutschen in IPA-Notation.[55] Die Zungenstellung wird durch den Ort des Luftstromhindernisses beschrieben. Sie ist bei jenen Lauten, bei denen sie relevant ist, in der Tabelle eingetragen.

[55] Quelle: Online Duden – Aussprache
http://www.duden.de/hilfe/aussprache (Stand: 18. 09. 2012).

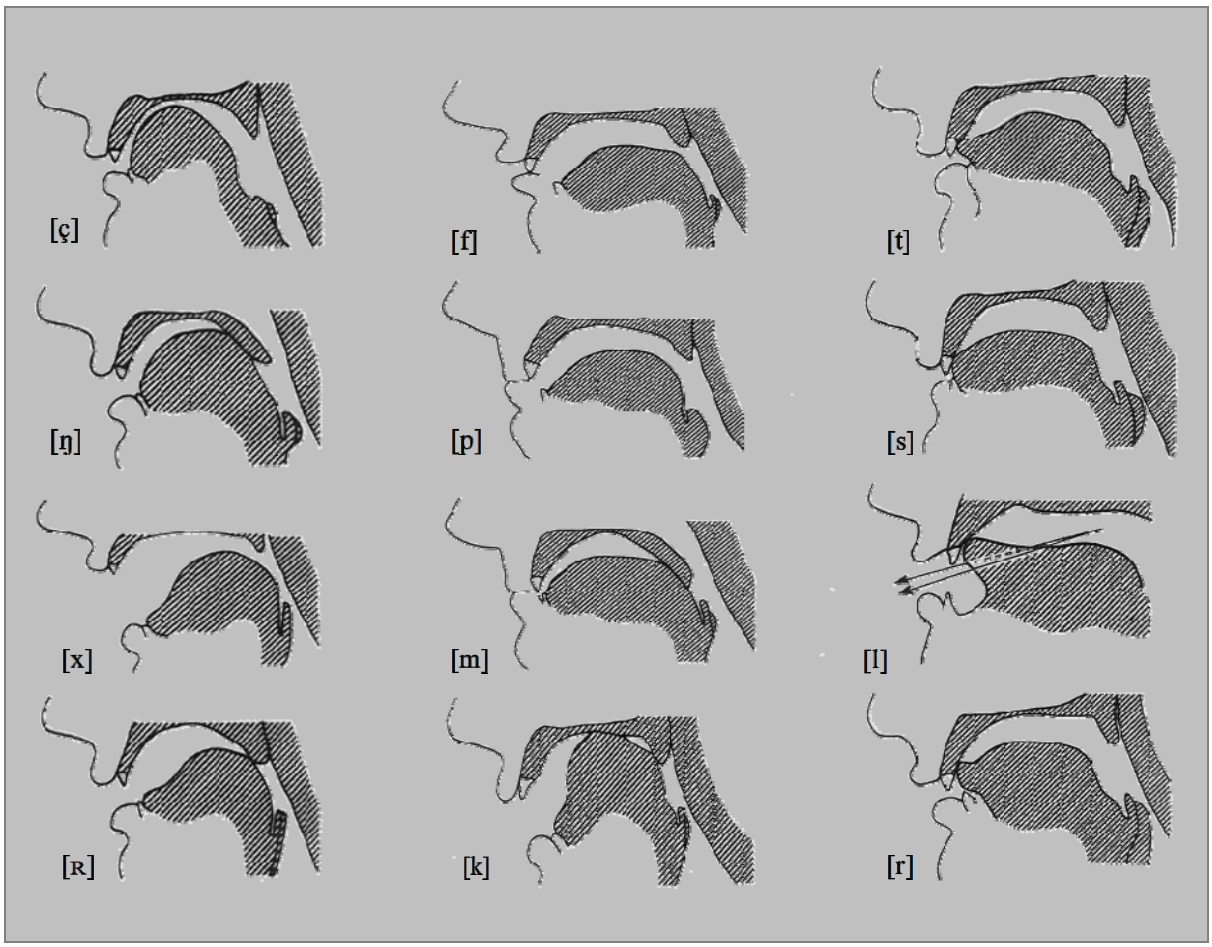

Abbildung 2-3: Querschnittdarstellungen der Lautbildung einzelner Konsonanten.[56]

Abbildung 2-3 zeigt Querschnittsdarstellungen bei der Lautbildung einiger Konsonanten. Dargestellt sind beispielsweise der mit vibrierender Zunge artikulierte Konsonant [r] und der mit vibrierendem Zäpfchen (Uvula) artikulierte Konsonant [ʀ], wobei es sich im Deutschen jeweils um Allophone des gleichen Phonems handelt.

2.1.2.1 Stimmstörungen

Dysphonie (Heiserkeit), Aphonie. Dabei handelt es sich um eine Beeinträchtigung des Stimmklangs und der stimmlichen Leistungsfähigkeit. Man unterscheidet die funktionelle Heiserkeit, die kehlkopfbedingte Heiserkeit sowie die endokrin bedingte Heiserkeit [Hildebrandt1998]. Die funktionelle Heiserkeit ist dadurch gekennzeichnet, dass keine primäre morphologische Veränderung des Stimmapparats vorhanden ist. Die kehlkopfbedingte Heiserkeit tritt auf bei: Formanomalien des Kehlkopfes,

[56] Quelle: Multimediale Referenz der deutschen Sprachlaute http://www.media-enterprise.de/dt/MM_REFERENZ/overview.htm (Stand: 10. 11. 2004).

Laryngitis (Kehlkopfentzündung), Stimmlippenknötchen, Stimmlippenlähmung, Kehlkopfkarzinom, Gewalteinwirkung auf den Kehlkopf (Larynx) sowie nach Intubation der Luftröhre. Auch der Zustand nach einer Laryngektomie sei in diesem Zuge angeführt. Die endokrin bedingte Heiserkeit tritt auf infolge von Hypothyreose (Schilddrüsenunterfunktion), Hypogonadismus und bei hormonaler Therapie.

Rhinophonie (Näseln). Synonym werden die Begriffe Rhinophonie, Hyperrhinophonie, Hyporhinophonie verwendet. Beim Näseln handelt es sich um eine Beeinträchtigung des Stimmklanges beziehungsweise der Artikulation. Die üblichen nasalen Phoneme in der deutschen Sprache sind das Phonem [m], das Phonem [n] und das Phonem [ŋ]. Bei den verschiedenen Formen des Näselns wird die Lautbildung der Phoneme und Phonemübergänge in charakteristischer Weise verändert. Die Rhinophonie führt zu einer Verlagerung der Artikulation nach „hinten“, was als „Rhinolalie“ bezeichnet wird. Man unterscheidet „Rhinolalia aperta“ (offenes Näseln) und „Rhinolalia clausa“ (geschlossenes Näseln). Beim offenen Näseln werden die Laute in unnatürlicher Weise durch die Nase artikuliert, was etwa bei Lippenspalten, Kieferspalten, offenen oder submukösen Gaumenspalten beziehungsweise Gaumensegellähmung zu beobachten ist. Beim geschlossenen Näseln wird der Resonanzraum der Nase nicht oder nur unzureichend genutzt, was etwa bei stockendem Schnupfen, Nasennebenhöhlenentzündungen, Nasenpolypen beziehungsweise Nasentumoren zu beobachten ist.

2.1.2.2 Lautbildungsstörungen

Phonetische Dyslalie. Unter „phonetischer Dyslalie“[57] versteht man eine Artikulationsstörung, welche dadurch gekennzeichnet ist, dass Phoneme beziehungsweise Phonemverbindungen nicht dem Alter entsprechend gebildet werden. Ein bekanntes Beispiel ist die gestörte Aussprache des Phonems [s], der so genante Sigmatismus (Lispeln). Man unterscheidet: Auslassungen (Elisionen), Ersetzungen (Substitutionen), Hinzufügungen (Adjunktionen) und Umstellungen (Permutationen). Je nach betroffenem Phonem unterscheidet man: Sigmatismus, Gammazismus, Kapazismus, und so weiter. Bei der „Audiogenen Dyslalie“ liegt die Ursache in einem beidseitigen Hörverlust. Als weitere Ursachen treten beispielsweise Dysglossien auf: Als Dysglossie bezeichnet man die für Beeinträchtigungen der Artikulation ursächliche Veränderungen der Sprechorgane, wie beispielsweise: Angeborene Missbildungen, Zahndefekte, Makroglossie (Zungenvergrößerung), Verletzungen oder Lähmungen an Lippen, Zunge, Gaumen und Rachen. Auch Kopf-Gesichts-Fehlbildungssyndrome können sich auf die Artikulation auswirken: Velumverkürzung, submuköse Gaumenspalte, Lippen-Kiefer-Gaumenspalte, Down-Syndrom, Pierre-Robin-Syndrom, Apert-Syndrom, Crouzon-

[57] Bezüglich des Fachbegriffs „Stammeln“ als Synonym für „Dyslalie“ gibt es etymologische Bedenken, da er im normalen Sprachgebrauch synonym mit dem Begriff „Stottern“ gebraucht wird.

Syndrom, Franceschetti-Syndrom, Goldenhar-Syndrom, Pfaundler-Syndrom, Waardenburg-Syndrom [Böhme1997]. Unter orofacialen Dysfunktionen versteht man Fehlfunktionen der Kaumuskulatur und der Gesichtsmuskulatur, die auch zu falschen Schluckgewohnheiten und damit zu Zahnfehlstellungen und Kieferfehlstellungen führen können. Orofaciale Dysfunktionen und die daraus resultierenden Sprechstörungen und Schluckstörungen beruhen vorwiegend auf funktionellen Veränderungen infolge eines Ungleichgewichts der Muskulatur im Mundbereich, Gesichtsbereich, Halsbereich und Nackenbereich [Böhme1997].

2.1.2.3 Redeflussstörungen

Stottern. Dabei handelt es sich um eine zeitweise auftretende Redeflussstörung, die sich durch Pausen, sowie Einschübe und Wiederholungen von Lauten, Silben und Wörtern äußert. Beim Stottern handelt es sich um eine zentrale Sprechstörung, deren Ursache oftmals unbekannt ist. Stottern tritt unabhängig vom Willen des Sprechers im Kindes- und Erwachsenenalter auf und äußert sich symptomatisch bezüglich Respiration, Phonation, Artikulation, Sprechablauf und Sprechmotorik.

Poltern. Man versteht darunter eine übermäßige Erhöhung des Sprechtempos, die zu undeutlicher Aussprache von Phonemen sowie zu Auslassungen oder Wiederholungen von Silben, Wörtern beziehungsweise Satzteilen führt. Das Poltern kann aus psychologischer Sicht als Beeinträchtigung der Planung temporaler Abfolgen oder als Beeinträchtigung der Wahrnehmung temporaler Abfolgen erklärt werden.

Bradylalie. Bradylalie wird auch als „Bradyarthrie“ oder „Bradyglossie“ bezeichnet. Man versteht darunter eine Verlangsamung des Sprechtempos, die etwa bei Multipler Sklerose zu beobachten ist.

Sprechapraxie. Unter einer „Apraxie“ versteht man die Beeinträchtigung der folgerichtigen Aneinanderreihung von Einzelbewegungen zu Bewegungsabläufen, obwohl die Einzelbewegungen separat durchgeführt werden können. Bei der Sprechapraxie handelt es sich um eine Einschränkung der Planung des Bewegungsablaufes der Sprechorgane, beispielsweise der Lippen und der Zunge. Die Sprechapraxie ist eine zentralmotorische Beeinträchtigung des Sprechens, bei der eine Beeinträchtigung der Planung der Abfolge der Sprechbewegungen angenommen wird. Demnach bestehen bei einer reinen Sprechapraxie keine Paresen oder Änderungen des Tonus, wie sie etwa bei Dysarthrien zu beobachten sind. Ebenso kann die Beeinträchtigung nicht durch Aphasie erklärt werden.

2.1.3 Sprachstörungen

Es werden Sprachentwicklungsstörungen und Sprachverluststörungen unterschieden. In den folgenden Abschnitten erfolgt die Einteilung in syntaktische, semantische und pragmatische Sprachstörungen. Je nach zugrunde liegenden Ursachen treten auch Kombinationen von Sprachstörungen der syntaktischen, semantischen und pragmatischen Ebenen auf, welche oftmals, im Sinne einer ursachenbezogenen Einteilung, eigens bezeichnet werden: Beispiele für ursachenbezogene Bezeichnungen sind unter anderen die verschiedenen „Dysphasien“ beziehungsweise „Aphasien“. Dabei handelt es sich um Sprachverluststörungen. Ursachen von Dysphasien beziehungsweise Aphasien können beispielsweise Durchblutungsstörungen des Gehirns, Schlaganfälle, Tumore, Hirnentzündungen oder Hirntraumata sein. Mit dem Sprachverlust können auch Beeinträchtigungen oder Ausfälle der Schreibfähigkeit (Dysgraphie, Agraphie), der Lesefähigkeit (Dyslexie, Alexie) und der Rechenfähigkeit (Dyskalkulie, Akalkulie) einhergehen. Man unterscheidet mehrere Formen der Aphasie: Bei der „globalen Aphasie“ handelt es sich um eine Kombination aus Sprachproduktionsstörung und Sprachverständnisstörung. Bei der „Broca-Aphasie“, welche auch als „motorische Aphasie“ bezeichnet wird, handelt es sich um eine Sprachproduktionsstörung ohne nennenswerte Sprachverständnisstörung. Das Leitbild der Broca-Aphasie ist der Agrammatismus. Bei der „Wernicke-Aphasie“, welche auch als „sensorische Aphasie“ bezeichnet wird, handelt es sich umgekehrt um eine Sprachverständnisstörung ohne nennenswerte Sprachproduktionsstörung. Das Leitbild der Wernicke-Aphasie ist der Paragrammatismus. Bei der „amnestischen Aphasie“ handelt es sich um eine Wortfindungsstörung ohne darüber hinausgehende Sprachproduktionsstörung und ohne nennenswerte Sprachverständnisstörung [Böhme1997].

Systemdesign von Sprachsteuerungen. Beim Design der Benutzerdialoge für Sprachstörungen ist darauf zu achten, die Benutzer nicht in ihren sprachlichen Möglichkeiten zu überfordern. Die Rückmeldungen der Sprachsteuerung können gegebenenfalls auch über eine Bildersprache erfolgen.

Phonologische Prozesse		Beispiele	
Silbenstrukturprozesse	Silbenauslassung	Banane	[baˈnaːnə] ⟶ [ˈnaːnə]
	Wortvereinfachung	Lokomotive	[lokomoˈtiːfə] ⟶ [ˈtiːfə]
	Reduplikation	Da	[daː] ⟶ [ˈdada]
	Mehrfachkonsonanzreduktion	Blume	[ˈbluːmə] ⟶ [ˈluːmə]
Harmonisierungsprozesse	Labialassimilation	Pudel	[ˈpuːdl̩] ⟶ [ˈpuːbl̩]
	Velarassimilation	Zunge	[ˈt͡sʊŋə] ⟶ [ˈkʊŋə]
	Prävokalstimmgebung	Tasche	[ˈtaʃə] ⟶ [ˈdaʃə]
Substitutionsprozesse	Alveolarisierung	Gabel	[ˈgaːbl̩] ⟶ [ˈdaːbl̩]
	Velarisierung	Bett	[bɛt] ⟶ [bɛk]
	Labialisierung	Hase	[ˈhaːzə] ⟶ [ˈhaːbə]
	Plosivierung	Tasse	[ˈtasə] ⟶ [ˈtatə]
	Frikativierung	Topf	[t͡ɔpf] ⟶ [tɔf]
	Affrizierung	Löffel	[ˈlœfl̩] ⟶ [ˈlœp͡fl̩]

Tabelle 5: Normale phonologische Prozesse beim Spracherwerb.[58]

Normale phonologische Prozesse beim Spracherwerb. Im Rahmen des normalen Spracherwerbs vereinfachen Kinder die Wörter, die sie aus ihrer Umwelt aufnehmen. Diese Vereinfachungen werden als „phonologische Prozesse“ bezeichnet. Die phonologischen Prozesse lassen sich, gemäß Tabelle 5, in folgende drei Kategorien einteilen: Silbenstrukturprozesse, Harmonisierungsprozesse und Substitutionsprozesse. Zu den Silbenstrukturprozessen gehören beispielsweise das Auslassen unbetonter Silben, die Vereinfachung mehrsilbiger Wörter, die Reduplikation und die Reduktion von Mehrfachkonsonanz. Zu den Harmonisierungsprozessen gehören beispielsweise Labial-assimilationen[59], Velarassimilationen und Prävokalstimmgebung. Zu den Substitutions-prozessen gehören beispielsweise die Alveolarisierung, die Velarisierung, die Labialisierung, die Plosivierung, die Frikativierung und die Affrizierung.

2.1.3.1 Syntaktische Sprachstörungen

Dysgrammatismus. Darunter versteht man die Unfähigkeit, Sätze nach den Regeln der Grammatik richtig zu bilden. Man spricht in diesem Zusammenhang auch vom „Telegrammstil“. Dabei liegen nur ganz einfache Satzstrukturen vor. Häufig handelt es sich um Einwortsätze oder Zweiwortsätze. Fehler der Wortbeugung, wie beispielsweise „Ein schön Baum“, und Fehler der Wortstellung, wie beispielsweise „Ich heim gehe bald“, sind Anzeichen für Dysgrammatismus. Es handelt sich um eine Beeinträchtigung der gesprochenen sowie der geschriebenen Sprache. Dysgrammatismus ist das Leitbild der „Broca-Aphasie“ [Böhme1997].

---

[58] Quelle: Sprachheilpädagogische und sprachtherapeutische Wissensplattform http://sprachheilwiki.dgs-ev.de/wiki/doku.php?id=therapie:phonetisch-phonologische_stoerungen (Stand: 25. 03. 2013)

[59] Unter „Assimilation“ versteht man die Beeinflussung eines Lautes durch einen Nachbarlaut.

Paragrammatismus. Paragrammatismus bei sonst unauffälliger Sprachproduktionsfähigkeit kombiniert mit einer Sprachverständnisstörung tritt etwa bei „Wernicke-Aphasie“ auf. Dabei ist der Satzbau zu komplex angelegt und es sind häufige Satzabbrüche sowie unnötige Verdopplungen und Verschränkungen in Hauptsätzen und Nebensätzen zu beobachten [Böhme1997].

Phonologische Dyslalie (Lautverwendungsstörung). Bei der „phonologischen Dyslalie“ können Laute zwar korrekt gebildet, aber nicht gemäß den sprachsystematischen Regeln angewandt werden. Häufig werden sie durch andere muttersprachliche Laute ersetzt oder ausgelassen. Seltener tritt auch eine Mischform aus „phonetischer Dyslalie“ und „phonologischer Dyslalie“ auf, die als „phonetisch-phonologische Störung“ bezeichnet wird. Bei dieser Form bedingen sich Lautbildungsstörungen und Lautverwendungsstörungen gegenseitig[60].

Dyslexie (Leseschwäche). Unter Dyslexie versteht man Probleme mit dem Lesen und Verstehen von Wörtern oder Texten bei normalem Sehvermögen. Oft tritt sie das erste Mal im Rahmen einer sogenannten „Lese-Rechtschreibschwäche“ (Legasthenie) in den ersten Schuljahren zutage, wobei die Betroffenen beim Lesenlernen und Schreibenlernen weit hinter die Altersgenossen zurückfallen, obwohl sie über eine normale Intelligenz verfügen. Die besondere familiäre Häufung von Dyslexie lässt vermuten, dass diese Störung zumindest teilweise erblich bedingt ist. Neben den genetischen Dyslexien spricht man auch von den erworbenen Dyslexien, welche bei Kindern und Jugendlichen infolge mangelnden Lesetrainings auftreten können. Dyslexien können auch im Erwachsenenalter durch Hirnschädigungen, wie beispielsweise Schlaganfälle oder Schädel-Hirn-Traumata, ausgelöst werden. Manche dieser Personen lesen Wörter, die so nicht dastehen, wie zum Beispiel „Katze“ statt „Hund“. Andere lesen mühsam buchstabierend[61].

Dysgraphie (Rechtschreibschwäche). Bei der Dysgraphie handelt es sich um eine Schreibstörung trotz intakter Motorik und normaler Intelligenz. Als Herdsymptom der Großhirnrinde, speziell des „Gyrus angularis“, stellt sie eine Sonderform der Apraxie dar und ist meist mit aphasischen Störungen kombiniert[62].

2.1.3.2 Semantische Sprachstörungen

Wortfindungsstörung. Es handelt sich um eine Störung der Fähigkeit, ein bestimmtes Wort zur Bezeichnung von Objekten, Ereignissen, Eigenschaften oder Tätigkeiten zur Verfügung zu haben. Eine Wortfindungsstörung kann durch sogenannte

---

[60] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Dyslalie (Stand: 28. 03. 2013).

[61] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Dyslexie (Stand: 14. 09. 2008).

[62] Quelle: Deutsche Wikipedia http://www.aerztlichepraxis.de/rw_4_Lexikon_HoleEintrag_14255_Eintrag.htm (Stand: 14. 09. 2008).

„Umwegstrategien“ kompensiert werden, wobei anstelle des speziellen Begriffs unter anderem Umschreibungen, allgemeine Floskeln und Gestik verwendet werden[63]. Eine Wortfindungsstörung ohne Sprachverständnisstörung bei unauffälliger Sprachproduktionsfähigkeit tritt etwa bei „amnestischer Aphasie“ auf [Böhme1997].

Semantische Sprachverständnisstörung. An dieser Stelle sei der semantische Aspekt verschiedener Aphasien, insbesondere der „globalen Aphasie“ und der Wernicke-Aphasie, erwähnt.

2.1.3.3 Pragmatische Sprachstörungen

Palilalie. Palilalie bezeichnet einen krankhaften Zwang, eigene Sätze und Wörter wiederholt zu sprechen. Dabei steigt die Sprechgeschwindigkeit, während die Lautstärke sinkt. Palilalie zählt wie die auch eine Variante der Echolalie (zwanghaftes Wiederholen von Wörtern des Gesprächspartners) zu den Ticstörungen. Sie tritt häufig in den Äußerungen hirngeschädigter Patienten auf. Besonders Patienten mit Tourette-Syndrom und Patienten mit fortgeschrittener Parkinson-Krankheit leiden unter dieser Ticstörung, aber auch bei Schizophrenie und Autismus ist das Vorkommen beschrieben[64].

Koprolalie. Koprolalie taucht einerseits als, meist bewusst eingesetztes, Stilmittel in der Literatur auf, bezeichnet aber andererseits auch ein neurologisch-psychiatrisches Symptom [Hildebrandt1998]. Eine besondere, ja geradezu kennzeichnende Bedeutung hat die Koprolalie beim Tourette-Syndrom, wo sie bei etwa 30 % der Betroffenen als komplexe vokale Ticstörung auftritt, die sich darin äußert, dass Betroffene plötzlich, ohne erkennbaren Grund oder Zweck sowie willkürlich unbeeinflussbar, sozial unangebrachte oder obszöne Wörter in regelrechten Salven von sich geben. Vom Tourette-Syndrom Betroffene erleben dies als impulsiven Zwang, dem sie machtlos ausgeliefert sind[65].

Echolalie. Der Begriff Echolalie hat mehrere Bedeutungen[66]: Echolalie bezeichnet einerseits einen krankhaften Zwang, vorgesagte Wörter und Sätze ohne Rücksicht auf den Inhalt oder die Situation, wörtlich oder abgewandelt nachzusprechen, was beispielsweise beim Tourette-Syndrom, bei Schizophrenie, bei Morbus Alzheimer und bei Autismus zu beobachten ist. Andererseits wird darunter auch die Beschränkung der Sprache auf das Nachsprechen vorgesagter Wörter verstanden, wie sie bei transkortikaler Aphasie zu beobachten ist[67].

---

[63] Quelle: Deutsche Wikipedia http://www.medrapid.info/krankheit/41889 (Stand: 14. 09. 2008).

[64] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Palilalie (Stand: 14. 09. 2008).

[65] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Koprolalie (Stand: 14. 09. 2008).

[66] Gelegentlich wird auch die Wiederholung vorgesagter Phrasen in der normalen frühkindlichen Sprachentwicklung als „Echolalie“ bezeichnet, wobei es sich dann nicht um eine Sprachstörung handelt.

[67] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Echolalie (Stand: 14. 09. 2008).

Psychogenes Schweigen (Mutismus). Der Mutismus ist ein vollständiges oder partielles Fehlen der stimmhaften Sprache bei erhaltener Sprechfähigkeit und abgeschlossener Sprachentwicklung. Der periphere und zentrale Sprechapparat sowie das Hörvermögen sind normal. Meist ist der Zustand gekennzeichnet durch ausbleibendes Lachen, Apathie beziehungsweise beeinträchtigtes Verhalten [Böhme1997]. In Abgrenzung dazu wird mittels des Begriffs „Taubstummheit“ auf Hörverlust als Ursache für Sprachstörungen hingewiesen. Ebenfalls in Abgrenzung dazu wird mittels des Begriffs „Hörstummheit“ (Audimutismus) auf eine zentrale Sprachentwicklungsstörung bei unbeeinträchtigtem Hörvermögen, intakten peripheren Sprechwerkzeugen, dem Alter entsprechendem Sprachverständnis und normaler Intelligenz, meist infolge eines frühkindlichen Hirnschadens, hingewiesen.

2.1.4 Visuelle Störungen

Zu den Symptomen visueller Störungen gehören: Refraktionsstörungen, Akkommodationsstörungen, Gesichtsfeldeinschränkungen, Gesichtsfeldausfälle, Phosphene, Augenrauschen[68] (Visual Snow, Visual Static), Floaters (Mouches volantes), Doppelbilder, Nachbilder[69], Beeinträchtigungen der Farbwahrnehmung.

Zu den Ursachen visueller Störungen gehören: Kurzsichtigkeit, Weitsichtigkeit, Alterssichtigkeit, Glaukom (Grüner Star), Katarakt (Grauer Star), Retinopathie, Makuladegeneration, Hemianopsie, Amblyopie (Rindenblindheit, visuelle Agnosie), Opticusatrophie sowie weitere neurologische Erkrankungen. Eine interessante Übersicht, auch hinsichtlich demografischer Daten innerhalb der Europäischen Union, findet sich in der Literatur [Zagler1997a].

Systemdesign von Sprachsteuerungen. Im Zuge von Systemdesignüberlegungen sollen folgende Richtlinien berücksichtigt werden: Große kontrastreiche Schriften sind zu verwenden. Die Notwendigkeit, Farben zu unterscheiden, vor allem Notwendigkeit Rot und Grün zu unterscheiden, ist zu vermeiden. Mindestens eine redundante Kommunikationsmöglichkeit, wie beispielsweise eine zusätzliche Sprachausgabe, ist vorzusehen.

---

[68] Es herrscht die Meinung vor, dass dieses Symptom bis zu einem gewissen Grad als normal anzusehen ist. Jedoch dürfte es in schwergradiger Ausprägung zu erheblichen Einschränkungen des täglichen Lebens führen. Es handelt sich jedoch um ein subjektives Symptom und die wissenschaftliche Rezeption ist beinahe nicht vorhanden.

[69] Auch Nachbilder sind bis zu einem gewissen Grad als normal anzusehen. Da es sich ebenfalls um ein subjektives Symptom handelt, ist die Abgrenzung bezüglich des Krankheitswerts schwierig.

2.1.5 Auditive Störungen

Zu den auditiven Beeinträchtigungen beziehungsweise Behinderungen gehören: Schallleitungsstörung, Schallempfindungsstörung, Recruitment-Störung[70], Tonhöhenwahrnehmungsstörung[71], Latenzzeitstörungen[72], Hyperakusis, Tinnitus sowie akustische Agnosie[73] (Rindentaubheit).

Als Ursachen kommen beispielsweise infrage: Perforation des Trommelfells, Unterbrechung der Gehörknöchelchenkette, Hörsturz, Knalltrauma, Lärmschwerhörigkeit, Presbyakusis (Altersschwerhörigkeit), Mittelohrentzündung, Barotrauma, Otosklerose, Schädigung durch Ototoxine[74], Durchblutungsstörungen, Schädigung durch Infektionskrankheiten, Schädigung durch Autoimmunkrankheiten, Schädigung durch Läsionen des Nervensystems, insbesondere durch mechanische Belastung von Hirnnerven. Beeinträchtigungen des Gehörs können unter Umständen, durch unzureichende auditive Rückmeldung des Gesprochenen, Ursache von Sprechstörungen und Sprachstörungen sein.

Systemdesign von Sprachsteuerungen. Im Zuge von Systemdesignüberlegungen sollte im Allgemeinen auf einen ausreichenden Variationsbereich einer für den Benutzer individuellen Lautstärkeeinstellung der wiedergegebenen Schallsignale geachtet werden. Gegebenenfalls können Rückmeldungen der Sprachsteuerung auch über eine Bildersprache erfolgen. Bei Sprachausgaben soll auf angenehmes Sprechtempo und gute Verständlichkeit geachtet werden. Zum Zwecke der Signalisierung sind Breitbandgeräusche den Sinustönen prinzipiell vorzuziehen. Alternative Kommunikationskanäle, wie etwa eine zusätzliche optische Anzeige sollen nach Möglichkeit ebenfalls eingesetzt werden. Aus Rücksicht auf Benutzer mit dekompensiertem[75] Tinnitus empfiehlt es sich, darauf zu achten, die Benutzer nicht massiv mit unnötigen Reizen zu überfluten, da in der Regel Konzentrationsstörungen infolge des Tinnitus-Geräuschs vorliegen. Es sollte auch vermieden werden, die Benutzer unnötigen Stresssituationen auszusetzen[76], wie sie beispielsweise durch Überforderung bei der Interaktion mit dem System auftreten können, da dekompensiert von Tinnitus Betroffene üblicherweise bereits von Schlafstörungen

---

[70] Dabei ist der Hörverlust bei leisen Geräuschen stärker als bei lauten Geräuschen.

[71] Hierbei werden beispielsweise unterschiedliche Tonhöhen wahrgenommen, je nachdem, ob der Ton von links, oder von rechts kommt.

[72] Hierbei werden Geräusche von „links“ und von „rechts“ als zeitversetzt erlebt.

[73] Unter „Agnosie“ versteht man die Beeinträchtigung des Erkennens trotz intakter Wahrnehmung. Die akustische Agnosie bezeichnet die Unfähigkeit, Gehörtes mit bekannten akustischen Informationen zu vergleichen und somit zu identifizieren.

[74] Dabei handelt es sich um toxische Substanzen, die das Gehör schädigen können. Auch einer Reihe von Medikamenten werden ototoxische Wirkungen zugesprochen.

[75] Die Attribute „kompensiert“ beziehungsweise „dekompensiert“ werden verwendet, um den individuellen Leidensdruck von Tinnitus-Patienten zu klassifizieren.

[76] Oftmals wird von Betroffenen mit dekompensiertem Tinnitus auch ein (unbewiesener) Zusammenhang zwischen Tinnitus und Stress zumindest konstruiert, sodass „stressige Situationen“ von vielen Betroffenen sicherheitshalber vermieden werden.

geschwächt sind und bei Überforderung leicht das Interesse verlieren oder verzweifelt reagieren können. Gelingt es jedoch, mit dem System eine Atmosphäre entspannender Ablenkung zu schaffen, so wird diese voraussichtlich gerne angenommen[77].

2.1.6 Psychische (seelische) Störungen

Man unterscheidet eine Vielfalt psychischer Beeinträchtigungen beziehungsweise Behinderungen. Zu unterscheiden sind grundsätzlich Psychosen und Neurosen sowie Mischformen. Große psychische Belastungen gehen beispielsweise allgemein von Behinderungen aus. Psychische Störungen sind daher häufig im ursächlichen Zusammenhang mit anderen ursprünglichen Beeinträchtigungen oder Behinderungen zu sehen. Weitere Ursachen können psychische Traumata oder soziale Konflikte sein. Unter Umständen kommen auch hirnorganische Veränderungen, wie etwa Tumore als Ursache infrage.

Systemdesign von Sprachsteuerungen. Im Zuge von Systemdesignüberlegungen sollte darauf geachtet werden, dass die Sprachsteuerung interessant und mit möglichst wenigen Verwendungshemmschwellen gestaltet wird. Es sollen nach Möglichkeit bei der Benutzung keinerlei Ängste aufkommen.

2.1.7 Kognitive (geistige) Störungen

Als „kognitive (geistige) Störung“, beziehungsweise „kognitive (geistige) Behinderung“ wird im Allgemeinen ein andauernder Zustand deutlich unterdurchschnittlicher kognitiver Fähigkeiten eines Menschen im Zusammenhang mit damit verbundenen Verhaltensauffälligkeiten bezeichnet. Eine eindeutige und allgemein akzeptierte Definition ist schwierig. Üblicherweise wird bei der Definition die Minderung der maximal erreichbaren Intelligenz in den Vordergrund gestellt. Die „International Classification of Diseases“ definiert in diesem Sinne die Bezeichnung „Intelligenzminderung“. Demnach lässt sich, rein auf die Intelligenz bezogen, eine kognitive Beeinträchtigung quasi als Steigerung der Lernschwäche verstehen. In anderen Definitionen rückt statt der Intelligenz eher die Interaktion des betroffenen Menschen mit seiner Umwelt in den Vordergrund.

Einige Krankheitsbilder ähneln oberflächlich betrachtet den kognitiven Störungen, sind jedoch eher den „psychischen (seelischen) Störungen“ zuzurechnen. Dazu zählen beispielsweise: Psychosoziale Deprivation (auch Deprivationssyndrom oder Hospitalismus) und die Pseudodebilität (Pseudodemenz, Ganser-Syndrom), bei der der Intelligenzmangel nur vorgetäuscht wird. Die Demenz, als altersbedingter oder krankheits-

[77] Quelle: Tinnitus-Forum http://www.tinnitus.de/ (Stand: 26. 07. 2010).

bedingter Verlust einmal besessener Fähigkeiten, und damit auch der Intelligenz, nimmt eine Sonderstellung ein, da sie gemäß den üblichen Definitionen, nicht den „kognitiven Behinderungen“ zugerechnet wird.

Am auffälligsten sind Lernschwierigkeiten in der Schule, Verzögerungen der kognitiv-intellektuellen Entwicklung im Kindesalter sowie herabgesetztes Abstraktionsvermögen. Beobachtet wird die Verminderung der Fähigkeit, Details in einen größeren Kontext einzuordnen, sowie Leichtgläubigkeit. Nicht nur die durchschnittlich erreichbare Intelligenz, sondern teilweise auch das Anpassungsvermögen und die soziale und emotionale Reife sind beeinträchtigt. Eine kognitive Beeinträchtigung ist häufig mit anderen Besonderheiten verbunden, wie zum Beispiel: Autismus, Lernschwäche, Beeinträchtigung der Motorik und der Sprache. Eine kognitive Störung beeinflusst nicht die Fähigkeit, Gefühle, wie zum Beispiel Freude, Wut oder Leid, zu empfinden. Jedoch beeinflusst sie zum Teil die Fähigkeit, mit diesen Gefühlen umzugehen und sie zu kommunizieren[78].

Als Ursachen für eine kognitive Beeinträchtigung gelten zum einen endogene Faktoren, wie zum Beispiel Erbkrankheiten beziehungsweise Chromosomenbesonderheiten, wie beispielsweise beim Down-Syndrom, Sotos-Syndrom oder Rett-Syndrom. Zum anderen gelten exogene Faktoren als ursächlich. Zu den exogenen Faktoren zählen vor allem erworbene cerebrale Schädigungen. Dabei kommen beispielsweise folgende Ursachen infrage: Schädel-Hirn-Trauma, Sauerstoffmangel während der Geburt, Einwirkung radioaktiver Strahlung, Gehirnentzündung oder Hirnhautentzündung. Nikotinkonsum und Alkoholkonsum während der Schwangerschaft werden ebenfalls als Ursachen diskutiert. Eindeutige Ursachenzuschreibungen sind mitunter schwierig, manchmal auch gar nicht möglich. In vielen Fällen sind „Schuldzuschreibungen“ auch für eine rechtzeitige Förderung eher hinderlich und kontraproduktiv.

Systemdesign von Sprachsteuerungen. Bei der Konzeption der Benutzerdialoge sollte darauf geachtet werden, die kognitiven (geistigen) Anforderungen so gering wie möglich zu halten. Eventuell kann sich als sinnvoll erweisen, neben der Mensch-Maschine-Schnittstelle für den Anwender, zusätzliche Mensch-Maschine-Schnittstellen für Betreuungspersonen bereitzustellen, welche anderen Anforderungen genügen müssen.

[78] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Geistige_Behinderung (Stand: 28. 03. 2013).

2.2 Mögliche Hilfestellungen bei der Rehabilitation

Ein Überblick über die Anwendung von Spracherkennern zur Unterstützung der Rehabilitation findet sich in der Literatur [Noyes1992] [Fuller1995]. Die hohen Kosten gelten als Ursache für das langsame Wachstums des Marktes für Systeme auf dem Gebiet der automatischen Spracherkennung für Menschen mit Behinderungen [Noyes1992].

Bedarf nach Grundlagenforschung. Obwohl Sprachsteuerungen bereits seit etlichen Jahren in verschiedenen Anwendungsbereichen mit mehr oder weniger großem Erfolg eingesetzt werden, scheint der Durchbruch der Technologie im täglichen Leben größerer Personengruppen derzeit noch nicht gelungen zu sein. Ein allgemein verbesserter Stand der Technik hätte jedoch durchaus das Potenzial, das Leben aller Benutzer gravierend zu verändern. Für das Image des Forschungsgebiets wäre es wohl wichtig, das bisher Erreichte zu perfektionierten und robust in die Praxis umzusetzen. Dies ist eindeutig eine Aufgabe der Grundlagenforschung, denn die gewinnorientierte Produktentwicklungsstrategie der Unternehmen erlaubt bezüglich der Werbewirksamkeit von Nachfolgeprodukten stets nur noch höher angesetzte Vorgaben, die oftmals nicht robust genug erfüllt werden können, wobei die Erwartungen der Benutzer stets aufs Neue enttäuscht werden. Nachholbedarf besteht auch bei der Normung bereits in hohem Maße ausgereifter Algorithmen.

Bedarf nach benutzerzentrierter Forschung. Auf dem Gebiet der Spracherkennung ist der Anteil der Publikationen, welche den praktischen Einsatz durch Menschen mit Behinderungen beschreiben, gemessen an der Gesamtzahl der Publikationen, relativ gering. Falls sich beispielsweise im Laufe der Zeit herauskristallisiert, dass die Benutzung von Sprachsteuerungen und Diktiersystemen durch Personen mit kognitiven Störungen Vorteile bringt, so würde sich zusätzlicher Bedarf ergeben, Untersuchungen darüber anzustellen, ob und wie die kognitiven Anforderungen der Benutzung von Diktiersystemen minimiert werden können. Bezüglich Sprachsteuerungen für Menschen, die nicht funktionell sprechend sind, wären vermutlich weitergehende Forschungsanstrengungen sinnvoll. Untersuchungen bezüglich Anzahl und Häufigkeit benötigter Übungssitzungen sowie bezüglich Methoden zur Förderung der tatsächlichen Verwendung im täglichen Leben wären ebenfalls hilfreich.

Akzeptanzverringerung bei Fehlfunktion. Obwohl Sprachsteuerungen bei der Steuerung der Umgebung viele Frustrationen ihrer Benutzer beseitigen können, besteht beim derzeit üblichen Stand der Technik das große Problem häufiger Falschakzeptanzfehler, Falschrückweisungsfehler sowie Verwechslungsfehler. Eine vorläufige Verbesserung kann darin bestehen, jene Schallereignisse, die zu wiederkehrenden Fehlfunktionen führen, zu identifizieren und zu vermeiden, oder die gespeicherten Steuerbefehle beziehungsweise die gespeicherten Hintergrundgeräusche entsprechend anzupassen.

Akzeptanzerhöhung bei Benutzungsdauerbegrenzung. Um die Meinung des Benutzers bezüglich der Qualität des Systems positiv zu beeinflussen, kann es sinnvoll sein, die Zeitdauer der Benutzung zu beschränken. Wenn der Benutzer ermüdet, kann ein Ansteigen der Fehlerraten die Folge sein. Eventuell empfiehlt es sich, um die Hilfsmittelakzeptanz zu optimieren, das System kürzer zu benutzen, anstatt es für eine lange Zeitdauer mit vielen Fehlfunktionen zu benutzen. Wenn die Ausdauer steigt, kann die Benutzungsdauer erhöht werden.

Akzeptanzverringerung bei „unpraktischen" Mikrofonen. Hintergrundgeräusche spielen bezüglich der Fehlfunktionen eine große Rolle. Obwohl die Platzierung des Mikrofons nahe am Mund das Problem bis zu einem gewissen Grad entschärfen kann, treten dabei andere Probleme auf: Atemgeräusche, Störgeräusche durch Berühren des Halterungsbügels, Hilfsmittelakzeptanzprobleme, weil das Aufsetzen eines Kopfbügelmikrofons unpraktisch ist oder ohne fremde Hilfe nicht möglich ist, sowie Lautstärkeschwankungen durch variierende Kopfbügeljustierung [Noyes1992].

Akzeptanzverringerung bei hohen Kosten. Bei Sprachsteuerungen handelt es sich um sehr komplexe Systeme mit hohen Entwicklungskosten. Neben der aufwändigen Individualisierung der Systeme müssen unter Umständen komplexe benutzerspezifische Dialogabläufe samt auszulösenden Aktionen verwaltet werden. Wartung und Instandhaltung sowie die Geräte selbst sind im Allgemeinen sehr teuer. Auch wenn die technischen Probleme der Umgebungssteuerungen beseitigt wären: Würden sie eingesetzt werden? Unglücklicherweise empfehlen, entsprechend den Ergebnissen einer Umfrage, Berufstherapeuten Umgebungssteuerungen für weniger als 25 % ihrer Klienten [Holme1997]. Der Hauptgrund für die wenig häufige Empfehlung sind hohe Kosten und kaum vorhandene Finanzierungen aus dritter Hand [Noyes1992]. Obwohl dieses Umfrageergebnis alarmierend ist, sollte die Möglichkeit der Benutzung von Diktiersystemen und Sprachsteuerungen durch Menschen mit Behinderungen nicht außer Acht gelassen werden.

2.2.1 Hilfestellung bei motorischen Störungen

Aus den Vorzügen von Sprachsteuerungen ergibt sich eine besondere Eignung für Personen, die nicht oder nur unter großem Aufwand befähigt sind, entsprechend adaptierte Schalter und Sensoren zu bedienen. Die therapeutischen Hoffnungen beim Einsatz von Spracherkennern beruhen für Personen mit schweren motorischen Behinderungen auf der verbesserten Möglichkeit, sich als selbstständig handelnd zu erleben.

Sprachsteuerungen werden seit längerer Zeit von mehrfach behinderten Menschen mit normaler Sprache verwendet. Ziel ist es, die Unabhängigkeit zu fördern, wobei die Sprachsteuerung benutzt wird, um Schallereignisse in effektive Aktionen umzusetzen. Unter Umständen ist die Sprache das einzige verbleibende Kommunikationsmittel für Menschen dieser Benutzergruppe.

2.2.1.1 Rollstuhlnavigation

Der erste mittels einer Sprachsteuerung navigierte Rollstuhl wurde in den späten Siebzigerjahren des vorigen Jahrhunderts in New York entwickelt[79]. Weiters wurde die Umgebungssteuerung von Telefon, Radio, Ventilatoren, Vorhängen, Blattwendegeräten vom Rollstuhl aus ermöglicht. Eine Gruppe von Testpersonen mit Cerebralparese bewertete den sprachgesteuerten Rollstuhl im Vergleich zu atemgesteuerten Systemen. Das Systemdesign des sprachgesteuerten Rollstuhls wurde besser angenommen, weil die Notwendigkeit des Scannens, das heißt, des Weiterschaltens eines Aktionsauswahl-Cursors, wegfällt und die direkte Auswahl der Aktion mit Sprache zumindest im Prinzip wesentlich schneller erfolgen kann.

2.2.1.2 Texterstellung

Hierbei kommen Diktiersysteme zum Einsatz, beispielsweise das „IBM ViaVoice Dictation System“. Es gilt jedoch zu bedenken, dass die Hilfsmittelakzeptanz sinkt, wenn sich Fehlfunktionen häufen. Im Vergleich zu Sprachsteuerungen handelt es sich bei Diktiersystemen um noch komplexere Systeme. Aufgrund des wesentlich größeren Wortschatzes und des dadurch zur Einschränkung der Perplexität notwendigen Grammatikmodells, ergeben sich zusätzliche Möglichkeiten für Fehlfunktionen.

2.2.1.3 Muskelstimulation

Sprachsteuerungen können von motorisch behinderten Patienten verwendet werden, um sogenannte „Funktionelle Elektrostimulationssysteme“ zu aktivieren. Dies erlaubt den Patienten, nach erfolgter Unterweisung durch den Physiotherapeuten, eine gewisse Routine im Rehabilitationsprozess ohne permanente Involvierung einer Krankenschwester zu erlangen. Es kann motivierend für die Patienten sein, einen gewissen Grad der Kontrolle im Rehabilitationsprozess zu erreichen.

2.2.1.4 Orthesenbewegungssteuerung

Bei „Orthesen“ handelt es sich um bewegliche Gelenksstützen, die unter anderem in der Schlaganfalltherapie eingesetzt werden. Ein wichtiger Bestandteil der Therapie nach einem Schlaganfall ist das ausreichende und gezielte Bewegen der

[79] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).

betroffenen Gliedmaßen. Solange der Patient die Bewegungen nicht selber ausführen kann, liegen sie in der Verantwortung des Pflegepersonals. Diese Pflegetätigkeit erfordert sehr viel Zeit. Eine mögliche Lösung besteht in der Anwendung automatisch bewegter Gelenksstützen.

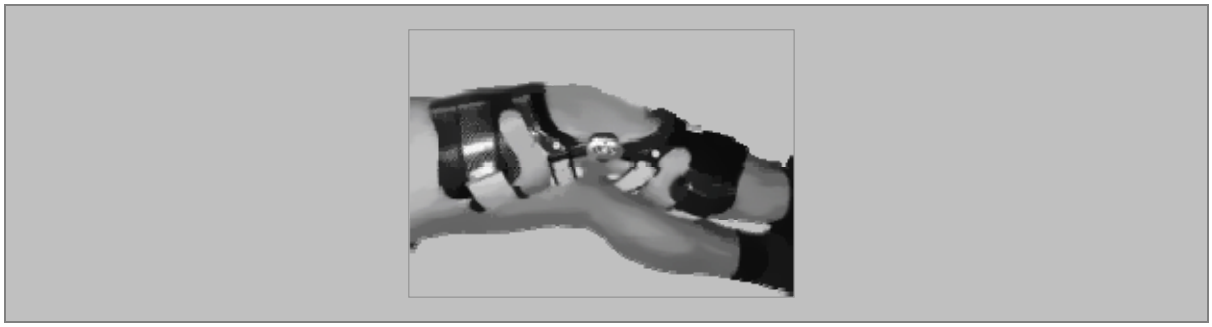

Abbildung 2-4: Orthese, gesteuert von der SICARE-Sprachsteuerung.

Bei speziell ausgerüsteten Orthesen besteht die Möglichkeit, zyklische Bewegungsfolgen sowie verschiedene Bewegungsparameter, wie beispielsweise die Bewegungsrichtung und die Bewegungsgeschwindigkeit, vom Patienten mittels Sprachbefehlen über die SICARE-Sprachsteuerung [Tschirk1999] einzustellen. Beispielsweise sind Ellbogenorthesen und Kniegelenkorthesen im Einsatz. Eine Kniegelenkorthese zeigt Abbildung 2-4.

2.2.1.5 Fragebogenausfüllhilfe

Das Ausfüllen von Fragebögen ist für Personen mit Mehrfachbehinderungen eine schwierige und manchmal unmögliche Aufgabe, da dabei im Allgemeinen der Bedarf zu Schreiben besteht. Es werden mündliche Antworten in Verbindung mit einem Spracheingabesystem als Alternative vorgeschlagen [Noyes1992].

2.2.1.6 Telefonzugang

DAISY-Sprachwahlgerät. Abbildung 2-5 zeigt das DAISY-Sprachwahlgerät zum Nachrüsten handelsüblicher Festnetztelefone. Das Abheben des Hörers verbleibt bei dieser Lösung als einzige motorische Interaktion zum Aufbau einer Verbindung. Falls dies nicht akzeptabel ist, sind weitere Lösungen in Zusammenhang mit sprachaktivierten Freisprecheinrichtungen möglich.

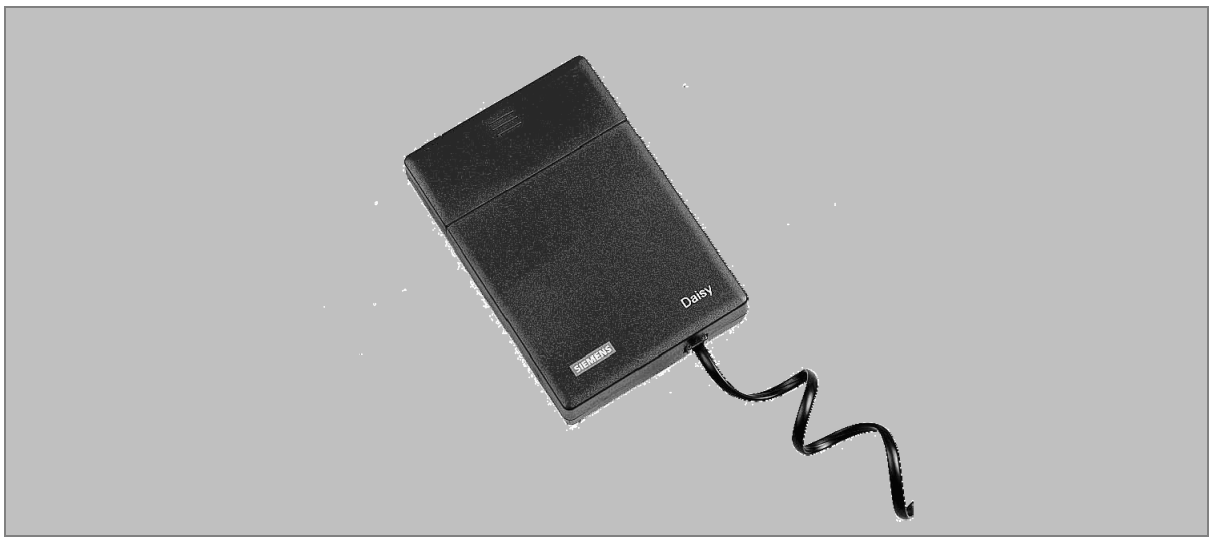

Abbildung 2-5: DAISY-Sprachwahlgerät zum Nachrüsten handelsüblicher Telefonapparate.

AUTONOM-System. Das AUTONOM-System [Zagler1993] [Flachberger1994] [Panek1996] [Zagler1997] [Panek2002] ermöglicht ebenfalls Zugang zum Telefon und eignet sich in Verbindung mit einer Sprachsteuerung besonders aufgrund der individuellen Konfigurierbarkeit und der Universalität des Systemkonzepts. Für eine detailliertere Beschreibung des AUTONOM-Systems sei auf Abschnitt 3.3 verwiesen.

2.2.1.7 Computerzugang und Internetzugang

Der Zugang zu Personal Computer und Internet kann für Menschen mit motorischen Beeinträchtigungen ein „Tor zur Welt“ sein und ist aufgrund der Interaktivität dem Fernsehen aus therapeutischer Sicht sicherlich vorzuziehen. Auch einfache oder komplexere Computerspiele, beispielsweise Kartenspiele, können Abwechslung bringen. Computerspiele weisen gegenüber nicht virtuellen Brettspielen den Vorteil auf, dass die motorischen Anforderungen geringer sind und das Wegräumen nach dem Spiel entfällt. Spracherkenner können die Anwendbarkeit dieser therapeutischen Perspektiven auf Personen mit schwereren Formen motorischer Beeinträchtigung erweitern. Computerzugang kann allgemein mittels eines Brain-Computer-Interface [Fichter2002] realisiert werden.

Sprachsteuerungsapplikationen. Der Einsatz kommerzieller Sprachsteuerungsapplikationen direkt am Personal Computer bietet sich zur Steuerung desselben an, jedoch gilt beim derzeit üblichen Stand der Technik nach wie vor, dass die Erwartungen bezüglich der Verwendbarkeit nicht zu hoch angesetzt werden sollten.

AUTONOM-System. Die allgemeine Steuerung eines Personal Computers kann mittels des AUTONOM-Systems [Zagler1993] [Flachberger1994] [Panek1996] [Zagler1997] [Panek2002] wesentlich unterstützt werden. In Verbindung mit einer Sprachsteuerung kann das Anwendungsgebiet weiter vergrößert werden. Für eine detailliertere Beschreibung des Autonom-Systems sei auf Abschnitt 3.3 verwiesen.

2.2.1.8 Umgebungssteuerung

Der Lösungsansatz des Einzelgerätekonzepts besteht darin, jedes einzelne Gerät für sich mit einer Sprachsteuerung auszustatten. Vorteile bietet dieser Ansatz vor allem bei mobilen Geräten, wie beispielsweise sprachgesteuerten Rollstühlen oder sprachgesteuerten Mobiltelefonen, da bei geforderter Mobilität eine zentrale Sprachsteuerung zur Umgebungssteuerung schwierig realisierbar ist. Als wichtige Voraussetzung für die Brauchbarkeit gilt für Menschen mit motorischen Beeinträchtigungen, dass zur Bedienung jedes einzelnen Gerätes möglichst keine motorische Interaktion, wie zum Beispiel manuelles Einschalten, erforderlich sein soll. Auch vorwiegend nichtmobile Geräte werden mit Sprachsteuerungen ausgestattet, wie zum Beispiel sprachgesteuerte Festnetztelefone. Bei Verwendung mehrerer sprachgesteuerter Einzelgeräte ist auf die Problematik Rücksicht zu nehmen, dass jedes Gerät unter Umständen auch die Kommandos mithört, die für die jeweils anderen Geräte bestimmt sind. Die erlebte, gemeinsame Falschakzeptanzrate kann unter Umständen drastisch ansteigen. Es ist daher gegebenenfalls auf möglichst unterschiedlich klingende Aktivierungswörter[80] der einzelnen Geräte zu achten. Im Gegensatz zum Einzelgerätekonzept übernimmt beim Mehrgerätekonzept ein zentraler Spracherkenner, welcher beispielsweise an einem Rollstuhl montiert ist, die Steuerung mehrerer Geräte.

„Voice Activated Control System“. Eine der ersten sprachgesteuerten sprecherabhängigen Umgebungssteuerungen stellt das „Voice Activated Control System“, aus den späten Siebzigerjahren des vorigen Jahrhunderts dar. Es besteht aus einem Mikrofon, einer Merkmalsextraktionseinheit, einem Minicomputer, einem elektronischen Display, einer Teletype-Schnittstelle und einer Relay-Schnittstelle. Neben dem „Steuerungsmodus“, welcher die Steuerung mehrerer angeschlossener Peripheriegeräte ermöglicht, gibt es einen „Texteingabemodus“, in welchem frei wählbare Texte buchstabiert werden können und einen „Rechenmodus“, welcher die Grundrechnungsarten bereitstellt. Obwohl die Funktionsweise keineswegs ausgereift ist, gibt die Einführung dieses Geräts in den Siebzigerjahren des vorigen Jahrhunderts die andauernde Hoffnung, dass diese Technologie die Leben von Personen mit Behinderungen verbessern könnte [Noyes1992].

---

[80] Ein sogenanntes „Aktivierungswort“ stellt eine Möglichkeit dar, um die Falschakzeptanzrate zu senken: Im Ruhezustand reagiert das Gerät ausschließlich auf ein Aktivierungswort. Dieses bringt das Gerät in den aktivierten Zustand, in welchem es auf die eigentlichen Steuerbefehle reagiert.

„Voice Activated Domestic Appliance System“. Die Anwendung von Umgebungssteuerungen zu Hause wird seit den Achtzigerjahren des vorigen Jahrhunderts ernsthaft untersucht. Bereits Mitte der Achtzigerjahre ist das „Voice Activated Domestic Appliance System“ verfügbar, welches eine größere Anzahl an Haushaltsgeräten steuern kann [Noyes1992].

Palo-Alto-Haushaltsroboterarmsteuerung. Ebenfalls seit den Achtzigerjahren des vorigen Jahrhunderts wird, beispielsweise initiiert vom „Palo Alto Veterans Administration Medical Center“, an der Sprachsteuerung von Roboterarmen geforscht. Das Ziel besteht darin, Personen bei Aufgaben im Haushalt zu unterstützen, wie zum Beispiel Küchenvorbereitungen, Entspannung, oder Therapie. Berichtet wird von häufigen Fehlfunktionen [Noyes1992].

Voice-Command-Sprachsteuerung. Bei dieser Sprachsteuerung werden ZCPA-Merkmalsvektoren und ein „Radial Basis Function Classifier Speech Recognizer“ eingesetzt. Die Rückweisung von Non-Keyword-Schallmustern basiert nicht auf der expliziten Modellierung solcher, sondern auf der Abweichung des Schallmusters von den durch Keyword-Schallmusterklassen [Lee1997].

SICARE-Sprachsteuerung. Bei der SICARE-Sprachsteuerung [Tschirk1999] [Tschirk2001] handelt es sich um eine Umgebungssteuerung bestehend aus einer sprachgesteuerten Universalfernbedienung und verschiedenen Peripheriekomponenten, wie beispielsweise: Antriebe zum Öffnen von Türen und Fenstern, Lichtschalter, Blattwendegeräte für Bücher, Fernseher, CD-Player, Videorekorder, Heizungen, Klimaanlagen, Jalousien, Rollstühle, elektrisch verstellbare Betten, Telefone.

Abbildung 2-6: SICARE-Sprachsteuerungen. Abgebildet sind Gehäuseformen verschiedener Varianten.

Die Steuerung der Peripheriegeräte erfolgt über Infrarotverbindungen, Funkverbindungen oder drahtgebundene Verbindungen. Abbildung 2-6 zeigt die Gehäuseformen verschiedener Varianten [Tschirk2001].

Im spanischen Krankenhaus „Hospital Nacional de Parapléjicos de Toledo“ wird eine Studie [GutierrezFayos2003] mit 30 Patienten durchgeführt, um zu evaluieren, inwieweit die SICARE-Sprachsteuerung helfen kann, durch Steuerung der Haushaltsinfrastruktur die Lebensqualität zu verbessern[81]. 82 % der Befragten sagten aus, dass die SICARE-Sprachsteuerung ihre Lebensqualität verbessert hat. 80 % können ihre persönliche Unabhängigkeit erhöhen. 63 % sagten aus, dass sie mit dem System generell besser zurechtkommen. Die Belastung der Patienten durch Stress beim Umgang mit der neuen Technologie wird als „gering“ bezeichnet [GutierrezFayos2003].

AUTONOM-System. Die Steuerung der Umgebung ist nur eine der Einsatzmöglichkeiten des AUTONOM-Systems [Flachberger1994] [Panek1996] [Zagler1997] [Panek2002]. Ein großes Sortiment von Peripheriegeräten lässt sich steuern. Die Steuerung der Peripheriegeräte erfolgt über Infrarotverbindungen, drahtgebundene Verbindungen, Telefonverbindungen, sowie Verbindungen mittels Trägerfrequenzverfahren über das 230V-Hausnetz [Zagler1993]. Für eine detaillierte Beschreibung sei auf Abschnitt 3.3 verwiesen.

2.2.2 Hilfestellung bei Sprechstörungen

Spracherkenner haben prinzipiell das Potenzial, Hilfestellungen zur Überwindung verschiedener Barrieren bei Sprechstörungen zu leisten. Obwohl es erfreulich ist, dass in verschiedenen Studien die Fehlerraten bei Sprechern mit Sprechstörungen und die Fehlerraten bei Sprechern ohne Sprechstörungen in den gleichen Größenordnungen liegen, ist dabei zu bedenken, dass die Brauchbarkeit derzeit üblicher Spracherkenner, insbesondere derzeit üblicher Diktiersysteme, noch oftmals hinter den bereits hohen Erwartungen der Gesellschaft zurückbleibt. Allgemeine Verbesserungen des Stands der Technik hätten sicherlich auch positive Auswirkungen auf die Fähigkeiten der Systeme zur Erkennung der Sprache von Personen mit Sprechstörungen. Beispielsweise besteht eine Barriere zum erfolgreichen Einsatz von Diktiersystemen durch Personen mit Dysarthrie in der Variabilität der Aussprache: Die Aussprache von Sprechern mit Dysarthrie variiert nicht nur zwischen den einzelnen Personen, sondern kann auch für einen einzelnen Sprecher variieren, abhängig von Uhrzeit, Ermüdung, Stress oder anderen Einflüssen. Daher schwankt die Effektivität der Diktiersysteme von Zeit zu Zeit und von Ort zu Ort.

[81] In Spanien sind etwa 3,5 Millionen Menschen, das sind etwa 9 % der Gesamtbevölkerung von einer Behinderung betroffen. Etwa 2,3 Millionen Menschen haben Schwierigkeiten, mit dem täglichen Leben zurechtzukommen.

Anpassung der Spracherkenner. Eine Studie berichtet über zwei holländische Sprecher mit leichter Dysarthrie, welche ein Continuous-Speech-System auf der Basis von Hidden-Markov-Modellen verwenden [Sanders2002]. Es wurden Versuche mit verschiedenen Vokabulargrößen sowie mit und ohne Grammatikmodell durchgeführt und jeweils die Wortfehlerraten bestimmt. Dabei wurde versucht, die Anzahl der Summanden, der „Gaussian Mixtures“, in den Sub-Word-Unit-Modellen optimal an die Sprache der Sprecher mit Dysarthrie anzupassen. Siehe Gleichung 24 im Zusammenhang mit Abbildung 1-21 auf Seite 52. Mit einem Vokabular von 516 Wörtern ohne Grammatikmodell ergaben sich, bei sprecherunabhängigem Training für die beiden Normalsprecher bezüglich der Ziffern von null bis zwölf, Verwechslungsfehlerraten von 12,8 % beziehungsweise 18 %. Für die beiden Sprecher mit Dysarthrie ergaben sich Verwechslungsfehlerraten von 64,1 % beziehungsweise 100 %. Durch sprecherabhängiges Modelltraining verbesserten sich die Verwechslungsfehlerraten der Sprecher mit Dysarthrie auf 28,2 % beziehungsweise 20,5 %. Eine andere Studie [Deller1991] beschreibt konkrete Vorschläge, vor allem betreffend des Zustandsmodells des Viterbi-Algorithmus, um Spracherkenner, die auf Hidden-Markov-Modellen beruhen, besser an die Gegebenheiten der Sprache von Sprechern mit Dysarthrie anzupassen. Siehe Gleichung 34 und Abbildung 1-13 auf Seite 38 im Zusammenhang mit der Literatur.

Qualitätssicherungsanalyse. Ob der Satz an Steuerbefehlen die Anforderungen bezüglich Falschakzeptanzrate, Falschrückweisungsrate sowie Verwechslungsfehlerrate erfüllt, wird bei der implementierten Sprachsteuerung mittels einer automatischen Qualitätssicherungsanalyse vorab geschätzt. Somit eignet sich der implementierte Spracherkenner [Hickersberger2004] unter bestimmten Umständen insbesondere für Personen mit deutlichen Sprechstörungen.

2.2.2.1 Sprachaufbesserung

Sprachausgabe von Diktiersystemen bei Sprechstörungen. Eine theoretische Möglichkeit besteht darin, ein geeignetes Diktiersystem mit einem Sprachausgabesystem zu verbinden, um schwer verständliche Sprache in eine verständlichere Form zu bringen [Alter1996]. Davon könnten vor allem Personen mit zusätzlicher motorischer Behinderung profitieren. Allerdings sollten derzeit die Erwartungen diesbezüglich nicht zu hoch angesetzt werden, da derzeit übliche Diktiersysteme noch keineswegs ausgereift sind.

Sprachausgabe des AUTONOM-Systems bei Sprechstörungen und Sprachstörungen. In Verbindung mit der im Rahmen dieser Arbeit implementierten Sprachsteuerung können mittels individuell konfigurierbarer Steuerbefehle, wobei unterscheidbare Laute oder Geräusche genügen, am AUTONOM-System verschiedene genormte oder benutzerspezifische Symbole angewählt werden, die die Sprachausgabe konfigurierbarer Wörter oder Sätze auslösen. Eine weitere Möglichkeit besteht darin,

einen Text Buchstaben für Buchstaben von einer Buchstabentafel zusammenzustellen. Techniken der Textvorhersage können dabei die Texterstellung beschleunigen.

Für eine detaillierte Beschreibung des AUTONOM-Systems sei auf Abschnitt 3.3 sowie auf Literatur verwiesen [Zagler1993] [Flachberger1994] [Panek1996] [Zagler1997] [Panek2002].

2.2.2.2 Umgebungssteuerung

Der Markt von Sprachsteuerungen zur Verwendung durch Sprecher mit Sprechstörungen hat vermutlich großes Wachstumspotenzial. Von Vorteil wären kontinuierliche Forschungsbemühungen, um die Effektivität verschiedener Systeme bei der Benutzung durch Sprecher mit Sprechstörungen, insbesondere mit schwankender Aussprache und schlechter Verständlichkeit zu evaluieren. Neue Entwicklungen finden sich in [Hawley2007].

Verwendung kleiner Vokabulargrößen. Es wird von Erfolgen bei der Entwicklung von Sprachsteuerungen für Personen mit Dysarthrie berichtet [Jayaram1995], aber andererseits wird von einem dramatischen Anstieg der Verwechslungsfehlerrate für Vokabulargrößen über dreißig Wörtern berichtet [Goodenough1991]. Eine Perspektive bezüglich der Benutzung von Sprachsteuerungen für diese Personengruppe besteht daher anscheinend in der Verwendung kleiner Vokabulargrößen, um Fehlfunktionen vermeiden. Wenn die Sprachsteuerung dafür geeignet ist, muss der Benutzer nicht unbedingt bestimmte Steuerbefehle sprechen, sondern ein Satz von lautlichen Äußerungen kann hinreichend sein.

Klassifizierung der Benutzer. Es besteht eine gewisse Hoffnung, die Benutzer in Klassen einteilen zu können, wobei für jede Klasse spezielle Strategien erfolgreich sein könnten. Zum Beispiel könnten Personen mit Stimmbandbeeinträchtigungen und Lautstärkeschwankungen mit anderen Steuerbefehlen ihre besten Ergebnisse erzielen, als Sprecher deren vordringliches Defizit im Erzielen der natürlichen Zeitdauern der Phone liegt. Wenn es gelänge, die Benutzer in Klassen einzuteilen, so wäre damit zu rechnen, dass weniger Zeit benötigt werden würde, um für jeden einzelnen Benutzer den, bezüglich der Benutzerzufriedenheit optimalen, Satz an Äußerungen zu finden [Goodenough1991].

Sprachsteuerung in Kombination mit „Scannen“. Eine Studie[82] untersucht die Benutzung von Sprachsteuerungen in Verbindung mit Scannen, um die Geschwindigkeit von Computereingaben funktionell nicht sprechender Personen zu erhöhen. Nebenbei werden die kognitiven Anforderungen beurteilt. Zwei Methoden, nämlich Scannen allein und Scannen in Verbindung mit einer Sprachsteuerung werden verglichen. Die Ergeb-

[82] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).

nisse eines zwölf Jahre alten Knaben werden beschrieben: Er kann drei diskrete, wiederholbare Äußerungen produzieren. Die Äußerungen „ma“, „hey“ und „heya“ werden folgenden Aktionen zugewiesen: Aufheben der letzten Auswahl, Auswahl der anderen Spaltenhälfte beziehungsweise der anderen Zeilenhälfte, Auswahl einer Zeile mit Verben. Es ergibt sich, dass der Teilnehmer durch die Kombination des Scannens mit einer Sprachsteuerung mehr ungewollte Aktionen auswählt, als bei alleinigem Scannen. Jedoch kann nach einigen Sitzungen immerhin die Anzahl der ausgewählten Aktionen pro Minute signifikant erhöht werden. Es sieht demnach so aus, als wären durch die Kombination des Scannens mit einer Sprachsteuerung Vorteile zu erzielen. Forschungsbemühungen mit einer größeren Anzahl an Personen wären jedoch sicherlich notwendig, um endgültige Schlüsse zu ziehen.

2.2.2.3 Texterstellung

Buchstabentafel des AUTONOM-Systems. Das AUTONOM-System stellt die Möglichkeit zur Verfügung, per Cursor auf einer Buchstabentafel längere Texte zusammenzustellen, die dann gespeichert beziehungsweise per Sprachausgabe vorgelesen werden können. Die Steuerung des Cursors ist mittels der im Rahmen dieser Arbeit entwickelten Sprachsteuerung auch mit wenigen aber genügend unterscheidbaren Äußerungen beziehungsweise Lauten möglich.

„Dragon Dictate Voice Recognition System“. In einer Studie wurde die Einzelwortverständlichkeit einer Person mit Dysarthrie mittels des Verständlichkeitstests „Computerized Assessment of Intelligibility of Dysarthric Speakers“ bestimmt. Dieser Verständlichkeitstest setzt normal hörende menschliche Schiedsrichter voraus. Der Messwert lag bei etwa 84 % und entsprach in etwa der Worterkennungshäufigkeit, die die entsprechende Person mit Dysarthrie bei der zweiten Sitzung mit dem „Dragon Dictate Voice Recognition System“ erreichte. Dies wird als Hinweis darauf gesehen, dass der Verständlichkeitstest „Computerized Assessment of Intelligibility of Dysarthric Speakers“ zur quantitativen Vorhersage dienen kann, wie erfolgreich ein Benutzer bei der Verwendung des „Dragon Dictate Voice Recognition System“ wäre [Ferrier1992]. In einer weiteren Studie [Ferrier1995] benutzen zehn Sprecher, die unter Dysarthrie infolge von Cerebralparese leiden, das „Dragon Dictate Voice Recognition System“. Im Rahmen der Studie wird versucht, folgende Fragestellungen zu klären: Hat die Verständlichkeit der Sprecher eine Auswirkung auf die Fehlerraten des Spracherkenners? Dabei zeigt sich ein starker Zusammenhang. Welche Faktoren haben neben der Verständlichkeit sonst noch Auswirkung auf die Fehlerraten des Spracherkenners? Dabei zeigen sich starke Zusammenhänge mit Müdigkeit, Pausenhäufigkeit und Störgeräuschhäufigkeit. Störungen des Redeflusses haben nur schwachen Einfluss. Wie weit weichen die Fehlerraten des Spracherkenners bei Patienten mit Dysarthrie infolge von Cerebralparese von den Fehlerraten normal sprechender Personen ab? Es zeigt sich interessanterweise, dass unter

Umständen sogar schwer verständliche Personen das Niveau der Normalsprechenden erreichen können.

„Microsoft Dictation System“. „Dragon Naturally Speaking Voice Recognition System“. „IBM VoicePad Speech Recognition System“. Eine vergleichende Studie behandelt die Fehlerraten verschiedener Spracherkenner mit Sprechern ohne Sprechstörungen und mit Sprechern mit leichten Sprechstörungen. Je nach verwendetem System und verwendeten Sätzen lagen jeweils nach der fünften Trainingseinheit die Worterkennungshäufigkeiten für die Patienten mit leichter Dysarthrie zwischen 39 % und 65 %. Für die normal sprechenden Kontrollpersonen lagen die Worterkennungshäufigkeiten zwischen 80 % und 92 %. Obwohl diese Ergebnisse auch bezüglich der normal sprechenden Personen enttäuschend erscheinen, weist der Autor der Studie darauf hin, dass unter gewissen Umständen, nämlich wenn Spracherkennung die einzige zur Verfügung stehende Möglichkeit der Texterstellung ist, auch eine Worterkennungshäufigkeit von 65 % akzeptabel sein kann [Hux2000].

„IBM VoiceType Dictation System“. In einer Studie[83] werden die Fehlerraten des „IBM VoiceType Dictation System“ bezüglich sechs Sprechern mit Dysarthrie mit den Fehlerraten, die sechs normal hörende erwachsene Zuhörer aufweisen, verglichen. Das „IBM VoiceType Dictation System“ liefert bei Sprechern ohne Sprechstörung bessere Fehlerraten als bei Sprechern mit Dysarthrie, die in Geschlecht und Alter dazupassen. Dennoch unterscheiden sich die Lernkurven zwischen beiden Gruppen nicht signifikant. Graduelle Verbesserungen der Fehlerraten des Spracherkenners wurden bei beiden Gruppen bei jeder Sitzung erzielt. Die menschlichen Zuhörer erzielten keine Verbesserungen ihrer Fehlerraten. Die Sprecher wurden mit dem Test „Computerized Assessment of Intelligibility of Dysarthric Speakers“ in verschiedene Verständlichkeitsklassen eingeteilt. Interessant ist, dass die normal hörenden Zuhörer in Gegensatz zu dieser Einteilung die Verständlichkeitsklasse „leichte Dysarthrie“ und die Verständlichkeitsklasse „mittlere Dysarthrie“ relativ ähnlich bewerteten. In einer anderen Studie [Kotler1997] wird der Einfluss logopädischer Trainingseinheiten der Benutzer auf die Fehlerraten des „IBM VoiceType Dictation System“ untersucht. Es zeigen sich signifikante Erfolge beim Training genau jener Vokabeln, mit denen der Spracherkenner Probleme hat.

2.2.2.4 Verständlichkeitsmessung und Sprechtraining

Ein weiteres interessantes Forschungsgebiet ist die automatisierte Verständlichkeitsmessung und die computergestützte Therapie bei Sprechstörungen. Es ist zu erwarten, dass Menschen mit Sprechstörungen ihre Kommunikationsfähigkeit verbessern können, wenn sie spezielle Sprechtrainingssysteme verwenden, die Rückmeldungen in geeigneter Art und Weise liefern. Auch die Benutzung von Sprachsteuerungen oder

[83] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).

Diktiersystemen scheint sich therapeutisch günstig auszuwirken, da der Benutzer beim Umgang mit dem System nebenbei trainiert, die Steuerbefehle stets in gleicher Weise auszusprechen.

Eine Studie[83] behandelt die Verwendung von Sprechtrainingssystemen durch Personen mit traumatischen Hirnverletzungen und anhaltenden Dysarthrien. Personen mit Hörverlust können ebenfalls von der Möglichkeit, ihre Verständlichkeit zu verbessern, profitieren [KewleyPort1991]. Der Übungsablauf sollte grundsätzlich so gestaltet werden, dass die kognitiven Anforderungen so gering wie möglich sind, um den Erfolg der Übungen gegebenenfalls diesbezüglich nicht zu gefährden.

Unkalibrierte Sprechtrainingssysteme. Einfache Sprechtrainingssysteme basieren auf der Transformation des Sprachsignals in eine sichtbare Darstellung, ohne jedoch kalibrierte Messergebnisse darzustellen. Solche Systeme können beispielsweise dazu verwendet werden, um die Beeinflussung der Lautstärke und der Tonhöhe zu üben, wobei das Display dem Benutzer die unkalibrierte Messgröße, beispielsweise farbcodiert, zeigt. Beim „Visi Pitch Speech Training System“ handelt es sich beispielsweise um ein solches Trainingssystem.

Kalibrierte Sprechtrainingssysteme. Kalibrierte Sprechtrainingssysteme stellen kalibrierte Messergebnisse, wie etwa Sonagramme, dar. Physikalische Signaleigenschaften können dargestellt werden, aber keine Information bezüglich der Verständlichkeit wird dargestellt.

Bewertende Sprechtrainingssysteme. Die Sprechtrainingssysteme dieser Gruppe bewerten die Verständlichkeit und werden als „Ziel für die Zukunft“ gesehen[84]. Kalibrierte akustische und physiologische Größen sowie Modelle der Sprachproduktion sowie der Psychoakustik werden bei der Analyse verwendet. Der Zusammenhang zwischen akustischen oder physiologischen Messgrößen und der Verständlichkeit wird dargestellt.

Verständlichkeitsmessung mittels des Münchner Verständlichkeitsprofils. Dieser Test basiert auf Listen sogenannter „Minimalpaare“. Dabei handelt es sich um Wörter, die sich lediglich um ein Phonem unterscheiden, beispielsweise „Name“ und „Dame“ oder „Butter“ und „Mutter“. Aus einer Liste von Minimalpaaren wird eine Liste von Einzelwörtern erstellt, indem jeweils ein Wort jedes Minimalpaares ausgewählt wird. Die entstandene Wortliste wird vom Patienten vorgelesen und ein normal hörender menschlicher Zuhörer notiert, was er verstanden hat. Aus den notierten Ergebnissen wird schließlich das Profil der Sprechstörung ermittelt [Ahrndt1992] [Ziegler1992]. Es liegt die Überlegung nahe, den menschlichen Zuhörer durch einen Spracherkenner zu ersetzen [Hickersberger1997] und den Test vollständig zu automatisieren. Dafür würde sich vorzugsweise ein Spracherkenner eignen, dessen „Aufmerksamkeit“ auf das unterschiedliche Phonem des Minimalpaars fokussiert werden kann. Insbesondere trifft des auf

---

[84] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).

Spracherkenner mit „Weighted Hidden Control Neural Networks“ zu [Na1996]. Verallgemeinerungen des Prinzips liegen jedoch auf der Hand.

Verständlichkeitsmessung mittels Sprachsteuerungen. In einer australischen Studie [Coleman1991] wird der Grad der Dysarthrie von zehn Personen mit Cerebralparese anhand der Verwechslungsfehlerrate einer Sprachsteuerung, nämlich der „Shadow-VET2-Sprachsteuerung“, bewertet. Als Quintessenz ergibt sich, dass zwar die Verwechslungsfehlerrate bei den Sprechern mit Dysarthrie durchschnittlich höher ist als bei den Sprechern der normal sprechenden Kontrollgruppe, aber in allen Gruppen die gleichen Muster bei den Verwechslungsfehlerraten zu erkennen sind: Beide Gruppen haben höhere Verwechslungsfehlerraten für Konsonantenminimalpaare als für Vokalminimalpaare. Beide Gruppen haben höhere Verwechslungsfehlerraten für Lautvertauschungen als für Falschbetonungen. Es wird daher angenommen, dass allgemeine Verbesserungen der Sprachsteuerung für Sprecher ohne Sprechstörung auch Verbesserungen für Sprecher mit Dysarthrie darstellen.

Verständlichkeitsmessung mittels „Dynamic Time Warping“. Im Rahmen einer Studie nahmen Personen mit Sprechstörungen infolge des Parkinson-Syndroms und infolge von Schlaganfällen teil [Gu2005]. Die Bewertung der Verständlichkeit erfolgt in dieser Studie mittels „Dynamic Time Warping“, wobei als Distanzfunktion die Itakura-Saito-Distanz Verwendung findet. Es wird ein spezieller „Mean Opinion Score“ zur Verständlichkeitsmessung durch menschliche Zuhörer definiert. Die automatische Verständlichkeitsmessung mittels „Dynamic Time Warping“ und Itakura-Saito-Distanz wird so kalibriert, dass sie zur statistischen Schätzung des „Mean Opinion Score“ herangezogen werden kann.

„Speech Training Station“ und „Speech Practice Station“. Dabei handelt es sich um zwei zusammengehörige Sprechtrainingssysteme, wobei sich die „Speech Training Station“ zur Verständlichkeitsmessung und für Übungen in der Klinik und sich die „Speech Practice Station“ für unabhängige spielerische Übungen zu Hause eignet[85]. Es handelt sich um bewertende Sprechtrainingssysteme. Die „Speech Training Station“ bietet gegenüber der „Speech Practice Station“ die Möglichkeit, dem Therapeuten kalibrierte physiologische Messwerte zur Verfügung zu stellen. Folgende Fähigkeiten werden im Rahmen von Spielen trainiert: Vokalisierungen, Produktion verwandter Silben, Steuerung der Stimmintensität und Steuerung der Stimmbandgrundfrequenz. An der „Speech Practice Station“ können die gleichen Parameter wie an der „Speech Training Station“ eingestellt werden, sodass Konsistenz zwischen den Übungen in der Klinik den Übungen zu Hause gegeben ist. Therapeuten berichten, dass die „Speech Training Station“ leicht zu erlernen ist und dass die Möglichkeiten, individuell auf die Bedürfnisse von Kindern einzugehen, von Vorteil sind. Probleme ergeben sich durch Abweichungen verschiedener Messkurven von den subjektiven Einschätzungen der Therapeuten. Dies trifft beispielsweise auf die Messkurve „Stimmgebung“ zu. Weitere Probleme betreffen Lautstärkeschwankungen infolge unterschiedlicher Mikrofonpositionierung und die Empfindlichkeit gegenüber Umgebungsgeräuschen. Die Beobachtung des Verhaltens der

Kinder deutet auf eine positive Einstellung zu den Übungen hin[85]. Die Kinder üben selbstständig und benutzen den Computer sogar, wenn keine Betreuung zur Verfügung gestellt wird. Es zeigt sich, dass die Zeitspanne, die die Kinder zum Üben aufbringen, mit dem Sprechtrainingssystem erhöht werden kann.

„Indiana Speech Training Aid System". Bei diesem Sprechtrainingssystem können die besten Aufnahmen beziehungsweise die besten Versuche des Übenden als Messlatte herangezogen werden. Die Verständlichkeitsmessung erfolgt über die Worterkennungshäufigkeit eines Low-Cost-Spracherkenners [KewleyPort1991]. Die Auswertungsmöglichkeiten beinhalten Stimmbandgrundfrequenzvergleiche und Amplitudenvergleiche im Zeitbereich sowie Vergleiche im Spektralbereich. Die Übungen erfolgen teilweise in Form von Spielen. Eine Studie untersucht, ob die Korrelation zwischen der Verständlichkeitsmessung des Sprechtrainingssystems und der Verständlichkeitsbewertung normal hörender Zuhörer hoch genug ist, sodass das Sprechtrainingssystem in den jeweiligen Übungssituationen eine gute Alternative zu menschlichen Zuhörern darstellen kann. Es wird eine moderate bis starke Korrelation festgestellt und die klinische Verwendung des „Indiana Speech Training Aid System" wird unter kontrollierten Bedingungen empfohlen [KewleyPort1991].

2.2.3 Hilfestellung bei Sprachstörungen

Eventuell können auch Menschen mit Sprachstörungen, von der Benutzung eines Spracherkenners profitieren. Es sind jedoch weitere Studien notwendig, um diese Hypothese zu evaluieren[86].

2.2.3.1 Dyslexietraining und Dysgraphietraining

Der therapeutische Effekt beim Einsatz von Spracherkennern bezüglich Dyslexie (Leseschwäche) und Dysgraphie (Schreibschwäche) ist ebenfalls Gegenstand von Untersuchungen. Es zeigt sich, dass der therapeutische Effekt bei diskreten Spracherkennern, welche eine Sprechweise mit Pausen vor und nach dem Wort voraussetzen, deutlicher ausgeprägt ist, als bei kontinuierlichen Spracherkennern, welche diese spezielle Sprechweise nicht voraussetzen. Als diskretes Spracherkennungssystem kommt in der Studie das „Dragon Dictate Voice Recognition System" zum Einsatz. Als kontinuierliches Spracherkennungssystem kommt das „Dragon Naturally Speaking Voice Recognition System" zum Einsatz [Higgins2000].

---

[85] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).
[86] Quelle: http://aac.unl.edu/Speech_Recognition.html (Stand: 28. 11. 2005).

2.2.3.2 Texterstellung

Ausgereifte Diktiersysteme haben theoretisch das Potenzial, Schülern mit Dyslexie und Dysgraphie bei der Erstellung von Aufsätzen und anderen Schularbeiten zu helfen. Mehrere Studien [MacArthur2004] [DeLaPaz1999] [Higgins1995] [Higgins1995a] [Wetzel1996] vergleichen die Effektivität der Texterstellung mittels Diktiersystemen mit anderen Arten der Texterstellung für verschiedene Altersklassen insbesondere für Personen mit Lernschwäche. Folgende Arten der Texterstellung wurden verglichen: Texterstellung ohne Hilfe, Texterstellung mithilfe eines menschlichen Schreibers, Texterstellung mithilfe eines Spracherkenners.

Fokussierung der Aufmerksamkeit auf den Inhalt. Wenn viele Rechtschreibfehler gemacht werden, können die häufigen Fehlerkorrekturpausen die betroffene Person wesentlich vom Inhalt ablenken. Oftmals wird die eigentliche Aussage des Satzes bei der Fehlerkorrektur vergessen, was zu Frustration führt. Weiters wird wegen des häufigen Ausbesserns sehr viel Zeit für die Erstellung des Textes benötigt [MacArthur2004]. Der Einsatz von Diktiersystemen könnte den Schülern erlauben, sich auf die Planung und den Inhalt des Textes zu konzentrieren, anstatt auf die Mechanik des Schreibens und auf die Rechtschreibung [DeLaPaz1999].

Vokabularerweiterung beim Wegfall der Rechtschreibfehlerangst. Schüler mit Dysgraphie benutzen beim Schreiben im Allgemeinen ein vereinfachtes Vokabular um Rechtschreibfehler zu vermeiden, obwohl sie eigentlich gerne komplexere Wörter verwenden würden [DeLaPaz1999]. Eine Studie [Higgins1995a] zeigt, dass die Texterstellung mithilfe eines Spracherkenners signifikant effektiver ist, als die Texterstellung ohne Hilfe. Es werden beim Diktieren mehr Wörter, die aus sieben oder mehr Buchstaben bestehen, verwendet, als beim Schreiben ohne Hilfe. Dennoch stellt sich das Schreiben mithilfe eines menschlichen Schreibers als mindestens genauso effektiv heraus, wie das Schreiben mit einem Spracherkenner [Higgins1995a].

Verbesserung der Motivation. Viele Schüler mit Dysgraphie beziehungsweise Dyslexie tendieren dazu, eine negative Einstellung zur Texterstellung und zum Schreiben im Allgemeinen zu haben. Diktiersysteme haben grundsätzlich das Potenzial, die Produktivität und die Motivation bei der Texterstellung positiv zu beeinflussen [DeLaPaz1999].

Verbesserung der Konzepterstellung. Die Bedeutung der Vorausplanung bei der Texterstellung ist unumstritten. Es ist von Vorteil, wenn das Diktiersystem es ermöglicht, Ideen beziehungsweise Schlüsselwortlisten zu erstellen, auf die beim Diktieren zurückgegriffen werden kann [DeLaPaz1999] [Wetzel1996].

Korrekturleseunterstützung durch Sprachausgabe. Spracherkennung und Sprachsynthese zu kombinieren, stellt einen Weg dar, Schülern mit Dysgraphie beziehungsweise Dyslexie zu helfen, das Korrigieren ihrer eigenen Fehler zu erlernen. Viele Diktiersysteme beinhalten die Möglichkeit, den eingegebenen Text wieder vorlesen

zu lassen. Das Vorlesen kann dazu beitragen, den Schülern grammatikalische Fehler, die sie bei stillem Lesen übersehen hätten, bewusst zu machen. Es wird betont, dass Spracherkenner kein Ersatz für das Erlernen der grammatikalischen Regeln sein sollen. Diese Fähigkeiten müssen zusätzlich erlernt werden [DeLaPaz1999].

Unabhängigkeit von menschlichen Schreibern. Der Einsatz von Diktiersystemen kann die Unabhängigkeit fördern und Honorarkosten vermeiden, die für menschliche Schreiber anfallen würden [Higgins1995a]. Die Unterstützung durch Lehrer beziehungsweise durch Spezialausbildner wird aber dennoch anfänglich benötigt, um die Techniken der Texterstellung mithilfe eines Diktiersystems zu erlernen, sowie zur Installation und zur Konfiguration des Diktiersystems.

Kognitive Herausforderung der Texterstellung. Es wird auf die beträchtliche kognitive Herausforderung hingewiesen, die in Betracht gezogen werden muss, wenn Spracherkenner von Personen mit Dyslexie und Dysgraphie eingesetzt werden, insbesondere wenn sie kognitive Störungen aufweisen. Sorgfältige, präzise Sprache muss verwendet werden, um brauchbare Fehlerraten zu erzielen [DeLaPaz1999]. Es kann schwierig sein, stets in Betracht zu ziehen, dass sogar das kleinste Räuspern über das Mikrofon aufgezeichnet wird und unter Umständen vom Spracherkenner interpretiert wird, was zu Wörtern führt, die nicht als Teil des zu erstellenden Textes gedacht sind. Diktiersysteme sind relativ schwierig zu bedienen und eine ganze Reihe verschiedener Kommandos muss erlernt werden, um eine effiziente Benutzung zu ermöglichen. Erschwerend kommt hinzu, dass der Wechsel der Betriebsart vom „Diktieren“ zum „Editieren“ und zurück erlernt werden muss. Üblicherweise ist es erforderlich, dem Diktiersystem einen neuen Benutzer bekannt zu machen, wobei eine größere Anzahl an Testsätzen gesprochen werden muss. Dieser Vorgang kann als lang und frustrierend empfunden werden. Mit positiver Einstellung und mit der Unterstützung durch einen Trainer ist es jedoch oftmals möglich, die Bekanntmachung mit dem Diktiersystem zu meistern.

Kognitive Herausforderung der Fehlerkorrektur. Das Erlernen der nötigen Vorgänge zu Korrektur von Fehlern, die ja oftmals durch das Diktiersystem selbst bedingt sind, ist notwendig. Wenn das gewünschte Wort im Auswahlmenü zu finden ist, so ist der eher einfach erlernbare Selektionsvorgang anwendbar. Wenn das gewünschte Wort aber nicht zu finden ist, so muss es buchstabiert werden, was für Schüler mit zusätzlichen kognitiven Störungen eine Herausforderung darstellt. Das „IBM VoiceType Dictation System“ benutzt Wortvorhersage, um den Buchstabiervorgang zu vereinfachen. Zu Beginn des Buchstabiervorgangs erscheint ein Korrekturmenü am Schirm. Das Korrekturmenü zeigt eine Liste von Vorschlägen an, aus der das korrekte Wort ausgewählt werden kann, falls es erscheint. Der gesamte Korrekturvorgang benötigt große Konzentration.

Teufelskreis der Frustration. Gelegentlich wird Frustration durch Seufzer zum Ausdruck gebracht. Seufzer und andere Geräusche wie Schnäuzen oder Räuspern

werden dann vom Diktiersystem als ungewollte Wörter aufgeschnappt, was zu einem Teufelskreis steigender Frustration führen kann [Wetzel1996].

Geringe effektive Diktiergeschwindigkeit. Eine Studie [Wetzel1996] beschreibt den Einsatz des „IBM VoiceType Dictation System“ durch einen Schüler. Nach vier Sitzungen ergibt sich, eine Worterkennungshäufigkeit von 74 %. Die Diktiergeschwindigkeit beträgt 5,5 Wörter pro Minute bei der letzten Sitzung. Für diesen speziellen Schüler scheinen die Vorteile der Verwendung des Diktiersystems gegenüber dem traditionellen Schreiben nach Absolvierung der begrenzten Anzahl von Trainingssitzungen nicht zu überwiegen [Wetzel1996].

2.2.4 Hilfestellung bei auditiven Störungen

2.2.4.1 Umgebungssprachniederschrift

Die Umsetzung von gesprochener Sprache in geschriebene Sprache mittels visuell dargestellter Schrift oder taktiler Blindenschrift könnte eine wesentliche Erleichterung für Personen mit Hörverlust darstellen. Jedoch ist eine eventuelle Realisierung aufgrund des derzeitigen Stands der Technik eher als Zukunftsthema anzusehen.

2.2.4.2 Umgebungssprachaufbesserung

Aufbesserung gesprochener Sprache könnte grundsätzlich mittels Kopplung eines Diktiersystems an ein Sprachausgabesystem erreicht werden. Vorteile bestünden in der vollständigen Störsignalbeseitigung und der Aussprache mittels bekannter Stimme, wobei eine optimale Ausnutzung des Resthörvermögens ermöglicht werden würde. Die Umsetzung scheitert jedoch derzeit am unzureichenden Stand der Technik.

2.2.5 Hilfestellung bei visuellen Störungen

Ähnlich den Personen mit motorischen Störungen, können auch Personen mit visuellen Störungen Vorteile aus motorischen Hilfestellungen mittels Spracherkennern ziehen. Siehe dazu in Analogie Abschnitt 2.2.1 auf Seite 120.

2.2.6 Hilfestellung bei psychischen (seelischen) Störungen

2.2.6.1 Ursachentherapie

Positive Effekte auf psychische (seelische) Störungen sind durch den Einsatz von Spracherkennern in erster Linie dann zu erwarten, wenn es gelingt, Barrieren, welche von anderen Störungen herrühren und für die psychische (seelische) Störung ursächlich sind, signifikant zu reduzieren. Therapeutische Effekte, wie etwa Sprechtrainingseffekte oder Trainingseffekte bei Sprachstörungen oder kognitiven Störungen, können sich somit positiv auf die Psyche des Betroffenen auswirken.

2.2.6.2 Unabhängigkeitssteigerung

Sprachsteuerungen haben grundsätzlich das Potenzial, die Unabhängigkeit von Personen, die Betreuung benötigen, zu steigern, sodass sich diese Personen vermehrt als selbstständig handelnd erleben können. Dies gilt bezüglich der Routinetätigkeiten des täglichen Lebens, wobei beispielsweise mittels des Zugangs zu Personal Computer und Internet berufliche und soziale Integrationsprozesse gefördert werden können, und auch insbesondere bezüglich der Bedienung von Equipment der Rehabilitationstechnik, beispielsweise bezüglich der Sprachsteuerung von „Funktionellen Elektrostimulations-systemen“ oder „Orthesen“.

2.2.7 Hilfestellung bei kognitiven (geistigen) Störungen

Kognitive Hilfestellungen durch Spracherkenner sind prinzipiell durch die Kompensation anderer Störungen möglich, wodurch die Aufmerksamkeit auf die eigentliche Aufgabe fokussiert werden kann. Die Verwendung von Spracherkennern durch Personen mit kognitiven Störungen ist ein relativ neues Anwendungsgebiet. In einer Studie wird untersucht, inwieweit Spracherkenner Menschen mit kognitiven Störungen unterstützen können. Das Ziel liegt darin, einen höheren Schulabschluss zu erreichen. Es wird allgemein für Menschen mit Behinderungen über eine hohe Korrelation zwischen der Höhe des erreichten Schulabschlusses und der Wahrscheinlichkeit, ein Beschäftigungsverhältnis einzugehen, berichtet [Roberts2002].

Diktiersysteme haben prinzipiell das Potenzial, Personen mit kognitiven Störungen, sofern sie auch von Dyslexie beziehungsweise Dysgraphie betroffen sind, die Texterstellung zu erleichtern, indem die kognitive Last der Rechtschreibfehlerkorrektur sowie des motorischen Schreibvorgangs und des Lesevorgangs verringert wird, wobei das Verstehen des Inhalts in den Vordergrund rückt. Die Anforderungen der Systeme an Wahrnehmung, Intelligenz, Konzentration und Gedächtnis lassen den Einsatz bei schwereren kognitiven Störungen in der klassischen Produktform jedoch fraglich erscheinen. Für Personen mit leichten kognitiven Störungen beruhen die therapeutischen Hoffnungen auf der Erweiterung der Möglichkeiten des selbstständigen Lernens und auf der Unterstützung von beruflichen und sozialen Integrationsprozessen.

Wegen der hohen kognitiven Anforderungen, die notwendig sind, um ein Diktiersystem effizient und präzise zu benutzen, wird allerdings empfohlen, allgemeine Fortschritte des Stands der Technik abzuwarten, bevor Diktiersysteme für Personen mit kognitiven Störungen zur Texterstellung empfohlen werden können [Wetzel1996].

2.3 Abgeleitete Anforderungen an die Sprachsteuerung

Als Schlussfolgerungen aus Abschnitt 2.2 werden Anforderungen an das Systemdesign festgelegt: Im Sinne des „Design-for-all-Prinzips“ soll eine möglichst große Benutzerzielgruppe angesprochen werden. Diesbezügliche Systemdesignvorgaben werden in Abschnitt 2.3.1 angeführt. Systemdesignvorgaben, welche die benutzerspezifische Konfigurierbarkeit betreffen werden im Abschnitt 2.3.2 auf Seite 142 angeführt. Systemdesignvorgaben, welche die Interaktion mit Betreuungspersonen und Entwicklern betreffen, werden in Abschnitt 2.3.3 auf Seite 143 angeführt.

2.3.1 Anforderungen bezüglich des Design-for-all-Prinzips

2.3.1.1 Redundanz der Eingabegeräte

Es soll eine Sprachsteuerung entwickelt werden, die in erster Linie als Eingabegerät für das AUTONOM-System dienen soll. Siehe auch Abschnitt 3.3 auf Seite 156. Es stehen den Benutzern freilich auch die weiteren, speziell für das AUTONOM-System entwickelten, Eingabegeräte zusätzlich zur Verfügung. Darunter befinden sich teilweise sehr benutzerspezifische Eingabegeräte, wie etwa Neigungssensorschalter am Brillengestell, Saug-Blas-Schalter, Spezialtaster, Augenbewegungssensorschalter, und viele andere mehr.

2.3.1.2 Keine Notwendigkeit motorischer Eingaben

Um den Bedürfnissen von Menschen mit motorischen Beeinträchtigungen gerecht zu werden, sollen stimmliche Äußerungen zur Erreichung des Ziels, das System bedienen zu können, hinreichend sein. Es sollen also insbesondere keine Tastendrücke auf der Computertastatur oder andere motorische Eingaben notwendig sein.

2.3.1.3 Audiovisuelle Rückmeldungen

Abbildung 2-7: Beim implementierten Benutzerdialog werden die Sprachausgaben zusätzlich gut lesbar eingeblendet.

Um den Bedürfnissen von Menschen mit auditiven Beeinträchtigungen oder visuellen Beeinträchtigungen oder Dyslexie (Leseschwäche) gerecht zu werden, sollen Sprachausgaben stets zusätzlich in gut lesbarer Form eingeblendet werden.

2.3.1.4 Tragbarkeit und Verfügbarkeit

Sprachsteuerungen für Menschen mit Behinderungen sind im Idealfall rund um die Uhr und an möglichst jedem Ort einsatzbereit[87]. Zum Beispiel könnte das Absetzen von Notrufen auch während Stromausfällen erforderlich sein. Die Implementierung soll für Personal Computer unter „Microsoft Windows“ erfolgen. Die Zukunft dieser Plattform scheint vom derzeitigen Standpunkt aus langfristig gesichert und es sind auch bereits mehrere Miniaturnotebooks am Markt, welche die gefragte Portabilität in hohem Maße

[87] Wenn Tragbarkeit gefragt ist, so ist letztlich auch der Stromaufnahme des Systems eine bestimmte Grenze gesetzt, was den Einsatz wenig rechenintensiver Algorithmen notwendig machen kann.

bereitstellen können. Batterien, welche naturgemäß regelmäßig gewechselt werden müssen, sollen nicht eingesetzt werden. Insbesondere sollen USB-Mikrofone verwendet werden, die vom Personal Computer aus versorgt werden.

2.3.1.5 Qualität der Sprachsteuerung

Die zuverlässige Erkennung von Steuerbefehlen ist unter allen Umständen der Benutzung gefragt. Ungewolltes Auslösen von Aktionen soll unbedingt vermieden werden. Im speziellen ist Robustheit bezüglich Hintergrundgeräuschen, Musik, Gesprächen und Atemgeräuschen notwendig. Das gewollte Auslösen von Aktionen soll ohne Probleme möglich sein. Ausreichende Hilfsmittelakzeptanz kann im Allgemeinen nur mit qualitativ hochwertiger und stabiler Funktion erreicht werden. Es empfiehlt sich daher die Optimierung der in Abschnitt 1.3 auf Seite 61 definierten Fehlerraten.

2.3.1.6 Robustheit durch Qualitätssicherungsanalyse

Die Aufnahme von Schallmustern der Steuerbefehle eines neuen Benutzers soll einfach und dialoggeführt erfolgen. Die freie Wahl der Steuerbefehle wirft ein Problem bezüglich der Robustheit auf, da der Benutzer beabsichtigt oder unbeabsichtigt beliebig ähnliche Schallmuster für unterschiedliche Steuerbefehle aufnehmen kann. Dies soll durch eine automatische Qualitätssicherungsanalyse zuverlässig verhindert werden: Die aufgenommenen Schallmuster müssen bestimmte Bedingungen bezüglich geschätzter Verwechslungsfehlerrate, Falschrückweisungsrate und Falschakzeptanzrate erfüllen, bevor die Benutzung freigeschaltet wird.

2.3.1.7 Sprechtraining durch Bildschirmdarstellung der Benutzerstimme

Bei der Benutzung einer Sprachsteuerung ist Sprechdisziplin bis zu einem gewissen Grad erforderlich. Die Steuerbefehle müssen beispielsweise jedes Mal relativ ähnlich klingend ausgesprochen werden. Dieser Umstand bietet umgekehrt wiederum eine potenzielle Chance auf Steigerung der Selbstkontrolle und Sprechdisziplin. Um die Chance optimal bereitzustellen, und auch um die Hilfsmittelakzeptanz zu erhöhen, soll eine Möglichkeit der Bildschirmdarstellung der Benutzerstimme implementiert werden, wobei das Schallsignal in einer rollenden Zeit-Frequenz-Darstellung, der sogenannten „Visual-Speech-Darstellung“ sichtbar gemacht werden soll. Das Einblenden des Verlaufs der Momentanlautstärke, sowie einer Frequenzskala soll Übungen ermöglichen.

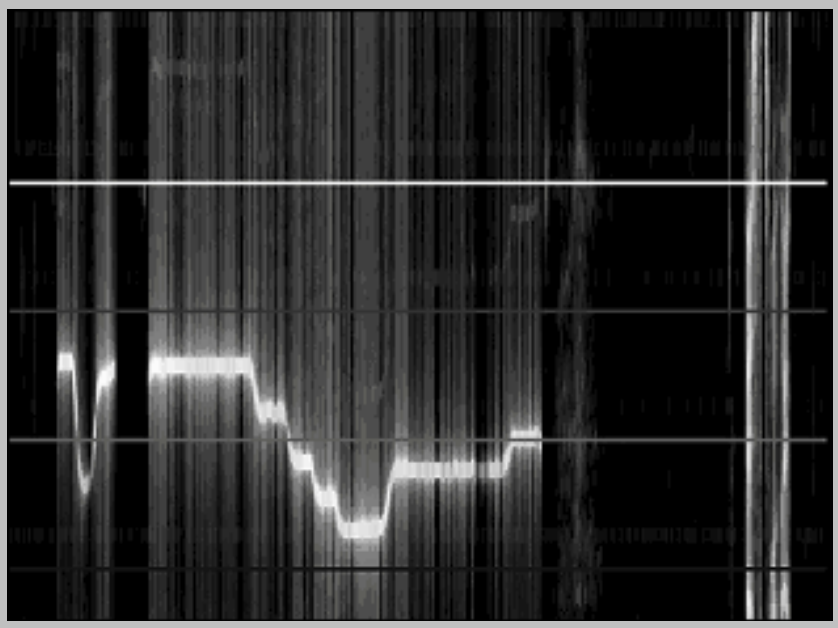

Abbildung 2-8: Die Visual-Speech-Darstellung beim Pfeifen einer Melodie. Die horizontalen Linien markieren 1 kHz, 2 kHz, 3 kHz und 4 kHz. Der Verlauf der Momentanlautstärke ist nicht eingeblendet.

2.3.2 Anforderungen bezüglich benutzerspezifischer Konfigurierbarkeit

2.3.2.1 Konfigurierbarkeit der Dialogabläufe

Die Sprachdialoge sollen einfach auf die Bedürfnisse der Anwender und auf die Steuerung anderer Anwendungen neben dem AUTONOM-System angepasst werden können. Die Sprachausgaben sollen konfigurierbar sein. Die zusätzlich zu den Sprachausgaben eingeblendeten Texte, die im Normalfall den Sprachausgaben entsprechen, sollen benutzerspezifisch konfigurierbar sein.

2.3.2.2 Konfigurierbarkeit der Steuerbefehle

Eine Anpassung an die vorhandenen Sprechmöglichkeiten des Benutzers ist in der Regel gefragt, daher soll eine sprecherabhängige Sprachsteuerung entwickelt werden, wobei die Steuerbefehle individuell und frei wählbar sein sollen. Die Änderung des Wortschatzes beziehungsweise das Ergänzen des Wortschatzes um weitere Steuerbefehle soll einfach durchführbar sein und soll nicht zu langen Trainingszeiten führen, um die Hilfsmittelakzeptanz nicht unnötig zu strapazieren.

2.3.3 Anforderungen bezüglich Interaktion mit Betreuungspersonen und Entwicklern

2.3.3.1 Schnittstelle für Betreuungspersonen

Betreuungspersonen sollen in die Lage versetzt werden, effizient in den Mensch-Maschine-Dialog eingreifen zu können. Konkret soll eine Pausetaste vorgesehen sein, mittels der Betreuungspersonen den Dialog anhalten können, um Erklärungen durchzuführen. Die Betreuungspersonen sollen jederzeit die Möglichkeit haben, auf einen der anderen vorbereiteten Mensch-Maschine-Dialoge umzuschalten, wenn das als sinnvoll erachtet wird.

2.3.3.2 Mikrofonaustausch ohne erneute Stimmprobenaufnahme

Amplitudengangkalibrierungsdaten der Schallaufnahmesysteme, insbesondere USB-Headsets in Verbindung mit Betriebssystemtreibern, sollen in einer Datenbank gespeichert und bei Bedarf automatisch geladen werden. Bei analoger Signaleinspeisung in den Personal Computer kann das verwendete Mikrofon beziehungsweise der verwendete Mikrofonvorverstärker von der Applikation nicht automatisch erkannt werden. Eine zuverlässig richtige Konfiguration durch den Benutzer beziehungsweise die Betreuungsperson ist notwendig und soll ermöglicht werden. Der verwendete Betriebssystemtreiber soll dennoch automatisch identifiziert werden und in Verbindung mit dem manuell konfigurierten Mikrofon soll das Laden der Amplitudengangkompensationsdaten aus der Datenbank ermöglicht werden.

2.3.3.3 Evaluierungsdialog zur Rückmeldung erlebter Fehlerraten

Die vom Benutzer erlebten Fehlerraten „Verwechslungsfehlerrate“, „Falschrückweisungsrate“ und „Falschakzeptanzrate“, können leider vom Prinzip her nicht allein aus den üblichen Systemprotokollen ermitteln werden. Das liegt daran, dass einer Sprachsteuerung ja in der Regel keine direkt protokollierbare Information darüber vorliegt, ob eine vergangene Systemreaktion korrekt war, ob sie überhaupt hätte erfolgen sollen oder ob sie nicht erfolgt ist, obwohl sie hätte erfolgen sollen.

Menschlicher Beobachter. Eine Möglichkeit der Ermittlung der Fehlerraten bestünde darin, einen menschlichen Beobachter einzusetzen, der die „Verwechslungsfehlerrate“ und die „Falschrückweisungsrate“, welche ja auf die Anzahl der dem System präsentierten Keyword-Schallmuster bezogen sind, mitprotokolliert. Diese Vorgangsweise würde allerdings einige Schwierigkeiten bezüglich der „Falschakzeptanzrate“ mit sich

bringen: Falschakzeptanzfehler, die auf Hintergrundgeräusche zurückzuführen sind, sind relativ schwierig zu protokollieren, weil nicht offensichtlich ist, wie die Hintergrundgeräusche jeweils zu zählen sind. Eine naheliegende Definition bestünde darin, die Anzahl jener Hintergrundgeräusche und vokabularfremden Wörter heranzuziehen, die vom Pattern-Builder-Funktionsblock in Abbildung 1-1 auf Seite 16 als Schallmuster ausgegeben werden. Dazu müsste für den menschlichen Beobachter allerdings ein entsprechendes Signal aus der Sprachsteuerung herausgeführt werden, das anzeigt, wenn der Pattern-Builder-Funktionsblock ein Schallmuster erzeugt. Es würde dabei weiters ein perfekt klassifizierender menschlicher Beobachter vorausgesetzt. Ein weiterer Nachteil bestünde darin, dass die nachträgliche Auszählung relativ aufwändig wäre und die Beobachtungsperson über einen längeren, repräsentativen Zeitraum zur Verfügung stehen müsste.

Evaluierungsdialog. Daher sollen die Benutzer der Sprachsteuerung selbst zur Evaluierung herangezogen werden: Die Sprachsteuerung soll unter Ausführung eines Evaluierungsdialogs gelegentlich zurückfragen, ob sie richtig reagiert hat. Automatische Systemprotokolle sollen dabei entstehen, sodass die Fehlerraten nachträglich bestimmt werden können. Die Häufigkeit der Systemrückfragen soll benutzerindividuell konfiguriert werden können. Es soll darauf geachtet werden, dass die kognitiven Anforderungen die Möglichkeiten der Benutzer nicht übersteigen. Daher sollen klare und einfache Mensch-Maschine-Dialoge realisiert werden. Eine gewisse Kooperationsbereitschaft der Benutzer ist notwendig. Um die kognitive Last zu senken, ist es sinnvoll, das aufgenommene Schallereignis vor der Rückfrage wiederzugeben. Dabei ist jedoch die Einschränkung zu bedenken, dass Personen mit auditiven Beeinträchtigungen die Wiedergabe unter Umständen nicht ausreichend wahrnehmen können. Es soll bei den Antworten der Benutzer eine Timeout-Funktionalität implementiert werden, um den Dialog normal weiterlaufen zu lassen, wenn der Benutzer auf die Systemrückfrage nicht antwortet. Bei einer Sprachsteuerung mit individuell aufnehmbaren Steuerbefehlen ist es natürlich wichtig, dass das System stets korrekt darüber informiert ist, welcher Benutzer es gerade verwendet[88]. Die Fehlerraten würden bei einem unerwarteten Benutzer im Allgemeinen drastisch ansteigen und die, aus den Systemprotokollen des Evaluierungsdialogs bestimmten Fehlerraten, wären für den erwarteten Benutzer verfälscht. Um dieses Problem zu vermeiden, soll der aktuelle Benutzer stets deutlich lesbar angezeigt werden und die Ansprache der Benutzer mit Namen vorgesehen werden. Es verbleiben jedoch folgende Schwierigkeiten: Werden die Antworten des Benutzers auf die Rückfragen von der Sprachsteuerung korrekt klassifiziert? Antworten die menschlichen Benutzer stets korrekt?

[88] Ideal wäre es natürlich, wenn der Benutzer vom System eindeutig erkannt werden könnte. Dies könnte beispielsweise durch berührungslose Identifikation mittels eines „Schlüssels“ erfolgen, den der Benutzer bei sich trägt.

3. Systemdesign und Implementierung

Schallaufnahmesystem

Qualitativ hochwertige Schallaufnahmehardware ist sehr wichtig, denn ein qualitativ unzureichendes Sprachsignal kann mittels Methoden der digitalen Signalverarbeitung nur innerhalb gewisser Grenzen beziehungsweise unter gewissen Annahmen über die Ausprägung der Störung aufgebessert werden. Batteriegespeiste Mikrofonverstärker sind zu vermeiden, um Einflüsse wechselnder Batteriespannung auf die Übertragungseigenschaften sowie den Zustand „leere Batterie“, als in der Praxis bedeutende Fehlerquelle, zu vermeiden. Bei der Konstruktion netzgespeister Mikrofonverstärker ist auf hohe Dämpfung des Netzbrumms zu achten.

Spracherkennermodul

Die entwickelte Applikation bietet neben der eigentlichen Sprachsteuerungsfunktionalität auch die Sprachdatenbanktestfunktionalität zum Zwecke der Evaluierung und laufenden Verbesserung des Spracherkennermoduls direkt auf der Zielplattform. Dadurch ist keine abschließende Portierung auf eine Zielplattform notwendig, und es wird somit diese in der Praxis bedeutende Fehlerquelle ausgeschaltet. Sämtliche sprachverarbeitende Algorithmen sind im Spracherkennermodul zusammengefasst und mittels einer „Dynamic Link Library“ realisiert, welche modular ausgetauscht werden kann. Dadurch sind für zukünftige Forschungsbemühungen Fehlerratenvergleiche unterschiedlicher Spracherkennermodule leicht machbar.

AUTONOM-Schnittstellenmodul

In diesem Abschnitt wird die Schnittstelle zum AUTONOM-System beschrieben, welche möglichst allgemein ausgelegt ist, um auch die Sprachsteuerung einer Vielzahl anderer Applikationen zu ermöglichen: Den gesteuerten Applikation werden Tastendrücke auf der Computertastatur vorgegaukelt, indem „Keyboard Events“ an das Betriebssystem gemeldet werden.

Dialogablaufmodul

Der Ablauf der Benutzerdialoge ist nicht hart codiert, sondern wird mittels ASCII-Dateien in einer eigens definierten State-Machine-Bescheibungssprache konfiguriert und kann damit leicht an bestimmte Benutzerbedürfnisse angepasst werden.

3.1 Schallaufnahmesystem

Beim Systemdesign von Sprachsteuerungen treten eine Reihe kritischer Fragestellungen auf. Einige der üblichen Stolpersteine liegen wohl in der Konzeption der Schallaufnahmehardware. Mit dem Ziel, eine möglichst robuste Sprachsteuerung zu erreichen, sollten möglichst wenige Kompromisse gemacht werden.

Bezeichnung	Hersteller	Empfehlenswerter Mundabstand	Bauform	Anschluss
Q400Mk3T	AKG	circa 80 cm	Mikrofon zum Festkleben an ebenen Flächen	Vorverstärker nach Abbildung 3-1 am Line-In-Eingang
Audio.10	Plantronics	circa 80 cm	Schwanenhalsmikrofon	Vorverstärker nach Abbildung 3-1 am Line-In-Eingang
AK5370	Logitech	circa 80 cm	Schwanenhalsmikrofon	USB-Schnittstelle
HSC-150	AKG	circa 7 cm	Am Ohr befestigtes Mikrofon	Vorverstärker nach Abbildung 3-1 am Line-In-Eingang
DSP-100	Plantronics	circa 7 cm	Kopfbügelmikrofon	USB-Schnittstelle
PC166-USB	Sennheiser	circa 7 cm	Kopfbügel-Headset	USB-Schnittstelle

Tabelle 6: Einige empfehlenswerte Mikrofone zum Anschluss an Personal Computer, die im Rahmen dieser Arbeit Verwendung finden.

In Tabelle 6 sind einige empfehlenswerte Mikrofone zum Anschluss an Personal Computer angeführt, die im Rahmen dieser Arbeit Verwendung finden. Für die Benutzung der im Rahmen dieser Arbeit implementierten Sprachsteuerung wird zum jetzigen Zeitpunkt ausschließlich das Kopfbügel-Headset PC-166USB von Sennheiser empfohlen.

3.1.1 Qualitätssichernde Überlegungen

Im Folgenden seien die wichtigsten prinzipiellen Stolpersteine kurz erläutert. Gelegentlich hört man das Argument, dass einige der folgenden Stolpersteine auch per Software, zum Beispiel mittels Kanalkompensationsalgorithmen und Rauschunterdrückungsalgorithmen überwunden oder zumindest teilweise überwunden werden können. Dies trifft zwar grundsätzlich zu, doch soll man wohl im Sinne eines robusten Systems von vornherein versuchen, Störeinflüsse jeglicher Art schon durch das Systemdesign zu vermeiden.

Zeitvariante Übertragungseigenschaften. Zeitvariante Eigenschaften des Übertragungskanals können die erlebten Fehlerraten negativ beeinflussen. Das Absinken der Batterieversorgungsspannung eines Mikrofonvorverstärkers kann dessen Übertragungseigenschaften unter Umständen signifikant beeinflussen.

Elektromagnetische Einstreuungen. Elektromagnetische Einstreuungen können zu schlechtem Signal-Rausch-Verhältnis führen, das sich wiederum ungünstig auf die erlebten Fehlerraten auswirken kann.

Exemplarstreuung der Übertragungseigenschaften. Die Exemplarstreuung der Übertragungseigenschaften der Schallaufnahmehardware[89] ist vor allem dann problematisch, wenn vortrainierte Modelle verwendet werden. Vortrainierte Modelle kommen etwa bei sprecherunabhängiger Erkennung zum Einsatz. Bei sprecherabhängiger Erkennung kommen sie zum Beispiel im Rahmen von Non-Keyword-Modellen[90] zum Einsatz. Siehe dazu Abschnitt 1.3.3 auf Seite 76.

Suboptimale Auswahl der Mikrofonbauform. Die Auswahl einer für den Benutzer ungünstigen Mikrofonbauform kann einerseits direkt dazu führen, dass die Hilfsmittelakzeptanz sinkt. Andererseits können dadurch die Fehlerraten ansteigen, weil das Mikrofon dann eventuell nicht wie vorgesehen benutzt wird. Beispielsweise können starke Hintergrundgeräusche und Pegelschwankungen auftreten, wenn Kopfbügelmikrofone in der Hand gehalten werden, weil das Aufsetzen zu unbequem ist. Auch andere Mikrofonbauformen, wie beispielsweise Schwanenhalsmikrofone, sollten nicht in der Hand gehalten werden. Ist das Aufsetzen von Kopfbügelmikrofonen aufgrund körperlicher Beeinträchtigungen nicht möglich, beziehungsweise ist dazu Hilfe erforderlich, so wirkt sich dieser Umstand oftmals negativ auf die Hilfsmittelakzeptanz aus.

Suboptimale Positionierung des Mikrofons. Das Mikrofon sollte möglichst weit von sämtlichen Störschallquellen entfernt positioniert werden. Weiters ist auf gute Schalldämmung und Vibrationsdämpfung der Befestigung zu achten. Beispielsweise ist bei der Mikrofonaufstellung auf einem Tisch des Öfteren zu beobachten, dass sich Personen direkt neben dem Mikrofon aufstützen, was zu starken Störgeräuschen führt. Außerdem kann sich unter Umständen Trittschall vom Boden über den Tisch auf das Mikrofon übertragen.

Suboptimale Benutzung des Mikrofons. Beim Aufsetzen von Kopfbügelmikrofonen ist zu beachten, dass die Atemluft beim Ausatmen durch den Mund oder durch die Nase oftmals zu starken Störgeräuschen führt. Es ist deshalb meist ratsam, wenn möglich, den Mikrofonkapselbügel in eine Position oberhalb der Nasenlöcher zu biegen. Die Position der Mikrofonkapsel sollte bei der Verwendung der Sprachsteuerung grundsätzlich der Position bei der Aufnahme der Steuerbefehle entsprechen.

---

[89] Bei Sprachsteuerungen für Personal Computer kommen Exemplarstreuungen einerseits durch unterschiedliche Soundkarten (oder auch am Mainboard integrierte Soundsysteme) beziehungsweise durch unterschiedliche externe Mikrofonvorverstärker und Mikrofone und andererseits durch unterschiedliche Software-Treiber der Betriebssysteme zustande. Einen Fortschritt in dieser Richtung stellen die mittlerweile etablierten USB-Mikrofone dar. Low-Cost-Spracherkenner für Personal Computer werden gelegentlich ohne Mikrofon ausgeliefert, wobei es dem Benutzer obliegt, ein qualitativ hochwertiges Signal bereitzustellen. Diese Vorgangsweise führt jedoch oftmals zu enttäuschten Erwartungen des Benutzers.

[90] Non-Keyword-Modelle dienen beispielsweise zur Reduzierung der Falschakzeptanzrate.

3.1.2 Anschluss analoger Mikrofone an Personal Computer

Derzeit ist es zwar üblich, dass Personal Computer Mikrofonvorverstärker im Rahmen eines Mikrofoneingangs als Teil der Audiohardware enthalten[91], jedoch sind die damit erreichbaren Signalqualitäten höchst unterschiedlich: Die einzelnen Soundsysteme unterscheiden sich neben der Signaldynamik auch bezüglich des Frequenzgangs, der Eingangsimpedanz sowie der Mikrofonversorgungsspannung. Weiters bieten die zugehörigen Betriebssystemtreiber verschiedene, unterschiedlich konfigurierbare Funktionselemente, wie beispielsweise Elemente mit Automatic-Gain-Control-Funktionalität oder Elemente mit Filterungsfunktionalität, an. Eine automatische Deaktivierung dieser Funktionselemente durch die Sprachsteuerungssoftware, die ja für möglichst viele verschiedene Personal Computer, und somit auch für möglichst viele verschiedene Betriebssystemtreiber, funktionieren soll, ist kaum in den Griff zu bekommen. Weniger konfigurierbare Funktionselemente bieten Betriebssystemtreiber im Allgemeinen für sogenannte „Line-In-Eingänge“, daher werden eben diese im Rahmen dieser Arbeit den Mikrofoneingängen bevorzugt. Das heißt, es werden beim Anschluss analoger Mikrofone grundsätzlich externe Mikrofonvorverstärker verwendet.

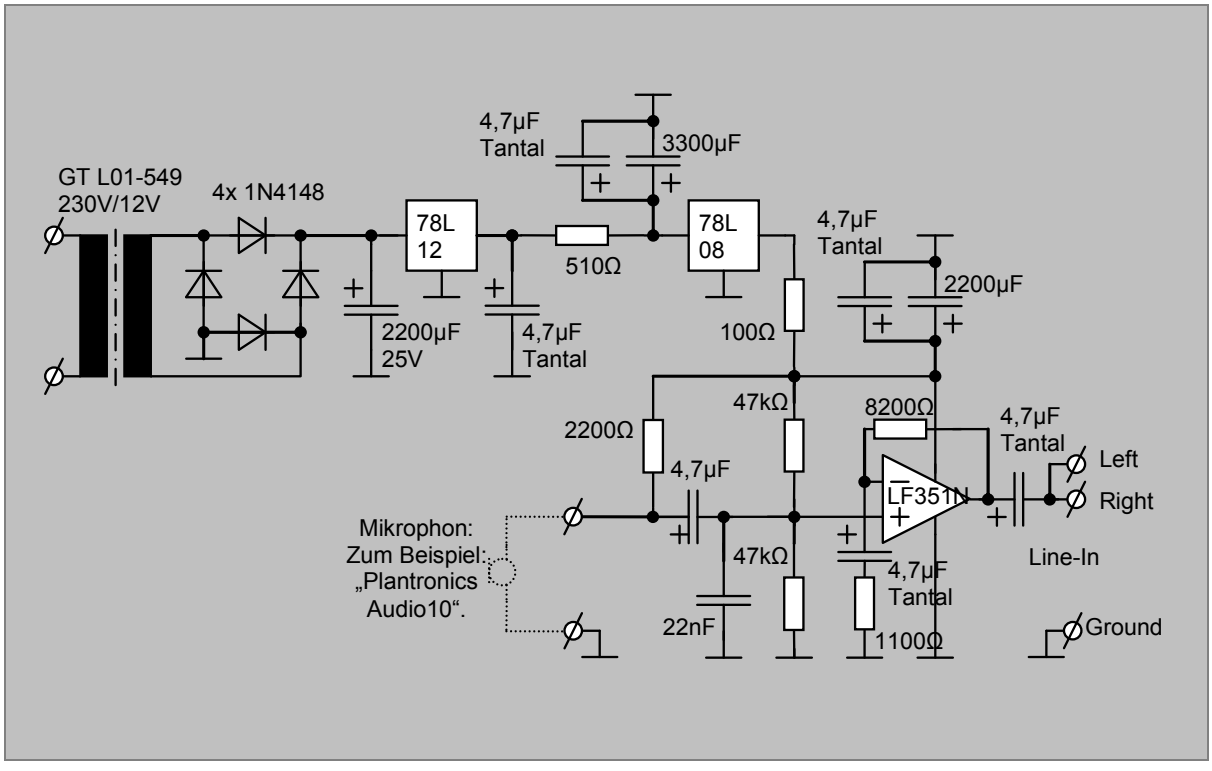

Abbildung 3-1: Netzgespeister Mikrofonvorverstärker mit hoher Dynamik zum Anschluss an Line-In-Eingänge der Soundkarten für Personal Computer.

[91] Unter Umständen müssen sie eingeschaltet werden, wobei gelegentlich spezielle Software notwendig ist.

Abbildung 3-1 zeigt die entwickelte Schaltung eines netzgespeisten Mikrofonverstärkers zum Anschluss an Line-In-Eingänge der Soundkarten für Personal Computer. Das Besondere an der Schaltung ist die hohe Unterdrückung des Netzbrumms, ohne dabei Induktivitäten zu verwenden. Es kommt ein ohmscher Widerstand in der Versorgungszuleitung zum Einsatz, welcher die Glättungszeitkonstante erhöht. Der dabei auftretende Spannungsabfall ist bezüglich der Versorgungsspannung der Operationsverstärkerschaltung einkalkuliert.

Beim Aufbau ist auf eine gewisse räumliche Trennung der Operationsverstärkerschaltung vom Netztrafo zu achten, um Einstreuungen zu vermeiden, wobei einige Zentimeter genügen. Bei weiterer Miniaturisierung wäre eine Schirmung notwendig. Weiters ist darauf zu achten, dass das zuleitende Netzkabel räumlich getrennt vom Audiokabel am Line-In-Eingang geführt wird, um elektromagnetische Einstreuungen zu verhindern. Diesbezügliche Überlegungen sind wichtig, weil sich Dynamikeinschränkungen fatal auf die Fehlerraten der Sprachsteuerung auswirken können.

Der Mikrofonvorverstärker nach Abbildung 3-1 bietet eine niedrige Ausgangsimpedanz und daher eine gute Unabhängigkeit von der Eingangsimpedanz der Soundkarte. Die Eingangsimpedanz der Operationsverstärkerschaltung ist so hochohmig dimensioniert, dass der Spannungsteiler, bestehend aus dem Pull-up-Widerstand zur Versorgung des Mikrofons und der Ausgangsimpedanz des Mikrofons, damit kaum belastet wird.

3.1.3 Anschluss von USB-Mikrofonen an Personal Computer

Während des Bearbeitungszeitraums dieser Arbeit sind Mikrofone mit USB-Schnittstelle am Consumer-Electronic-Markt eingeführt worden und haben sich schnell als Standard für qualitativ hochwertige Spracherkenner etabliert.

Entscheidende Vorteile von USB-Mikrofonen sind: Das Modell des angeschlossenen Mikrofons kann per Software mittels einer USB-Gerätekennung identifiziert werden. Solange die USB-Gerätekennung bei Änderungen der Übertragungseigenschaften der USB-Mikrofone vom Hersteller verändert wird und die Versionskennung des Betriebssystemtreibers bei Änderungen der Übertragungseigenschaften des Betriebssystemtreibers vom Hersteller verändert wird, ist es möglich, Amplitudengangkalibrierungsdaten automatisch aus einer Datenbank zu laden. Dieser Vorteil wird bei der im Rahmen dieser Arbeit implementierten Sprachsteuerung für jene USB-Mikrofone umgesetzt, die in Tabelle 6 angeführt sind[92]. Der zweite entscheidende Vorteil von USB-Mikrofonen besteht darin, dass keine Soundkarten verwendet werden, die in den Personal Computer eingebaut sind. Bei der Verwendung von USB-Mikrofonen ist im Prinzip ein Benutzerwechsel von einem Personal Computer zum nächsten, ohne erneute Aufnahme der Steuerbefehle, möglich[93].

[92] Während des Bearbeitungszeitraums dieser Arbeit wurde lange Zeit, für die Benutzung der im Rahmen dieser Arbeit implementierten Sprachsteuerung, das USB-Mikrofon „DSP-100“ von Plantronics empfohlen. Leider begab es sich, dass bei einem von Plantronics herausgegebenen Update der Betriebssystemtreiber die Übertragungseigenschaften geändert wurden, ohne die Versionskennung des Betriebssystemtreibers zu modifizieren. Dieses Mikrofon kann daher leider nicht mehr zur Verwendung mit der Sprachsteuerung empfohlen werden, weil die Amplitudengangkompensationsdaten nun nicht mehr eindeutig aus der Datenbank geladen werden können.

[93] Im Prinzip ist das Laden der Ampitudengangkompensationsdaten aus der Datenbank auch bei analogen Mikrofonen möglich, allerdings ist diese Vorgangsweise mit Unsicherheiten verbunden, da das angeschlossene Mikrofon der Applikation stets korrekt, manuell mitgeteilt werden muss.

3.1.4 Kalibrierung des Schallaufnahmesystems

3.1.4.1 Gesamtamplitudengangkalibrierung bei USB-Mikrofonen

Messung mit dem Schallpegelmessgerät. Mittels eines Schallpegelmessgerätes lässt sich der Gesamtamplitudengang „Effektivwert der Zahlenwerte der Samples, die die Applikation erhält, bezogen auf den Effektivwert des Schalldrucks an der Mikrofonkapsel in Abhängigkeit der Frequenz“ bestimmen.

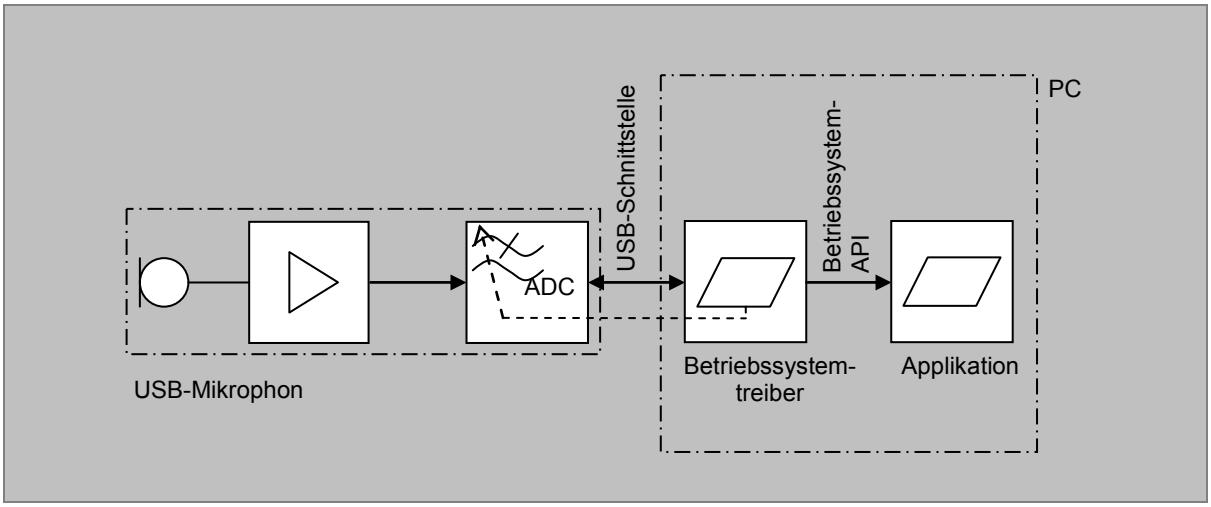

Abbildung 3-2: Signalverarbeitungskette bei der Schallaufnahme mittels eines USB-Mikrofons, das an den Personal Computer direkt über eine USB-Schnittstelle angeschlossen wird.

Abbildung 3-2 zeigt das Blockschaltbild der Signalverarbeitungskette beim Anschluss eines USB-Mikrofons an einen Personal Computer. Der Vorteil liegt auf der Hand: Der Gesamtamplitudengang ist dabei nicht mehr von der im Personal Computer vorhandenen Soundkarte und einem eventuellen Vorverstärker abhängig, sondern kann aus der USB-Gerätekennung normalerweise eindeutig bestimmt werden. Die USB-Gerätekennung steht der Applikation relativ einfach zur Verfügung. Dennoch ist bei der Programmierung der Applikation darauf zu achten, dass im Allgemeinen auch das Betriebssystem und der Betriebssystemtreiber zusätzliche Funktionselemente mit relevanten Übertragungseigenschaften einbringen, welche teilweise konfigurierbar sind. Dazu gehören Elemente mit relevanten Amplitudengängen oder auch Elemente mit Automatic-Gain-Control-Funktionalität. Diese zusätzlichen Übertragungseigenschaften müssen von der Applikation unbedingt deaktiviert oder kompensiert werden, was oftmals einen nicht unbeträchtlichen programmtechnischen Aufwand nach sich zieht. Eine diesbezüglich ideale Lösung ist durch eine Zusammenarbeit mit dem Hersteller des USB-Mikrofons, der oftmals auch den Betriebssystemtreiber bereitstellt, erreichbar. Ein ungünstiger Fall tritt ein, wenn der Hersteller des USB-Mikrofons die Übertragungseigen-

schaften der Serie ändert, ohne die USB-Gerätekennung beziehungsweise gegebenenfalls die Versionskennung des Betriebssystemtreibers zu ändern, wobei bezüglich der Datenbank der Ampitudengangkompensationsdaten eine Mehrdeutigkeit entsteht und der korrekten Amplitudengangkompensationsdaten nicht mehr automatisch aus der Datenbank geladen werden können.

3.1.4.2 Gesamtamplitudengangkalibrierung bei analogen Mikrofonen

Der Gesamtamplitudengang zerfällt in die Teilamplitudengänge, die dem Mikrofon, dem Vorverstärker, der Soundkarte und dem Betriebssystemtreiber zuzuschreiben sind.

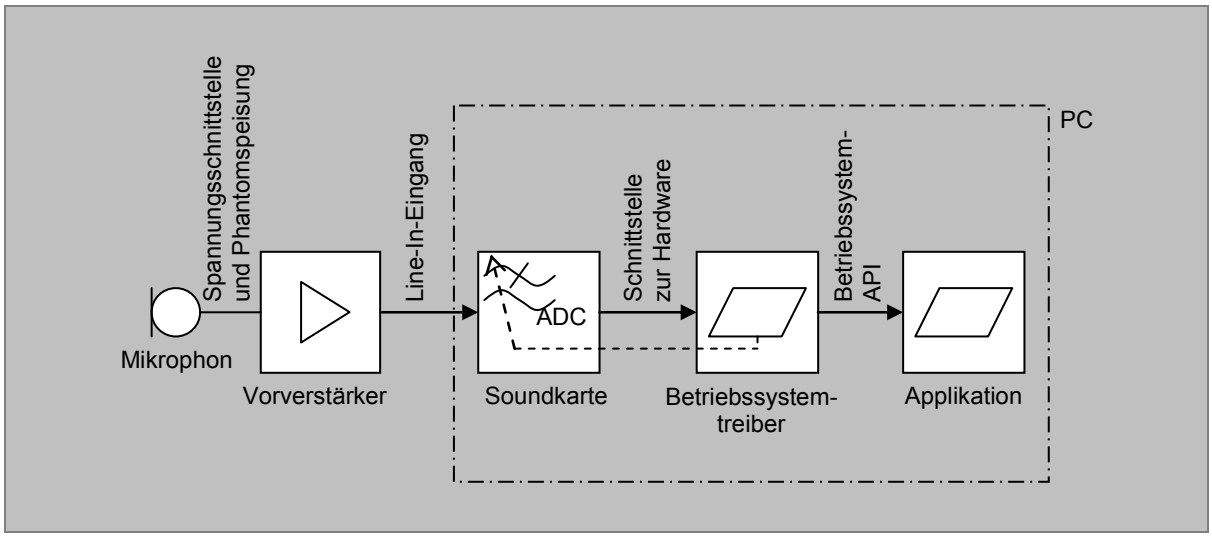

Abbildung 3-3: Signalverarbeitungskette bei der Schallaufnahme mittels eines phantomgespeisten Mikrofons, das über einen externen Vorverstärker an einen Personal Computer angeschlossen wird, wobei der Line-In-Eingang verwendet wird.

Wenn der Amplitudengang der Übertragungsfunktion „Effektivwert der Mikrofonspannung bezogen auf den Effektivwert des Schalldrucks an der Mikrofonkapsel in Abhängigkeit der Frequenz“ für das jeweilige Mikrofon laut Datenblatt bekannt ist, kann eine rechnerische Kontrolle der mit dem Schallpegelmessgerät gemessenen Werte des Gesamtamplitudengangs durchgeführt werden. Dies ist deswegen möglich, weil der Amplitudengang „Effektivwert der Zahlenwerte der Samples, die die Applikation erhält, bezogen auf den Effektivwert der Spannung am Eingang des Vorverstärkers, in Abhängigkeit der Frequenz“ im Prinzip bekannt ist oder relativ einfach gemessen werden kann.

Es bietet sich, neben dem Zweck der Kontrolle der mit dem Schallpegelmessgerät gemessenen Werte an, die Teilamplitudengänge der Funktionsblöcke nach Abbildung 3-3 auf Seite 152 zu bestimmen, um die Austauschbarkeit der Funktionsblöcke zu ermöglichen.

Amplitudengang der Soundkarte inklusive Betriebssystemtreiber. Der Amplitudengang der Übertragungsfunktion „Effektivwert der Zahlenwerte der Samples, die die Applikation erhält, bezogen auf den Effektivwert der Spannung am Line-In-Eingang der Soundkarte“ wird vorteilhafterweise bestimmt. Dazu wird eine spezielle Messeinrichtung entworfen, um schnell und einfach die Amplitudengänge beliebiger Soundkarten inklusive Betriebssystemtreiber bestimmen zu können.

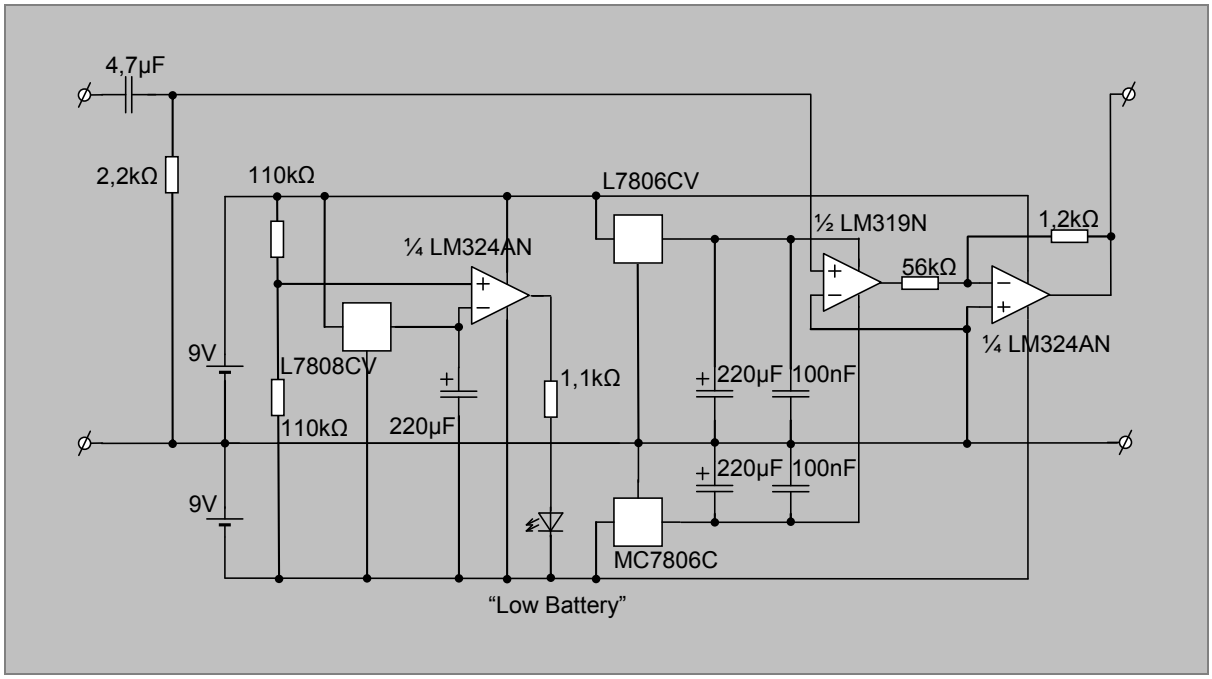

Abbildung 3-4: Soundkartenkalibrierung: Signalformung zur Amplitudengangmessung.

Die Schaltung dieser Messeinrichtung ist in Abbildung 3-4 dargestellt. Sie dient dazu, ein Signal, welches einer Sinusschwingung ähnlich ist und dessen Amplitude nur ungefähr bekannt ist, in ein möglichst exaktes Rechtecksignal mit sehr geringer und genau definierter Amplitude umzuformen, wobei die Grundfrequenz exakt erhalten bleibt.

Die Messapplikation gibt an einem Kanal des Line-Out-Ausgangs Sinustöne aus und nimmt das Rechtecksignal am zu messenden Line-In-Eingang wieder auf. Das aufgenommene Signal wird einer diskreten Kurzzeit-Fourier-Transformation unterworfen. Da die Amplitude des aufgenommenen Rechtecksignals exakt bekannt ist, ist auch die Amplitude dessen Grundwelle exakt bekannt und damit auch der Wert des Amplitudengangs bei dieser Frequenz. Die Frequenzen der ausgegebenen Sinustöne werden so gewählt, dass sie exakt auf den „Bins“ der diskreten Kurzzeit-Fourier-Transformation liegen. Dadurch werden unerwünschte Messfehler vermieden, deren Ursache in

der bei der Transformation verwendeten Fensterfunktion liegt. Dennoch empfiehlt es sich, eine speziell für Amplitudenmessungen optimierte Flat-Top-Fensterfunktion zu verwenden.

Es ist darauf zu achten, dass die Einstellungen des Betriebssystemtreibers inklusive sämtlicher Sampling-Einstellungen, wie zum Beispiel der Abtastrate, so wie im späteren Betrieb vorgenommen werden. Die meisten Soundkarten ändern nämlich ihren Amplitudengang zumindest insofern, dass je nach Abtastrate ein anderes Anti-Aliasing-Filter vorgeschaltet wird. Auch der Betriebssystemtreiber kann, je nach Einstellung, zusätzliche Übertragungseigenschaften einbringen. Zeitvariante Übertragungseigenschaften, wie beispielsweise eine Automatic-Gain-Control-Funktionalität, sind zu deaktivieren.

Amplitudengang des Vorverstärkers. Der Amplitudengang des Vorverstärkers ist im Allgemeinen bekannt. Es kommt für eine große Auswahl an Mikrofonen beispielsweise jener nach Abbildung 3-1 auf Seite 148 infrage.

Amplitudengang des Mikrofons. Der Amplitudengang des Mikrofons ist oftmals laut Datenblatt bekannt. Jedenfalls aber kann er rechnerisch bestimmt werden, wenn der Gesamtamplitudengang mittels eines Schallpegelmessgeräts gemessen wird und die anderen bekannten Teilamplitudengänge davon rechnerisch entfernt werden.

Berücksichtigung der Benutzungsgewohnheiten. Mit jedem der kalibrierten Schallaufnahmesysteme erhält man zwar einen ähnlichen und bekannten Gesamtamplitudengang. Das eigentliche Ziel besteht aber darin, den Austausch der Schallaufnahmesysteme zu ermöglichen. Dabei gilt es, den Umstand zu berücksichtigen, dass unterschiedliche Schallaufnahmesysteme aufgrund unterschiedlicher Mikrofonbauformen mit unterschiedlichen Benutzungsgewohnheiten vergesellschaftet sind: Je nach Mikrofonbauform sind unterschiedliche Mundabstände zu erwarten. Einerseits gilt es daher, die Übertragungsstrecke vom Mund zur Mikrofonkapsel zu berücksichtigen. Andererseits tendieren die Sprecher dazu, mit aufgesetztem Kopfbügelmikrofon leiser zu sprechen, als mit einem entfernten Mikrofon. Diese beiden Effekte können zum Beispiel empirisch untersucht werden und für jedes Schallaufnahmesystem mittels eines Korrekturwerts für den Gesamtampitudengang berücksichtigt werden.

3.2 Spracherkennermodul

Den mit Abstand größten Anteil am Arbeitsaufwand im Rahmen dieser Dissertation bringt die Entwicklung des Spracherkennermoduls mit sich. Es wird auf eine saubere Trennung des Spracherkennermoduls von der restlichen Anwendung geachtet, um dieses Modul – oder aber auch die restliche Anwendung mit anderen Spracherkennermodulen – zukünftig modular weiterentwickeln zu können:

Die im Rahmen dieser Arbeit geschaffene Anwendung, welche etwa dreißigtausend Zeilen Pascal-Quellcode umfasst, eignet sich im Prinzip auch sehr gut, um andere Spracherkennermodule unter gleichen, definierten Bedingungen zu testen. Einer der wesentlichen Vorteile der geschaffenen Anwendung besteht nämlich darin, dass sie sowohl die Funktionalität zur Durchführung von Sprachdatenbanktests als auch die Funktionalität zur Sprachsteuerung enthält. Bei Systementwicklungsprozessen, bei denen das Spracherkennermodul nach dem Sprachdatenbanktest noch auf eine andere Zielplattform, wie beispielsweise auf einen digitalen Signalprozessor, portiert werden muss, besteht die Gefahr, dass sich Fehler einschleichen, da im Allgemeinen keine „Bit-true-Portierung“ der komplizierten Algorithmen erfolgen kann, und daher vergleichendes Debugging nur teilweise möglich ist.

3.3 AUTONOM-Schnittstellenmodul

AUTONOM-System. Im Rahmen dieser Arbeit gelangt das AUTONOM-System zum Einsatz. Es handelt sich um ein Assistenzsystem zur Förderung der Selbstständigkeit für ältere Menschen und für Menschen mit Behinderungen, das an der Technischen Universität Wien von der „Forschungsgruppe Rehabilitationstechnik“ entwickelt wird [Flachberger1994] [Panek2002] [Panek1996] [Zagler1997]. Bereits 1995 wird auf Basis des „Neural Net Speech Recognizers“ [Tschirk1999] eine Sprachsteuerung für das AUTONOM-System implementiert [Loidolt1995], welche als Vorläufer für das im Rahmen dieser Arbeit entwickelte System gesehen werden kann. Das Grundpaket des AUTONOM-Systems ist in Abbildung 3-5 dargestellt[94].

Abbildung 3-5: Das Grundpaket des AUTONOM-Systems.

Um den deutlich verschiedenen Bedürfnissen der potenziellen Benutzer gerecht zu werden, ist hohe Flexibilität bezüglich der Konfigurierbarkeit der bereitgestellten Funktionalität und der Benutzerschnittstelle notwendig. Ein großes Sortiment von Peripheriegeräten sowie eine konfigurierbare Sprachausgabe lassen sich steuern. Ins System integriert ist eine sogenannte „Telehilfe“. Die Steuersignalübertragung erfolgt über Infrarot, per Trägerfrequenzverfahren über das 230V-Hausnetz, über die Telefonleitung, sowie über direkte Drahtverbindungen [Zagler1993]. Leicht können individuelle Einstellungen geändert werden, etwa ob beim Läuten des Telefons der Fernseher und die Stereoanlage leiser gedreht werden oder mit welchem der benutzerspezifischen

[94] Quelle: Produktinformation des AUTONOM-Systems http://www.fortec.tuwien.ac.at/autonom (Stand: 16. 11. 2005).

Eingabegeräte das Abheben des Telefons durch den Benutzer erfolgen soll. Durch die hochgradige Konfigurierbarkeit ergibt sich ein enorm weites Anwendungsgebiet. Ein benutzerkonfiguriertes Menü zeigt Abbildung 3-6.

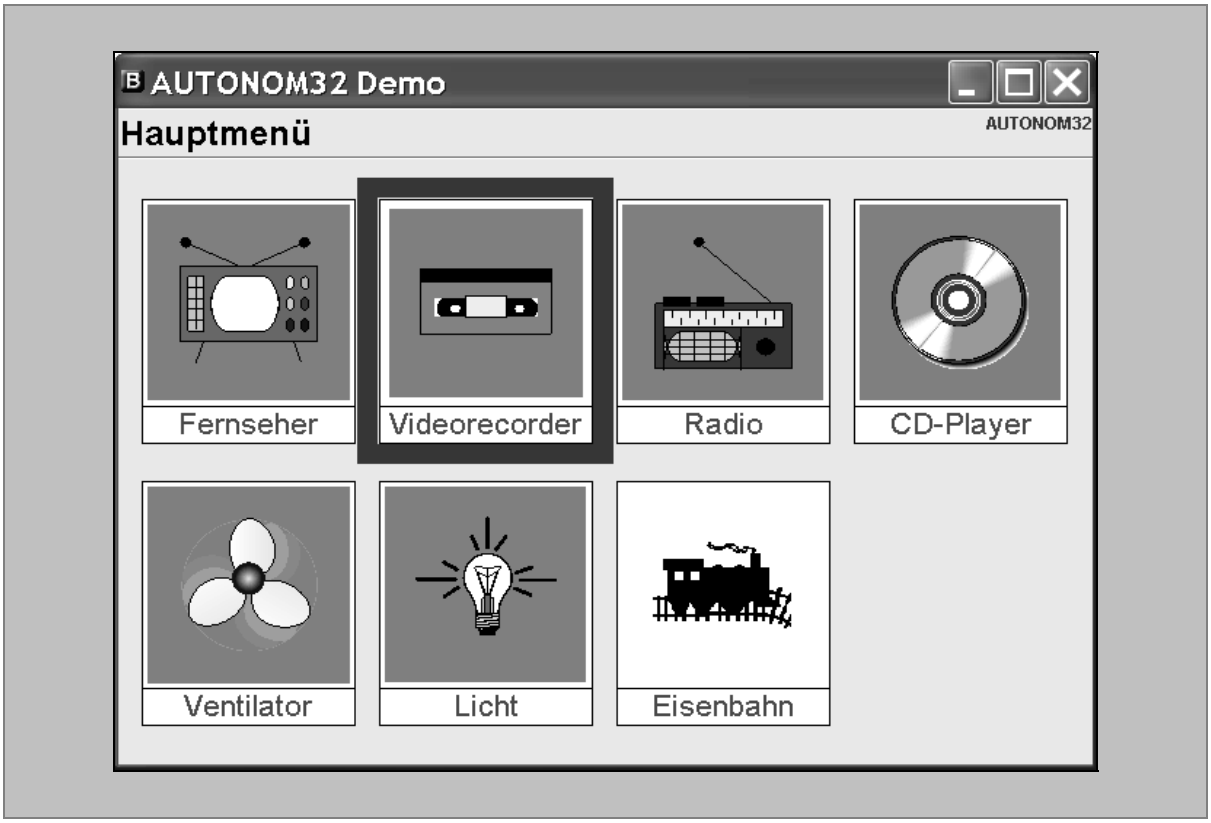

Abbildung 3-6: Ein benutzerspezifisch konfigurierbares Menü des AUTONOM-Systems.

Steuerung mittels Tastendruckemulation. Die Bedienung des AUTONOM-Systems kann über eine Vielzahl benutzerspezifischer Eingabegeräte erfolgen. Die Bedienung über die Cursor-Tasten einer Computertastatur ist natürlich ebenfalls möglich. Mit dem Hintergedanken, die Schnittstelle zur Steuerung von Applikationen, der im Rahmen dieser Arbeit entwickelten Sprachsteuerung, möglichst universell auszulegen, wird das Konzept gewählt, sogenannte „Win32-API Keyboard Events“ zu erzeugen, sodass dem AUTONOM-System Tastendrücke auf der Computertastatur vorgegaukelt werden. Diese einfache Schnittstelle kann natürlich auch leicht zur Steuerung anderer Applikationen verwendet werden. Weiters bietet dieses Konzept den Vorteil, dass keine Änderungen am bestehenden AUTONOM-System notwendig sind, da die Schnittstelle natürlicherweise bereits vorhanden ist. Eine Spezialität des AUTONOM-Systems besteht allerdings darin, dass die zur Steuerung erzeugten „Keyboard Events“ ein bestimmtes Timing bezüglich des Drückens und des Loslassens der virtuellen Tasten einhalten müssen, denn das AUTONOM-System vermeidet durch benutzerspezifische Restriktionen dieses Timings Fehlauslösungen, die auf einer tatsächlichen Tastatur beispielsweise durch zitternde Finger zustande kommen können.

3.4 Dialogablaufmodul

Grundsätze für Benutzerdialoge. Um unterschiedliche Peripheriegeräte einheitlich steuern zu können und um die kognitiven Anforderungen für den Benutzer vor allem während der Aufnahme der Schallmuster zu senken, werden einfache einheitliche Dialoge verwendet, welche unabhängig von Steuerbefehlen und Peripheriegeräten sind. Die genaue Wortwahl an bestimmten Dialogstellen wird beispielsweise zufällig variiert, in der Hoffnung, damit die Hilfsmittelakzeptanz durch den Benutzer zu erhöhen. Auch benutzerabhängige Konfiguration der Stimme der Sprachausgabe kann vermutlich dazu beitragen, eine Beziehung zum Hilfsmittel aufzubauen und damit die Hilfsmittelakzeptanz zu steigern.

Konfigurierbarkeit der Benutzerdialoge. Um die Benutzerdialoge leicht an die Bedürfnisse der Benutzer und an unterschiedliche zu steuernde Applikationen anpassen zu können, wurde eine einfache State-Machine-Beschreibungssprache definiert, in welcher die Benutzerdialoge in Form von Textdateien festgelegt sind. Die Textdateien der Benutzerdialoge werden vom Dialogablaufmodul interpretiert.

Warum wurde nicht auf eine standarisierte Lösung zurückgegriffen? Als alternativer Trend würde sich zwar das standardisierte „VoiceXML“ anbieten, allerdings beschreitet Microsoft durch die Etablierung der „Microsoft Workflow Foundation“ für „Microsoft Visual Studio“ einen eigenen Weg, der voraussichtlich zur Kundenbindung an „Microsoft Visual Studio“ beziehungsweise in weiterer Folge zur Kundenbindung an den „Microsoft Speech Server“ führt [Goth2003]. Da „Microsoft Workflow Foundation“ derzeit für die Pascal-Entwicklung im Rahmen dieser Arbeit mit „Borland Delphi“ nicht infrage kommt und auch die Zukunft von „VoiceXML“ fraglich scheint, wird im Rahmen dieser Arbeit die Entscheidung getroffen, eine eigene State-Machine-Beschreibungssprache zu definieren und zu implementieren. Vorteile bestehen außerdem darin, dass die selbstdefinierte State-Machine-Beschreibungssprache gut an die Applikation gut angepasst werden kann, dass sie einfach zu erlernen ist und dass sie bezüglich zukünftiger Bedürfnisse leicht erweiterbar ist.

4. Evaluierung

Evaluierung mit Sprachdatenbanken
Dieser Abschnitt beschreibt die Ermittlung der Fehlerraten der im Rahmen dieser Arbeit entwickelten Sprachsteuerung zum Zwecke des Auslotens eines sprecherunabhängigen Einsatzes. Zunächst werden verschiedene Nullhypothesen und Alternativhypothesen formuliert. An den entsprechenden Varianten der Sprachsteuerung werden Klassifikationstests mittels Sprachdatenbanken durchgeführt. Anschließend wird getestet, ob die beobachteten Unterschiede in den Verwechslungsfehlerraten statistisch signifikant sind, das heißt, ob die jeweiligen Nullhypothesen abzulehnen sind, oder nicht.

Vorbereitung begleitender Evaluierung im praktischen Einsatz
Dieser Abschnitt behandelt die Ermittlung der Fehlerraten im praktischen sprecherabhängigen Einsatz. Es wird ein spezieller Evaluierungsdialog vorgesehen, der mittels Rückfragen an die Benutzer die Ermittlung der erlebten Fehlerraten aus den automatischen Systemprotokollen ermöglicht. Weiters werden strukturierte Interviews mit Benutzern und Betreuungspersonen vorgeschlagen.

4.1 Evaluierung mit Sprachdatenbanken

Für den praktischen Einsatz der im Rahmen dieser Dissertation entwickelten Sprachsteuerung werden lediglich sprecherabhängige Fähigkeiten gefordert. Dennoch werden Klassifikationstests mit bestimmten sprecherunabhängigen Sprachdatenbanken durchgeführt. Der Zweck liegt im Ausloten der sprecherunabhängigen Fähigkeiten und in der Möglichkeit des Vergleichs mit anderen Sprachsteuerungen, insbesondere mit Siemens-Sprachsteuerungen[95].

Überanpassung an die Testdatenbank vermeiden. Zur Evaluierung von Mustererkennungssystemen ist es allgemein üblich, eine zur Verfügung stehende Datenbank in eine Trainingsdatenbank und eine Testdatenbank aufzuteilen, wobei die Testdatenbank dazu dient, das Verhalten des Mustererkennungssystems im späteren Einsatz abzuschätzen. Zu beachten ist dabei jedoch, dass die Testdatenbank nicht des Öfteren nach Änderungen am Mustererkennungssystem zum Einsatz kommen sollte, da jede Änderung am Mustererkennungssystem, die darauf abzielt, die Testdatenbankklassifikationsergebnisse zu verbessern, auch die Aussagekraft der Testdatenbankklassifikationsergebnisse schmälert. Das heißt, die Testdatenbankklassifikationsergebnisse spiegeln schließlich unter Umständen nur mehr unzureichend das Verhalten des Mustererkennungssystems im späteren Einsatz wieder. Das Problem besteht nämlich in einer unbeabsichtigten Optimierung bezüglich der Testdatenbank. Der Ablauf ähnelt einer Random-Search-Optimierung: Wird das Mustererkennungssystem während der jahrelangen Algorithmenentwicklung immer wieder – mehr oder wenig zufällig – verändert, wobei sich jene Veränderungen durchsetzen, die die besten Testdatenbankklassifikationsergebnisse liefern, so kommt es im Laufe der Zeit zu einer Überanpassung an die Testdatenbank. Siehe dazu auch Abschnitt I.1 auf Seite 185 im Anhang.

Keine Aufnahmen aus den Datenbanken entfernen. Natürlich kann eine Überanpassung auch leicht künstlich erzeugt werden: Man stelle sich vor, eine Sprachsteuerung liefere beim Klassifikationstest an einer Datenbank eine relativ hohe Verwechslungsfehlerrate von 50 %. Würde man nun jene Aufnahmen, die falsch klassifiziert werden, aus der Datenbank entfernen, so würde das Testergebnis dem einer perfekten Sprachsteuerung entsprechen, aber nicht die Qualität im realen Einsatz widerspiegeln. Daher werden im Rahmen dieser Arbeit die Datenbanken absolut unangetastet belassen und somit auch offensichtlich fehlerhafte Aufnahmen, die beispielsweise abgeschnittenen oder sogar leeren Inhalt aufweisen, soweit sie zur Kenntnis gelangen, nicht aus den Datenbanken entfernt. Denn würde man anfangen, Aufnahmen aus den Datenbanken zu entfernen: Wo würde man die Grenze ziehen? Freilich wäre es

[95] Sprachsteuerungen der „Siemens AG Österreich“ werden beispielsweise in der Telefonie (sprachgesteuerte Telefoniezusatzdienste, Sprachwahl für Schnurlostelefone und Handys, sprachgesteuerte Freisprecheinrichtungen), der Rehabilitationstechnik (sprachgesteuerte Prothesen, Sprachsteuerungen zur Umgebungssteuerung) sowie als sprachgesteuerte Fernbedienungen für Consumer-Electronic-Produkte (Fernseher, Stereoanlagen) eingesetzt.

wünschenswert, würden die Aufnahmen in den Datenbanken möglichst die Aufnahmeumstände des praktischen Einsatzes widerspiegeln. Eine gute Möglichkeit wäre, die Aufnahmen der Datenbanken als Nebenprodukt des praktischen Einsatzes der Sprachsteuerung zu erzeugen.

4.1.1 Beschreibung der Sprachdatenbanken

Allgemeine Beschreibung der Datenbanken. Die Sprachdatenbanken werden dankenswerter Weise von der Firma Siemens zur Verfügung gestellt [Hickersberger1997]. Es handelt sich um eine deutsche Sprachdatenbank und eine englische Sprachdatenbank mit jeweils 15 Steuerbefehlen. Bei den Sprechern der Sprachdatenbanken handelt es sich hauptsächlich um Männer mit vorwiegend deutscher Muttersprache. Die Aufnahmen der deutschen Sprachdatenbanken weisen alle korrekten Inhalt auf und es sind wenige Hintergrundgeräusche hörbar. Die Aufnahmen der englischen Sprachdatenbank sind jedoch vereinzelt abgeschnitten und weisen teilweise einen sehr schlechten Signalgeräuschabstand auf. Weiters sind bei den Aufnahmen der englischen Sprachdatenbank oftmals starke Hintergrundgeräusche zu hören. Demgemäß stellt die englische Sprachdatenbank die weitaus größere Herausforderung für die Sprachsteuerung dar, was aber nicht umgekehrt heißen soll, dass die Klassifikationsaufgabe der deutschen Sprachdatenbank einfach zu meistern wäre.

	Beschreibung der Sprecher	Sprecheranzahl	Anzahl der Steuerbefehle	Steuerbefehle	Subjektive Einschätzung der Qualität der Aufnahmen
Deutsche Trainingsdatenbank DE_TRN	Vorwiegend männliche Sprecher deutscher Muttersprache im Alter von etwa 25 bis 55 Jahren.	70	15	„Null“, „Eins“, „Zwei“,… „Neun“, „Nummer“, „Name“, „Löschen“, „Wählen“, „Ende“.	Alle Aufnahmen mit korrektem Inhalt, wenige Hintergrundgeräusche hörbar.
Deutsche Testdatenbank DE_TST		14	15		
Englische Trainingsdatenbank GB_TRN		199..204 (je nach Steuerbefehl)	15	„Zero“, „One“, „Two“, … „Nine“, „Number“, „Name“, „Delete“, „Telephone“, „Exit“.	Teilweise abgeschnittener Inhalt, teilweise sehr schlechter Signal-Geräusch-Abstand, starke Hintergrundgeräusche.
Englische Testdatenbank GB_TST		49..53 (je nach Steuerbefehl)	15		

Tabelle 7: Beschreibung der verwendeten Sprachdatenbanken.

Aufteilung in Testdatenbank und Trainingsdatenbank. Bei der Aufteilung der Sprachdatenbanken in Testdatenbanken und Trainingsdatenbanken zum Zwecke sprecherunabhängiger Klassifikationstests sind einige Bedingungen einzuhalten, damit die anhand der Testdatenbank bestimmten Kennzahlen auch für den auszulotenden sprecherunabhängigen praktischen Einsatz repräsentativ sind. Es sollen männliche und weibliche Sprecher in beiden Teildatenbanken im selben Verhältnis vorhanden sein. Kein Sprecher der Testdatenbank darf in der Trainingsdatenbank vorhanden sein. Die Reihenfolge der Sprecher soll vor der Aufteilung in Testsprecher und Trainingssprecher gemischt werden, sodass sich unabsichtlich geänderte Aufnahmebedingungen während der Abwicklung der Aufnahmesitzungen zur Erstellung der Sprachdatenbank in beiden Teildatenbanken gleichmäßig auswirken.

4.1.2 Die getesteten Varianten des Spracherkenners

„Schalterstellung“ (Konfigurationsdatei)				Benennung
LDA-Funktionalität	HCNN-Funktionalität	Fenster-funktion	RLL-Funktionalität	
Nein	Nein	Hamming	Nein	„Centroid Sequence Speech Recognizer“
Nein	Nein	**Flat-Top**	Nein	⋮
Nein	Nein	Rectangle	Nein	⋮
Ja	Nein	Hamming	Nein	⋮
Ja	Nein	**Flat-Top**	Nein	⋮
Ja	Nein	Rectangle	Nein	⋮
Nein	**Ja**	Hamming	Nein	„Centroid Sequence Hidden Control Neural Network Speech Recognizer“
Nein	**Ja**	**Flat-Top**	Nein	⋮
Nein	**Ja**	Rectangle	Nein	⋮
Ja	**Ja**	Hamming	Nein	⋮
Ja	**Ja**	**Flat-Top**	Nein	⋮
Ja	**Ja**	Rectangle	Nein	⋮
Nein	Nein	Hamming	**Ja**	„Centroid Sequence Speech Recognizer“ (mit Run-Length-Limited-Sprechtempokompensator, Österr. Pat. Nr. 414060)
Nein	Nein	**Flat-Top**	**Ja**	⋮
Nein	Nein	Rectangle	**Ja**	⋮
Ja	Nein	Hamming	**Ja**	⋮
Ja	Nein	**Flat-Top**	**Ja**	⋮
Ja	Nein	Rectangle	**Ja**	⋮
Nein	**Ja**	Hamming	**Ja**	„Centroid Sequence Hidden Control Neural Network Speech Recognizer“ (mit Run-Length-Limited-Sprechtempokompensator, Österr. Pat. Nr. 414060)
Nein	**Ja**	**Flat-Top**	**Ja**	⋮
Nein	**Ja**	Rectangle	**Ja**	⋮
Ja	**Ja**	Hamming	**Ja**	⋮
Ja	**Ja**	**Flat-Top**	**Ja**	⋮
Ja	**Ja**	Rectangle	**Ja**	⋮

Tabelle 8: Benennung der getesteten Varianten des Spracherkennermoduls.

Konfiguration der realisierten Sprachsteuerung mittels „Schaltern“. Tabelle 8 zeigt, wie durch verschiedene „Schalter“ unterschiedliche, im theoretischen Teil dieser Arbeit beschriebene Verfahren konfiguriert werden können. Der Schalter „LDA-Funktionalität“ aktiviert beziehungsweise deaktiviert die LDA-Transformation im Pattern-Builder-Funktionsblocks nach Abbildung 1-3 auf Seite 20. Der Schalter „HCNN-Funktionalität“ schaltet zwischen dem Sub-Word-Unit-Modell-Funktionsblock nach Abbildung 1-14 auf Seite 42 und dem nach Abbildung 1-19 auf Seite 49 um, indem das „Hidden Control Neural Network“ aktiviert beziehungsweise deaktiviert wird. Der Schalter „Fensterfunktion“ schaltet die Fensterfunktion der Kurzzeit-Fourier-Transformation im Pattern-Builder-Funktionsblocks nach Abbildung 1-3 auf Seite 20 um. Der Schalter „RLL-Funktionalität“ schaltet zwischen dem „Viterbi-Dynamic-Programming-Sprechtempo-kompensator“ nach Abbildung 1-12 auf Seite 37 und dem „Run-Length-Limited-Sprechtempokompensator“ nach Abbildung 1-25 auf Seite 60 um. Die Stellungen der „Schalter“ haben natürlich weitreichende Auswirkungen auf den Trainingsablauf. Zumindest für den Klassifikationsvorgang lassen sich die umgeschalteten Funktionalitäten aber leicht in den grundlegenden Blockschaltbildern in Abbildung 1-2 auf Seite 17, Abbildung 1-3 auf Seite 20, Abbildung 1-6 auf Seite 30 sowie Abbildung 1-7 auf Seite 31 erkennen.

Bezeichnung der konfigurierbaren Verfahren. Durch die konkreten Schalterstellungen ergeben sich die Funktionalitäten der Verfahren „Centroid Sequence Speech Recognizer“ nach Abschnitt 1.2.1.4 und „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ nach Abschnitt 1.2.1.5. Das österreichische Patent Nummer 414060 „Klangfolgenerkenner“ erstreckt sich im Speziellen auf Verfahren mit aktiviertem „Run-Length-Limited-Sprechtempokompensator“ nach Abbildung 1-25 auf Seite 60.

4.1.3 Alternativhypothesen und Nullhypothesen

Durch die Konfigurationsmöglichkeit des Spracherkenners mittels „Schaltern“ (Konfigurationsdatei) wurde unmittelbar die Überprüfung folgender Alternativhypothesen vorbereitet: LDA-Hypothese, HCNN-Hypothese, Flat-Top-Hypothese, RLL-Hypothese.

Alternativhypothesen. Die Alternativhypothesen bestehen jeweils in der Annahme, dass die Datenbanktests für die Verfahren mit jeweils aktiviertem Feature besser ausfallen als für die Verfahren mit deaktiviertem Feature. Die Flat-Top-Hypothese besteht in der Annahme, dass die Datenbanktests bei den Verfahren mit aktiviertem Flat-Top-Fenster besser ausfallen als für die zugehörigen Verfahren mit aktiviertem Hamming-Fenster.

Nullhypothesen. Die zugehörigen Nullhypothesen bestehen jeweils in der Annahme, dass die Datenbanktests für die Verfahren mit jeweils aktiviertem Feature gleich gut oder schlechter ausfallen als für die Verfahren mit deaktiviertem Feature. Die zugehörige Nullhypothese zur Flat-Top-Hypothese besteht in der Annahme, dass die Datenbanktests bei den Verfahren mit aktiviertem Flat-Top-Fenster gleich gut oder schlechter ausfallen, als für die zugehörigen Verfahren mit aktiviertem Hamming-Fenster.

Wilcoxon-Vorzeichen-Rangtest. In Analogie zu medizinischen Fragestellungen, bei denen die Wirksamkeit einer Therapie geprüft werden soll, wird der Wilcoxon-Vorzeichen-Rangtest gewählt. Die Patienten entsprechen den einzelnen Spracherkennervarianten. Die Therapie besteht im Einschalten des jeweiligen Features. Der Wilcoxon-Vorzeichen-Rangtest ist ein nichtparametrischer statistischer Test zur Prüfung der zentralen Tendenzen zweier gepaarter Stichproben, wobei im Vergleich zum t-Test keine Annahme bezüglich Normalverteilung getroffen werden muss[96].

Bonferroni-Korrektur. Da mit den gewonnenen Daten vier verschiedene Alternativhypothesen getestet werden, wird sicherheitshalber die konservative Bonferroni-Korrektur[97] angewendet, obwohl jeweils eine andere Teilmenge der gewonnenen Daten zum Test herangezogen wird. Die jeweiligen Teilmengen haben jedoch nichtleere Schnittmengen. Als „signifikant“ wird im Folgenden ein unterschrittenes korrigiertes Signifikanzniveau von 0.05/4 = 0.0125 angesehen.

---

[96] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Wilcoxon-Vorzeichen-Rang-Test (Stand: 15. 08. 2012).

[97] Quelle: Deutsche Wikipedia http://de.wikipedia.org/wiki/Bonferroni-Methode (Stand: 15. 08. 2012).

4.1.4 Ergebnisse der Sprachdatenbanktests

4.1.4.1 Klassifikation der Testdaten

Variante				Verwechslungsfehlerrate beim Test mit Datenbank			
LDA-Funktionalität	HCNN-Funktionalität	Fenster-funktion	RLL-Funktionalität	DE_TRN (Training mit DE_TRN)	⬇ DE_TST (Training mit DE_TRN)	GB_TRN (Training mit GB_TRN)	GB_TST (Training mit GB_TRN)
Ja	Nein	Rectangle	**Ja**	3,00 %	1,00 %	10,30 %	9,50 %
Ja	Nein	Hamming	**Ja**	3,30 %	1,00 %	10,10 %	10,20 %
Ja	**Ja**	Rectangle	**Ja**	1,70 %	1,40 %	9,00 %	9,30 %
Nein	Nein	Rectangle	**Ja**	4,40 %	1,40 %	12,70 %	10,50 %
Ja	Nein	Flat-Top	**Ja**	3,60 %	1,90 %	10,10 %	11,30 %
Nein	Nein	Flat-Top	**Ja**	4,60 %	1,90 %	12,60 %	10,50 %
Nein	**Ja**	Flat-Top	**Ja**	0,20 %	2,40 %	3,20 %	8,70 %
Ja	**Ja**	Hamming	**Ja**	2,60 %	2,90 %	8,70 %	7,50 %
Nein	Nein	Hamming	**Ja**	4,70 %	2,90 %	12,40 %	10,50 %
Nein	**Ja**	Hamming	Nein	0,20 %	3,30 %	4,50 %	14,40 %
Nein	**Ja**	Hamming	**Ja**	0,20 %	3,30 %	3,20 %	8,30 %
Ja	**Ja**	Flat-Top	**Ja**	2,40 %	3,30 %	7,70 %	10,50 %
Nein	**Ja**	Rectangle	Nein	0,10 %	3,80 %	5,60 %	16,30 %
Nein	**Ja**	Rectangle	**Ja**	0,20 %	3,80 %	3,40 %	6,60 %
Nein	**Ja**	Flat-Top	Nein	0,40 %	4,30 %	4,40 %	13,30 %
Nein	Nein	Rectangle	Nein	11,30 %	5,20 %	22,80 %	23,70 %
Nein	Nein	Hamming	Nein	10,00 %	5,70 %	21,10 %	21,90 %
Ja	Nein	Rectangle	Nein	10,10 %	5,70 %	23,40 %	21,90 %
Nein	Nein	Flat-Top	Nein	10,20 %	6,20 %	19,70 %	18,50 %
Ja	**Ja**	Flat-Top	Nein	5,70 %	7,10 %	15,80 %	16,80 %
Ja	Nein	Flat-Top	Nein	9,60 %	7,10 %	18,20 %	18,70 %
Ja	**Ja**	Rectangle	Nein	6,20 %	7,60 %	19,00 %	20,00 %
Ja	Nein	Hamming	Nein	10,00 %	7,60 %	21,90 %	21,50 %
Ja	**Ja**	Hamming	Nein	6,50 %	9,00 %	18,00 %	22,00 %

Tabelle 9: Verwechslungsfehlerraten der getesteten Varianten des Spracherkennermoduls (Version 080630). Sortiert bezüglich der Verwechslungsfehlerraten unter Verwendung der Datenbank DE_TST.

Tabelle 9 zeigt die Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung, wobei die gesamte Tabelle bezüglich der Verwechslungsfehlerraten unter Verwendung der deutschen Testdatenbank sortiert ist. Augenscheinlich ist dabei der günstige Einfluss der Aktivierung der RLL-Funktionalität.

Die besten Ergebnisse werden mit aktivierter LDA-Funktionalität und deaktivierter HCNN-Funktionalität erreicht, also mit einem einfachen „Centroid Sequence Speech Recognizer“ mit aktivierter LDA-Transformation entsprechend Tabelle 8 auf Seite 163 und aktiviertem Run-Length-Limited-Sprechtempokompensator.

Variante				Verwechslungsfehlerrate beim Test mit Datenbank			
LDA-Funktionalität	HCNN-Funktionalität	Fenster-funktion	RLL-Funktionalität	DE_TRN (Training mit DE_TRN)	DE_TST (Training mit DE_TRN)	GB_TRN (Training mit GB_TRN)	⬇ GB_TST (Training mit GB_TRN)
Nein	**Ja**	Rectangle	**Ja**	0,20 %	3,80 %	3,40 %	6,60 %
Ja	**Ja**	Hamming	**Ja**	2,60 %	2,90 %	8,70 %	7,50 %
Nein	**Ja**	Hamming	**Ja**	0,20 %	3,30 %	3,20 %	8,30 %
Nein	**Ja**	Flat-Top	**Ja**	0,20 %	2,40 %	3,20 %	8,70 %
Ja	**Ja**	Rectangle	**Ja**	1,70 %	1,40 %	9,00 %	9,30 %
Ja	Nein	Rectangle	**Ja**	3,00 %	1,00 %	10,30 %	9,50 %
Ja	Nein	Hamming	**Ja**	3,30 %	1,00 %	10,10 %	10,20 %
Nein	Nein	Rectangle	**Ja**	4,40 %	1,40 %	12,70 %	10,50 %
Nein	Nein	Flat-Top	**Ja**	4,60 %	1,90 %	12,60 %	10,50 %
Nein	Nein	Hamming	**Ja**	4,70 %	2,90 %	12,40 %	10,50 %
Ja	**Ja**	Flat-Top	**Ja**	2,40 %	3,30 %	7,70 %	10,50 %
Ja	Nein	Flat-Top	**Ja**	3,60 %	1,90 %	10,10 %	11,30 %
Nein	**Ja**	Flat-Top	Nein	0,40 %	4,30 %	4,40 %	13,30 %
Nein	**Ja**	Hamming	Nein	0,20 %	3,30 %	4,50 %	14,40 %
Nein	**Ja**	Rectangle	Nein	0,10 %	3,80 %	5,60 %	16,30 %
Ja	**Ja**	Flat-Top	Nein	5,70 %	7,10 %	15,80 %	16,80 %
Nein	Nein	Flat-Top	Nein	10,20 %	6,20 %	19,70 %	18,50 %
Ja	Nein	Flat-Top	Nein	9,60 %	7,10 %	18,20 %	18,70 %
Ja	**Ja**	Rectangle	Nein	6,20 %	7,60 %	19,00 %	20,00 %
Ja	Nein	Hamming	Nein	10,00 %	7,60 %	21,90 %	21,50 %
Nein	Nein	Hamming	Nein	10,00 %	5,70 %	21,10 %	21,90 %
Ja	Nein	Rectangle	Nein	10,10 %	5,70 %	23,40 %	21,90 %
Ja	**Ja**	Hamming	Nein	6,50 %	9,00 %	18,00 %	22,00 %
Nein	Nein	Rectangle	Nein	11,30 %	5,20 %	22,80 %	23,70 %

Tabelle 10: Verwechslungsfehlerraten der getesteten Varianten des Spracherkennermoduls (Version 080630). Sortiert bezüglich der Verwechslungsfehlerraten unter Verwendung der Datenbank GB_TST.

Tabelle 10 zeigt die Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung, wobei die gesamte Tabelle bezüglich der Verwechslungsfehlerraten unter Verwendung der englischen Testdatenbank sortiert ist. Auch hier zeigt sich der eindeutig günstige Einfluss der Aktivierung der RLL-Funktionalität.

Die besten Verwechslungsfehlerraten werden hier, im Gegensatz zu den Verwechslungsfehlerraten unter Verwendung der deutschen Testdatenbank, bei aktivierter HCNN-Funktionalität erreicht, was wohl daran liegt, dass die englische Datenbank, aufgrund ihres größeren Umfangs, für das Training der relativ größeren Anzahl an

Modellparametern, vorteilhafter ist. Diese Konfiguration entspricht einem „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ nach Tabelle 8 auf Seite 163 mit aktiviertem Run-Length-Limited-Sprechtempokompensator.

4.1.4.2 Reklassifikation der Trainingsdaten

Variante				Verwechslungsfehlerrate beim Test mit Datenbank			
LDA-Funktionalität	HCNN-Funktionalität	Fenster-funktion	RLL-Funktionalität	⬇ DE_TRN (Training mit DE_TRN)	DE_TST (Training mit DE_TRN)	GB_TRN (Training mit GB_TRN)	GB_TST (Training mit GB_TRN)
Nein	**Ja**	Rectangle	Nein	0,10 %	3,80 %	5,60 %	16,30 %
Nein	**Ja**	Hamming	Nein	0,20 %	3,30 %	4,50 %	14,40 %
Nein	**Ja**	Flat-Top	**Ja**	0,20 %	2,40 %	3,20 %	8,70 %
Nein	**Ja**	Hamming	**Ja**	0,20 %	3,30 %	3,20 %	8,30 %
Nein	**Ja**	Rectangle	**Ja**	0,20 %	3,80 %	3,40 %	6,60 %
Nein	**Ja**	Flat-Top	Nein	0,40 %	4,30 %	4,40 %	13,30 %
Ja	**Ja**	Rectangle	**Ja**	1,70 %	1,40 %	9,00 %	9,30 %
Ja	**Ja**	Flat-Top	**Ja**	2,40 %	3,30 %	7,70 %	10,50 %
Ja	**Ja**	Hamming	**Ja**	2,60 %	2,90 %	8,70 %	7,50 %
Ja	Nein	Rectangle	**Ja**	3,00 %	1,00 %	10,30 %	9,50 %
Ja	Nein	Hamming	**Ja**	3,30 %	1,00 %	10,10 %	10,20 %
Ja	Nein	Flat-Top	**Ja**	3,60 %	1,90 %	10,10 %	11,30 %
Nein	Nein	Rectangle	**Ja**	4,40 %	1,40 %	12,70 %	10,50 %
Nein	Nein	Flat-Top	**Ja**	4,60 %	1,90 %	12,60 %	10,50 %
Nein	Nein	Hamming	**Ja**	4,70 %	2,90 %	12,40 %	10,50 %
Ja	**Ja**	Flat-Top	Nein	5,70 %	7,10 %	15,80 %	16,80 %
Ja	**Ja**	Rectangle	Nein	6,20 %	7,60 %	19,00 %	20,00 %
Ja	**Ja**	Hamming	Nein	6,50 %	9,00 %	18,00 %	22,00 %
Ja	Nein	Flat-Top	Nein	9,60 %	7,10 %	18,20 %	18,70 %
Nein	Nein	Hamming	Nein	10,00 %	5,70 %	21,10 %	21,90 %
Ja	Nein	Hamming	Nein	10,00 %	7,60 %	21,90 %	21,50 %
Ja	Nein	Rectangle	Nein	10,10 %	5,70 %	23,40 %	21,90 %
Nein	Nein	Flat-Top	Nein	10,20 %	6,20 %	19,70 %	18,50 %
Nein	Nein	Rectangle	Nein	11,30 %	5,20 %	22,80 %	23,70 %

Tabelle 11: Verwechslungsfehlerraten der getesteten Varianten des Spracherkennermoduls (Version 080630). Sortiert bezüglich der Verwechslungsfehlerraten unter Verwendung der Datenbank DE_TRN.

Tabelle 11 zeigt die Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung, wobei die gesamte Tabelle bezüglich der Verwechslungsfehlerraten unter Verwendung der deutschen Trainingsdatenbank sortiert ist.

Man sieht, dass die Aktivierung der HCNN-Funktionalität eine sehr gute Reklassifikation der deutschen Trainingsdatenbank bewirkt, die sich aber nicht in vollem

Umfang auf die deutsche Testdatenbank überträgt. Dies deutet darauf hin, dass möglicherweise eine schwache Überanpassung an die Trainingsdatenbank, möglicherweise begründet durch die relativ große Anzahl freier Modellparameter des Neuronalen Netzes, vorliegt.

Weiters sieht man, dass die LDA-Funktionalität bezüglich der Reklassifikation der deutschen Trainingsdatenbank nicht besonders hilfreich zu sein scheint, ihre Vorzüge aber bei der Klassifikation der deutschen Testdatenbank ausspielt, wie man Tabelle 9 auf Seite 166 entnehmen kann.

Variante				Verwechslungsfehlerrate beim Test mit Datenbank			
LDA-Funktionalität	HCNN-Funktionalität	Fenster-funktion	RLL-Funktionalität	DE_TRN (Training mit DE_TRN)	DE_TST (Training mit DE_TRN)	⬇ GB_TRN (Training mit GB_TRN)	GB_TST (Training mit GB_TRN)
Nein	**Ja**	Hamming	**Ja**	0,20 %	3,30 %	3,20 %	8,30 %
Nein	**Ja**	Flat-Top	**Ja**	0,20 %	2,40 %	3,20 %	8,70 %
Nein	**Ja**	Rectangle	**Ja**	0,20 %	3,80 %	3,40 %	6,60 %
Nein	**Ja**	Flat-Top	Nein	0,40 %	4,30 %	4,40 %	13,30 %
Nein	**Ja**	Hamming	Nein	0,20 %	3,30 %	4,50 %	14,40 %
Nein	**Ja**	Rectangle	Nein	0,10 %	3,80 %	5,60 %	16,30 %
Ja	**Ja**	Flat-Top	**Ja**	2,40 %	3,30 %	7,70 %	10,50 %
Ja	**Ja**	Hamming	**Ja**	2,60 %	2,90 %	8,70 %	7,50 %
Ja	**Ja**	Rectangle	**Ja**	1,70 %	1,40 %	9,00 %	9,30 %
Ja	Nein	Hamming	**Ja**	3,30 %	1,00 %	10,10 %	10,20 %
Ja	Nein	Flat-Top	**Ja**	3,60 %	1,90 %	10,10 %	11,30 %
Ja	Nein	Rectangle	**Ja**	3,00 %	1,00 %	10,30 %	9,50 %
Nein	Nein	Hamming	**Ja**	4,70 %	2,90 %	12,40 %	10,50 %
Nein	Nein	Flat-Top	**Ja**	4,60 %	1,90 %	12,60 %	10,50 %
Nein	Nein	Rectangle	**Ja**	4,40 %	1,40 %	12,70 %	10,50 %
Ja	**Ja**	Flat-Top	Nein	5,70 %	7,10 %	15,80 %	16,80 %
Ja	**Ja**	Hamming	Nein	6,50 %	9,00 %	18,00 %	22,00 %
Ja	Nein	Flat-Top	Nein	9,60 %	7,10 %	18,20 %	18,70 %
Ja	**Ja**	Rectangle	Nein	6,20 %	7,60 %	19,00 %	20,00 %
Nein	Nein	Flat-Top	Nein	10,20 %	6,20 %	19,70 %	18,50 %
Nein	Nein	Hamming	Nein	10,00 %	5,70 %	21,10 %	21,90 %
Ja	Nein	Hamming	Nein	10,00 %	7,60 %	21,90 %	21,50 %
Nein	Nein	Rectangle	Nein	11,30 %	5,20 %	22,80 %	23,70 %
Ja	Nein	Rectangle	Nein	10,10 %	5,70 %	23,40 %	21,90 %

Tabelle 12: Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630). Sortiert bezüglich der Verwechslungsfehlerraten unter Verwendung der Datenbank GB_TRN. Das Training erfolgt in allen Fällen mit der jeweiligen Trainingsdatenbank DE_TRN beziehungsweise GB_TRN.

Tabelle 12 zeigt die Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung, wobei die gesamte Tabelle bezüglich der Verwechslungsfehlerraten unter Verwendung der englischen Trainingsdatenbank sortiert ist.

Man sieht, dass die Aktivierung der HCNN-Funktionalität wiederum eine sehr gute Reklassifikation der Trainingsdatenbank bewirkt, die sich jedoch wiederum nicht in vollem Umfang auf die Testdatenbank überträgt, was darauf hindeutet, dass möglicherweise eine schwache Überanpassung an die Trainingsdatenbank, möglicherweise begründet durch die relativ große Anzahl freier Modellparameter des Neuronalen Netzes, vorliegt.

Weiters sieht man, dass die Aktivierung der LDA-Funktionalität bezüglich der Reklassifikation der englischen Trainingsdatenbank wiederum nicht besonders hilfreich zu sein scheint. Die Aktivierung der LDA-Funktionalität bietet im Gegensatz zum guten Abschneiden bezüglich der deutschen Testdatenbank, gemäß Tabelle 9, bezüglich der englischen Testdatenbank, gemäß Tabelle 10, keinen eindeutigen Vorteil.

4.1.5 Hypothesenprüfung (deutsche Datenbank)

4.1.5.1 LDA-Hypothese (deutsche Datenbank, nicht signifikant)

HCNN-Funktionalität	Fensterfunktion	RLL-Funktionalität	***vorher*** (DE_TST, ohne LDA)	***nachher*** (DE_TST, (mit LDA)
Nein	Rectangle	Nein	5,20%	5,70%
Nein	Hamming	Nein	5,70%	7,60%
Nein	Flat-Top	Nein	6,20%	7,10%
Ja	Rectangle	Nein	3,80%	7,60%
Ja	Hamming	Nein	3,30%	9,00%
Ja	Flat-Top	Nein	4,30%	7,10%
Nein	Rectangle	**Ja**	1,40%	1,00%
Nein	Hamming	**Ja**	2,90%	1,00%
Nein	Flat-Top	**Ja**	1,90%	1,90%
Ja	Rectangle	**Ja**	3,80%	1,40%
Ja	Hamming	**Ja**	3,30%	2,90%
Ja	Flat-Top	**Ja**	2,40%	3,30%

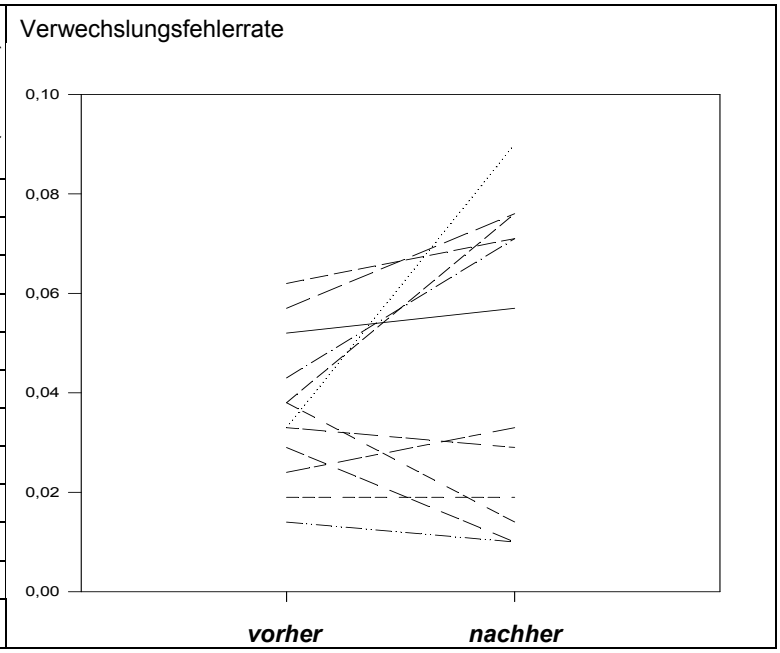

Tabelle 13: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei der Aktivierung der LDA-Funktionalität.

Wilcoxon Signed Rank Test
Normality Test (Shapiro-Wilk) Passed (P = 0,813)

Group	N	Missing	Median	25%	75%
vorher	12	0	0,0355	0,0253	0,0497
nachher	12	0	0,0450	0,0153	0,0747

W= 30,000 T+ = 48,000 T-= -18,000
Z-Statistic (based on positive ranks) = 1,334
P(est.)= 0,197 P(exact)= 0,206

The change that occurred with the treatment is not great enough to exclude the possibility that it is due to chance (P = 0,206).

Abbildung 4-1: Wilcoxon-Vorzeichen-Rangtest zur LDA-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-1 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur LDA-Hypothese, getestet an der deutschen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „nicht signifikant“, das heißt, die Nullhypothese wird angenommen.

4.1.5.2 HCNN-Hypothese (deutsche Datenbank, nicht signifikant)

LDA-Funktionalität	Fensterfunktion	RLL-Funktionalität	***vorher*** (DE_TST, ohne HCNN)	***nachher*** (DE_TST, mit HCNN)
Nein	Rectangle	Nein	5,20%	3,80%
Nein	Hamming	Nein	5,70%	3,30%
Nein	Flat-Top	Nein	6,20%	4,30%
Ja	Rectangle	Nein	5,70%	7,60%
Ja	Hamming	Nein	7,60%	9,00%
Ja	Flat-Top	Nein	7,10%	7,10%
Nein	Rectangle	**Ja**	1,40%	3,80%
Nein	Hamming	**Ja**	2,90%	3,30%
Nein	Flat-Top	**Ja**	1,90%	2,40%
Ja	Rectangle	**Ja**	1,00%	1,40%
Ja	Hamming	**Ja**	1,00%	2,90%
Ja	Flat-Top	**Ja**	1,90%	3,30%

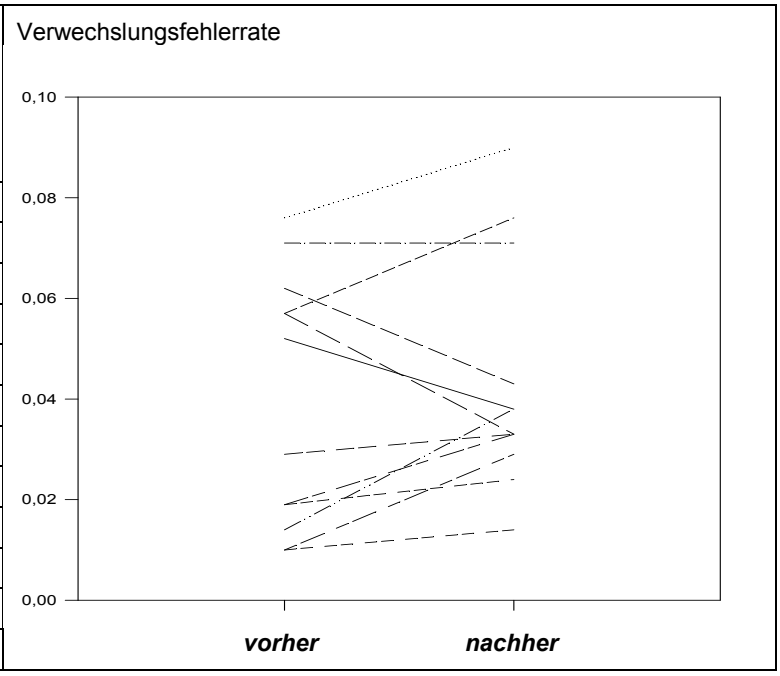

Tabelle 14: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei Aktivierung der HCNN-Funktionalität.

Wilcoxon Signed Rank Test
Normality Test (Shapiro-Wilk) Passed (P = 0,274)

Group	N	Missing	Median	25%	75%
vorher	12	0	0,0405	0,0153	0,0607
nachher	12	0	0,0355	0,0300	0,0640

W= 19,000 T+ = 42,500 T-= -23,500
Z-Statistic (based on positive ranks) = 0,846
P(est.)= 0,423 P(exact)= 0,413

The change that occurred with the treatment is not great enough to exclude the possibility that it is due to chance (P = 0,413).

Abbildung 4-2: Wilcoxon-Vorzeichen-Rangtest zur HCNN-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-2 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur HCNN-Hypothese, getestet an der deutschen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „nicht signifikant“, das heißt, die Nullhypothese wird angenommen.

4.1.5.3 Flat-Top-Hypothese (deutsche Datenbank, nicht signifikant)

LDA-Funktionalität	HCNN-Funktionalität	RLL-Funktionalität	***vorher*** (DE_TST, mit Hamming)	***nachher*** (DE_TST, mit Flat-Top)
Nein	Nein	Nein	5,70%	6,20%
Ja	Nein	Nein	7,60%	7,10%
Nein	**Ja**	Nein	3,30%	4,30%
Ja	**Ja**	Nein	9,00%	7,10%
Nein	Nein	**Ja**	2,90%	1,90%
Ja	Nein	**Ja**	1,00%	1,90%
Nein	**Ja**	**Ja**	3,30%	2,40%
Ja	**Ja**	**Ja**	2,90%	3,30%
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

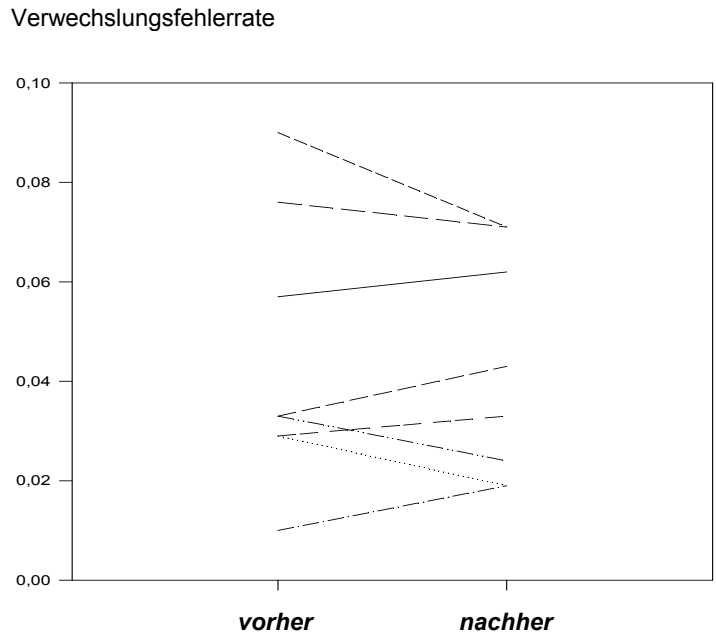

Tabelle 15: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei der Aktivierung des Flat-Top-Fensters anstelle des Hamming-Fensters.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,464)

Group	N	Missing	Median	25%	75%
vorher	8	0	0,0330	0,0290	0,0712
nachher	8	0	0,0380	0,0203	0,0687

W= -10,000 T+ = 13,000 T-= -23,000
Z-Statistic (based on positive ranks) = -0,700
P(est.)= 0,529 P(exact)= 0,547

The change that occurred with the treatment is not great enough to exclude the possibility that it is due to chance (P = 0,547).

Abbildung 4-3: Wilcoxon-Vorzeichen-Rangtest zur HCNN-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-3 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur Flat-Top-Hypothese, getestet an der deutschen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „nicht signifikant“, das heißt, die Nullhypothese wird angenommen.

4.1.5.4 RLL-Hypothese (deutsche Datenbank, signifikant)

LDA-Funktionalität	HCNN-Funktionalität	Fensterfunktion	***vorher*** (DE_TST, ohne RLL)	***nachher*** (DE_TST, (mit RLL)
Nein	Nein	Rectangle	5,20%	1,40%
Nein	Nein	Hamming	5,70%	2,90%
Nein	Nein	Flat-Top	6,20%	1,90%
Nein	**Ja**	Rectangle	3,80%	3,80%
Nein	**Ja**	Hamming	3,30%	3,30%
Nein	**Ja**	Flat-Top	4,30%	2,40%
Ja	Nein	Rectangle	5,70%	1,00%
Ja	Nein	Hamming	7,60%	1,00%
Ja	Nein	Flat-Top	7,10%	1,90%
Ja	**Ja**	Rectangle	7,60%	1,40%
Ja	**Ja**	Hamming	9,00%	2,90%
Ja	**Ja**	Flat-Top	7,10%	3,30%

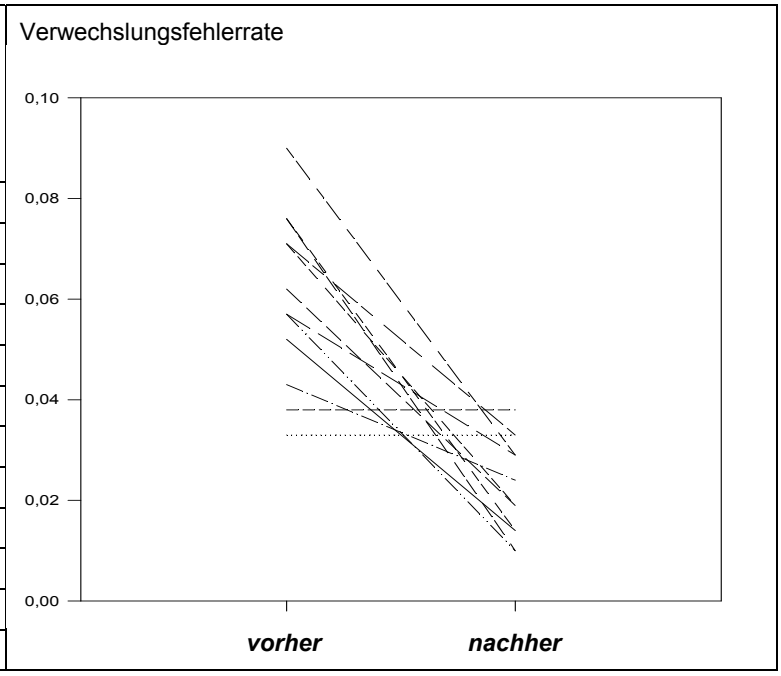

Tabelle 16: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei Aktivierung der RLL-Funktionalität.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,288)

Group	N	Missing	Median	25%	75%
vorher	12	0	0,0595	0,0452	0,0747
nachher	12	0	0,0215	0,0140	0,0320

W= -55,000 T+ = 0,000 T-= -55,000
Z-Statistic (based on positive ranks) = -2,803
P(est.)= 0,006 P(exact)= 0,002

The change that occurred with the treatment is greater than would be expected by chance; there is a statistically significant difference (P = 0,002).

Abbildung 4-4: Wilcoxon-Vorzeichen-Rangtest zur RLL-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-4 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur RLL-Hypothese, getestet an der deutschen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „signifikant“, das heißt, die Nullhypothese wird zugunsten der Alternativhypothese verworfen.

4.1.6 Wiederholung der Hypothesenprüfung (englische Datenbank)

4.1.6.1 LDA-Hypothese (englische Datenbank, nicht signifikant)

HCNN-Funktionalität	Fensterfunktion	RLL-Funktionalität	***vorher*** (GB_TST, ohne LDA)	***nachher*** (GB_TST, mit LDA)
Nein	Rectangle	Nein	23,70%	21,90%
Nein	Hamming	Nein	21,90%	21,50%
Nein	Flat-Top	Nein	18,50%	18,70%
Ja	Rectangle	Nein	16,30%	20,00%
Ja	Hamming	Nein	14,40%	22,00%
Ja	Flat-Top	Nein	13,30%	16,80%
Nein	Rectangle	**Ja**	10,50%	9,50%
Nein	Hamming	**Ja**	10,50%	10,20%
Nein	Flat-Top	**Ja**	10,50%	11,30%
Ja	Rectangle	**Ja**	6,60%	9,30%
Ja	Hamming	**Ja**	8,30%	7,50%
Ja	Flat-Top	**Ja**	8,70%	10,50%

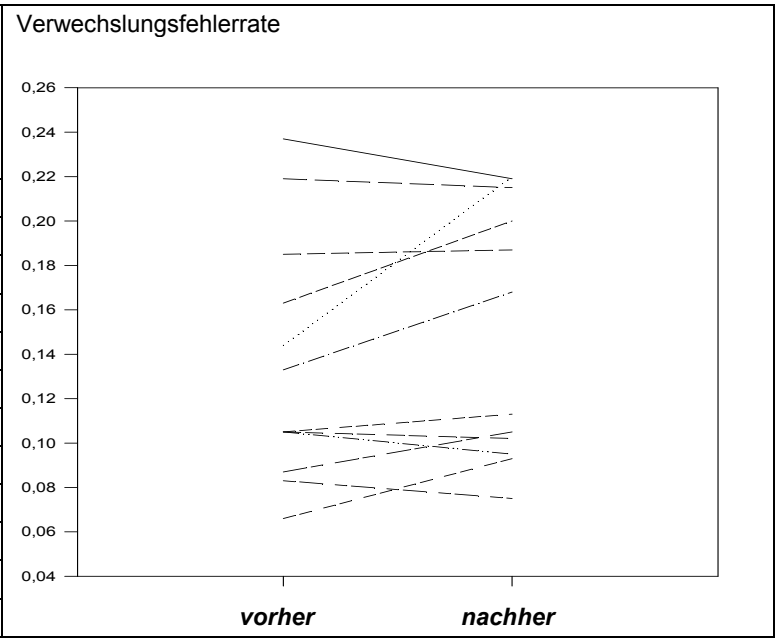

Tabelle 17: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei Aktivierung der LDA-Funktionalität.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,165)

Group	N	Missing	Median	25%	75%
Col 1	12	0	0,119	0,0915	0,179
Col 2	12	0	0,141	0,0968	0,211

W= 33,000 T+ = 55,500 T-= -22,500
Z-Statistic (based on positive ranks) = 1,295
P(est.)= 0,209 P(exact)= 0,204

The change that occurred with the treatment is not great enough to exclude the possibility that it is due to chance (P = 0,204).

Abbildung 4-5: Wilcoxon-Vorzeichen-Rangtest zur LDA-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-5 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur LDA-Hypothese, getestet an der englischen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „nicht signifikant“, das heißt, die Nullhypothese wird angenommen.

4.1.6.2 HCNN-Hypothese (englische Datenbank, signifikant)

LDA-Funktionalität	Fensterfunktion	RLL-Funktionalität	***vorher*** (GB_TST, ohne HCNN)	***nachher*** (GB_TST, mit HCNN)
Nein	Rectangle	Nein	23,70%	16,30%
Nein	Hamming	Nein	21,90%	14,40%
Nein	Flat-Top	Nein	18,50%	13,30%
Ja	Rectangle	Nein	21,90%	20,00%
Ja	Hamming	Nein	21,50%	22,00%
Ja	Flat-Top	Nein	18,70%	16,80%
Nein	Rectangle	**Ja**	10,50%	6,60%
Nein	Hamming	**Ja**	10,50%	8,30%
Nein	Flat-Top	**Ja**	10,50%	8,70%
Ja	Rectangle	**Ja**	9,50%	9,30%
Ja	Hamming	**Ja**	10,20%	7,50%
Ja	Flat-Top	**Ja**	11,30%	10,50%

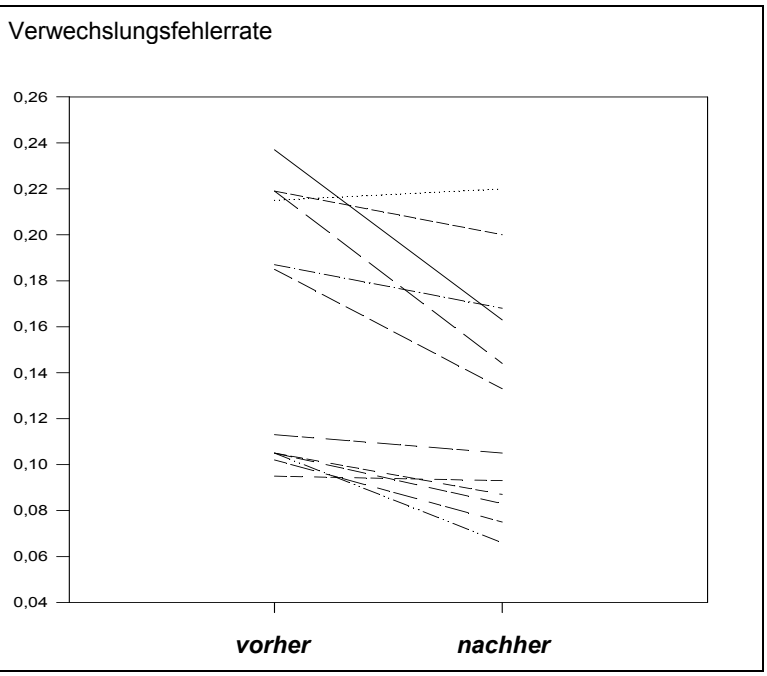

Tabelle 18: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei Aktivierung der HCNN-Funktionalität.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,185)

Group	N	Missing	Median	25%	75%
vorher	12	0	0,149	0,105	0,218
nachher	12	0	0,119	0,0840	0,167

W= -74,000 T+ = 2,000 T-= -76,000
Z-Statistic (based on positive ranks) = -2,904
P(est.)= 0,004 P(exact)= 0,001

The change that occurred with the treatment is greater than would be expected by chance; there is a statistically significant difference (P = 0,001).

Abbildung 4-6: Wilcoxon-Vorzeichen-Rangtest zur HCNN-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-6 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur HCNN-Hypothese, getestet an der englischen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „signifikant“, das heißt, die Nullhypothese wird zugunsten der Alternativhypothese verworfen. An der deutschen Sprachdatenbank ergibt sich jedoch das Ergebnis „nicht signifikant“ (siehe Seite 172).

4.1.6.3 Flat-Top-Hypothese (englische Datenbank, nicht signifikant)

LDA-Funktionalität	HCNN-Funktionalität	RLL-Funktionalität	***vorher*** (GB_TST, mit Hamming)	***nachher*** (GB_TST, mit Flat-Top)
Nein	Nein	Nein	21,90%	18,50%
Ja	Nein	Nein	21,50%	18,70%
Nein	**Ja**	Nein	14,40%	13,30%
Ja	**Ja**	Nein	22,00%	16,80%
Nein	Nein	**Ja**	10,50%	10,50%
Ja	Nein	**Ja**	10,20%	11,30%
Nein	**Ja**	**Ja**	8,30%	8,70%
Ja	**Ja**	**Ja**	7,50%	10,50%
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

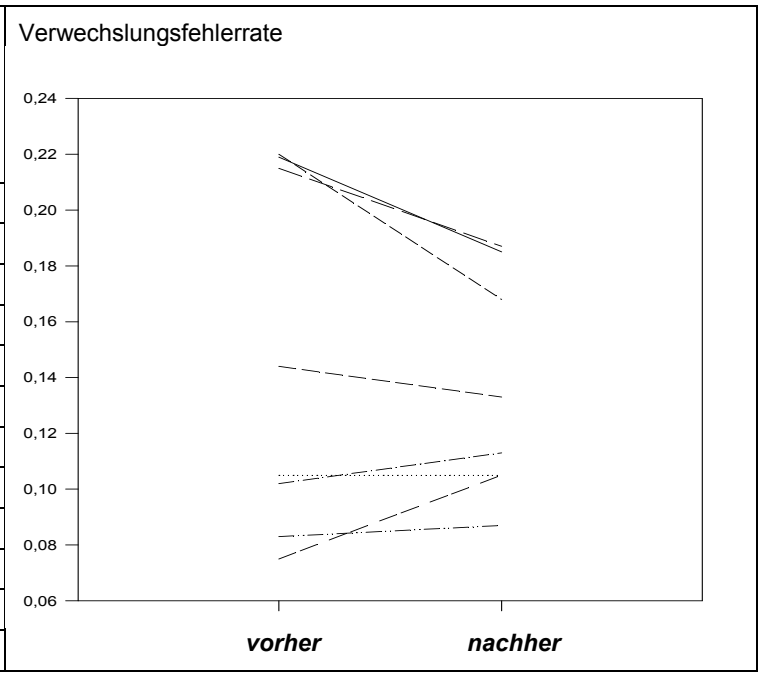

Tabelle 19: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei der Aktivierung des Flat-Top-Fensters anstelle des Hamming-Fensters.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,971)

Group	N	Missing	Median	25%	75%
Col 1	8	0	0,125	0,0877	0,218
Col 2	8	0	0,123	0,105	0,181

W= -10,000 T+ = 9,000 T-= -19,000
Z-Statistic (based on positive ranks) = -0,845
P(est.)= 0,447 P(exact)= 0,469

The change that occurred with the treatment is not great enough to exclude the possibility that it is due to chance (P = 0,469).

Abbildung 4-7: Wilcoxon-Vorzeichen-Rangtest zur Flat-Top-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0".

Abbildung 4-7 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur LDA-Hypothese, getestet an der englischen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „nicht signifikant", das heißt, die Nullhypothese wird angenommen.

4.1.6.4 RLL-Hypothese (englische Datenbank, signifikant)

LDA-Funktionalität	HCNN-Funktionalität	Fensterfunktion	***vorher*** GB_TST (ohne RLL)	***nachher*** GB_TST (mit RLL)
Nein	Nein	Rectangle	23,70%	10,50%
Nein	Nein	Hamming	21,90%	10,50%
Nein	Nein	Flat-Top	18,50%	10,50%
Nein	**Ja**	Rectangle	16,30%	6,60%
Nein	**Ja**	Hamming	14,40%	8,30%
Nein	**Ja**	Flat-Top	13,30%	8,70%
Ja	Nein	Rectangle	21,90%	9,50%
Ja	Nein	Hamming	21,50%	10,20%
Ja	Nein	Flat-Top	18,70%	11,30%
Ja	**Ja**	Rectangle	20,00%	9,30%
Ja	**Ja**	Hamming	22,00%	7,50%
Ja	**Ja**	Flat-Top	16,80%	10,50%

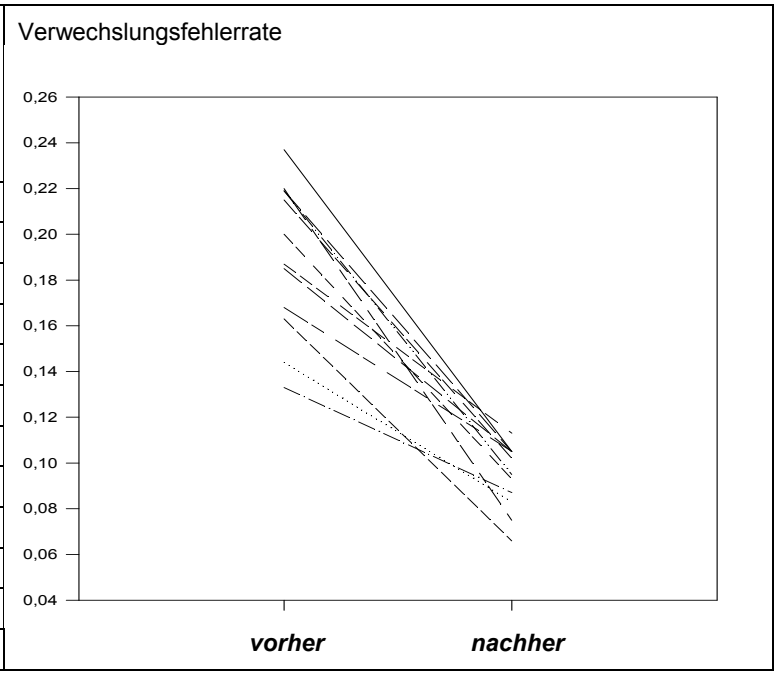

Tabelle 20: Gepaarte Auflistung der Verwechslungsfehlerraten der getesteten Varianten der Sprachsteuerung (Version 080630) sowie Vorher-Nachher-Line-Plot zur Visualisierung der Veränderung der Verwechslungsfehlerraten bei der Aktivierung der RLL-Funktionalität.

Wilcoxon Signed Rank Test

Normality Test (Shapiro-Wilk) Passed (P = 0,823)

Group	N	Missing	Median	25%	75%
vorher	12	0	0,194	0,164	0,219
nachher	12	0	0,0985	0,0840	0,105

W= -78,000 T+ = 0,000 T-= -78,000
Z-Statistic (based on positive ranks) = -3,059
P(est.)= 0,003 P(exact)= <0,001

The change that occurred with the treatment is greater than would be expected by chance; there is a statistically significant difference (P = <0,001).

Abbildung 4-8: Wilcoxon-Vorzeichen-Rangtest zur RLL-Hypothese (deutsche Sprachdatenbank). Bildschirmausgabe der Statistik-Software „SigmaPlot Version 11.0“.

Abbildung 4-8 zeigt das Ergebnis des Wilcoxon-Vorzeichen-Rangtests zur RLL-Hypothese, getestet an der englischen Sprachdatenbank. Das Signifikanzniveau wurde unter Berücksichtigung der Bonferroni-Korrektur auf 1,25% festgesetzt. Das Ergebnis ist „signifikant“, das heißt, die Nullhypothese wird zugunsten der Alternativhypothese verworfen.

4.2 Vorbereitung begleitender Evaluierung im praktischen Einsatz

Usability-Tests. Vor der Freigabe einer neuen Softwareversion empfiehlt es sich, verschiedene Usability-Tests durchzuführen: Zunächst werden Usability-Tests im Selbstversuch sowie mit Testpersonen ohne Sprechstörungen oder Sprachstörungen durchgeführt. Spezielle Tests betreffen die Robustheit bezüglich 50Hz-Netzbrumm bei sehr stiller Umgebung sowie die Robustheit bezüglich der Fahrgeräusche im Auto[98].

Akzeptanzsteigerung durch Optimierung bezüglich geringer Wartezeiten. Bei der Evaluierung durch Testpersonen scheint vor allem die Hilfsmittelakzeptanz ein kritischer Erfolgsfaktor: Der Ablauf der Sitzungen sollte nach Möglichkeit gut vorbereitet werden und ein hoher Reifegrad der Systemimplementierung ist notwendig, da oftmals auch kleinere Fehlfunktionen dazu führen können, dass das Interesse verloren geht. Der „Centroid Sequence Speech Recognizer“ nach Tabelle 8 auf Seite 163 benötigt beispielsweise im Vergleich zum komplizierteren „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ – jeweils mit aktiviertem Run-Length-Limited-Dynamic-Programming-Sprechtempokompensator – nur etwa ein Zehntel der Trainingszeit. Damit vergeht, von der Aufnahme der benutzerspezifischen Steuerbefehle bis zur Verwendung des Systems wesentlich weniger Zeit und die Geduld des Benutzers wird nicht unnötig strapaziert. Wenn die Rechenleistung der Personal Computer, wie in Zukunft zu erwarten ist, so weit angestiegen ist, dass die Dauer des Trainings nicht mehr zu einer relevanten Verringerung der Hilfsmittelakzeptanz führt, oder falls es sich um einen geduldigen Benutzer handelt, so empfiehlt es sich, den „Centroid Sequence Hidden Control Neural Network Speech Recognizer“ mit aktiviertem Run-Length-Limited-Sprechtempokompensator zu verwenden.

Akzeptanzsteigerung durch Qualitätssicherung der Stimmproben. Es empfiehlt sich, durch verschiedene konstruktive Maßnahmen sicherzustellen, dass bei der Aufnahme der benutzerspezifischen Steuerbefehle auch unter schwierigen Umgebungsbedingungen qualitativ hochwertige Schallmuster gewonnen werden. Die Erfahrung zeigt, dass man beinahe davon ausgehen kann, dass während der relativ langen Zeitspanne der Aufnahme der Steuerbefehle irgendwann das Telefon läutet oder die Tür zufällt oder andere Hintergrundgeräusche auftreten. Die im Rahmen dieser Arbeit realisierte Lösung besteht darin, dass mehr als zwei Schallmuster pro Steuerbefehl aufgenommen werden, beispielsweise fünf, und für jeden Steuerbefehl jenes Schallmuster wieder gelöscht wird, welches der Klasse, repräsentiert durch die jeweils anderen vier Schallmuster am unähnlichsten ist. Diese Vorgangsweise hat sich sehr bewährt. Weiters empfiehlt es sich, sicherzustellen, dass die zum Training verwendeten Schallmuster selbst richtig klassifiziert werden. Weiters wird automatisch sichergestellt, dass die zum Training ver-

[98] Diese Tests werden im Rahmen dieser Arbeit unter anderem im Kaisermühlentunnel in Wien bei einer Fahrgeschwindigkeit von circa 80 km/h durchgeführt.

wendeten Schallmuster nicht auf die Non-Keyword-Modelle abgebildet werden und umgekehrt die vorbereiteten Schallmuster für Hintergrundgeräusche nicht auf die Keyword-Modelle abgebildet werden. Nur wenn all diese Bedingungen erfüllt sind, wird der Spracherkenner für den Benutzer zur Verwendung freigeschaltet. Andernfalls werden automatisch und dialoggeführt bestimmte Neuaufnahmen der zum Training verwendeten Schallmuster eingeleitet.

4.2.1 Interviews mit Benutzern und Betreuungspersonen

Präzise Hypothesenformulierung und die Auswahl einer repräsentativen Personengruppe bilden das Fundament für quantitative und qualitative Methoden der Sozialforschung. Individuelle Erwartungen und Anforderungen an die Sprachsteuerung können sehr stark variieren.

Fragen zur Hilfsmittelakzeptanz. Interessant scheinen beispielsweise folgende Fragen an Benutzer beziehungsweise Betreuungspersonen: „Wie zufrieden sind sie mit dem System insgesamt? Bietet die Sprachsteuerung die Möglichkeit, das tägliche Leben zu erleichtern? Wenn nicht: Warum nicht? Wird die Sprachsteuerung als Hilfsmittel angenommen? Wenn nicht: Warum nicht? Wird die Aufnahme der individuellen Steuerbefehle als lästig empfunden? Wenn ja: Was könnte man verbessern? Wird der Umgang mit dem Mikrofon, beispielsweise das Aufsetzen des Kopfbügelmikrofons, als unpraktisch empfunden? Wenn ja: Was könnte man verbessern? Haben Sie Verbesserungsvorschläge, wie man die Hilfsmittelakzeptanz verbessern könnte?“

Fragen zur Bildschirmdarstellung der Benutzerstimme. Interessant scheinen beispielsweise folgende Fragen an Benutzer beziehungsweise Betreuungspersonen: „Wie häufig wird die Bildschirmdarstellung der Benutzerstimme verwendet? Wird die Bildschirmdarstellung der Benutzerstimme auch zum Zwecke des Stimmtrainings verwendet? Haben Sie Verbesserungsvorschläge bezüglich der Bildschirmdarstellung der Benutzerstimme?“

Fragen zur subjektiven Beurteilung der Fehlerraten. Interessant scheinen beispielsweise folgende Fragen an Benutzer beziehungsweise Betreuungspersonen: „Ist die Häufigkeit der Fehlauslösungen durch Wörter, bei denen es sich nicht um Steuerbefehle handelt, für Sie akzeptabel? Ist die Häufigkeit der Fehlauslösungen durch Geräusche für Sie akzeptabel? Ist die Häufigkeit des Umstands, dass Steuerbefehle zu keiner Reaktion führen, für Sie akzeptabel? Ist die Häufigkeit des Umstands, dass Steuerbefehle zu falschen Reaktionen führen, für Sie akzeptabel? Haben Sie weitere allgemeine oder benutzerspezifische Verbesserungsvorschläge?“

4.2.2 Automatische Systemprotokolle

Aus den automatischen Systemprotokollen kann entnommen werden, wann welcher Dialog gestartet und beendet wurde. Stimmt der Benutzer der Verwendung seiner Schallaufnahmen zu Forschungszwecken zu, so können bei Bedarf sämtliche Schallaufnahmen, die den durchgeführten Klassifikationsvorgängen zugrunde liegen, weiter analysiert werden. Dadurch können allgemeine Verbesserungen des Spracherkenners spezielle Verbesserungen des Spracherkenners für den jeweiligen Benutzer erreicht werden. Beispielsweise kann die Rückweisung bestimmter Hintergrundgeräusche oder die Zuverlässigkeit der Klassifikation der benutzerspezifischen Steuerbefehle verbessert werden. Aus den automatischen Systemprotokollen kann weiters entnommen werden, wann die Bildschirmdarstellung der Benutzerstimme genutzt wird. Die implementierten automatischen Systemprotokolle dienen der Vorbereitung zukünftiger Forschungsvorhaben.

4.2.3 Evaluierungsdialog

Dieser Dialog sollte nur eingesetzt werden, wenn es dem Benutzer ein wirkliches Bedürfnis ist, zur Verbesserung der Sprachsteuerung aktiv beizutragen. Die Schallmuster werden dabei teilweise vom Benutzer selbst per Rückfrage im Rahmen des Mensch-Maschine-Dialogs klassifiziert, wobei in Aussicht gestellt werden kann, Verbesserungen der Sprachsteuerung allgemein oder spezielle Anpassungen für den jeweiligen Benutzer später durchzuführen. Falls die Rückfragen der Sprachsteuerung doch in manchen Situationen als lästig empfunden werden, ist für den Benutzer selbstverständlich auch stets ein Dialog verfügbar, der die gleiche Funktionalität ohne Benutzerrückfragen zur Verfügung stellt. Die Häufigkeit der Rückfragen im Evaluierungsdialog ist weiters benutzerspezifisch einstellbar.

Annahmen bezüglich des Benutzers. Das Prinzip der Rückmeldung mittels eines Evaluierungsdialogs setzt die Kooperation des Benutzers voraus. Da die benötigen kognitiven Fähigkeiten trotz diesbezüglicher Optimierung des Dialogs nicht unbeträchtlich sind, ist der Einsatz dieses Verfahrens nicht immer möglich.

Annahmen bezüglich des Systems. Es wird angenommen, dass die Steuerbefehle für „Ja.“ und „Nein.“ stets richtig klassifiziert werden. Aufgrund der geringen Perplexität in dieser Dialogsituation sind hier allerdings wenige Probleme zu erwarten. Bei hohem Grad an Sprechstörungen können die Protokolleinträge, aus denen die Fehlerraten berechnet werden, durch einen menschlichen Hörer, anhand der Schallaufnahmen, korrigiert werden, wenn der Benutzer der Verwendung seiner Schallaufnahmen zu Forschungszwecken zustimmt. Eine geeignete Serviceschnittstelle zum Abhören der Schallmuster und zum Einsehen der Klassifikationsergebnisse sowie zum

Einsehen verschiedener, zum Klassifikationsvorgang gehörender Details steht zur Verfügung.

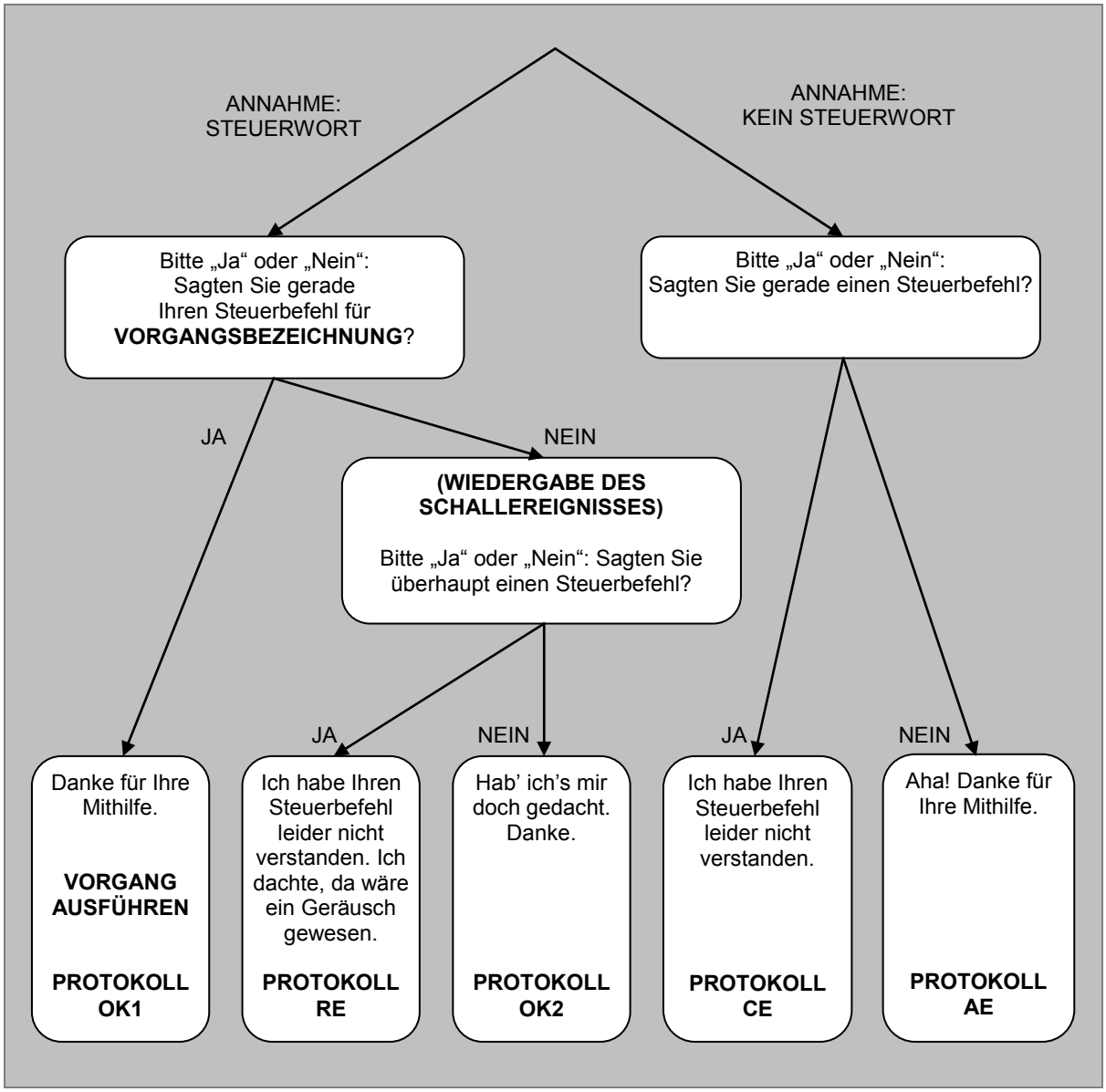

Abbildung 4-9: Beispiel eines Mensch-Maschine-Dialogs zur Systemevaluierung per Rückfragen an den oder die Benutzer.

Der implementierte Evaluierungsdialog dient der Vorbereitung zukünftiger Forschungsvorhaben. Es wurde bislang lediglich die Machbarkeit gezeigt, allerdings keine systematischen Auswertungen der Protokolleinträge vorgenommen.

Zusammenfassung und Ausblick

Basierend auf den Algorithmen der Diplomarbeit [Hickersberger1998] und basierend auf den Erfahrungen mit der Vorgängerversion [Loidolt1995], welche ein Sprachsteuergerät von Tschirk [Tschirk2001] verwendet, wird im Rahmen dieser Arbeit eine Sprachsteuerung für das AUTONOM-System [Zagler1997] [Panek2002] entwickelt, implementiert und mittels Sprachdatenbanktests (siehe Abschnitt 4.1 auf Seite 160) sowie mittels praktischer Versuche für den Einsatz vorbereitet. Das österreichische Patent Nummer 414060 wird erteilt [Hickersberger2004].

Evaluierung von Verbesserungsmaßnahmen. Die Hauptthese der Arbeit besteht in der Annahme, dass der beschriebene „Run-Length-Limited-Dynamic-Programming-Algorithmus“ (siehe Abschnitt 1.2.2 auf Seite 57) eine Verbesserung gegenüber dem üblichen „Viterbi-Dynamic-Programming-Algorithmus“ [Forney1973] (siehe Abschnitt 1.2.1 auf Seite 37) darstellt, sodass die Verwechslungsfehlerrate (siehe Abschnitt 1.3.1 auf Seite 61) reduziert wird. Bezüglich der Nebenthesen wird evaluiert, ob sich jeweils folgende Maßnahmen statistisch signifikant auf die Verwechslungsfehlerrate auswirken: Lineare Transformation der Merkmalsvektoren gemäß linearer Diskriminanzanalyse (siehe Abschnitt 4.1.5.1 auf Seite 171), Verwendung alternativer Fensterfunktionen bei der Kurzzeit-Fourier-Transformation (siehe Abschnitt 4.1.5.3 auf Seite 173), Verbesserung der Wortmodellierung mittels zusätzlicher „Prediction Models“ bestehend aus „Hidden Control Neural Networks“ (siehe Abschnitt 4.1.5.2 auf Seite 172). Die Verbesserungsmaßnahmen werden anhand zweier Sprachdatenbanken (siehe Abschnitt 4.1.1 auf Seite 161) getestet und jeweils mittels des Wilcoxon-Vorzeichen-Rangtests mit Bonferroni-Korrektur auf statistische Signifikanz geprüft. Die Verbesserungsmaßnahme der Hauptthese (siehe Abschnitt 4.1.5.4 auf Seite 174) stellt sich als einzige der vier geprüften Verbesserungsmaßnahmen als signifikant heraus.

Vorbereitung praktischer Studien. Auf Basis der Bedürfnisse von Menschen mit Behinderungen werden Anforderungen für das Systemdesign der Sprachsteuerung abgeleitet (siehe Abschnitt 2.3 auf Seite 139). Die entwickelte Sprachsteuerung ist explizit für die Benutzung durch Menschen mit Sprechstörungen vorbereitet: Die jeweiligen Steuerbefehle können individuell und frei gewählt werden, soweit das implementierte automatische Qualitätssicherungsanalyse-Modul (siehe Abschnitt 2.2.2 auf Seite 126 und Abschnitt 4.2 auf Seite 179) die gewählten Steuerbefehle als hinreichend befindet, um eine gute Klassifikationsqualität sicherzustellen). Es werden spezielle Dialoge mit Protokollierung (siehe Abschnitt 4.2.2 auf Seite 181), welche Rückfragen beinhalten, ob das System richtig verstanden hat (siehe Abschnitt 4.2.3 auf Seite 181), in Vorbereitung zukünftiger Forschungsvorhaben vorkonfiguriert.

Vorgeschlagene zukünftige Forschungsbemühungen (Anwendung). Die entwickelte Sprachsteuerung ermöglicht die Steuerung der Umgebung mittels des AUTONOM-Systems (siehe Abschnitt 3.3 auf Seite 156). Sie ist jedoch derzeit nur für Sprecher ohne Sprechstörungen getestet (siehe Abschnitt 4.1.1 auf Seite 161). Nichtsdestotrotz ist die Implementierung für Benutzer mit Sprechstörungen vorbereitet (siehe Abschnitt 2.2.2 auf Seite 126 und Abschnitt 4.2 auf Seite 179). Weitere Forschungsbemühungen wären zu empfehlen, um zu evaluieren, inwiefern die Sprachsteuerung für Benutzer mit Sprechstörungen nützlich ist, wobei zu erwarten ist, dass die alternativen und augmentativen Kommunikationsmöglichkeiten des AUTONOM-Systems von Vorteil sind. Die therapeutischen Hoffnungen beim Einsatz der Sprachsteuerung bestehen hauptsächlich wie folgt: Für Personen mit schweren motorischen Behinderungen ergibt sich die Möglichkeit, sich selbst als autonom handelnde Personen zu erleben [Zagler1997]. Für leicht kognitiv beeinträchtigte Personen ergeben sich Möglichkeiten zu selbstbestimmtem Lernen und bezüglich der Unterstützung von beruflichen und sozialen Integrationsprozessen. Es wird weiters angenommen, dass sich für Personen mit Sprechstörungen ein möglicher Sprechtrainingseffekt im Umgang mit der Sprachsteuerung ergibt.

Vorgeschlagene zukünftige Forschungsbemühungen (Algorithmen). State-of-the-Art-Spracherkenner verwenden derzeit „Deep Neural Networks“ beziehungsweise „Deep Belief Networks“ [Do2011] [Do2011a] [Dahl2012] [Jaitly2012] für die Sub-Word-Unit-Modelle von Hidden-Markov-Modell-Hybrid-Systemen (HMM/ANN, HMM/DBN). Üblicherweise wird dabei der „Viterbi-Dynamic-Programming-Algorithmus“ (siehe Abschnitt 1.2.1 auf Seite 37) als Bestandteil der Hidden-Markov-Modelle verwendet. Es wäre interessant, im Rahmen weiterer Forschungsbemühungen zu evaluieren, ob die Sub-Word-Unit-Modelle dieser Hybrid-Systeme zu weiteren Verbesserungen in Kombination mit dem vorgeschlagenen „Run-Length-Limited-Dynamic-Programming-Algorithmus“ führen. Die beschriebenen „Centroid-Hidden-Control-Neural-Network-Sub-Word-Unit-Modelle“ (siehe Abschnitt 1.2.1.5 auf Seite 50) können eingesetzt, weiterentwickelt beziehungsweise zum Vergleich herangezogen werden. Zwar sind diese Sub-Word-Unit-Modelle sehr rechenintensiv und es konnte im Rahmen dieser Arbeit nicht eindeutig gezeigt werden, dass ihre Aktivierung zu einer signifikanten Verbesserung führt, aber dennoch lässt eine Post-hoc-Analyse der Daten darauf schließen, dass das Verfahren vermutlich die Robustheit gegenüber Störungen an den zum Training verwendeten Sprachaufnahmen weiter erhöht, was für die praktische Anwendung durchaus vorteilhaft ist.

Anhang I. Neuronale Funktionennetze

Neuronale Funktionennetze spielen unter anderem eine wichtige Rolle als funktioneller Bestandteil von Sub-Word-Unit-Modellen. Im Folgenden werden die wichtigsten Optimierungsverfahren beschrieben, um die freien Parameter der Funktionennetze zu trainieren, das heißt, möglichst optimal festzulegen.

Das Training der freien Parameter besteht im näherungsweisen Lösen eines überbestimmten Gleichungssystems, welches allgemein in Gleichung 107 notiert ist, wobei j die Nummer der Gleichung sei.

$$\mathbf{f}\left(\mathbf{x}_j, \mathbf{w}\right) \approx \mathbf{d}_j \qquad \textbf{Gleichung 107}$$

Die freien Parameter $\mathbf{w}$, welche auch „Netzgewichte“ genannt werden, werden derart bestimmt, dass eine vorab definierte Fehlerfunktion ein Minimum annimmt. Bei der Fehlerfunktion handelt es sich um ein Skalarfeld $E\left(\mathbf{w}\right)$, also einer Funktion des Fehlers in Abhängigkeit der freien Parameter $\mathbf{w}$. Eine häufig benutzte Fehlerfunktion ist der Mean-Square-Fehler nachGleichung 108, es sind aber durchaus andere Fehlerfunktionen denkbar. Die natürliche Zahl J steht im Folgenden für die Anzahl der Gleichungen.

$$E\left(\mathbf{w}\right) = \frac{1}{J}\sum_{j=1}^{J}(\mathbf{y}_j - \mathbf{d}_j)^2 \qquad \textbf{Gleichung 108}$$

$$\mathbf{y}_j = \mathbf{f}\left(\mathbf{x}_j, \mathbf{w}\right) \qquad \textbf{Gleichung 109}$$

I.1 Random-Search-Optimierung

Bei der Random-Search-Optimierung [Artlieb2000] wird folgendermaßen vorgegangen: Zunächst wird die Fehlerfunktion $E\left(\mathbf{w}_0\right)$ an einer zufällig bestimmten Stelle $\mathbf{w}_0$ ausgewertet. Sodann wird eine zufällige Richtung $\boldsymbol{\delta}_0$ erzeugt und $E\left(\mathbf{w}_0 + \boldsymbol{\delta}_0\right)$ ausgewertet. Ist der Fehler an dieser Stelle größer, wird die Richtung verworfen und eine neue Richtung erzeugt. Ist eine Richtung gefunden, sodass sich der Fehler verringert, so wird die neue Stelle akzeptiert und der Vorgang wiederholt.

Eine Schwierigkeit bei der Random-Search-Optimierung besteht darin, dass auch lokale Minima die Suche zum Stillstand bringen können, wobei keine Information darüber vorliegt, ob es sich bei der gefundenen Stelle um das globale Minimum handelt. Eine Verbesserung des Verfahrens besteht beispielsweise in dessen mehrfacher Anwendung

mit verschiedenen Startpunkten. Dies ist insbesondere dann vorteilhaft, wenn Möglichkeiten der Parallelverarbeitung bestehen.

Eine weitere Möglichkeit der Verbesserung besteht darin, auch Richtungen, in denen sich der Fehler erhöht, unter gewissen Umständen, beispielsweise mit gewisser Wahrscheinlichkeit, zu akzeptieren, wobei die Stelle mit dem kleinsten bekannten Fehler als aktuelles Ergebnis gespeichert bleibt. Mittels dieser Vorgangsweise können Täler mit lokalen Minima dann grundsätzlich wieder verlassen werden.

I.2 Genetische Optimierung

Bei der Genetischen Optimierung wird ein Evolutionsprozess simuliert [Rojas1996] [Artlieb2000]:

1.) Zunächst wird innerhalb des Definitionsbereichs eine Menge von Startpunkten zufällig festgelegt. Die Punkte dieser sogenannten „Population“ werden üblicherweise mittels Bitketten repräsentiert, wobei die Bitkette im einfachsten Fall aus aneinandergereihten Binärdarstellungen der freien Parameter besteht.

2.) Die zu minimierende Fehlerfunktion $E(\mathbf{w})$ wird für die aktuelle „Population“ ausgewertet und die Punkte gemäß ihrer Qualität sortiert. Mittels eines „Mutationsoperators“ werden die Punkte der Population verändert. Dies geschieht zum Beispiel, indem Bits verändert werden. Der Zweck der Mutation besteht in der Erweiterung des Suchbereichs.

3.) Mittels „Vererbungsoperatoren“ werden neue Punkte aus den Punkten der aktuellen „Population“ erzeugt. Die Erzeugung kann zum Beispiel dadurch erfolgen, dass der Anfang einer Bitkette mit dem Ende einer anderen Bitkette zu einer neuen Bitkette verbunden wird.

4.) Weiter bei 2.)

Die Punkte der Population nähern sich mit der Zeit den lokalen Minima beziehungsweise dem globalen Minimum der Funktion. Der Definitionsbereich der zu optimierenden Funktion wird an vielen Stellen gleichzeitig berücksichtigt. Gegenüber anderen Algorithmen ergibt sich der Vorteil, dass größere Robustheit bezüglich eines Hängenbleibens des Optimierungsvorgangs in lokalen Minima besteht.

I.3 Backpropagation-Optimierung

Bei diesem Optimierungsverfahren [Widrow1990] [Rojas1996] [Artlieb2000] wird der Gradient der zu optimierenden Funktion verwendet. Der Nachteil des Verfahrens besteht daher darin, dass die zu optimierende Funktion differenzierbar sein muss. Backpropagation-Optimierung wird erst seit der Mitte der Achtzigerjahre angewendet, obwohl das Konzept relativ naheliegend ist. Der Grund dafür ist wohl darin zu suchen, dass vom Anfang der Sechzigerjahre bis zur Mitte der Neunzigerjahre vorigen Jahrhunderts hauptsächlich neuronale Netze implementiert wurden, die Signum-Funktionen anstelle der heute üblichen Sigmoid-Funktionen verwendeten. Für diese Neuronalen Netze ist die heutige Backpropagation-Optimierung nicht geeignet ist, da die Signum-Funktion an der Stelle Null nicht differenzierbar ist.

Im Allgemeinen wird die Funktion $E(\mathbf{w})$ aus diskreten Berechnungselementen zusammengesetzt. Man spricht man von einem Funktionennetz für $E(\mathbf{w})$, dem sogenannten „Vorwärtsnetz“. Zur numerischen Berechnung des Gradienten $\mathbf{G}_k$ dieses Skalarfeldes an der Stelle $\mathbf{w}_k$ nach Gleichung 110, wird vorab ein sogenanntes „Rückwärtsnetz“ implementiert, das ebenfalls aus diskreten Berechnungselementen zusammengesetzt ist[99]. Die Anzahl der Komponenten des Vektors $\mathbf{G}_k$ entspricht der Anzahl der Netzgewichte. Der Index k wählt eine bestimmte Stelle, beispielsweise im Rahmen der iterativen Optimierung.

$$\mathbf{G}_k = \left(\vec{\nabla} E(\mathbf{w})\right)\Big|_{\mathbf{w}=\mathbf{w}_k} \qquad \textbf{Gleichung 110}$$

Die Bestimmung des Rückwertsnetzes geschieht mittels Methoden der Differenzialrechnung. Bemerkenswert ist hierbei ein Verfahren, bei dem jedes Vorwärtsberechnungselement durch ein zugehöriges Rückwärtsberechnungselement anhand einer Korrespondenztabelle ersetzt wird und das Rückwärtsnetz direkt hingezeichnet werden kann[Rojas1996]. Das Verfahren basiert freilich bei näherer Betrachtung auf der Kettenregel der Differenzialrechnung und das gezeichnete Rückwärtsnetz besteht aus Berechnungselementen, die Jacobi-Matrizen berechnen, die mittels zusätzlicher Berechnungselemente multipliziert werden. Da das Rückwärtsnetz die Jakobi-Matrizen in an bestimmten Stellen ausgewerteter Form benötigt, ist das ursprüngliche Vorwärtsnetzwerk zur numerischen Berechnung des Gradienten zusätzlich notwendig. Vor dem Backpropagation-Schritt wird im Allgemeinen eine Feed-Forward-Berechnung durchgeführt.

---

[99] Wie im Folgenden dargelegt wird, leitet sich die Bezeichnung „Backpropagation-Optimierung“ in Anspielung auf die Auswertung des Rückwärtsnetzwerks her.

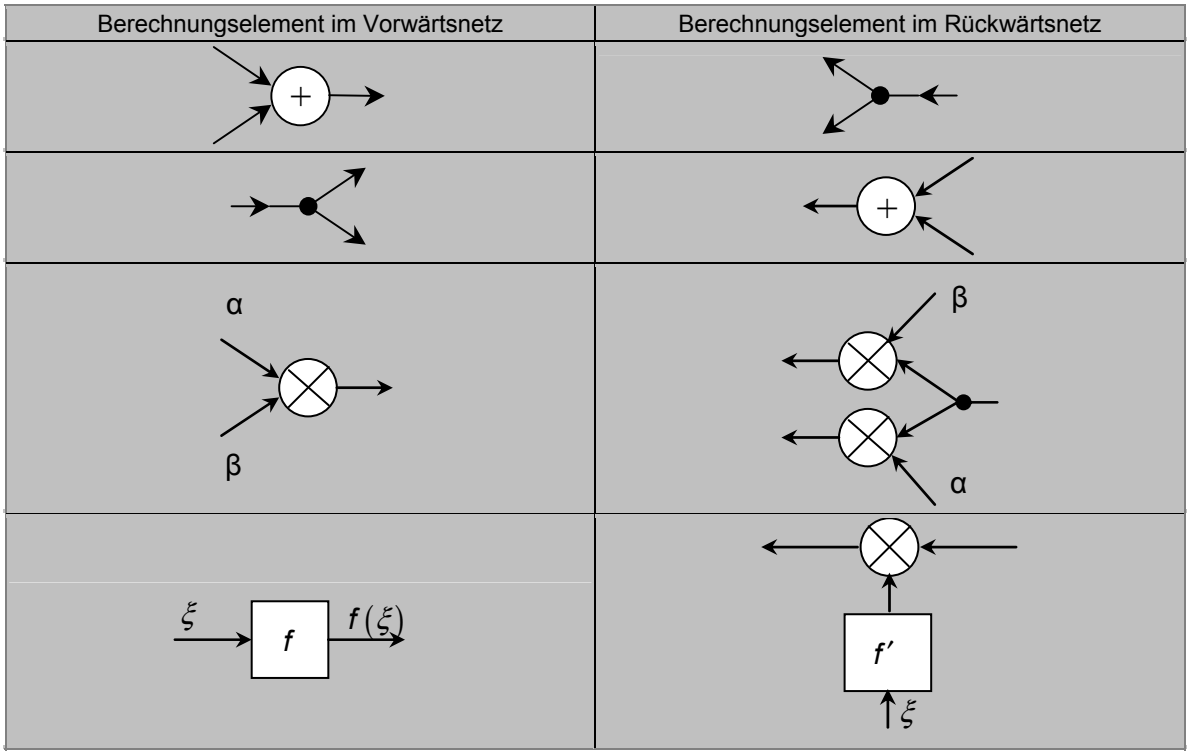

Abbildung I-1: Beispiele aus der Korrespondenztabelle zur Gradientenbestimmung [Rojas1996]. Die Zahlenwerte α, β, ξ als Eingänge der Rückwärtsberechnungselemente werden mittels des Vorwärtsnetzes berechnet, das damit Teil des Rückwärtsnetzes wird.

Wertet man zunächst das Vorwärtsnetz an der Stelle $\mathbf{w}_k$ und dann das Rückwärtsnetz aus, so erhält man den gesuchten Gradienten an der Stelle $\mathbf{w}_k$. Man spricht dabei vom sogenannten „Forward-Calculation-Schritt" und vom sogenannten „Backpropagation-Schritt". Die im „Backpropagation-Schritt" benötigte Information aus dem Vorwärtsnetzwerk, also zum Beispiel die Stelle ξ an der die Funktion *f* in Abbildung I-1 ausgewertet wird, wird zunächst im „Forward-Calculation-Schritt" berechnet. Nach der Berechnung des Gradienten erfolgt die Veränderung der Netzgewichte im sogenannten „Weight-Update-Schritt".

Die Backpropagation-Optimierung besteht also, nachdem ein Startpunkt $\mathbf{w}_0$ für die freien Parameter festgelegt ist, was üblicherweise mittels eines Zufallszahlen-generators geschieht, in der iterativen Ausführung folgender drei Schritte: Forward-Calculation-Schritt, Backpropagation-Schritt und Weight-Update-Schritt.

Man spricht von einem linearen Gradientenverfahren, wenn der Korrekturvektor $\mathbf{w}_{k+1} - \mathbf{w}_k$ aus dem negativen Gradienten durch Multiplikation mit einem konstanten Faktor μ berechnet wird. Der Faktor μ beeinflusst maßgeblich die Schrittweite.

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \cdot \left(-\mathbf{G}_k\right) \qquad \textbf{Gleichung 111}$$

I.3.1 Beispiel: Adaptiver Linearkombinierer

Abbildung I-2 zeigt ein Neuron ohne Sigmoid-Funktion. Dabei handelt es sich sozusagen um einen adaptiven Linearkombinierer. Dieser bildet die theoretische Grundlage des Neuronen-Typs „Adaline“ [Widrow1990]. Der Wert y_j wird durch Linearkombination der Eingangswerte $\mathbf{x}_j$ berechnet. Gleichung 112 zeigt, dass die Netzgewichte $\mathbf{w}_j$ als Linearkoeffizienten in die Berechnung einfließen.

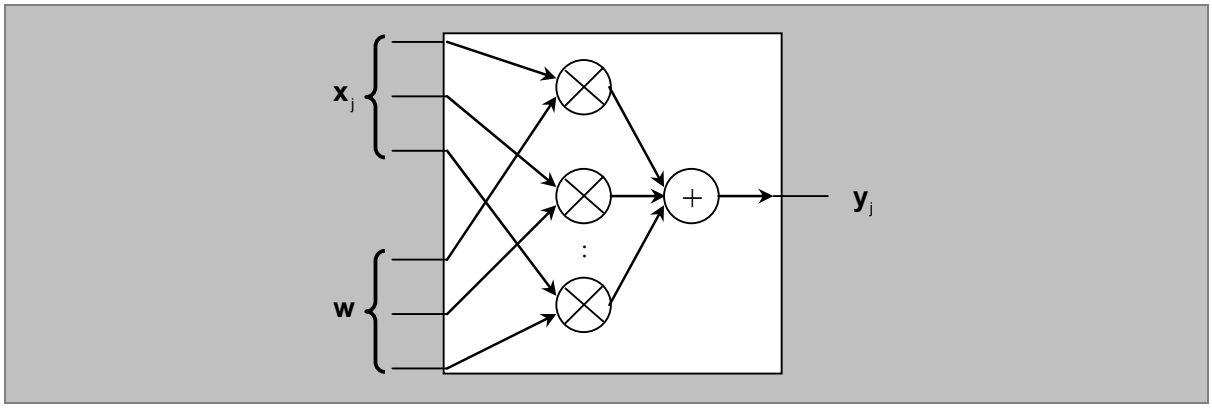

Abbildung I-2: Adaptiver Linearkombinierer.

Gleichungen: $y_j = \mathbf{x}_j^T \cdot \mathbf{w}_j$ Matrix-Schreibweise: $\mathbf{y} = \mathbf{X}\mathbf{w}$ **Gleichung 112**

Wird der Mean-Square-Fehler nach Gleichung 108 eingezeichnet, so ergibt sich das Vorwärtsnetz für $E(\mathbf{w})$, wie in Abbildung I-3 dargestellt:

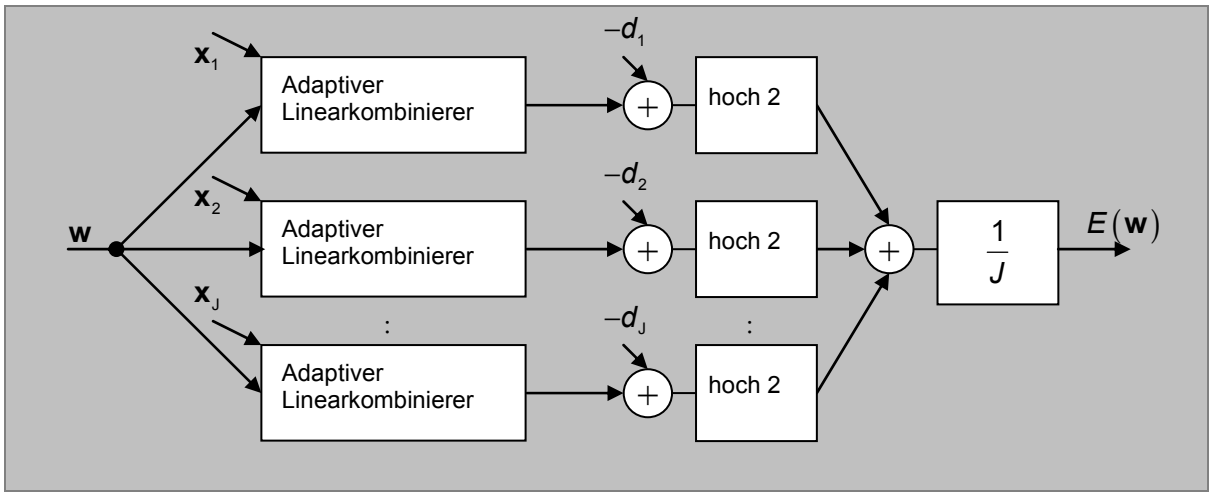

Abbildung I-3: Vorwärtsnetz des adaptiven Linearkombinierers bei quadrierter euklidischer Distanz als Fehlermaß.

Es existiert ein einziges globales Minimum der Mean-Square-Fehlerfunktion des Linearkombinierers, nämlich der optimale Gewichtsvektors $\mathbf{w}_{opt}$, welcher auch „Wiener Vektor“ genannt wird. Zur Bestimmung des „Wiener Vektors“ kann beispielsweise eine Backpropagation-Optimierung verwendet werden: Dazu ist das Funktionennetz des adaptiven Linearkombinierers nach Abbildung I-2 auf Seite 189 in das Vorwärtsnetz nach Abbildung I-3 einzusetzen und das Vorwärtsnetz nach Abbildung I-1 auf Seite 188 in ein Rückwärtsnetz zu transformieren [Rojas1996]. Die Netzgewichte, welchen den „Wiener Vektor“ bilden, werden sodann mittels Backpropagation-Optimierung bestimmt. Eine andere Möglichkeit besteht darin, den Wiener Vektor analytisch zu berechnen, indem der Gradient der Mean-Square-Fehlerfunktion Null gesetzt wird: Für die zu minimierende Größe $J \cdot E(\mathbf{w})$ gilt Gleichung 113.

$$J \cdot E(\mathbf{w}) = (\mathbf{y}-\mathbf{d})^{T} \cdot (\mathbf{y}-\mathbf{d}) = (\mathbf{X}\mathbf{w}-\mathbf{d})^{T} \cdot (\mathbf{X}\mathbf{w}-\mathbf{d}) \qquad \textbf{Gleichung 113}$$

Gradientenbildung nach $\mathbf{w}$ und Nullsetzen führt nach [Widrow1990] auf die Wiener-Hopf-Gleichung. Der Wiener-Vektor folgt sodann zu:

$$\mathbf{w}_{opt} = (\mathbf{X}^{T}\mathbf{X})^{-1}\mathbf{X}^{T}\mathbf{d} \qquad \textbf{Gleichung 114}$$

I.3.2 Beispiel: Adaptiver Matrix-Multiplizierer

Der adaptive Matrix-Multiplizierer kann als Menge adaptiver Linearkombinierer aufgefasst werden. Ausgangspunkt ist Gleichung 115 in Analogie zurGleichung 112.

$$\text{Gleichungen: } \mathbf{y}_{j} = \mathbf{x}_{j}\mathbf{W} \quad \text{Matrix-Schreibweise: } \mathbf{Y} = \mathbf{X}\mathbf{W} \qquad \textbf{Gleichung 115}$$

Wenn die J Vektoren voneinander linear unabhängig sind, existiert die Pseudoinverse $\mathbf{X}^{+}$. Das heißt, $\mathbf{X}^{+}\mathbf{X} = \mathbf{I}$, wobei $\mathbf{I}$ die Einheitsmatrix mit J Zeilen und J Spalten sei. Es gilt dann für die optimale Matrix $\mathbf{W}_{opt}$:

$$\mathbf{W}_{opt} = \mathbf{X}^{+}\mathbf{D} \qquad \textbf{Gleichung 116}$$

Auch in diesem Fall kann Backpropagation-Optimierung eingesetzt werden, um die optimale Matrix $\mathbf{W}_{opt}$ zu finden. So kann also mittels Backpropagation-Optimierung die Pseudoinverse einer Matrix berechnet werden, wenn als erwünschter Ausgang des Matrix-Multiplizierers die Zeilen der Einheitsmatrix $\mathbf{I}$ herangezogen werden, also $\mathbf{D} = \mathbf{I}$ gewählt wird. Dann entspricht $\mathbf{W}_{opt} = \mathbf{X}^{+}$. Wenn die Matrix $\mathbf{X}$ invertierbar ist, entspricht die Pseudoinverse $\mathbf{X}^{+}$ der invertierten Matrix $\mathbf{X}^{-1}$. Derart kann Backpropagation-Optimierung zur Matrixinversion herangezogen werden.

I.3.3 Beispiel: Adaptives Kolmogorov-Netz

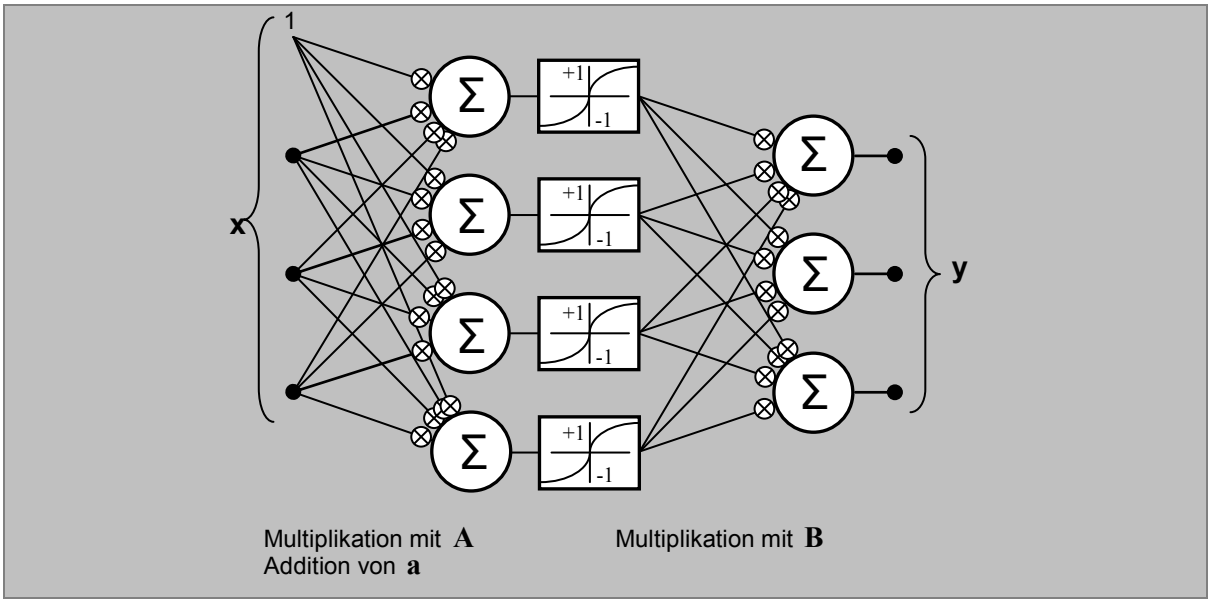

Abbildung I-4: Kolmogorov-Netz: Das einfachst-mögliche Multi-Layer-Perceptron-Netz, welches bei theoretischer, unbegrenzter Anzahl an Hidden-Layer-Neuronen der Klasse der „Universellen Approximatoren“ angehört.

Das Kolmogorov-Netz ist ein spezielles Multi-Layer-Perceptron-Netz, wie es Abbildung I-4 darstellt. Die Multiplikation mit einem Netzgewicht wird mittels des Symbols $\otimes$ dargestellt. Der zweite Faktor, also das Netzgewicht, ist in Abbildung I-4 unterdrückt. Das implementierte Vektorfeld hängt von den aktuellen Netzgewichten ab. Jedem Netzgewichtsvektor $\mathbf{w}$ ist also ein Vektorfeld $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{w})$ zugeordnet.

Die Abbildung $\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{w})$ zerfällt also in Teilabbildungen gemäß Gleichung 117: Der Vektor $\mathbf{x}$ wird mit der Matrix $\mathbf{A}$ multipliziert. Der Vektor $\mathbf{a}$ wird addiert. Dieser Vektor wird mittels des Vektorfelds **sgm** transformiert. Mittels neuerlicher Matrixmultiplikation mit $\mathbf{B}$ wird der Vektor $\mathbf{y}$ berechnet. Die Matrizen $\mathbf{A}$ und $\mathbf{B}$ und der Vektor $\mathbf{a}$ werden zum Vektor der Netzgewichte $\mathbf{w}$ zusammengestellt.

$$\mathbf{y} = \mathbf{B} \cdot \mathbf{sgm}(\mathbf{a} + \mathbf{A} \cdot \mathbf{x}) = \mathbf{f}(\mathbf{x}, \mathbf{w}) \qquad \textbf{Gleichung 117}$$

Das Vektorfeld **sgm** in Gleichung 117 ist folgendermaßen definiert: Jede Komponente des Argumentvektors wird einer Sigmoid-Funktion unterworfen, welche laut Definition die Bedingungen nach Gleichung 118 erfüllen und an jeder Stelle differenzierbar sein muss.

$$\lim_{\xi \to +\infty} sgm(\xi) = +1, \quad \lim_{\xi \to -\infty} sgm(\xi) = -1 \qquad \textbf{Gleichung 118}$$

Abbildung I-5 zeigt typische Sigmoid-Funktionen. Die Berechnung von $\xi/(1+|\xi|)$ kann zeitsparender implementiert werden, als die Berechnung von $2/(1+e^{-2\xi})-1$, die aber wiederum den geringen Vorteil einer etwas einfacheren Implementierung der Backpropagation-Optimierung hat [Widrow1990]. Das liegt daran, dass sich die Ableitung dieser Funktion nach Gleichung 120 durch die Funktion selbst ausdrücken lässt, wodurch das Rückwärtsnetzwerk optimiert werden kann.

$$sgm(\xi) = \frac{2}{1+e^{-2\xi}} - 1 \qquad \textbf{Gleichung 119}$$

$$\frac{d}{d\xi} sgm(\xi) = 1 - sgm^2(\xi) \qquad \textbf{Gleichung 120}$$

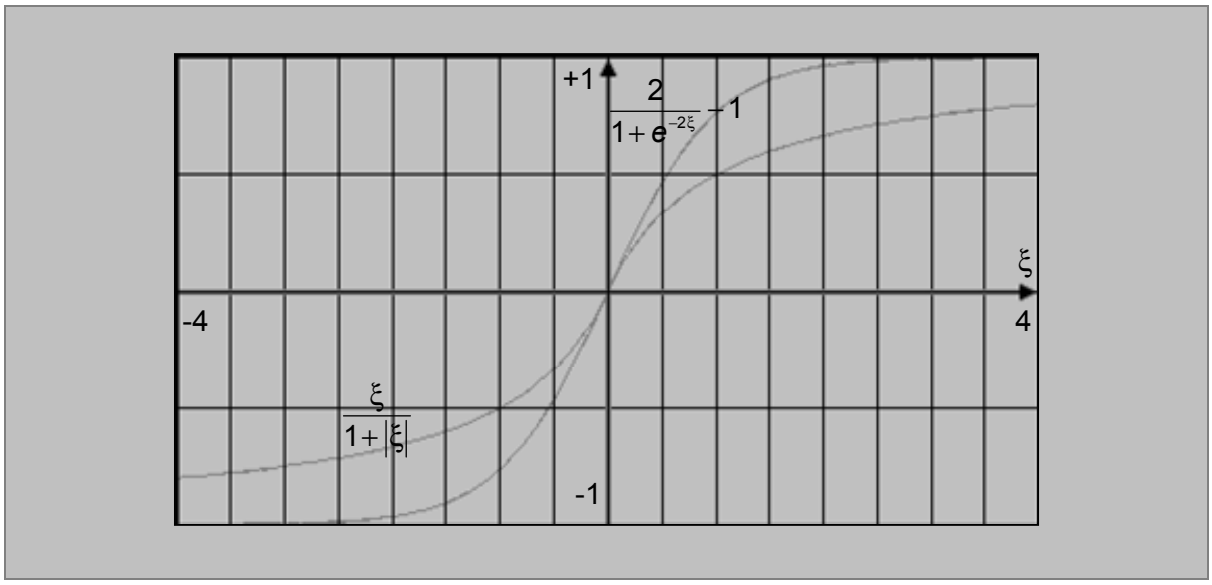

Abbildung I-5: Zwei übliche Sigmoid-Funktionen.

Eine interessante Eigenschaft der Kolmogorov-Netze ist nun, dass ein solches Netz, welches nur einen einzigen „Hidden Layer“ aber Hidden-Layer-Neuronen in theoretischer, unbegrenzter Anzahl enthält, stetige Funktionen prinzipiell beliebig genau annähern kann [Cybenko1989]. In der Praxis erfolgt die Approximation durch Superposition lediglich endlich vieler stetiger Sigmoid-Funktionen.

Die Aussage des sogenannten „Kolmogorov-Theorems“ beschränkt sich auf die Existenz eines Netzgewichtsvektors. Die wichtige Frage, bezüglich der praktischen Machbarkeit bleibt jedoch unbeantwortet: Wie viele Summanden, das heißt wie viele Hidden-Layer-Neuronen, werden benötigt, um eine Approximation mit gegebener Qualität erreichen zu können? In der Praxis ist wohl die Anzahl der Hidden-Layer-Neuronen als auch der Wertebereich der Netzgewichte beschränkt [Cybenko1989].

I.4 Online-Backpropagation-Optimierung

Die Berechnung des oftmals implementierten Mean-Square-Fehlers führt bei großer Anzahl J von Gleichungen $\mathbf{f}(\mathbf{x}_j, \mathbf{w}) = \mathbf{d}_j$, also einer großen Trainingsdatenbank, auf ein sehr großes Funktionennetz für $E(\mathbf{w}) = MSE(\mathbf{w})$. Vielfach wird die Backpropagation-Optimierung daher nur näherungsweise implementiert. Man nennt diese folgende Variante „Online-Backpropagation-Optimierung“ [Artlieb2000]:

Der Mean-Square-Fehler ist der Mittelwert der quadratischen Fehler der einzelnen Funktionennetze $\mathbf{y}_j = \mathbf{f}(\mathbf{x}_j, \mathbf{w})$, von denen jedes einem Trainingspaar $(\mathbf{x}_j, \mathbf{d}_j)$ entspricht. Anstatt den Mittelwert zu minimieren, werden in vielen Implementierungen die einzelnen Summanden, das heißt, die quadratischen Fehler nach Gleichung 121, nacheinander minimiert.

$$SE_j(\mathbf{w}) = (\mathbf{y}_j - \mathbf{d}_j)^2 \qquad \textbf{Gleichung 121}$$

Für das jeweilige Vorwärtsnetz eines Trainingspaares wird ein Forward-Calculation-Schritt durchgeführt und für das zugehörige Rückwärtsnetz ein Backpropagation-Schritt durchgeführt. Sodann folgt bereits ein Weight-Update-Schritt und des wird zum nächsten Trainingspaar übergegangen. Sind alle Trainingspaare an der Reihe gewesen, so wird wieder beim Ersten fortgesetzt. Einen Durchgang, wobei jedes Trainingspaar einmal an die Reihe kommt, nennt man auch „Pass“. Die „Online-Backpropagation-Optimierung“ verläuft gemäß Abbildung I-6.

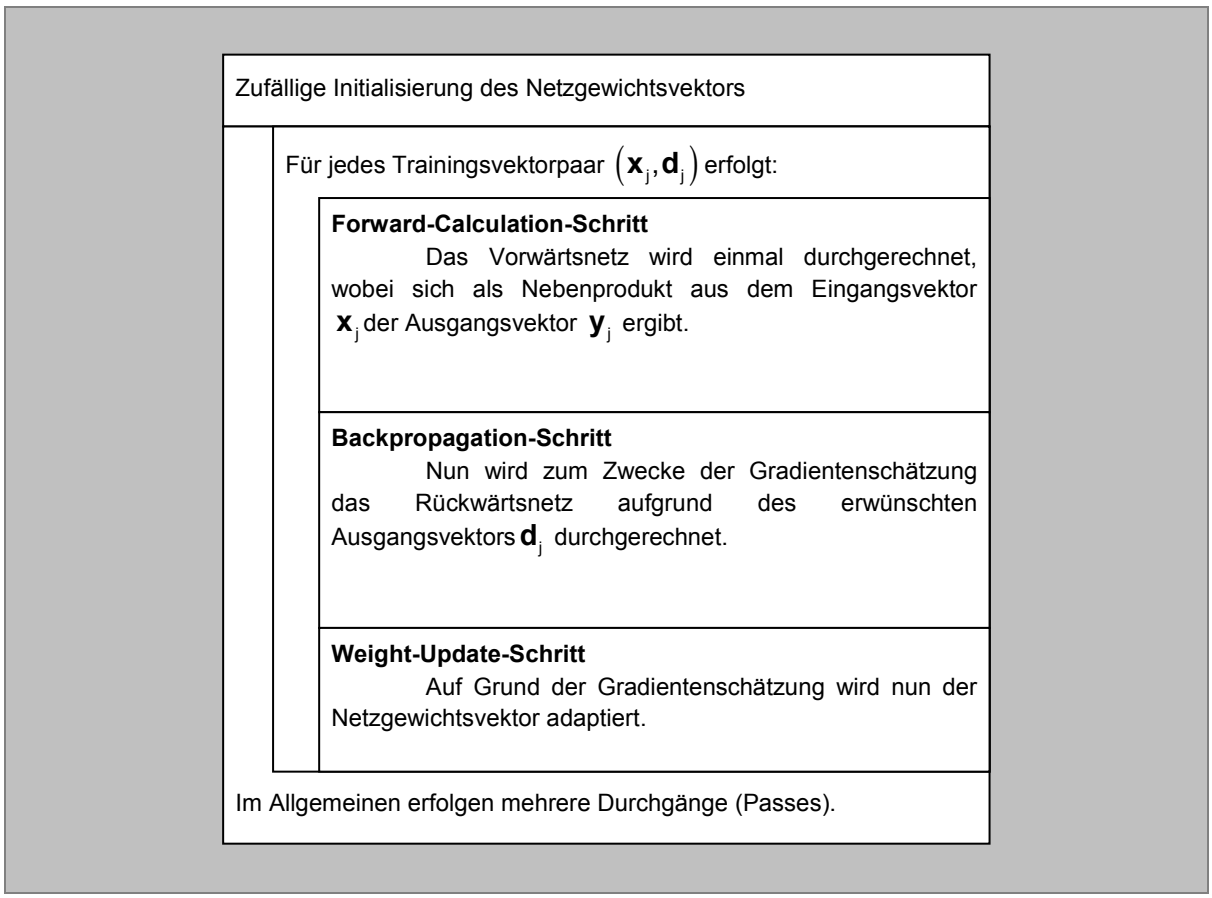

Abbildung I-6: Ablauf der Online-Backpropagation-Optimierung.

Die Gradienten-Schätzungen zur Optimierung der Netzgewichte hängen also vom aktuellen Trainingspaar ab und würden sogar ständig wechseln, wenn die Stelle $\mathbf{w}$ festgehalten würde, also kein Weight-Update-Schritt stattfinden würde. Für jeden Weight-Update-Schritt k ist also eine zum entsprechenden Trainingspaar gehörende Schätzung des Gradienten $\hat{\mathbf{G}}_k$ nach Gleichung 122 relevant.

$$\hat{\mathbf{G}}_k = \left(\vec{\nabla} \cdot \mathrm{SE}_k(\mathbf{w})\right)\Big|_{\mathbf{w}=\mathbf{w}_k} \qquad \textbf{Gleichung 122}$$

In diesem Falle wäre der Mittelwert der wechselnden Schätzungen – gemittelt über einen „Pass“ – gleich dem Gradienten nach Gleichung 110. Mittels kleiner Werte für die Konstante μ in Gleichung 111 wird das Festhalten der Stelle $\mathbf{w}$ näherungsweise realisiert, sodass der Mean-Square-Fehler näherungsweise optimiert wird.

I.5 Momentum-Backpropagation-Optimierung

Bei der Momentum-Backpropagation-Optimierung [Artlieb2000] handelt es sich um eine Variante[100] der Backpropagation-Optimierung. Gleichung 111 wird modifiziert, indem ein Momentum-Term addiert wird [Rumelhart1986]. Es erfolgt, wie in Abbildung I-7 dargestellt, eine IIR-Filterung erster Ordnung, wobei aus den Gradienten die Änderungen des Netzgewichtsvektors bestimmt werden.

$$\Delta \mathbf{w}_k = \mu \cdot \left(-\mathbf{G}_k\right) + \alpha \cdot \Delta \mathbf{w}_{k-1}$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \Delta \mathbf{w}_k \qquad \textbf{Gleichung 123}$$

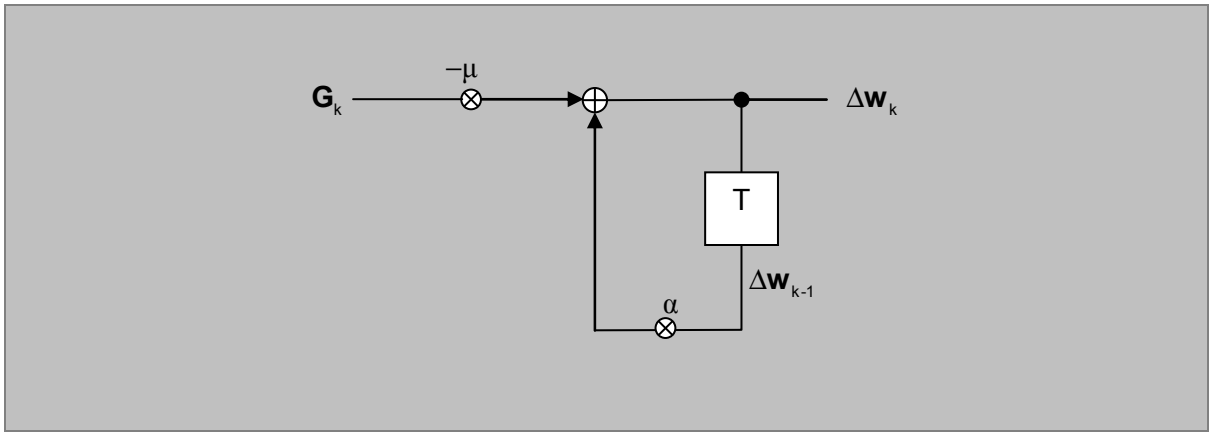

Abbildung I-7: IIR-Filterung im Rahmen der Momentum-Backpropagation-Optimierung.

[100] Es existieren eine Reihe von Varianten der Backpropagation-Optimierung, wie beispielsweise die Conjugate-Gradient-Backpropagation-Optimierung und die Quasi-Newton-Backpropagation-Optimierung. Die Bedingungen, unter denen sich der erhöhte Aufwand lohnt, sind in der Praxis kaum abzuschätzen [Widrow1990].

Anhang II. Kurzzeit-Fourier-Transformation

Die diskrete Kurzzeit-Fourier-Transformation [Schröder1962] [Allen1977] [Portnoff1981] ist ein bei Abtastsystemen weitverbreitetes Verfahren zur Analyse von Signalen, welche vorteilhafterweise quasistationär sind. Es spielt unter anderem eine wichtige Rolle als funktioneller Bestandteil verschiedener Algorithmen zur Merkmalsextraktion aus Sprachsignalen. In diesem Anhang wird der bekannte Zusammenhang mit der kontinuierlichen Fourier-Transformation herausgearbeitet.

Die kontinuierliche Fourier-Transformation. Es wird im Folgenden von der Definitionsgleichung der kontinuierlichen Fourier-Transformation in symmetrischer Schreibweise nach Gleichung 124 sowie der Rücktransformation nach Gleichung 125 ausgegangen:

$$S(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-j2\pi ft} dt \qquad \textbf{Gleichung 124}$$

Die Rücktransformation erfolgt ebenfalls durch Bildung eines verallgemeinerten Skalarprodukts:

$$s(t) = \int_{-\infty}^{\infty} S(f) \cdot e^{j2\pi ft} df \qquad \textbf{Gleichung 125}$$

Wird ein Signal $s(t)$ abgetastet, so geht dabei im allgemeinen Fall Information verloren. Diese wird durch Annahmen über den Signalverlauf ersetzt, wie im Folgenden dargestellt wird.

Der Signalverlauf zwischen den Stützstellen. Da es in der Natur eines Abtastsystems liegt, dass der Signalverlauf nur zu diskreten Zeitpunkten bekannt ist[101], müssen Annahmen über den Signalverlauf zwischen den Abtastzeitpunkten getroffen werden. Man könnte etwa auf die Idee kommen, den Signalverlauf zwischen den Abtastzeitpunkten mittels Geradenstücken zu interpolieren. Eine andere Möglichkeit besteht in der Annahme einer Treppenfunktion[102].

$$i \cdot \Delta t = t \qquad \textbf{Gleichung 126}$$

Dirac-Impulse sind im Folgenden so definiert, dass als Argument die dimensionsbehaftete Zeitgröße verwendet wird und das Gewicht als dimensionsloser Skalar zu eins normiert ist.

---

[101] Die bei realen Analog-Digital-Konvertern auftretenden zeitlichen Mittelungsvorgänge fügen sich auf natürliche Weise in die Theorie ein.

[102] Ein solches Signal wäre etwa direkt hinter einer Abtast-Halte-Schaltung denkbar.

$$\int_{-\infty}^{\infty} \delta(t)\,dt = 1 \qquad \textbf{Gleichung 127}$$

Die einfachsten Gleichungen ergeben sich im Folgenden, wenn der Signalverlauf zwischen den Abtastzeitpunkten einer Si-Interpolation genügt[103]. Sowohl ein als Treppenfunktion interpoliertes Signal, als auch ein mittels Si-Funktion interpoliertes Signal, lässt sich jedenfalls nach Gleichung 128 darstellen: Die Funktion $adc(t)$ repräsentiert die gewünschte Art der Interpolation.

$$s(t) = adc(t) \otimes \sum_{i=-\infty}^{\infty} \left[s_i \cdot \delta(t - i \cdot \Delta t) \right] \qquad \textbf{Gleichung 128}$$

Umgekehrt kann diese Gleichung zur Rekonstruktion des kontinuierlichen Signals $s(t)$ aus den Abtastwerten s_i verwendet werden. Dabei wird oftmals eine Interpolation mittels Si-Funktion nach Gleichung 129 verwendet, was der impliziten Annahme entspricht, dass das Signal ideal bandbegrenzt sei.

$$adc(t) = \frac{\sin\left(\frac{\pi t}{\Delta t}\right)}{\frac{\pi t}{\Delta t}} \qquad \textbf{Gleichung 129}$$

Aus Gleichung 128 und Gleichung 129 ergibt sich Gleichung 130. Dabei handelt es sich um die Whittaker-Shannon-Interpolationsformel, die auf Arbeiten von Whittaker aus dem Jahre 1915 und auf Arbeiten von Shannon aus dem Jahre 1948 zurückgeht.

$$s(t) = \sum_{i=-\infty}^{\infty} \left[s_i \cdot Si\left(\frac{\pi t}{\Delta t} - \pi i\right) \right] \qquad \textbf{Gleichung 130}$$

Nullwerte außerhalb des Analyseintervalls. Aus Gründen der praktischen Implementierbarkeit stellt sich zudem die Notwendigkeit der Annahme heraus, dass das Signal gleich null außerhalb eines endlichen Intervalls wäre. Da diese Annahme oft nicht ausreichend zutreffend ist, kann man sich mit der Vorstellung behelfen, nicht das eigentliche Signal sondern das Signal multipliziert mit einer Fensterfunktion analysieren zu wollen. Die bei der Analyse entstehende Abweichung lässt sich abschätzen, wenn man sich vergegenwärtigt, dass es sich sodann beim Resultat um die Faltung des

[103] Ein solches Signal würde theoretisch direkt hinter einer idealen Bandbegrenzung auftreten.

interessierenden Spektrums mit dem Spektrum der Fensterfunktion handelt. Man setzt jedenfalls den kontinuierlichen Signalverlauf mittels N diskreter Abtastwerte an[104]:

$$s(t) = \sum_{i=0}^{N-1} \left[s_i \cdot \delta(t - i \cdot \Delta t) \right] \otimes adc(t) \qquad \textbf{Gleichung 131}$$

Transformation mittels Summation. Setzt man Gleichung 131 in Gleichung 124 ein, so kommt man ohne Integration aus, denn es ergibt sich:

$$\begin{aligned} S(f) &= \int_{-\infty}^{\infty} \sum_{i=0}^{N-1} \left[s_i \cdot \delta(t - i \cdot \Delta t) \right] \otimes adc(t) \cdot e^{-j2\pi f t} \cdot dt = \\ &= ADC(f) \cdot \sum_{i=0}^{N-1} \int_{-\infty}^{\infty} s_i \cdot \delta(t - i \cdot \Delta t) \cdot e^{-j2\pi f t} \cdot dt = \\ &= ADC(f) \cdot \sum_{i=0}^{N-1} s_i \cdot e^{-j2\pi f i \Delta t} \end{aligned} \qquad \textbf{Gleichung 132}$$

Wählt man für $adc(t)$ die Si-Interpolation, so ergibt sich also für $ADC(f)$ ein symmetrisches Rechteckfenster im Spektralbereich, das heißt das Spektrum $S(f)$ einer Zeitfunktion, deren Werte sich zwischen den Abtastwerten nach einer Si-Interpolation verhalten[105], ist ideal bandbegrenzt. Wählt man $adc(t)$ gemäß einer Interpolation durch eine Treppenfunktion, so ergibt sich jedoch kein bandbegrenztes Spektrum.[106] Die bekannte Formel für die diskrete Kurzzeit-Fourier-Transformation ergibt sich unter Annahme der Verwendung der Si-Interpolation zu Gleichung 133:

$$S(f) = \sum_{i=0}^{N-1} s_i \cdot e^{-j2\pi f i \Delta t} \qquad \textbf{Gleichung 133}$$

Auswertung an äquidistanten Stellen. Man erhält die bekannte Gleichung 136 der diskreten Fourier-Transformation, indem Gleichung 133 an wiederum genau N äquidistanten Stellen $f = k \cdot \Delta f$ ausgewertet wird.

$$\Delta f \cdot \Delta t \cdot N = 1 \qquad \textbf{Gleichung 134}$$

---

[104] Wie gesagt, kann ein allgemeiner Signalverlauf von diesem Ansatz einerseits insofern abweichen, als weitere Abtastwerte außerhalb des Intervalls ungleich null sein könnten und andererseits eine bestimmte Art der Interpolation zwischen den Abtastwerten angesetzt wird.

[105] Dies ergibt sich bei den meisten Realisierungen als Nebeneffekt des Anti-Aliasing-Filters näherungsweise.

[106] Was auch zu vermuten ist, da das Treppensignal aufgrund der Sprünge kein bandbegrenztes Signal darstellt.

$$f \cdot t = \frac{i \cdot k}{N}$$ **Gleichung 135**

$$S_k = \sum_{i=0}^{N-1} s_i \cdot e^{-j2\pi \frac{ik}{N}}$$ **Gleichung 136**

Fensterfunktionen. Wie im Zusammenhang mit Gleichung 131 ausgeführt, werden die Abtastwerte des zu analysierenden Signals außerhalb des Fensterintervalls notwendigerweise zu null angenommen. Da diese Forderung in der Praxis oftmals nicht ausreichend zutreffend ist, analysiert man nicht das eigentliche Signal, sondern das Signal multipliziert mit einer Fensterfunktion. Beim Resultat handelt es sich dann um das interessierende Spektrum gefaltet mit dem Spektrum der Fensterfunktion.

$$S'_k = \sum_{i=0}^{N-1} s_i \cdot w_i \cdot e^{-j2\pi \frac{ik}{N}}$$ **Gleichung 137**

Die konkrete Definition der Fensterfunktion bestimmt nun eine Reihe interessanter Eigenschaften der Kurzzeit-Fourier-Transformation [Harris1978] [Nuttall1981] [Schlang1990] [Adams1991]. Bereits in den Sechzigerjahren des vorigen Jahrhunderts wurden spezielle Eigenschaften der Fensterfunktionen analysiert, um dem biologischen Vorbild der Spektralanalyse im menschlichen Innenohr nahe zu kommen [Gambardella1968]. Vom heutigen Standpunkt aus sind diese Erkenntnisse im Lichte der Wavelet-Analyse zu sehen.

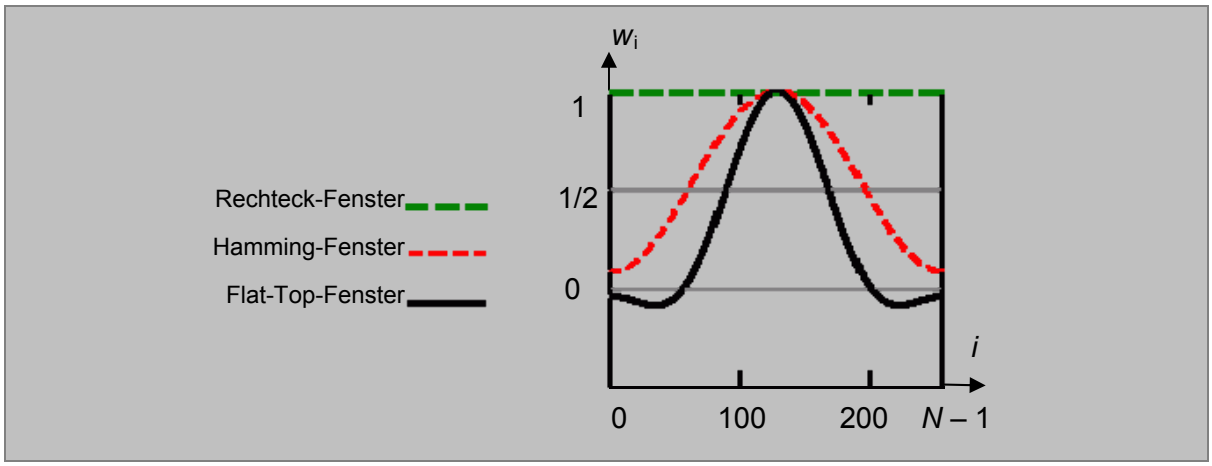

Abbildung II-1: Verschiedene Fensterfunktionen.

Abbildung II-1 zeigt verschiedene Fensterfunktionen für N = 256: Das Rechteck-Fenster, das Hamming-Fenster und das Flat-Top-Fenster[107]. Flat-Top-Fenster sind bei Spektrum-Analyzer-Geräten verbreitet und für die Amplitudenmessung optimiert. Sie weisen im Frequenzbereich einen geringen „passband ripple“, das heißt eine geringe Welligkeit im Durchlassbereich auf, sodass auch bei harmonischen Signalen mit tatsächlichen Frequenzen zwischen zwei Spektrallinien, der relativ genaue Amplitudenmesswert auf einer der beiden Spektrallinien zu finden ist. Es sind auch Flat-Top-Fenster-Funktionen mit mehr als zwei Cosinus-Termen üblich[108].

$$w_i^{\text{Hamming}} = 0{,}540 + 0{,}460 \cdot \cos\left(2\pi \frac{i}{N}\right) \qquad \textbf{Gleichung 138}$$

$$w_i^{\text{Flat-Top}} = 0{,}281 - 0{,}521 \cdot \cos\left(2\pi \frac{i}{N}\right) + 0{,}198 \cdot \cos\left(4\pi \frac{i}{N}\right) \qquad \textbf{Gleichung 139}$$

Dennoch wird den Fensterfunktionen der Kurzzeit-Fourier-Transformation auf dem Gebiet der Spracherkennung vielfach wenig Beachtung geschenkt. Aus Tradition wird meist das bewährte Hamming-Fenster verwendet. Das liegt wohl daran, dass die Technik der Spracherkennung noch keineswegs ausgereift ist, und wesentlich dringendere Probleme anzustehen scheinen [Rozman2003]. Möglicherweise wird der Auswahl der Fensterfunktion jedoch derzeit zu wenig Bedeutung beigemessen, denn zweifelsohne stellt die Merkmalsextraktion eine „sehr empfindliche Schraube“ bezüglich der Qualität von Spracherkennern dar.

[107] Quelle: DADiSP Documentation
http://www.dadisp.com/webhelp/mergedprojects/refman2/fncreffk/FLATTOP.htm (Stand: 25. 03. 2013).

[108] Quelle: Matlab R2013a Documentation
http://www.mathworks.de/de/help/signal/ref/sigwin.flattopwinclass.html (Stand: 25. 03. 2013).

Anhang III. Lineare Diskriminanzanalyse

In der Literatur werden verschiedene Varianten der linearen Diskriminanzanalyse beschrieben. Im Rahmen dieser Arbeit wurde die folgende Variante implementiert. Gegeben seien N Punktwolken $\mathbf{x}_{i,n}$, mit jeweils I_n Punkten.

$$n \in \{1,2,\ldots N\}$$ **Gleichung 140**

Berechnet werden zunächst die Zentroide der Punktwolken sowie der Zentroid der Ortsvektoren aller Punktwolken:

$$\boldsymbol{\mu}_n = \frac{1}{I_n}\sum_{i=1}^{I_n}\mathbf{x}_{i,n}, \quad \boldsymbol{\mu} = \sum_{n=1}^{N} p_n \cdot \boldsymbol{\mu}_n$$ **Gleichung 141**

$$p_n = \frac{I_n}{\sum_{n=1}^{N} I_n}$$ **Gleichung 142**

Bei der implementierten Variante werden die Punkte der Punktwolken allerdings derart „mehrfach gezählt“, sodaß sich gleich viele Punkte I_n für jede Punktwolke n ergeben und jede Punktwolke mit gleichem Gewicht p_n einfließt. Es gilt also für den im Rahmen dieser Arbeit implementierten Algorithmus:

$$p_n = \frac{1}{N}$$ **Gleichung 143**

Die „Between-Class Scatter Matrix“ und die „Within-Class Scatter Matrix“ seien definiert gemäß [Dahmen2001]:

$$\mathbf{S}_{between} = \sum_{n=1}^{N} I_n \left(\boldsymbol{\mu}_n - \boldsymbol{\mu}\right)^T \cdot \left(\boldsymbol{\mu}_n - \boldsymbol{\mu}\right)$$ **Gleichung 144**

$$\mathbf{S}_{within} = \sum_{n=1}^{N}\sum_{i=1}^{I_n}\left(\mathbf{x}_{i,n} - \boldsymbol{\mu}_n\right)^T \cdot \left(\mathbf{x}_{i,n} - \boldsymbol{\mu}_n\right)$$ **Gleichung 145**

Gesucht ist eine gemeinsame lineare Transformation, d.h. eine Koordinatentransformation, für die Vektoren der Punktwolken, sodaß es zu einer „Simultaneous Diagonalization“ kommt, d.h, daß beide Matrizen, und damit auch ihre Inversen, im transformierten Koordinatensystem Diagonalform annehmen. Da durch die Koordinatentransformation der Within-Class-Scatter minimiert und der Between-Class-Scatter

maximiert werden soll, wird folgende Gleichung zur Bestimmung der Eigenwerte und Eigenvektoren herangezogen [HaebUmbach1992]:

$$\mathbf{S}_{\text{within}}^{-1}\mathbf{S}_{\text{between}} \cdot \mathbf{a} = \lambda \cdot \mathbf{a} \qquad \textbf{Gleichung 146}$$

Die Anforderungen an die gesuchte Koordinatentransformation können erreicht werden, indem die Transformation aus einer Whitening-Transformation bezüglich einer der beiden Scatter-Matrizen und einer Hauptachsentransformation bezüglich der anderen Scatter-Matrix zusammengesetzt wird. In der implementierten Variante wird die Inverse der Within-Class-Scatter-Matrix für die Whitening-Transformation herangezogen. Die Whitening-Transformation setzt sich ihrerseits aus einer Hauptachsentransformation sowie Streckungen in Richtung der Hauptachsen zusammen:

$$\mathbf{S}_{\text{within}}^{-1} \cdot \mathbf{u} = \lambda_{\text{within}}^{-1} \cdot \mathbf{u} \qquad \textbf{Gleichung 147}$$

$$\widehat{\mathbf{U}} = \mathbf{\Lambda}_{\text{within}}^{-\frac{1}{2}} \cdot \mathbf{U} \qquad \textbf{Gleichung 148}$$

Bezüglich der anderen Scatter-Matrix, im bereits transformierten Koordinatensystem, wird eine Hauptachsentransformation durchgeführt,

$$\left(\mathbf{U}^{\mathrm{T}}\mathbf{S}_{\text{between}}\mathbf{U}\right) \cdot \mathbf{v} = \lambda_{\text{between}} \cdot \mathbf{v} \qquad \textbf{Gleichung 149}$$

Die zusammengesetzte Transformation, die sogenannte „LDA-Transformation“ ergibt sich als Produkt der Transformationsmatrizen, wobei die Spaltenvektoren Gleichung 146 erfüllen.

$$\mathbf{A} = \left(\mathbf{a}_1, \mathbf{a}_2, \ldots\right) = \mathbf{V}\widehat{\mathbf{U}} \qquad \textbf{Gleichung 150}$$

Die Vektoren der Punktwolken werden wie folgt transformiert:

$$\widehat{\mathbf{x}}_{\mathrm{i,n}} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{x}_{\mathrm{i,n}} \qquad \textbf{Gleichung 151}$$

Die LDA-Transformation kann auch zur Dimensionsreduktion herangezogen werden. Dazu werden die Spaltenvektoren in Gleichung 150 entsprechend fallender Eigenwerte sortiert, und nicht alle, sondern nur eine gewisse Anzahl an Spaltenvektoren zu einer nicht-quadratischen Transformationsmatrix zusammengestellt.

Literatur

[Adams1991] Adams J. W., „A New Optimal Window“, *IEEE Transactions on Signal Processing*, Vol.39, No.8, August, pp.1753-1769, 1991.

[Adamy2003] Adamy J., Voutsas K., Willert V., „Ein binaurales Richtungshörsystem für mobile Roboter in echoarmer Umgebung (A Binaural Sound Localization System for Mobile Robots in Low-reflecting Environments) “, *Automatisierungstechnik*, Vol.51, No.9, pp.387-395, 2003.

[Ahrndt1992] Ahrndt T., Ziegler W., „Sprechen und Verstehen am Computer: Ein PC-basiertes Verfahren zur Verständlichkeitsprüfung mit dem "Münchner Verständlichkeitsprofil (MVP)“, *Biomedical Journal*, 35, pp.4-8, 1992.

[Allen1977] Allen J. B., Rabiner L. R., „A Unified Approach to Short-Time Fourier Analysis and Synthesis“, *Proceedings of the IEEE*, Vol.65, No.11, November, pp.1558-1564, 1977.

[Alter1996] Alter K., Buchberger E., Matiasek J., Niklfeld G., Trost H., „VIECTOS - The Vienna Concept to Speech System“, *Natural Language and Speech Technology - Results of the 3rd KONVENS Conference*, October, pp.166-170, 1996.

[Anemüller2001] Anemüller J., *Across-Frequency Processing in Convolutive Blind Source Separation*, Dissertation der Universität Oldenburg, 2001.

[Anemüller2002] Anemüller J., Kollmeier B., „Adaptive Separation of Acoustic Sources for Anechoic Conditions: A Constrained Frequency Approach“, *Speech Communication*, 39 (1-2), pp.79-95, 2002.

[Artlieb2000] Artlieb M., View Interpolation mit Radial-Basis-Function-Networks, Diplomarbeit der Technischen Universität Wien, 2000.

[Atal1974] Atal B. S., „Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification“, *The Journal of the Acoustical Society of America*, Vol.55, No.6, June, pp.1304-1312, 1974.

[Aubert1993] Aubert X., Heab-Umbach R., Ney H., „Continuous Mixture Densities and Linear Discriminant Analysis for Improved Context-Dependent Acoustic Models“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, Minneapolis NM, pp.648-651, 1993.

[Barbier1991] Barbier L., Chollet G., „Robust Speech Parameters Extraction for Word Recognition in Noise using Neural Networks“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, May, Toronto, Canada, pp.145-148, 1991.

[Beaufays2003] Beaufays F., Boies D., Weintraub M., Zhu Q., „Using Speech/Non-Speech Detection to Bias Recognition Search on Noisy Data“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.424-427, 2003.

[Beh2003] Beh J., Ko H., „A Novel Spectral Subtraction Scheme for Robust Speech Recognition: Spectral Subtraction using Spectral Harmonics of Speech“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.648-651, 2003.

[Bellman1957] Bellman R. E., *Dynamic Programming*, Princeton University Press, 1957.

[Biem2001] Biem A., Katagiri S., Mc Dermott E., Juang B.-H., „An Application of Discriminative Feature Extraction to Filter-Bank-Based Speech Recognition“, *IEEE Transactions on Speech and Audio Processing*, Vol.9, No.2, February, pp.96-110, 2001.

[Biem2003] Biem A., „Optimizing Features and Models using the Minimum Classification Error Criterion“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.868-871, 2003.

[Blauert1996] Blauert J., *Spatial Hearing - The Psychophysics of Human Sound Location*, ISBN 0-262-02413-6, MIT Press Cambridge, 1996.

[Bodden1995] Bodden M., Anderson T., „Binaurale automatische Spracherkennung im Störschall“, *Fortschritte der Akustik DAGA*, März, Saarbrücken, pp.951-954, 1995.

[Böhme1997] Böhme G., *Sprach-, Sprech-, Stimm- und Schluckstörungen - Band 1: Klinik*, ISBN 3-437-21018-1, Gustav Fischer Verlag Stuttgart Jena Lübeck Ulm, 1997.

[Bottou1989] Bottou L., Fogelman-Soulie F., Blanchet P., Lienard J. S., „Experiments with Time Delay Networks and Dynamic Time Warping for Speaker Independent Isolated Digit Recognition“, *Eurospeech*, September, Paris France, pp.537-540, 1989.

[Bourlard1990] Bourlard H., Wellekens C. J., „Links between Markov Models and Multilayered Perceptrons“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.12, No.12, December, pp.1167-1178, 1990.

[Bridle1992] Bridle J. S., „Neural Networks or Hidden Markov Models for Automatic Speech Recognition: Is there a Choice?“, *Speech Recognition and Understanding. Recent Advances, Trends and Applications. NATO ASI Series F75*, Springer Verlag Berlin Heidelberg, pp.225-236, 1992.

[Carrol1989] Carrol S. M., Dickinson B. W., „Construction of Neural Networks using the Radon Transform“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.1, June, Washington DC, pp.607-611, 1989.

[Chang1992] Chang P.-C., Juang B.-H., „Discriminative Template Training for Dynamic Programming Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, March, San Francisco CA, pp.493-496, 1992.

[Chen2004] Chen J., Huang Y., Li Q., Paliwal K. K., „Recognition of Noisy Speech using Dynamic Spectral Subband Centroids“, *IEEE Signal Processing Letters*, Vol.11, No.2, February, pp.258-261, 2004.

[Chetouani2002] Chetouani M., Gas B., Zarader J. l., „The Modular Neural Predictive Coding Architecture“, *Proceedings of the 9th International Conference on Neural Information Processing ICONIP*, Vol.1, November, pp.452-456, 2002.

[Chetouani2002a] Chetouani M., Gas B., Zarader J. L., „Discriminative Training for Neural Predictive Coding applied to Speech Features Extraction“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.1, Honolulu, HI USA, Mai, pp.852-857, 2002.

[Chetouani2003] Chetouani M., Gas B., Zarader J., „Modular Neural Predictive Coding for Discriminative Feature Extraction“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, pp.32-36, 2003.

[Clemins2003] Clemins P. J., Johnson M. T., „Application of Speech Recognition to African Elephant (Loxodonta Africana) Vocalizations“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.484-487, 2003.

[Colburn1996] Colburn H. S., „Computational Models of Binaural Processing“, *Springer Handbook of Auditory Research - Volume: Auditory Computation*, ISBN 0-387-97843-7, Springer-Verlag New York, pp.332-400, 1996.

[Coleman1991] Coleman C. L., Meyers S., „Computer Recognition of the Speech of Adults with Cerebral Palsy and Dysarthria“, , Vol.7, March, pp.34-42, 1991.

[Cox2000] Cox R. V., Kamm C. A., Rabiner L. R., Schröter J., Wilpon J. G., „Speech and Language Processing for Next-Millennium Communications Services“, *Proceedings of the IEEE*, Vol.88, No.8, August, pp.1314-1337, 2000.

[Cybenko1989] Cybenko G., „Approximation by Superpositions of a Sigmodial Function“, *Mathematics of Control, Signals, and Systems*, Vol. 2, No.4, pp.303-314, 1989.

[Dahl2012] Dahl, G. E., Yu, D., Deng, L., Acero, A., „Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition“. *IEEE Transactions on Audio, Speech, and Language Processing* Vol.20, No.1, pp.30-42, 2012.

[Dahmen2001] Dahmen J., *Invariant Image Object Recognition using Gaussian Mixture Densities*, Dissertation der Rheinisch-Westfälischen Technischen Hochschule Aachen, 2001.

[Dau1996] Dau T., Püschel D., Kohlrausch A., „A Quantitative Model of the "Effective" Signal Processing in the Auditory System. I. Model Structure“, *The Journal of the Acoustical Society of America*, Vol.99, No.6, June, pp.3615-3622, 1996.

[Dau1996a] Dau T., Püschel D., Kohlrausch A., „A Quantitative Model of the "Effective" Signal Processing in the Auditory System. II. Simulations and Measurements“, *The Journal of the Acoustical Society of America*, Vol.99, No.6, June, pp.3623-3631, 1996.

[Dau1997] Dau T., Kollmeier B., „Modeling Auditory Processing of Amplitude Modulation. I. Detection and Masking with Narrow-Band Carriers“, *The Journal of the Acoustical Society of America*, Vol.102, No.5, November, pp.2892-2905, 1997.

[Dau1997a] Dau T., Kollmeier B., „Modeling Auditory Processing of Amplitude Modulation. II. Spectral and Temporal Integration“, *The Journal of the Acoustical Society of America*, Vol.102, No.5, November, pp.2906-2919, 1997.

[DeLaPaz1999] De La Paz S., „Composing via Dictation and Speech Recognition Systems: Comensatory Technology for Students with Learning Disabilities“, *Learning Disability Quarterly*, Vol.22, Summer, pp.173-182, 1999.

[Deller1991] Deller J. R., Hsu D., Ferrier L. J., „On the use of Hidden Markov Modelling for Recognition of Dysarthric Speech“, *Computer Methods and Programs in Biomedicine*, Vol.35, No.2, pp.125-139, 1991.

[Deshmukh1999] Deshmukh N., Ganapathiraju A., Picone J., „Hierarchical Search for Large Vocabulary Conversational Speech Recognition“, *IEEE Signal Processing Magazine*, Vol.16, No. 5, September, pp.84-107, 1999.

[Dharanipragada2002] Dharanipragada S., „Feature Extraction for Robust Speech Recognition“, *IEEE International Symposium on Circuits and Systems ISCAS*, Vol.2, Phoenix-Scottsdale AZ USA, May, pp.855-858, 2002.

[Do2011] Do, V. H., Xiao, X., Chng, E. S., „Comparison and Combination of Multilayer Perceprons and Deep Belief Networks in Hybrid Automatic Speech Recognition Systems“, *Proceedings of the Asia-Pacific Signal and Information Processing Annual Summit and Conference APSIPA ASC*, 2011.

[Do2011a] Do, V. H., „Hybrid Architectures for Speech Recognition“, PhD Thesis at the Nanyang Technological University, pp.1-81, 2011.

[Eggink2001] Eggink J., Wahrnehmungsbasierte Trennung und Gruppierung auditorischer Objekte - Ein Vergleich aktueller computergestützter Modelle, Diplomarbeit der Universität Hamburg, 2001.

[Eier1994] Eier R., „Markov Chains for Modeling and Analyzing Digital Data Signals“, *Proceedings of the IEEE-IMS-Workshop - Information Theory and Statistics*, October, p.86, 1994.

[Eier1999] Eier R., „Mapped Markov Chains for Modeling Non-Markovian Processes“, *World Multiconference on Systemics, Cybernetic and Informatics*, July, Orlando Florida, pp.348-355, 1999.

[Eppinger1993] Eppinger B., Herter E., *Sprachverarbeitung*, ISBN 3-446-16076-0, Carl Hanser Verlag München Wien, 1993.

[Falaschi1993] Falaschi A., Baldassarra A., Martinelli G. F., Prina Ricotti L., „Ergodic Hidden Control Neural Network for Modelling of the Speech Process“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Minneapolis NM, pp.605-608, 1993.

[Feldbauer2005] Feldbauer C., Kubin G., Kleijn W. B., „Andropomorphic Coding of Speech and Audio“, *Journal on Applied Signal Processing EURASIP*, Vol.2005, No.9, pp.1334-1349, 2005.

[Ferrier1992] Ferrier L. J., Jarrell N., Carpenter T., Shane H. C., „A Case Study of a Dysarthric Speaker using the DragonDictate Voice Recognition System“, *Journal for Computer Users in Speech and Hearing*, Vol.8, No.1, pp.33-52, 1992.

[Ferrier1995] Ferrier L. J., Shane H. C., Ballard H. F., Carpenter T., Benoit A., „Dysarthric Speakers' Intelligibility and Speech Characteristics in Relation to Computer Speech Recognition“, *Augmentative and Alternative Communication*, Vol.11, No.3, September, pp.165-175, 1995.

[Fichter2002] Fichter C., *Brain-Computer Interfaces - Geschichte, aktueller Stand und Perspektiven*, Nebenfacharbeit Neurophysiologie der Philosophischen Fakultät der Universität Zürich, 2002.

[Fink2000] Fink G. A., Sagerer G., „Zeitsynchrone Suche mit n-Gramm-Modellen höherer Ordnung“, *Konvens 2000/Sprachkommunikation, ITG-Fachbericht*, 161, pp.145-150, 2000.

[Flachberger1994] Flachberger C., Panek P., Zagler W. L., „AUTONOMY - A Flexible and Easy-to-Use Assistive System to Support the Independence of Handicapped and Elderly Persons“, *Proceedings of the 4th International Conference on Computers for Handicapped Persons ICCHP*, September, Vienna, Austria, pp.65-75, 1994.

[Forney1973] Forney G. D., „The Viterbi Algorithm“, *Proceedings of the IEEE*, Vol.61, No.3, March, pp.268-278, 1973.

[Fukuda2003] Fukuda T., Yamamoto W., Nitta T., „Distinctive Phonetic Feature Extraction for Robust Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, pp.25-28, 2003.

[Fuller1995] Fuller P., Lysley A., Colven D., „Trees in the Forest or Seeing the Wood for the Trees“, *The European Context for Assistive Technology - Proceedings of the 2nd TIDE Congress*, Paris, IOS Press, pp.3-16, 1995.

[Gajic2003] Gajic B., Paliwal K. K., „Robust Speech Recognition using Features based on Zero Crossings with Peak Amplitudes“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.64-67, 2003.

[Gambardella1968] Gambardella G., „Time Scaling and Short-Time Spectral Analysis“, *The Journal of the Acoustical Society of America*, Vol.44, No.6, pp.1745-1747, 1968.

[Giuliani2003] Giuliani D., Gerosa M., „Investigating Recognition of Children's Speech“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, pp.137-140, 2003.

[Goth2003] Goth G., „Speech Recognition's Evolution Continues“, *IEEE Intelligent Systems*, Vol.18, No.5, September, October, pp.5-7, 2003.

[Grassi1998] Grassi S., *Optimized Implementation of Speech Processing Algorithms*, Dissertation der Universität Neuchâtel, 1998.

[Graupe1992] Graupe D., „Blind Adaptive Filtering of Unknown Speech from Unknown Noise in a Single Receiver Situation“, *Proceedings of the International Symposium on Circuits and Systems ISCAS*, Vol.6, May, pp.2617-2620, 1992.

[Gray1918] Gray H., *Anatomy of the Human Body*, Philadelphia: Lea & Febiger, 1918.

[Gray2001] Gray R. M., „Gauss Mixture Vector Quantization“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.3, May, Salt Lake City, Utah, pp.1769-1772, 2001.

[Gu2003] Gu L., Gao J., Harris J. G., „Endpoint Detection in Noisy Environment using a Poincare Recurrence Metric“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.428-431, 2003.

[Gu2005] Gu L., Harris J. G., Shrivastav R., Sapienza C., „Disordered Speech Assessment Using Automatic Methods Based on Quantitative Measures“, *Journal on Applied Signal Processing EURASIP*, Vol.2005, No.9, pp.1400-1401, 2005.

[GutierrezFayos2003] Gutierrez Fayos R., Valdizan M. C., Labarta Bertol C., Jara Sanz M. P., Pozuelo Gomez M. A., Mareque M. A., Pinto A., Sarmiento J. M., De Mercado Fraguas J., De Torres Iglesias R., *Dispositivos de Control del Entorno y Aplicaciones Domóticas en Personas con Lesion Medular de Nivel alto*, Investigacion: Hospital Nacional de Parapléjicos de Toledo, Pronissa, 2003.

[HaebUmbach1992] Haeb-Umbach R., Ney H., „Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, March, San Francisco CA, pp.13-16, 1992.

[HaebUmbach1993] Haeb-Umbach R., Geller D., Ney H., „Improvements in Connected Digit Recognition using Linear Discriminant Analysis and Mixture Densities“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, Minneapolis NM, pp.239-242, 1993.

[Hansen1996] Hansen M., Dau T., Kollmeier B., „Objektive Sprachqualitätsvorhersage mittels einer gehörorientierten Vorverarbeitung“, *Fortschritte der Akustik DAGA*, Februar, Bonn, pp.496-497, 1996.

[Harris1978] Harris F. J., „On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform“, *Proceedings of the IEEE*, Vol.66, No.1, January, pp.51-83, 1978.

[Hartmann2000] Hartmann C., Kleinschmidt M., Tchorz J., Kollmeier B., „Gehörgerechte Vorverarbeitung für die robuste Spracherkennung auf Basis von Wortuntereinheiten“, *Fortschritte der Akustik DAGA*, März, Oldenburg, pp.380-381, 2000.

[Hawley2007] Hawley, M.S., Enderby, P., Green, P., Cunningham, S., Brownsell, S., Carmichael, J., Parker, M., Hatzis, A., O'Neill, P., Palmer, R., „A speech-controlled environmental control system for people with severe dysarthria“, *Medical Engineering & Physics*, Vol. 29, No.5, p.586-593, 2007.

[Hermansky1990], „Perceptual linear prediction (PLP) analysis of speech“, *The Journal of the Acoustical Society of America*, Vol.87, No.4, April, pp.1738-1752, 1990.

[Hickersberger1997] Hickersberger H., *Spracherkennung mit Hidden Control Neural Networks*, Diplomarbeit der Technischen Universität Wien, 1997.

[Hickersberger1998] Hickersberger H., „Spracherkennung mit Hidden Control Neural Networks“, *e&i Elektrotechnik und Informationstechnik - ÖVE-Verbandszeitschrift*, 115.Jahrgang, Heft 5, Mai, pp.245-250, 1998.

[Hickersberger2004] Hickersberger H., „Sound Sequence Recognizer, Klangfolgenerkenner“, *Österreichisches Patentamt - Patentanmeldung*, A260-2004, pp.1-8, 2004.

[Higgins1995] Higgins E. L., Zvi J. C., „Assistive Technology for Postsecondary Students with Learning Disabilities: From Research to Practice“, *Annals of Dyslexia*, Vol.45, pp.123-143, 1995.

[Higgins1995a] Higgins E. L., Raskind M. H., „Compensatory Effectiveness of Speech Recognition on the Written Composition Performance of Postsecondary Students with Learning Disabilities“, *Learning Disability Quarterly*, Vol.18, No.2, pp.159-174, 1995.

[Higgins2000] Higgins E. L., Raskind M. H., „Speaking to Read: The Effects of Continuous vs. Discrete Speech Recognition Systems on the Reading and Spelling of Children with Learning Disabilities“, *Journal of Special Education Technology*, Vol.15, No.1, pp.19-30, 2000.

[Hildebrandt1998] Hildebrandt H., *Pschyrembel Klinisches Wörterbuch*, ISBN 3-11-014824-2, de Gruyter Berlin, 1998.

[Hohmann2002] Hohmann V., „Frequency Analysis and Synthesis using a Gammatone Filterbank“, *Acustica - acta acustica*, Vol.41, No.1, pp.22-46, 2002.

[Holme1997] Holme S., Kanny E., Guthrie M., Johnson K., „The use of Environmental Control Units by Occupational Therapists in Spinal Cord Injury and Disease Services“, *American Journal of Occupational Therapy*, Vol.51, No.1, pp.42-48, 1997.

[Hornik1989] Hornik K., Stinchcombe M., White H., „Multilayer Feedforward Networks are Universal Approximators“, *Neural Networks*, Vol.2, No.5, pp.359-366, 1989.

[Huang2003] Huang K.-C., Juang Y.-T., „Feature Weighting in Noisy Speech Recognition“, *Electronics Letters*, Vol.39, No.12, June, pp.938-939, 2003.

[Hux2000] Hux K., Rankin-Erickson J., Manasse N., Lauritzen E., „Accuracy of Three Speech Recognition Systems: Case Study of Dysarthric Speech“, *Augmentative and Alternative Communication*, Vol.16, September, pp.186-196, 2000.

[Iso1990] Iso K.-I., Watanabe T., „Speaker-Independent Word Recognition using a Neural Prediction Model“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Albuquerque NM, pp.441-444, 1990.

[Ivanov2005] Ivanov A. V., Petrovsky A. A., „Analysis of the ICH Adaption for the Anthropomorphic Speech Processing Systems“, *Journal on Applied Signal Processing EURASIP*, Vol.2005, No.9, pp.1323-1333, 2005.

[Jaitly2012] Jaitly, N., Nguyen, P., Senior, A., Vanhoucke, V.: „Application of Pretrained Deep Neural Network to Large Vocabulary Conversational Speech Recognition“, Proceedings of Interspeech, 2012.

[Jayaram1995] Jayaram G., Abdelhamied K., „Experiments in Dysarthric Speech Recognition using Artificial Neural Networks“, *Journal of Rehabilitation Research and Development*, Vol.32, No.2, May, pp.162-169, 1995.

[Jelinek1976] Jelinek F., „Continuous Speech Recognition by Statistical Methods“, *Proceedings of the IEEE*, Vol. 64, No.4, April, pp.532-556, 1976.

[Jorgensen2003] Jorgensen C. Lee, D. D. Agabon, S. , „Sub Auditory Speech Recognition based on EMG Signals“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.4, July, pp.3128-3133, 2003.

[Juang1992] Juang B.-H., Katagiri S., „Discriminative Learning for Minimum Error Classification“, *IEEE Transactions on Signal Processing*, Vol.40, No.12, December, pp.3043-3054, 1992.

[Juang1992a] Juang B.-H., Rabiner L. R., „Spectral Representations for Speech Recognition by Neural Networks - A Tutorial“, *Proceedings of the IEEE-SP-Workshop - Neural Networks for Signal Processing*, August-September, Piscataway, pp.214-222, 1992.

[Kasper1997] Kasper K., Reininger H., Wolf D., „Exploiting the Potential of Auditory Preprocessing for Robust Speech Recognition by Locally Recurrent Neural Networks“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, Munich Germany, pp.1223-1226, 1997.

[KewleyPort1991] Kewley-Port D., Watson C. S., Elbert M., Maki D., Reed D., „The Indiana Speech Training Aid (ISTRA) II: Training Curriculum and Selected Case Studies“, *Clinical Linguistics & Phonetics*, Vol.5, No.1, pp.13-38, 1991.

[Kim1993] Kim H., Parker J. K., „Hidden Control Neural Network Identification-Based Tracking Control of a Flexible Joint Robot“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.2, October, Nagoya, Japan, pp.1825-1828, 1993.

[Kleinschmidt1998] Kleinschmidt M., Tchorz J., Wittkop T., Hohmann V., Kollmeier B., „Robuste Spracherkennung durch binaurale Richtungsfilterung und gehörgerechte Vorverarbeitung“, *Fortschritte der Akustik DAGA*, März, Zürich, pp.396-397, 1998.

[Kleinschmidt1999] Kleinschmidt M., Marzinzik M., Kollmeier B., „Combining Monaural Noise Reduction Algorithms and Perceptive Preprocessing for Robust Speech Recognition“, *Psychophysics, Physiology, and Models of Hearing*, World Scientific Singapore, 1999.

[Kleinschmidt2000] Kleinschmidt M., Hohmann V., „Perzeptive Vorverarbeitung und automatische Selektion sekundärer Merkmale zur robusten Spracherkennung“, *Fortschritte der Akustik DAGA*, März, Oldenburg, pp.382-383, 2000.

[Kohonen1990] Kohonen T., „The Self-Organizing Map“, *Proceedings of the IEEE*, Vol.78, No.9, September, pp.1464-1480, 1990.

[Kollmeier2002] Kollmeier B., „Cocktail-Partys und Hörgeräte: Biophysik des Gehörs - Physikalisch inspirierte Hörmodelle weisen den Weg zu intelligenten Hörgeräten“, *Physik Journal*, Vol.1, No.4, pp.39-45, 2002.

[Kotler1997] Kotler A.-L., Thomas-Stonell N., „Effects of Speech Training on the Accuracy of Speech Recognition for an Individual with a Speech Impairment“, *Augmentative and Alternative Communication*, Vol.13, June, pp.71-80, 1997.

[Kwon2003] Kwon O.-W., Lee T.-W., „Optimizing Speech/Non-Speech Classifier Design using AdaBoost“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.436-439, 2003.

[Lapedes1987] Lapedes A., Farber R., „Nonlinear Signal Processing using Neural Networks: Prediction and System Modeling“, *Los Alamos National Laboratory Technical Report*, LA/UR87/2662, pp.1-51, 1987.

[Lee1997] Lee S.-Y., Kim D.-S., Ahn K.-H., Jeong J.-H., Kim H., Park S.-Y., Kim L.-Y., Lee J.-S., Lee H.-Y., „Voice Command II: A DSP Implementation of Robust Speech Recognition in Real-World Noisy Environments“, *Proceedings of the International Conference on Neural Information Processing ICONIP*, November, Dunedin, New Zealand, pp.1051-1054, 1997.

[Lee2002] Lee J., Seo C., Lee K. Y., „A New Nonlinear Prediction Model based on the Recurrent Neural Predictive Hidden Markov Model for Speech Enhancement“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, May, pp.1037-1040, 2002.

[Lee2003] Lee C., Hyun D., Choi E., Go J., Lee C., „Optimizing Feature Extraction for Speech Recognition“, *IEEE Transactions on Speech and Audio Processing*, Vol.11, No.1, January, pp.80-87, 2003.

[Levin1989] Levin E., Gewirtzman R., Inbar G. F., „Neural Network Architecture for Adaptive System Modeling and Control“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.2, June, Washington DC, pp.311-316, 1989.

[Levin1990] Levin E., „Word Recognition using Hidden Control Neural Architecture“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Albuquerque NM, pp.433-436, 1990.

[Levin1993] Levin E., „Hidden Control Neural Architecture Modeling of Nonlinear Time Varying Systems and its Applications“, *IEEE Transactions on Neural Networks*, Vol.4, No.1, January, pp.109-116, 1993.

[Li2001] Li Q., Zheng J., Zhou Q., Lee C.-H., „A Robust, Real-Time Endpoint Detector with Energy Normalization for ASR in Adverse Environments“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, May, Salt Lake City, Utah, pp.233-236, 2001.

[Loidolt1995] Loidolt, G., „AUTONOM III: Spracherkennung“, Diplomarbeit an der Technischen Universität Wien, p.1-71, 1995.

[MacArthur2004] Mac Arthur C. A., Cavalier A.R., „Dictation and Speech Recognition Technology as Test Accommodations“, *Exceptional Children*, Vol.71, No.1, pp.43-58, 2004.

[Mak2004] Mak B. K.-W., Tam Y.-C., Li P. Q., „Discriminative Auditory-Based Features for Robust Speech Recognition“, *IEEE Transactions on Speech and Audio Processing*, Vol.12, No.1, January, pp.27-36, 2004.

[Mangold2005] Mangold M., *Duden - Das Aussprachewörterbuch*, ISBN 3-411-04066-1, Dudenverlag Mannheim, 2005.

[Martinelli1994] Martinelli G., „Hidden Control Neural Network“, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.41, No.3, March, pp.245-247, 1994.

[Martinez2001] Martinez R., Álvarez A., Gómez P., Nieto V., Rodellar V., „Combination of Adaptive Filtering and Spectral Subtraction for Noise Removal“, *The 2001 IEEE International Symposium on Circuits and Systems ISCAS*, Vol.2, May, pp.793-796, 2001.

[Martinez2001a] Martinez A. M., Kak A. C., „PCA versus LDA“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.23, No.2, February, pp.228-233, 2001.

[Marzinzik2000] Marzinzik M., Noise Reduction Schemes for Digital Hearing Aids and their Use for the Hearing Impaired, Dissertation der Universität Oldenburg, 2000.

[McDermott1994] Mc Dermott E., Katagiri S., „Prototype-Based Minimum Classification Error/Generalized Probablistic Descent Training for Various Speech Units“, *Computer Speech and Language*, 8, pp.351-368, 1994.

[Merhav1991] Merhav N., Ephraim Y., „Maximum Likelihood Hidden Markov Modeling using a Dominant Sequence of States“, *IEEE Transactions on Signal Processing*, Vol.39, No.9, September, pp.2111-2115, 1991.

[Mika1999] Mika S., Rätsch G., Weston J., Schölkopf B., Müller K.-R., „Fisher Discriminant Analysis with Kernels“, *Proceedings of the IEEE-SP-Workshop - Neural Networks for Signal Processing IX*, August, pp.41-48, 1999.

[Miyatake1990] Miyatake M., Sawai H., Minami Y., Shikano K., „Integrated Training for Spotting Japanese Phonemes using Large Phonemic Time-Delay Neural Networks“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Albuquerque NM, pp.449-452, 1990.

[Molau2003] Molau S., Hilger F., Ney H., „Feature Space Normalization in Adverse Acoustic Conditions“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.656-659, 2003.

[Moller2002] Moller A.R., Rollins P.R., “The non-classical auditory pathways are involved in hearing in children but not in adults”, *Neuroscience Letters*, Vol.319, pp.41-44, 2002.

[Moore2003] Moore D. C., Mc Cowan I. A., „Microphone Array Speech Recognition: Experiments on Overlapping Speech in Meetings“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.5, April, pp.497-500, 2003.

[Na1995] Na K. M., Chae S.-I., Ann S. G., „Corrective Training of Hidden Control Neural Network“, *Proceedings of the International Conference on Neural Networks*, Vol.5, November, December, pp.2867-2870, 1995.

[Na1996] Na K. M., Chae S.-I., „GPD Training of the State Weighting Functions in Hidden Control Neural Network“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.6, May, Atlanta, Georgia, pp.3366-3369, 1996.

[Narayanan2002] Narayanan S., Potamianos A., „Creating Conversational Interfaces for Children“, *IEEE Transactions on Speech and Audio Processing*, Vol.10, No.2, February, pp.65-78, 2002.

[Noyes1992] Noyes J. M., Frankish C. R., „Speech Recognition Technology for Individuals with Disabilities“, *Augmentative and Alternative Communication*, Vol.8, December, pp.297-303, 1992.

[Nuttall1981] Nuttall A. H., „Some Windows with Very Good Sidelobe Behavior“, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.29, No.1, February, pp.84-91, 1981.

[Ohno1994] Ohno S., Fujisaki H., Hirose K., „A Method for Efficient DP Matching in Spoken Word Recognition“, *International Symposium on Speech, Image Processing and Neural Networks ISSIPNN*, April, Hong Kong, pp.714-717, 1994.

[ONeill1988] O'Neill M. A., „Faster Than Fast Fourier“, *BYTE*, April, pp.293-300, 1988.

[Panek1996] Panek P., Flachberger C., Zagler W. L., „The Integration of Technical Assistance into the Rehabilitation Process - a Field Study“, *Interdisciplinary Aspects on Computers Helping People with Special Needs - 5th International Conference ICCHP*, July, Linz, Austria, pp.529-537, 1996.

[Panek2002] Panek P., Beck C., Mina S., Seisenbacher G., Zagler W. L., „Technical Assistance for Severely Motor- and Multiple Disabled Children - Some Long Term Experiences“, *Interdisciplinary Aspects on Computers Helping People with Special Needs - 8th International Conference ICCHP - Lecture Notes in Computer Science*, Vol.2398, July, pp.181-188, 2002.

[Parsons1976] Parsons T. W., „Separation of Speech from Interfering Speech by means of Harmonic Selection“, *The Journal of the Acoustical Society of America*, Vol.60, No.4, October, pp.911-918, 1976.

[Patterson1995] Patterson R. D., Allerhand M., Giguère C., „Time-Domain Modeling of Peripheral Auditory Processing“, *The Journal of the Acoustical Society of America*, Vol.98, No.4, October, pp.1890-1894, 1995.

[Petek1992] Petek B., Tebelskis J., „Context-Dependent Hidden Control Neutral Network Architecture for Continuous Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, March, San Francisco CA, pp.397-400, 1992.

[Petek1993] Petek B., Ferligoj A., „Exploiting Prediction Error in a Predictive-Based Connectionist Speech Recognition System“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, April, Minneapolis NM, pp.267-270, 1993.

[Petek2000] Petek B., „On the Predictive Connectionist Models for Automatic Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.6, June, Istanbul, Turkey, pp.3442-3445, 2000.

[Picone1993] Picone J., „Signal Modeling Techniques in Speech Recognition“, *Proceedings of the IEEE*, Vol.81, No.9, September, pp.1215-1247, 1993.

[Portnoff1981] Portnoff M. R., „Short-Time Fourier Analysis of Sampled Speech“, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.29, No.3, June, pp.364-373, 1981.

[Prante2001] Prante H. U., *Modeling Judgements of Environmental Sounds by means of Artificial Neural Networks*, Disseration der Technischen Universität Berlin, 2001.

[Rabiner1986] Rabiner L. R., Juang B. H., „An Introduction to Hidden Markov Models“, *IEEE ASSP Magazine*, Vol.3, No.1, January, pp.4-16, 1986.

[Rabiner1989] Rabiner L. R., Wilpon J. G., Soong F. K., „High Performance Connected Digit Recognition using Hidden Markov Models“, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.37, No.8, August, pp.1214-1225, 1989.

[Rabiner1989a] Rabiner L. R., „A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition“, *Proceedings of the IEEE*, Vol.77, No.2, February, pp.257-286, 1989.

[Rabiner1993] Rabiner L. R., Juang B. H., *Fundamentals of Speech Recognition*, ISBN 0-13-015157-2, Prentice Hall London, 1993.

[Ramirez2004] Ramirez J., Segura J. C., Benitez C., De la Torre A., Rubio A. J., „A New Kullback-Leibler VAD for Speech Recognition in Noise“, *IEEE Signal Processing Letters*, Vol.11, No.2, February, pp.266-269, 2004.

[Raphael1999] Raphael C., „Automatic Segmentation of Acoustic Musical Signals using Hidden Markov Models“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.4, April, pp.360-370, 1999.

[Rattay1997] Rattay F., Lutter P., “Speech Sound Representation in the Auditory Nerve: Computer Simulation Studies on Inner Ear Mechanisms”, *ZAMM - Journal of Applied Mathematics and Mechanics*, Vol.77, No.12, pp.935-943, 1997.

[Ravindran2005] Ravindran S., Schlemmer K., Anderson D. V., „A Physiologically Inspired Method for Audio Classification“, *Journal on Applied Signal Processing EURASIP*, Vol.9, pp.1374-1381, 2005.

[Roberts2002] Roberts K. D., „Voice Recognition Software as a Compensatory Strategy for Postsecondary Students with Learning Disabilities“, *Proceedings of the International Conference on Computers in Education*, Vol.2, December, pp.1506-1507, 2002.

[Rojas1996] Rojas R., *Theorie der Neuronalen Netze - Eine systematische Einführung*, ISBN 3-540-56353-9, Springer-Verlag Berlin, 1996.

[Rouat2005] Rouat J., Pichevar R., „Source Separation with one ear: Proposition for an Anthropomorphic Approach“, *Journal on Applied Signal Processing EURASIP*, Vol.9, pp.1365-1373, 2005.

[Rozman2003] Rozman R., Kodek D. M., „Improving Speech Recognition Robustness using Non-Standard Windows“, *The IEEE Region 8 EUROCON - Computer as a Tool*, Vol.2, September, pp.171-174, 2003.

[Rumelhart1986] Rumelhart D. E., Hinton G. E., Williams R. J., „Learning Internal Representation by Error Propagation“, *Parallel Distributed Processing - Explorations in the Microstructure of Cognition - Volume 1: Foundations*, MIT-Press Cambridge, pp.318-363, 1986.

[Ruske1988] Ruske G., Automatische Spracherkennung - Methoden der Klassifikation und Merkmalsextraktion, ISBN 3-486-20877-2, R.-Oldenburg-Verlag München, 1988.

[Sanders2002] Sanders E., Ruiter M., Beijer L., Strik H., „Automatic Recognition of Dutch Dysarthric Speech - A Pilot Study“, *Proceedings of the 7th International Conference on Spoken Language Processing ICSLP*, September, Denver, Colorado, pp.1-4, 2002.

[Schlang1990] Schlang M., Mummert M., „Die Bedeutung der Fensterfunktion für die Fourier-t-Transformation als gehörgerechte Spektralanalyse“, *Fortschritte der Akustik DAGA*, April, Wien, pp.1043-1046, 1990.

[Schlüter2001] Schlüter R., Ney H., „Using Phase Spectrum Information for Improved Speech Recognition Performance“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, May, Salt Lake City Utah, pp.133-136, 2001.

[Schmid1990] Schmid P., *Explicit N-Best Formant Features for Segment-Based Speech Recognition*, Dissertation des Oregon Graduate Institute of Science and Technology, 1990.

[Schoknecht1999] Schoknecht R., Spott M., Liekweg F., Riedmiller M., „Search Space Reduction for Strategy Learning in Sequential Decision Processes“, *Proceedings of the 6th International Conference on Neural Information Processing ICONIP*, Vol.1, November, pp.148-153, 1999.

[Schölkopf1999] Schölkopf B., Müller K.-R., Smola A. J., „Lernen mit Kernen - Support-Vektor-Methoden zur Analyse hochdimensionaler Daten“, *Informatik Forschung und Entwicklung*, Vol.14, pp.154-163, 1999.

[Schröder1962] Schröder M. R., Atal B. S., „Generalized Short-Time Power Spectra and Autocorrelation Functions“, *The Journal of the Acoustical Society of America*, Vol.34, No.11, November, pp.1679-1683, 1962.

[Seeger1997] Seeger M., Unterstützung eingeschränkter Mixturemodelle durch lineare Transformationen des Merkmalsraumes im Rahmen des EM-Algorithmus, Studienarbeit der Universität Karlsruhe, 1997.

[Shafran2003] Shafran I., Rose R., „Robust Speech Detection and Segmentation for Real-Time ASR Applications“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.432-435, 2003.

[Shao2003] Shao X., Milner B., „Clean Speech Reconstruction from Noisy Mel-Frequency Cepstral Coefficients using a Sinusoidal Model“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, pp.704-707, 2003.

[Skowronski2003] Skowronski M. D., Harris J. G., „Improving the Filter Bank of a Classic Speech Feature Extraction Algorithm“, *Proceedings of the International Symposium on Circuits ans Systems ISCAS*, Vol.4, May, pp.281-284, 2003.

[Sorensen1992] Sorensen H. B. D., Hartmann U., „Self-Structuring Hidden Control Neural Model for Speech Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.2, March, San Francisco CA, pp.353-356, 1992.

[Sorensen1992a] Sorensen H. B. D., „Speech Recognition in Noise using a Self-Structuring Noise Reduction Model and Hidden Control Models“, *Proceedings of the International Joint Conference on Neural Networks IJCNN*, Vol.2, June, pp.279-284, 1992.

[Sorensen1992b] Sorensen H. B. D., Hartmann U., „Self-Structuring Hidden Control Neural Models“, *Proceedings of the IEEE-SP-Workshop - Neural Networks for Signal Processing*, August-September, Piscataway, pp.149-156, 1992.

[Sorensen1993] Sorensen H. B. D., Hartmann U., „Pi-Sigma and Hidden Control Based Self-Structuring Models for Text-Independent Speaker Recognition“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Minneapolis NM, pp.537-540, 1993.

[Sorensen1995] Sorensen H. B. D., Hartmann U., Hunnerup P., „Discriminative Training of Self-Structuring Hidden Control Neural Models“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.5, May, Detroid, Michigan, pp.3379-3382, 1995.

[Sovka1996] Sovka P., Pollak P., Kybic J., „Extended Spectral Subtraction“, *European Signal Processing Conference EUSIPCO*, September, Trieste Italy, pp.963-966, 1996.

[Storm2001] Storm R., Wahrscheinlichkeitsrechnung, mathematische Statistik und statistische Qualitätskontrolle, ISBN 3-446-21812-2, Fachbuchverlag Leipzig, 2001.

[Tchorz1996] Tchorz J., Wesselkamp M., Kollmeier B., „Gehörgerechte Merkmalsextraktion zur robusten Spracherkennung in Störgeräuschen“, *Fortschritte der Akustik DAGA*, Februar, Bonn, pp.532-533, 1996.

[Tchorz1997] Tchorz J., Kasper K., Reininger H., Kollmeier B., „On the Interplay Between Auditory-Based Features and Locally Recurrent Neural Networks for Robust Speech Recognition in Noise“, *Eurospeech*, September, Rhodes Greece, pp.2075-2078, 1997.

[Tchorz1999] Tchorz J., Kollmeier B., „A Model of Auditory Perception as Front End for Automatic Speech Recognition“, *The Journal of the Acoustical Society of America*, Vol.106, No.4, October, pp.2040-2050, 1999.

[Tchorz2001] Tchorz J., Kleinschmidt M., Kollmeier B., „Noise Suppression based on Neurophysiologically-Motivated SNR Estimation for Robust Speech Recognition“, *Advances in Neural Information Processing Systems NIPS*, 13, MIT-Press Cambridge, pp.821-827, 2001.

[Tebelskis1990] Tebelskis J., Waibel A., „Large Vocabulary Recognition using Linked Predictive Neural Networks“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Albuquerque NM, pp.437-440, 1990.

[Tishby1990] Tishby N., „A Dynamical Systems Approach to Speech Processing“, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing ICASSP*, Vol.1, April, Albuquerque NM, pp.365-368, 1990.

[Trentin2003] Trentin E., Gori M., „Robust Combination of Neural Networks and Hidden Markov Models for Speech Recognition“, *IEEE Transactions on Neural Networks*, Vol.14, No.6, November, pp.1519-1531, 2003.

[Tschirk1999] Tschirk W., „Neural Networks in Action. SICARE pilot - A Voice Remote Control for Disabled People“, *ÖGAI Journal - Österreichische Gesellschaft für Artificial Intelligence*, Nr.1, April, pp.22-24, 1999.

[Tschirk2001] Tschirk W., „Neural Net Speech Recognizers - Voice Remote Control Devices for Disabled People“, *e&i Elektrotechnik und Informationstechnik - ÖVE-Verbandszeitschrift*, 118.Jahrgang, Heft 7-8, pp.367-370, 2001.

[Tschirk2004] Tschirk W., „Self-optimizing Voice Control User Interface“, *Proceedings of the European Signal Processing Conference EUSIPCO*, September, Vienna, pp.1-4, 2004.

[Tubach1991] Tubach J. P., Doignon P., „A System for Natural Spoken Language Queries Design, Implementation and Assessment“, *Eurospeech*, September, Genova Italy, pp.1473-1476, 1991.

[Vaseghi1997] Vaseghi S. V., Milner B. P., „Noise Compensation Methods for Hidden Markov Models Speech Recognition in Adverse Environments“, *IEEE Transactions on Speech and Audio Processing*, Vol.5, No.1, January, pp.11-21, 1997.

[Wachsmuth1997] Wachsmuth S., Kombination von Grammatiken und statistischen Sprachmodellen zur automatischen Spracherkennung, Diplomarbeit der Universität Bielefeld, 1997.

[Waibel1989] Waibel A., Hanazawa T., Hinton G., Shikano K., Lang K. J., „Phoneme Recognition using Time-Delay Neural Networks“, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.37, No.3, March, pp.328-339, 1989.

[Wei2003] Wei J., Du L., Yan Z., Zeng H., „A New Algorithm for Voice Activity Detection“, *Proceedings of the International Symposium on Circuits ans Systems ISCAS*, Vol.2, May, pp.588-591, 2003.

[Wetzel1996] Wetzel K., „Speech-Recognizing Computers: A Written-Communication Tool for Students with Learning Disabilities?“, *Journal of Learning Disabilities*, Vol.29, No.4, pp.371-380, 1996.

[Wever1930] Wever E.G., Bray C.W., "Action currents in the auditory nerve in response to acoustical stimulations.", *Proceedings of the National Academy of Sciences*, Vol.16, pp.344-350, 1930.

[Widrow1990] Widrow B., Lehr M. A., „30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation“, *Proceedings of the IEEE*, Vol.78, No.9, September, pp.1415-1442, 1990.

[Wittkop1997] Wittkop T., Hohmann V., „Parameteroptimierung eines Richtungsfilters für Hörgerätealgorithmen“, *Fortschritte der Akustik DAGA*, März, Kiel, pp.246-247, 1997.

[Wittkop1997a] Wittkop T., Albani S., Hohmann V., Peissig J., Woods W. S., Kollmeier B., „Speech Processing for Hearing Aids: Neinise Reduction Motivated by Models of Binaural Interaction“, *Acustica - acta acustica*, Vol.83, No.4, pp.684-699, 1997.

[Yamada2002] Yamada T., Nakamura S., ShikaNein K., „Distant-Talking Speech Recognition Based on a 3-D Viterbi Search Using a Microphone Array“, *IEEE Transactions on Speech and Audio Processing*, Vol.10, Nein.2, February, pp.48-56, 2002.

[Yaniv2003] Yaniv R., Burshtein D., „An Enhanced Dynamic Time Warping Model for Improved Estimation of DTW Parameters“, *IEEE Transactions on Speech and Audio Processing*, Vol.11, No.3, May, pp.216-228, 2003.

[Zagler1993] Zagler W. L., Flachberger C., „AUTONEINM - A Remote Control System: An example for what techNeinlogy can do, but what technicians don't kNeinw“, *COST A5 Workshop on GerontechNeinlogies - Österreichische Akademie der Wissenschaften*, Vienna, pp.1-9, 1993.

[Zagler1997] Zagler W. L., Panek P., Flachberger C., „Technical Assistance for Severely Motor- and Multiple Impaired Children“, *Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems*, June, Maribor, Slovenia, pp.232-237, 1997.

[Zagler1997a] Zagler W. L., Edelmayer G., Mayer P., „Studies of Vision Enhancement with Optoelectronical Devices“, *Proceedings of the 10th IEEE Symposium on Computer-Based Medical Systems*, June, Maribor, Slovenia, pp.28-33, 1997.

[Zheng2001] Zheng F., Zhang G., Song Z., „Comparison of Different Implementations of MFCC“, *Journal of Computer Science and TechNeinlogy*, Vol.16, No.6, September, pp.582-589, 2001.

[Ziegler1992] Ziegler W., Hartmann E., Wiesner I., „Dysarthriediagnostik mit dem "Münchner Verständlichkeitsprofil" (MVP) - Konstruktion des Verfahrens und Anwendungen“, *Der Nervenarzt*, Vol.63, pp.602-608, 1992.

[Zwicker1999] Zwicker E., Fastl H., *Psychoacoustics - Facts and Models*, ISBN 3-540-65063-6, Springer-Verlag Berlin, 1999.

Abkürzungen

AFCC … Auditory-based Feature Cepstral Coefficients

CER … Confusion Error Rate

DPF … Distinctive Phonetic Features

EMG … Electromyogram

ETSI … European Telecommunication Standardization Institute

FAR … False Acceptance Rate

FFT … Fast Fourier Transformation

FRR … False Rejection Rate

HCNN … Hidden Control Neural Network

HFCC … Human Factor Cepstral Coefficients

LDA … Linear Discriminant Analysis

LPC … Linear Predictive Coding

MFCC … Mel Scale Frequency Cepstral Coefficients

NPC … Neural Predictive Coding

PARCOR … Partial Correlation

PEMO … Perception Model

PLP … Perceptual Linear Prediction

RLL … Run-Length Limit

SSC … Spectral Subband Centroids

TDNN … Time Delay Neural Network

ZCPA … Zero Crossings Peak Amplitudes

Stichwortverzeichnis

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Lebenslauf Helmut Hickersberger

Name:	Helmut Hickersberger
Geburtstag:	25.05.1972
Geburtsort:	Wien
Staatsbürgerschaft:	Österreich
Führerschein:	Klassen A, B
Sprachkenntnisse:	Englisch (fließend), Deutsch (Muttersprache)

1986/09 – 1991/05	HTL A-1200 Wien TGM: „Nachrichtentechnik“
1988/08, 1989/07	Betriebspraktika bei WIBEBA & AEG
1990/06	Eintragung in die „Ehrentafel der TGM-Besten“
1991/05	Matura mit „ausgezeichnetem Erfolg“
1991/09 – 1999/10	Studium der Elektrotechnik (Stzw. Computertechnik) TU-Wien
1992/07, 1993/07	Betriebspraktika bei PHILIPS
1992/07, 1993/07	Betriebspraktika bei SIEMENS Position: Software-Entwickler (Pascal) Projekt: „Terminal-Emulator zur Kommunikation mit Line-Trunk-Baugruppen von Siemens-Telefonvermittlungsanlagen“
1994/01-02	Werkvertrag mit ALCATEL Position: Algorithmen-Entwickler, Software-Entwickler (Pascal) Projekt: „Algorithmus zur Suche der optimalen Anordnung von Räumen bei der Gebäudeplanung“
1994/07, 1995/08	Betriebspraktika bei SIEMENS Position: Algorithmen-Entwickler (DSP-Assembler, Pascal) Projekt: „Siemens-Sprachsteuerung SICARE“
1996/02-04, 1997/08	Betriebspraktika (Diplomarbeit) bei SIEMENS Position: Diplomand (Pascal) Projekt: „Spracherkennung mit Hidden Control Neural Networks“
1997/11	Förderpreis der Gesellschaft für IT im ÖVE
1998/07	Betriebspraktikum bei IBM Position: Software-Entwickler (C++), Tester Projekt: „Performance-Testumgebung für den IBM-Spracherkenner ViaVoice“
ab 1999/10	Doktoratstudium der technischen Wissenschaften an der TU-Wien
2000/04 – 2001/03	Freier Dienstvertrag mit SIEMENS Position: Software-Configuration-Manager (DSP-Assembler, ANSI-C) Projekt: „Sprachwahl-Funktionalität für Gigaset-Anrufbeantworter“ Position: Software-Entwickler (ANSI-C) Projekt: „Flash-File-System für den Unispeech-8051/Oak-Prozessor (Infineon)“ Position: Software-Entwickler (Pascal, C++) Projekt: „PC-Benutzeroberflächen zur Messepräsentation des Unispeech-8051/Oak-Evaluation-Boards“
ab 2001/04	Anstellungsvertrag mit SIEMENS Position: Software-Entwickler (Pascal) Projekt: „PC-Benutzeroberfläche zum Modultest der Siemens-Sprachsteuerung SICARE“ Position: System-Architekt, Software-Entwickler (Pascal) Projekt: „Automatische V.92-Modemtestanlage“
2006/08	Patent „Klangfolgen-Erkenner“ (Pat. Nr.414060)
	Position: System-Architekt, Software-Entwickler (Pascal) Projekt: „Automatische VoIP-Gateway-Testanlage: Codec-Tests, Echo-Canceller-Tests ITU/G.168, Jitterbuffer-Tests“
2007/03	Patent „PC-Audio-Kalibriervorrichtung“ (SIEMENS, Pat. angemeldet)
	Position: System-Architekt, Software-Entwickler (Pascal) Projekt: „Automatische Sprachqualitätstestanlage für den Microsoft Office Communicator“ Position: Software-Tester (C++) Projekt: „Parametrierung und Regressionstests eines weltweit vereinheitlichten DTMF-Receivers“ Position: Algorithmen-Entwickler, Software-Entwickler (Visual Basic) Projekt: „Automatische Klirrfaktor-Messanlage für die Fertigungsprüfung“
2008/10	Kurs „Microsoft UC Developer Platform TAP“ (Redmond)
2009/03	Kurs „Microsoft .NET/C#, Microsoft SQL Server 2008“ (Wien)
	Position: System-Architekt, Software-Entwickler (C#, XAML, T-SQL) Projekt: „Siemens Materialmanagement-System“
2009/05	Kurs „Microsoft Office Communication Development“ (München)
2009/11	Kurs „Architecture Experts Curriculum: Analyse & Design“ (Wien)
2009/12	Kurs „Architecture Experts Curriculum: Software-Architekturen“ (Wien)
	Position: Software-Entwickler (C#, XAML) Projekt: „Konfigurator-Applikation zur Unterstützung des Vertriebs bei der Angebotslegung von Bürokommunikationslösungen: Telefonanl., PC-Arbeitsplätze, VoIP-Gateways und Helpdesk-Lösungen“ Position: Projekt-Manager für das Gewerk „Elektrotechnik, Mess- und Regeltechnik“ Projekt: „HRSG Gas- und Dampfkraftwerke Knapsack II, Lausward“