

Implementation kopfpositionsbezogener Binauraltechnik

Diplomarbeit

durchgeführt von

Piotr Majdak

Markus Noisternig

am

Institut für Elektronische Musik und Akustik
an der Universität für Musik und darstellende Kunst in Graz

Betreuer:

o. Univ. Prof. Mag DI Dr. techn. Robert Höldrich

Univ. Ass. DI Alois Sontacchi

Graz, im Februar 2002

Danksagung

Wir möchten ganz besonders unserem Betreuer, Herrn Alois Sontacchi, für die anregenden Diskussionen und die vielseitige Unterstützung beim Entstehen dieser Diplomarbeit danken.

Unser Dank gilt auch Herrn Robert Höldrich, der durch zahlreiche Anregungen viel zur Entstehung der in dieser Diplomarbeit behandelten Ideen beigetragen hat.

Weiters danken wir allen IEM-Mitarbeitern, sowie den Versuchspersonen für Ihre Unterstützung.

Ich möchte ganz besonders meiner Frau Ali für Ihre Unterstützung danken.

(Piotr)

Zusammenfassung

Die originalgetreue Reproduktion eines aufgezeichneten Schallfeldes, sowie die Spatialisation von Monoquellen, erfordert bei Kopfhörerwiedergabe eine Filterung der virtuellen Quellen mittels Außenohrübertragungsfunktionen (HRTFs). Diese beschreiben die Signale an den Trommelfellen beider Ohren in Abhängigkeit vom Einfallswinkel der Schallwelle. Die menschliche Hörwahrnehmung erlaubt eine Verbesserung der Lokalisation von Schallquellen mittels kleiner Kopfbewegungen. Um dieses Phänomen binauralen Applikationen zugänglich zu machen ist die Implementierung eines Headtrackers notwendig. Bei dynamischen Systemen entsteht die Problematik der zeitvarianten Interpolation zwischen den verschiedenen HRTFs. Eine Lösungsmöglichkeit bietet die Kodierung der Quellen in Ambisonic-Signale, wobei Änderungen der Kopfposition durch zeitvariante Manipulation (Rotations- und Translationsmatrix) der Ambisonic-Signale realisiert werden. Die Implementierung der benötigten HRTFs kann mittels zeitinvarianter Filter realisiert werden. Es wird ein objektives mathematisches Modell vorgestellt, welches eine Optimierung des Systems bezüglich der Lokalisation unter Betrachtung psychoakustischer Effekte ermöglicht. Die Verifizierung des mathematischen Modells erfolgt anhand von Hörversuchen. Ziel ist die Implementierung des optimierten Algorithmus auf einem DSP als Vorentwicklung eines *Consumer*-Produktes.

Abstract

Convincing sound reproduction via headphones requires filtering of virtual sources with head related impulse responses (HRIRs) which describe the signals at the two ear drums as a function of the source angle. Regarding hearing in natural sound fields humans are able to improve source localisation capabilities due to small head movements. To benefit from this, head tracking has to be incorporated in the time-varying binaural sound reproduction system. This leads to the problem of high-quality time-varying interpolation between different HRIRs. The proposed method solves this problem with a virtual Ambisonic approach which results in time-invariant filters. The influence of head position is taken into account with a cheap time-variant rotation matrix. A mathematical model is presented to objectively derive sound localisation performance using HRIR based binaural reproduction systems. To evaluate the significance of the proposed mathematical model, listening tests have been carried out. Furthermore, the optimised algorithm is implemented on a DSP to estimate the feasibility to develop a consumer product.

Inhaltsverzeichnis

1 Einleitung.....	1
2 Räumliches Hören.....	3
2.1 Allgemeines.....	3
2.1.1 Medianebene, Horizontalebene und Frontalebene.....	3
2.1.2 Azimut, Elevation und Distanz.....	4
2.1.3 Ipsilaterales und kontralaterales Ohr.....	5
2.2 Faktoren zur Bestimmung einer Schallquellenposition.....	5
2.2.1 Monaurale Faktoren.....	5
2.2.2 Binaurale Faktoren.....	6
2.2.3 Weitere Begriffsdefinitionen.....	8
3 Modellierung von HRTF-Kurven.....	10
3.1 Implementierung von HRTF–Filtern mittels direkten FIR- und IIR–Filtern.....	10
3.1.1 FIR Filterdesign.....	11
3.1.2 IIR Filterdesign.....	11
3.1.3 Probleme bei der zeitvarianten Implementation.....	12
3.2 Implementierung von HRTF Kurven mittels Hauptkomponentenmethode.....	13
3.2.1 Prinzip.....	13
3.2.2 Extraktion der spatialen Eigenschaften im Frequenzbereich.....	16
3.2.3 Extraktion der spatialen Eigenschaften im Zeitbereich.....	16
3.2.4 Implementierung der Hauptkomponentenmethode	17
4 AMBISONIC	20
4.1 Ambisonic Grundlagen.....	20
4.1.1 Verwendetes Koordinatensystem.....	21
4.1.2 Ambisonic Kodierung - Allgemein.....	21
4.1.3 Ambisonic Dekodierung – Allgemein	25
4.1.4 Ambisonic Kodierung und Dekodierung im 2D – Fall.....	28
4.2 Binaurales Wiedergabesystem (2D).....	31
4.3 Optimierungen bezüglich der DSP - Implementation eines 2D Binauralsystems.....	34
4.3.1 Anzahl der virtuellen Lautsprecher.....	34
4.3.2 Positionierung der virtuellen Lautsprecher	35
4.3.3 Fensterung der Ambisonic-Kanäle bei der Dekodierung.....	37
4.3.4 Länge der HRIR-Filter.....	43
4.4 Erweiterungen eines binauralen 2D Wiedergabesystems.....	43
4.4.1 Akustische Linse.....	43
4.4.2 Translatorische Bewegung.....	46
4.4.3 Raumsimulation.....	50

5	Objektives Fehlermaß.....	54
5.1	Berechnung von Lokalisationsfehlern.....	54
5.1.1	Berechnung der Interauralen Laufzeitdifferenz.....	55
5.1.2	Lokalisation aufgrund der Interauralen Laufzeitdifferenzen.....	57
5.1.3	Berechnung der Interauralen Pegeldifferenz.....	58
5.1.4	Lokalisation aufgrund der Interauralen Pegeldifferenzen.....	59
5.1.5	Zusammenführung der Lokalisationsparameter.....	59
5.1.6	Lokalisationsunschärfe.....	61
5.1.7	Mittlere Fehler.....	61
5.1.8	Zusammenfassung der Berechnungen.....	62
5.2	Betrachtung und Analyse ausgesuchter Systeme.....	62
5.2.1	Ordnung des Ambisonic-Systems und Anzahl der Lautsprecher.....	63
5.2.2	Vergleich der Referenz-HRIRs (Kemar vs. AKG).....	67
5.2.3	Auswirkung unterschiedlicher Lautsprecheranordnung.....	71
5.2.4	Auswirkung der Gewichtung der Ambisonic-Ordnungen.....	73
6	Hörversuche.....	77
6.1	Einleitung.....	77
6.2	Hörversuchsdesign.....	77
6.3	Versuchspersonen.....	79
6.4	Verwendete Testsignale.....	79
6.5	Versuchsaufbau und Hardware.....	79
6.6	Versuchsablauf (chronologisch).....	82
6.7	Mathematische Auswertung der Hörversuche.....	83
6.7.1	Aufbereiten der Daten.....	83
6.7.2	Vorne/Hinten Ambiguität.....	83
6.7.3	Ambiguitätsbefreiung der Daten.....	84
6.7.4	Berechnen der Streuparametermatrix.....	85
6.7.5	Bewertung der Offsetmessung.....	85
6.7.6	Auswertung der verbalen Befragung.....	86
6.8	Ergebnisse.....	87
6.8.1	Beurteilung der Versuchspersonen.....	87
6.8.2	Unterschiede zwischen den Settings.....	88
6.8.3	Auswertung der Ambiguität.....	90
6.8.4	Auswertung der verbalen Beschreibung.....	92
7	Implementation des Ambisonic-Algorithmus.....	96
7.1	Motorola DSP56366.....	96
7.1.1	Der Prozessorkern.....	97
7.1.2	Die Speicherkonfiguration.....	98
7.1.3	Peripheriebausteine.....	98

7.1.4	Blockschaltbild.....	99
7.2	Evaluation Module DSP56367EVM.....	100
7.3	Die Benutzerschnittstelle.....	102
7.3.1	Hardware.....	102
7.3.2	Bedienung.....	104
7.4	Audio-Routing.....	105
7.5	Software.....	106
7.5.1	Allgemeiner Aufbau.....	106
7.5.2	Audio-Treiber.....	107
7.5.3	User Interface.....	108
7.5.4	Ambisonic-Algorithmus.....	109
7.6	Rechenleistung.....	115
7.6.1	Abschätzung des voraussichtlichen Bedarfes.....	115
7.6.2	Abschätzung der DSP-Belastung.....	116
7.6.3	Messung der DSP-Belastung für das implementierte System.....	116
8	Zusammenfassung und Ausblick.....	118
8.1	Zusammenfassung.....	118
8.2	Ausblick.....	119
	Anhang A - MAJNOI_Toolbox für MATLAB.....	122
	Anhang B - Berechnung der Ambisonic-Systeme.....	144
	Anhang C - Hörversuche.....	162
	Anhang D - User Interface.....	170

1 Einleitung

Die Darstellung virtueller Schallquellen mittels binauraler Wiedergabetechniken stellt einen sehr rechenintensiven Prozess dar. Erst der rasche Anstieg der zur Verfügung stehenden Rechenleistung hat die Entwicklung der Binauraltechnik in Form von Echtzeitsystemen ermöglicht.

Um eine virtuelle Hörwahrnehmung generieren zu können, muss zuerst eine Analyse der tatsächlichen Wahrnehmung erfolgen. Aus der Analyse werden Hörparameter extrahiert. Der Syntheseprozess besteht nun in der Zusammenfügung der extrahierten Parameter. Die begrenzte Genauigkeit der Analyse, sowie die Vereinfachung der verwendeten Algorithmen aufgrund der zur Verfügung stehenden Rechenleistung, führen zu Artefakten in der synthetisierten Abbildung der Schallquelle. Unter Zuhilfenahme mathematischer Modelle sowie Simulationstechniken kann eine Optimierung der zur Verfügung stehenden Parameter bezüglich der erzeugten Hörwahrnehmung erfolgen. Die Verifizierung der Steuerbarkeit der erzeugten Wahrnehmung kann nur anhand von Hörversuchen durchgeführt werden. Prinzipiell entsteht die Diskrepanz zwischen der Optimierung hinsichtlich der Abbildungsgenauigkeit und der zur Verfügung stehenden Rechenleistung. Laut Begault [4] kann eine optimale Lokalisation erzielt werden, wenn die synthetisierten binauralen Signale die gewohnte Hörwahrnehmung in natürlicher Umgebung so gut wie möglich reproduzieren. Folgende Ansätze bei der Implementierung dienen der Erzeugung einer natürlichen Hörwahrnehmung in binauralen Wiedergabesystemen:

- Implementierung eines Headtrackes, damit bei Drehung des Kopfes die Position der virtuellen Schallquelle konstant bleibt¹.

¹ Die menschliche Hörwahrnehmung erlaubt eine Verbesserung der Lokalisation von Schallereignissen mittels kleiner Kopfdrehungen (vgl. [5], [50], [4])

- Verwendung von individuellen HRIR²-Kurven zur Filterung der Signale
- Simulierung von Räumlichkeit durch Implementierung von frühen Reflexionen und Diffushall

In der folgenden Diplomarbeit erfolgt die Entwicklung eines binauralen Wiedergabesystems. Dabei erfolgt die Verwendung verschiedener nicht individualisierter HRIR-Kurven. Die Implementierung eines Headtrackers führt auf die Problematik der zeitvarianten Interpolation zwischen den verschiedenen HRTF-Kurven. Eine Lösungsmöglichkeit bietet die Kodierung der Signale in Ambisonic-Signale mittels zeitinvarianter Filter, wobei die Kopfdrehung mittels einer zeitvarianten Rotationsmatrix berücksichtigt wird.

Weiters erfolgt eine Optimierung des binauralen Wiedergabesystems anhand eines theoretischen Fehlermaßes. Die Verifizierung der Aussagekraft dieses theoretischen Fehlermaß-Modells erfolgt mit Hörversuchen.

Abschließend erfolgt eine DSP-Implementierung des optimierten Algorithmus.

² HRIR (head related impulse responses) bzw. Außenohrübertragungsfunktionen: Diese beschreiben die Filterwirkung der Ohrmuschel (pinna), des Oberkörpers (torso) usw. in Abhängigkeit vom Einfallswinkel der Schallquelle.

2 Räumliches Hören

Das Erkennen der Position eines Schallereignisses beruht bei der menschlichen Hörwahrnehmung auf vielen Faktoren und stellt einen sehr komplexen Vorgang dar. Im Folgenden soll ein kurzer Überblick über die wichtigsten Faktoren gegeben werden, wobei die Betrachtung lediglich aus der Sicht der physikalischen Akustik sowie der Psychoakustik erfolgt. Auf neurophysiologische Modelle wird hier nicht näher eingegangen.

2.1 Allgemeines

Das kopfbezogene Koordinatensystem (vgl. [6]) wird eingeführt, sowie die notwendigen Begriffe und Bezeichnungen definiert.

2.1.1 Medianebene, Horizontalebene und Frontalebene

Die *Medianebene* bezeichnet jene Ebene im Raum, die durch den gleichen Abstand von linkem und rechtem Ohr gekennzeichnet ist. Als *Horizontalebene* wird jene waagrechte Raumebene verstanden, die durch beide Ohren verläuft, hingegen bezeichnet die *Frontalebene* die vertikale Raumebene die durch beide Ohren verläuft.

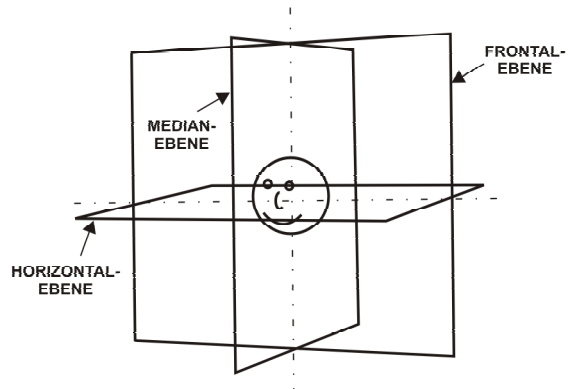


Abb. 2.1: Bezeichnung der räumlichen Bezugsebenen
(vgl. [25])

2.1.2 Azimut, Elevation und Distanz

- *Azimutwinkel:* Winkel in der Horizontalebene
- *Elevationswinkel:* Winkel in der Vertikalebene
- *Distanz:* Abstand zum Ursprung

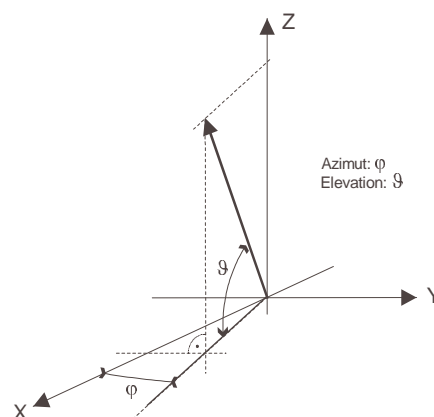


Abb. 2.2: Definition der Winkelbezeichnungen

Der Elevationswinkel wird bei Beschreibungen der Hörwahrnehmung von der Horizontalebene aus gemessen [6]. Im sphärischen Koordinatensystem (wichtig bei der Ableitung der Kodierungs- und

Dekodierungsvorschriften eines Ambisonic-Systems) wird der Elevationswinkel von der Z-Achse aus gemessen (vgl. Kap. 4.1.1).

2.1.3 Ipsilaterales und kontralaterales Ohr

Unter *ipsilateralem Ohr* wird das dem Schallereignis zugewandte Ohr verstanden, zu dem der Direktschall den kürzeren Weg besitzt. Als *kontralaterales Ohr* wird hingegen das dem Schallereignis abgewandte Ohr verstanden.

2.2 Faktoren zur Bestimmung einer Schallquellenposition

Eine systematische Behandlung akustischer sowie psychoakustischer Faktoren der Positionswahrnehmung geht von der Betrachtung nur einer Schallquelle aus. Durch die Linearität der Schallfeldgleichungen können die durch mehr Schallquellen entstehenden Signale überlagert werden, sofern diese dekorreliert sind. Eine Untersuchung der Faktoren bei korrelierten Signalen erfolgt getrennt. Die Positionswahrnehmung beruht auf der Qualität der Signale am jeweiligen Ohr (monaurale Faktoren) sowie auf den Unterschieden zwischen den Signalen an ipsilateralem und kontralateralem Ohr (binaurale Faktoren). Eine Beschreibung der Filterwirkung der Ohrmuschel (pinna), sowie weiterer Einflussfaktoren wie Oberkörper (torso) usw. erfolgt durch die sogenannte Außenohrübertragungsfunktion. Mit HRTF (*Head Related Transfer Function*) wird die Außenohrübertragungsfunktion in der Frequenzdomäne, mit HRIR (*Head Related Impulse Response*) in der Zeitdomäne bezeichnet.

2.2.1 Monaurale Faktoren

Monaurale Faktoren beschreiben Änderungen, die an beiden Ohren gleich sind. Die Änderung der Schallquellenposition führt zu einer Änderung des Amplitudenspektrums der Außenohrübertragungsfunktion im vorwiegend hochfrequenten Bereich. Ausreichender Frequenzinhalt des Quellspektrums sowie eine gute Kenntnis der spektralen Beschaffenheit der Schallquelle sind demnach Voraussetzung, um diesen Faktor zur Positionswahrnehmung durch den Menschen nutzen zu können.

2.2.2 Binaurale Faktoren

Binaurale Faktoren beschreiben die Unterschiede der HRTF- bzw. HRIR-Kurven von ipsi- und kontralateralem Ohr. Der Beitrag dieser Faktoren zur Positionswahrnehmung ist daher vom Quellspektrum weitgehend unabhängig. Jedoch gilt auch hier, dass bekannte Signale zu einer genaueren Lokalisation führen.

Interaurale Zeitdifferenzen (ITD, *Interaural Time Difference*)

Die Beugung tieffrequenter Schallwellen am Kopf führt zu Laufzeitunterschieden zwischen ipsi- und kontralateralem Ohr. Daher wird die ITD vorwiegend bei Frequenzen $< 1,5$ kHz wirksam, d.h. wenn die Wellenlänge größer als der Kopfdurchmesser ist ($\lambda > d_{\text{Kopf}}$). In Frequenzbereichen in denen die Phasenlage nicht mehr eindeutig ist, wird die ITD der Einhüllenden wahrgenommen.

Interaurale Pegeldifferenzen (ILD, *Interaural Level Difference*)

Durch Abschattung der Schallwelle am Kopf entstehen zwischen den Signalen an ipsi- und kontralateralem Ohr Pegeldifferenzen. Bei Differenzen > 25 dB wird die Schallquelle ganz aus ipsilateraler Richtung wahrgenommen [6]. Kleine Wellenlängen bezüglich des Kopfdurchmessers ($\lambda < d_{\text{Kopf}}$) sind Voraussetzung zur Entstehung von Abschattungen, daher werden ILDs vorwiegend bei Frequenzen ab 1,5 kHz wirksam.

Präzedenzeffekt (Haas Effekt, Gesetz der 1. Wellenfront)

Der Präzedenzeffekt stellt einen wichtigen Mechanismus dar, welcher in reflexionsbehafteten Räumen die Lokalisation einer Schallquellenposition aufgrund der ITD unterstützt. ITDs von 0 bis 0,6 ms führen zu einer Verschiebung der Positionswahrnehmung in Richtung des ipsilateralen Ohres. Zwischen etwa 0,6 und 35 ms wird die Schallquelle nur an der Seite des zuerst ankommenden Schalls geortet (Präzedenzeffekt), es kommt jedoch zu einer Quellaufweitung. Bei ITDs größer als 35 ms tritt eine Echowirkung ein [21].

Vorne / Hinten Ambiguität

Eine Betrachtung des Kopfes anhand eines Kugelmodells ergibt eine Mehrdeutigkeit der ITD (*Cone of Confusion*), welche über den Frequenzbereich annähernd gleich bleibt. Die dadurch entstehende Vorne-Hinten Ambiguität wurde anhand von Hörversuchen nachgewiesen [50]. Da jedoch die Mehrdeutigkeit der ILD über den Frequenzbereich nicht konstant ist (größter Unterschied zwischen 3 kHz und 7 kHz), kann die ILD als unterstützender Lokalisationsfaktor zur Auflösung der Vorne-Hinten Verwechslung herangezogen werden [6].

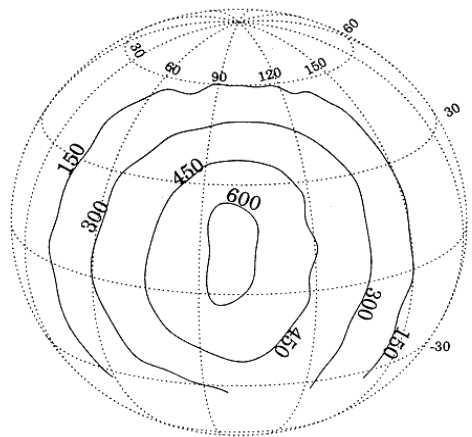


Abb. 2.3: Kurven gleicher ITD (in μs) [20]

Auswirkung spontaner Kopfbewegungen

Die unwillkürliche Reaktion des Menschen auf Lokalisationsunsicherheiten besteht in einer leichten Drehung des Kopfes („Peilbewegungen“), wobei die Azimut-Drehbewegung am häufigsten vorkommt ([4], [6], [50]). Kopfbewegungen relativ zur Schallquelle führen, aufgrund der Abhängigkeit der HRTFs bzw. HRIRs von der Schallquellenposition, zu Änderungen der Signale an den Ohren. Diese Änderung spielt als zusätzlicher Lokalisationsfaktor, vor allem zur Auflösung der Vorne-Hinten Ambiguität (Abb. 2.4), eine wesentliche Rolle ([4], [46], [47]). In einer weiteren Studie konnte nachgewiesen werden, dass lediglich von der Versuchsperson kontrollierte Bewegungen des Kopfes oder der Quelle zu einer signifikanten Verbesserung der Vorne/Hinten Ambiguität führt. Die Bewegung der Quelle selbst (nicht von der Versuchsperson kontrolliert) führt zu keiner Änderung [50].

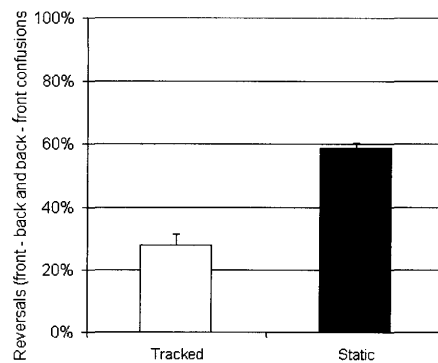


Abb. 2.4: Einfluss der spontanen Kopfbewegung zur Auflösung der Vorne-Hinten Ambiguität [4]

2.2.3 Weitere Begriffsdefinitionen

Minimal hörbare Winkel (MAA, *Minimum Audible Angle*)

Der MAA beschreibt den kleinsten Winkel bei dem *zwei* Schallquellen getrennt voneinander wahrgenommen werden können ([6], [20]), wobei eine starke Abhängigkeit vom Quellspektrum besteht.

Minimal hörbare Bewegungswinkel (MAMA, *Minimum Audible Movement Angle*)

Der MAMA beschreibt die minimale Winkelauslenkung *einer* pendelnden Schallquelle, bei der die Richtung der Bewegung wahrgenommen werden kann [20]. Wobei eine starke Abhängigkeit von der Geschwindigkeit besteht.

Absolute Lokalisation – Lokalisationsunschärfe (*Localisation Blur*)

Beschreibt die Genauigkeit der Lokalisation *einer* Schallquelle (nicht auf eine Referenzquelle bezogen). Die Lokalisationsunschärfe hängt sehr stark vom Quellspektrum ab. Abb. 2.5 zeigt als Beispiel die Lokalisationsunschärfe bei weißen Rauschimpulsen mit 100ms Dauer [6].

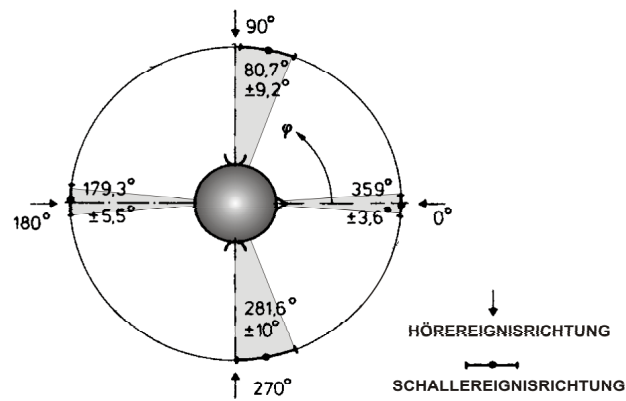


Abb. 2.5: Lokalisationsunschärfe (Blur) bei weißen Rauschimpulsen mit 100ms Dauer [6]

3 Modellierung von HRTF-Kurven

Im Folgenden Kapitel wird die direkte Implementation von HRTF-Kurven mittels FIR bzw. IIR Filtern kurz beschrieben³. Dabei entsteht bei der Implementierung bewegter Schallquellen das Problem der zeitvarianten Filterung. Eine Lösung bietet die Hauptkomponentenmethode (PCA, *Principal Components Analysis*), welche eine recheneffiziente Methode zur Nachbildung von HRTF-Kurven darstellt. Die Implementation zeitvarianter Filter zur Darstellung bewegter Quellen ist dabei nicht notwendig. Ein kurzer Überblick über die Berechnung, sowie über verschiedene Möglichkeiten der Implementation der Hauptkomponentenmethode wird gegeben.

3.1 Implementierung von HRTF-Filtern mittels direkten FIR- und IIR-Filtern

Folgende Eigenschaften werden laut [16] beim Design von Synthesefiltern genutzt (vgl. dazu auch [8] und [23]):

- Lineare Filter können in minimalphasige Filter und Allpaßfilter (*{magnitude, excess-phase}* Darstellung) zerlegt werden
- Annäherung des Allpaßfilters durch eine einfache Zeitverzögerung. Diese Vereinfachung gilt bei HRTF-Kurven in einem Frequenzbereich bis zu 10 kHz, da in diesem Bereich die 'excess-phase' eine lineare Funktion der Frequenz darstellt.

³ dieses Kapitel folgt vorwiegend [16]

3.1.1 FIR Filterdesign

Bei der direkten Implementation von HRTF-Kurven mittels FIR Filtern wird wie folgt vorgegangen:

- {magnitude, excess-phase} – Darstellung der HRTF-Kurven
- Näherung der excess-phase mittels einfacher Zeitverzögerungsglieder
- Diffusfeldentzerrung des Amplitudengangs
- Minimalphasige Rekonstruktion der FIR-Filterantwort
- Approximieren der ursprünglichen HRTF durch Kaskadierung der Zeitverzögerungsglieder und des minimalphasigen Filters

3.1.2 IIR Filterdesign

Bei der direkten Implementation von HRTF-Kurven mittels IIR Filter wird wie folgt vorgegangen:

- {magnitude, excess-phase} – Darstellung der HRTF-Kurven
- Näherung der excess-phase mittels einfacher Zeitverzögerungsglieder
- Diffusfeldentzerrung des Amplitudengangs
- Glätten des Amplitudenspektrums unter Verwendung frequenzproportionaler Hann-Fenster (Constant Q smoothing)
- Frequenzverzerren des Amplitudengangs entsprechend der Bilineartransformation (Warping)
- Berechnen einer Näherung der Übertragungsfunktion mittels Yule-Walker-Algorithmus
- Die Anwendung der Bilineartransformation ergibt die Übertragungsfunktion (Dewarping)

Um psychoakustischen Effekten der Hörwahrnehmung gerecht zu werden, kann die Frequenzverzerrung so gewählt werden, dass die neue Frequenzachse annähernd der Barkskala [6] entspricht. Der Verlust an Genauigkeit bei hohen Frequenzen wird für eine bessere Näherung bei tiefen Frequenzen in Kauf genommen.

Eine Approximation von FIR-Filtern mittels IIR-Filtern wird in [8] vorgestellt. Auf diese alternative Designmethode wird hier jedoch nicht näher eingegangen.

3.1.3 Probleme bei der zeitvarianten Implementation

Bei der binauralen Abbildung bewegter Schallquellen ergibt sich das Problem der zeitvarianten Implementation der HRIRs bzw. HRTFs. Dies erfordert eine Interpolation perceptiver Merkmale. Folgende zwei Prozesse bei der zeitvarianten Implementation können unterschieden werden:

Interpolation:

Da die gemessenen HRTFs bzw. HRIRs nur in begrenzter Winkelauflösung vorliegen, muss zur Erhöhung der Auflösung des Gesamtsystems zwischen jeweils zwei nebeneinanderliegenden Kurven interpoliert werden. Bei Verwendung der Hauptkomponentenmethode erfolgt die Interpolation zwischen den Richtungsübertragungsfunktionen (DTFs, vgl. Kap. 3.2.1).

Kommutation:

Bei einem zeitvarianten Filterungsvorgang werden zwischen zwei aufeinander folgenden Samples die Filterkoeffizienten aktualisiert. Dies kann jedoch zu hörbaren Artefakten in Folge von Unstetigkeiten führen („Klicks“).

Um dies zu vermeiden, werden die Filterkoeffizienten nicht an einem Zeitpunkt aktualisiert, sondern in Richtung der Zielkoeffizienten stetig inkrementiert bzw. dekrementiert. Die Quellposition wird an genau definierten Zeitpunkten $[T_S, 2T_S, \dots, m \cdot T_S]$ aktualisiert, wodurch eine Diskretisierung der Quellbewegung entsteht. Entsprechend der jeweiligen Quellposition werden die zugehörigen Filterkoeffizienten bestimmt. Innerhalb des gewählten Zeitintervalls ΔT_{Change} sollten die neuen Filterkoeffizienten erreicht werden. Da die Aktualisierung der Filterkoeffizienten nur zwischen der Berechnung zweier aufeinander folgender Samples, d.h. im Zeitraum ΔT_{Update} , stattfinden kann, ergeben sich

$$N = \frac{\Delta T_{Change}}{\Delta T_{Update}} \quad (3.1)$$

Aktualisierungsschritte. Der Differenzenquotient der neuen und momentanen Filterkoeffizienten ergibt die elementaren Dekrementierungs- bzw. Inkrementierungsschritte,

$$\Delta_i = \frac{k_i(t) - k_i(t - T_s)}{N} \quad (3.2)$$

die der zyklischen Aktualisierung der momentanen Filterkoeffizienten dienen:

$$k_i(t+1) = k_i(t) + \Delta_i \quad (3.3)$$

Daraus ist ersichtlich, dass die Filterwirkung der Quellbewegung somit auch von den Parametern Δ_i und ΔT_{Change} abhängt. Bei sehr groß gewählten Intervallen ΔT_{Change} entstehen kleine Dekremente bzw. Inkremente, d.h. die Kommutation erfolgt langsam. Dies verhindert bei abrupten Änderungen der Filterkoeffizienten das Erreichen der Quellposition innerhalb des Zeitintervalls ΔT_s . Es entsteht eine Tiefpasswirkung der Quellbewegung. Für klein gewähltes ΔT_{Change} wird die Kommutation sehr schnell durchgeführt, d.h. die HRTF der Quellposition wird innerhalb ΔT_s erreicht. Die großen Aktualisierungsschritte können jedoch hörbare Artefakte hervorrufen. Durch die Vorgabe eines maximalen Δ_i lassen sich jegliche Artefakte ausschließen, die Filterwirkung der Quellbewegung ist nicht mehr konstant – bei schnelleren Bewegungen steigt die Zeitkonstante des Systems.

3.2 Implementierung von HRTF Kurven mittels Hauptkomponentenmethode

3.2.1 Prinzip

Aus einem Satz von gemessenen Außenohrübertragungsfunktionen werden Basisvektoren extrahiert, deren richtungsabhängige Linearkombination die modellierten HRTFs bzw. HRIRs ergeben. Die Gewichte der Linearkombination repräsentieren die Richtungskomponenten, die Basisvektoren die Frequenzkomponenten.

Zuerst wird aus den gemessenen Außenohrübertragungsfunktionen der Mittelwert gebildet,

$$\bar{\mathbf{h}} = \frac{1}{N} \sum_{k=1}^n \mathbf{h}_k \quad (3.4)$$

wobei \mathbf{h}_k die k-te HRTF bzw. HRIR darstellt. Der Vektor $\bar{\mathbf{h}}$ enthält somit die richtungsunabhängigen Eigenschaften, die allen HRIRs bzw. HRTFs gemein sind. Durch Subtraktion dieses Mittelwertes von den jeweiligen Außenohrkurven können die Richtungsübertragungsfunktionen \mathbf{d}_k im Zeitbereich (DIRs, *Directional Impulse Responses*), wie auch im Frequenzbereich (DTFs, *Directional Transfer Functions*), berechnet werden. Diese beinhalten primär richtungsabhängige Information. Die Wahl des Datenformats wirkt sich auf die zu implementierende Filterstruktur aus. Durch Zusammenführen der DTF mit dem Mittelwert kann die ursprüngliche HRTF wieder synthetisiert werden.

$$\mathbf{h}_k = \mathbf{d}_k + \bar{\mathbf{h}} \quad (3.5)$$

Die Berechnung der Autokorrelationsmatrix \mathbf{S} liefert ein Maß für die Ähnlichkeit zweier DIRs bzw. DTFs. \mathbf{S} berechnet sich aus der $[p \times n]$ Matrix \mathbf{D} , deren Spalten jeweils die DIR bzw. DTF (mit p Samples) einer bestimmten Richtung enthalten, n ist die Anzahl der gemessenen Richtungen.

$$\mathbf{S} = \frac{1}{n} \mathbf{D} \mathbf{D}^H \quad H \dots \text{hermitesch} \quad (3.6)$$

Da die Autokorrelationsmatrix \mathbf{S} symmetrisch und nichtnegativ definit ist, gibt es p orthogonale Eigenvektoren \mathbf{c}_i (EFs, *Eigen Transfer Functions*) mit den korrespondierenden reellen und nichtnegativen Eigenwerten λ_i (SCFs, *Spatial Characteristic Transfer Functions*). Die orthonormale Transformationsmatrix \mathbf{C} berechnet sich nun aus den q Eigenvektoren, die mit den größten Eigenvektoren korrespondieren.

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_p \geq 0 \quad \Leftrightarrow \quad \mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \dots \ \mathbf{c}_q] \quad (3.7)$$

Der Gewichtsvektor \mathbf{w}_k berechnet sich mittels der zeitdiskreten Karhunen-Loève Transformation bzw. Hotelling Transformation (KLT) wie folgt:

$$\mathbf{w}_k = \mathbf{C}^T \mathbf{d}_k \quad (3.8)$$

Die KLT minimiert zugleich den quadratischen Fehler bei gegebener Zahl an Basisvektoren.

Für die Bedingung $q=p$ erfüllt die reell bzw. komplexwertige Matrix \mathbf{C}^T die Orthogonalitätsbedingung

$$[\mathbf{C}^T]^T = [\mathbf{C}^T]^{-1} \text{ bzw. } [\mathbf{C}^T]^H = [\mathbf{C}^T]^{-1} \quad (3.9)$$

Nach weiterer Umformung ergibt sich \mathbf{d}_k somit als gewichtete Summe der Basisvektoren, wobei das Gleichheitszeichen nur für $q=p$ gilt. Da in der Praxis $q \ll p$ ist, stellt folgende Gleichung nur eine Näherung dar, wobei sich mit steigender Anzahl der verwendeten Basisvektoren eine geringere Abweichung ergibt.

$$\mathbf{d}_k = \mathbf{C} \mathbf{w}_k \quad (3.10)$$

Das Ohrsignal \mathbf{o} für die Schallquellenposition (θ_k, ϕ_k) erhält man durch Filterung des Signals mit der synthetisierten Außenohrübertragungsfunktion. Es ergeben sich folgende Filterstrukturen, wobei der Operator \circ in der mathematischen Darstellung, je nach Betrachtung im Frequenz- bzw. Zeitbereich, einer Multiplikation bzw. Faltung entspricht.

Für HRIRs und lineare HRTFs gilt:

$$\mathbf{o} = (\bar{\mathbf{h}} + \mathbf{d}_k) \circ \mathbf{s} = \bar{\mathbf{h}} \circ \mathbf{s} + \sum_{i=1}^q w_i(\theta_k, \phi_k) (\mathbf{c}_i \circ \mathbf{s}) \quad (3.11)$$

wobei \mathbf{s} das zu spatialisierende Eingangssignal darstellt.

Für logarithmierte HRTFs gilt mit $\mathbf{h}_{k,lin} = \overline{\mathbf{h}_{lin}} \circ \mathbf{d}_{k,lin}$ die Beziehung,

$$\mathbf{o} = \mathbf{s} \circ \mathbf{h}_{lin} \circ \mathbf{d}_{k,lin} = W_i(\theta_k, \phi_k) (\mathbf{s} \circ \mathbf{h}_{lin} \circ \mathbf{c}_i) \quad (3.12)$$

wobei die Filterung des Signals mit \mathbf{c}_i im voraus berechnet werden kann, da \mathbf{c}_i nur frequenz- und nicht richtungsabhängig ist.

3.2.2 Extraktion der spatialen Eigenschaften im Frequenzbereich

Ausgehend von komplexwertigen HRTFs (Amplitude und Phase) erfolgt die Modellierung der Außenohrübertragungsfunktionen im Frequenzbereich⁴,

$$\mathbf{h}_m(\theta_k, \phi_k) = \sum_{i=1}^q w_i(\theta_k, \phi_k) \mathbf{c}_i + \mathbf{c}_0 \quad (3.13)$$

wobei $\mathbf{h}_m(\theta_k, \phi_k)$ die modellierte HRTF für die Richtung (θ_k, ϕ_k) darstellt, \mathbf{c}_i [$i=0, \dots, q$] die Eigenübertragungsfunktion und $w_i(\theta_k, \phi_k)$ die zugehörigen Gewichtungsfaktoren.

3.2.3 Extraktion der spatialen Eigenschaften im Zeitbereich

Die Modellierung im Zeitbereich erfolgt ausgehend von normalisierten HRIRs⁵:

$$\mathbf{h}(\theta, \phi) = \sum_{i=1}^M \omega_i(\theta, \phi) \mathbf{c}_i + \bar{\mathbf{h}} \quad (3.14)$$

$\mathbf{h}(\theta, \phi)$ stellt die richtungsbezogene HRIR dar, welche aus den mit $\omega_i(\theta, \phi)$ gewichteten Eigenfunktionen \mathbf{c}_i modelliert wird. Die Gewichtungsfaktoren $\omega_i(\theta, \phi)$ für beliebige Richtungen berechnen sich durch Interpolation der w_i der gemessenen Richtungen:

⁴ Dieser Abschnitt folgt [16], vgl. auch [13], [17], [27]

⁵ Dieser Abschnitt folgt [16], vgl. auch [51]

$$\begin{aligned}
\omega_i(\theta, \phi) \approx & \frac{(\theta - \theta_2)(\phi - \phi_2)}{A} w_i(\theta_1, \phi_1) \\
& + \frac{(\theta - \theta_2)(\phi - \phi_1)}{A} w_i(\theta_1, \phi_2) \\
& + \frac{(\theta - \theta_1)(\phi - \phi_2)}{A} w_i(\theta_2, \phi_1) \\
& + \frac{(\theta - \theta_1)(\phi - \phi_1)}{A} w_i(\theta_2, \phi_2)
\end{aligned} \tag{3.15}$$

3.2.4 Implementierung der Hauptkomponentenmethode

Nach der Aufbereitung der Rohdaten [16] kann die Implementierung wie unter 3.2.1 beschrieben, sowohl im Frequenz- als auch im Zeitbereich (vgl. Abb. 3.1, Abb. 4.1) erfolgen. Aufgrund der Minimierung des mittleren quadratischen Fehlers durch die Karhunen-Loève Transformation, erreicht die Implementation mit 5 Basisvektoren 90% der Varianz der HRTF – Amplitude.

Bei der Implementation zusätzlicher Schallquellen bleibt die Anzahl der Filter konstant. Für jede weitere Schallquelle müssen lediglich ein Zeitverzögerungsglied sowie 5 Multiplikatoren (jeweils pro Ohr) hinzugefügt werden (in Abb. 3.1, Abb. 4.1 strichliert dargestellt).

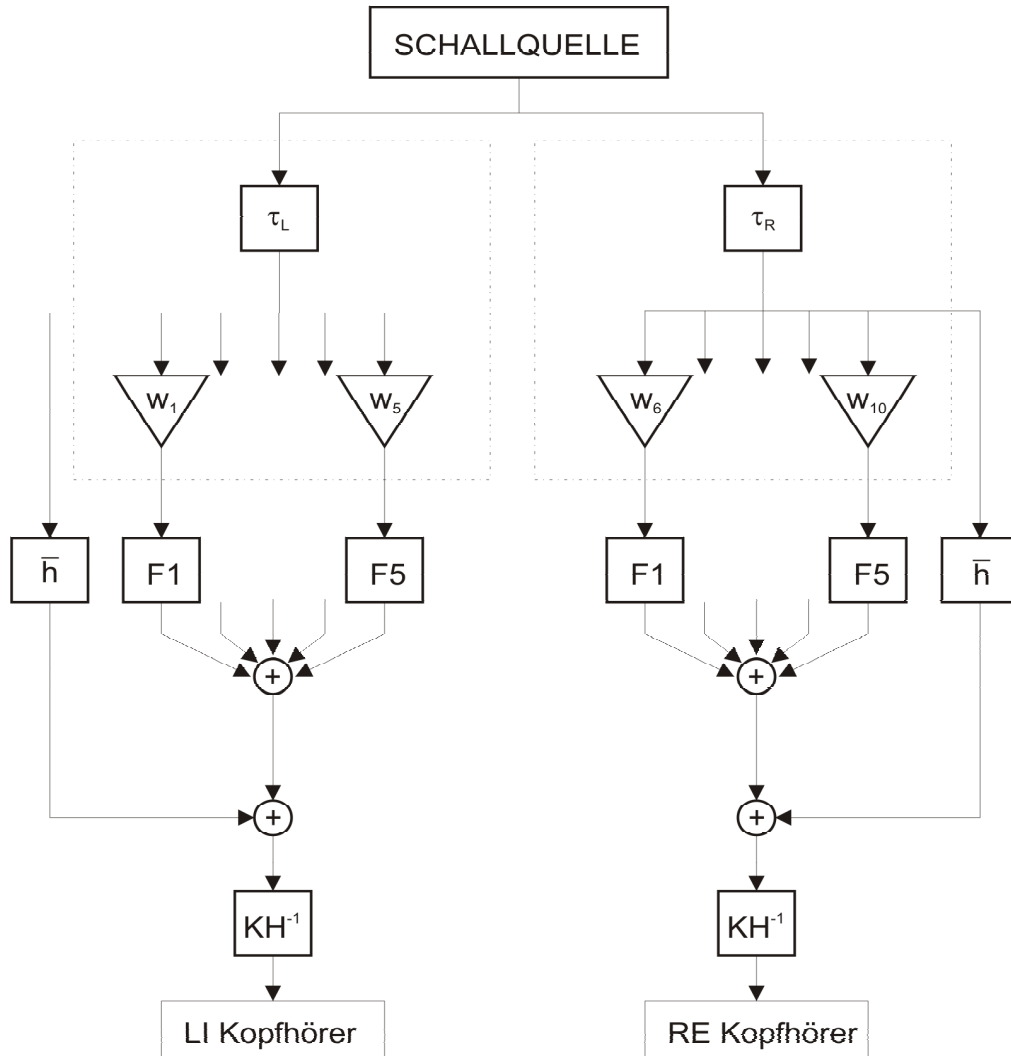


Abb. 3.1: Prinzip der Hauptkomponentenmethode mit linearen Daten im Zeit- und Frequenzbereich [16]. Dabei sind w_i die Gewichte, F_i die Basisvektoren und τ_L und τ_R die ITD für linkes und rechtes Ohr. \bar{h} ist die gemittelte Impulsantwort, KH^{-1} ist der inverse Kopfhörerfrequenzgang. Die Implementation zusätzlicher Schallquellen erfordert ein weiteres Zeitverzögerungsglied und 5 Multiplikatoren pro Ohr (in der Abbildung strichliert eingefasst)

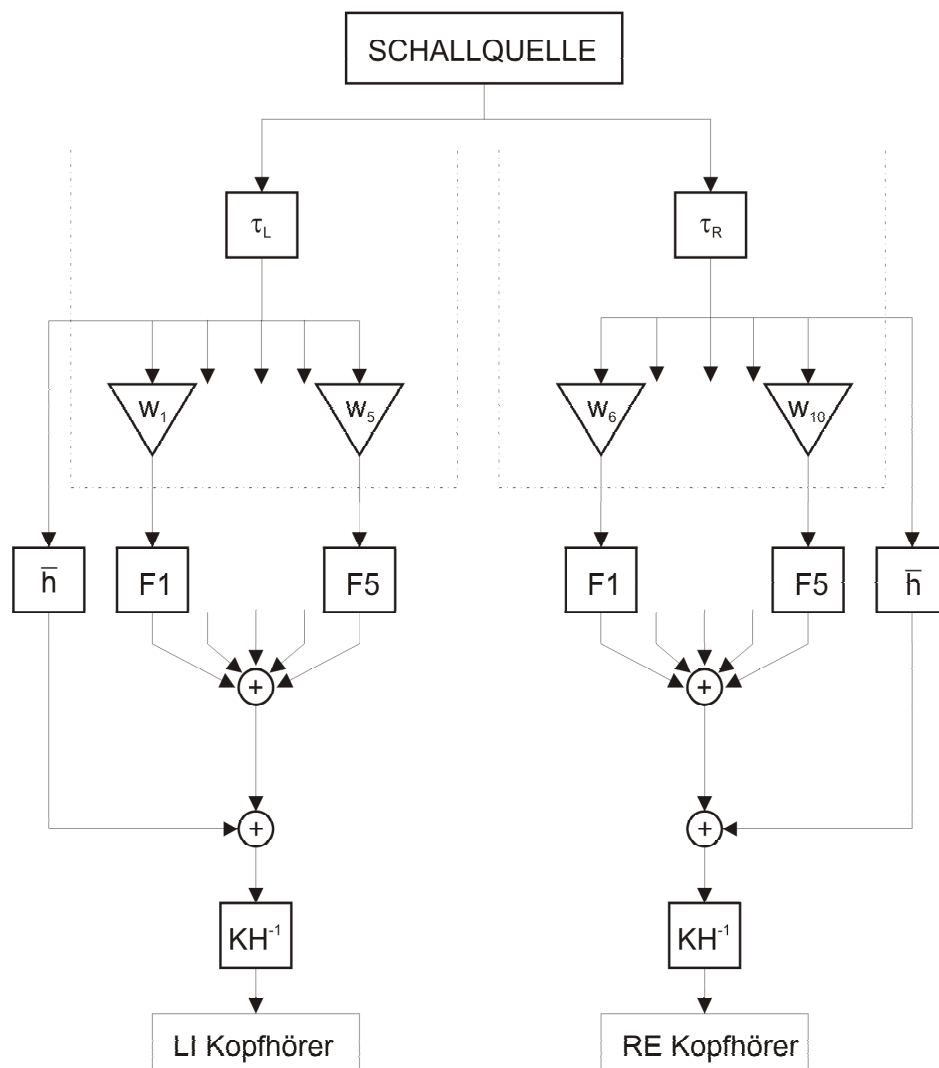


Abb. 4.1: Prinzip der Hauptkomponentenmethode mit logarithmischen Daten im Frequenzbereich [16]. Dabei sind w_i die Gewichte, F_i die Basisvektoren und τ_l und τ_r die ITD für linkes und rechtes Ohr. \bar{h} ist die gemittelte Impulsantwort, KH^{-1} ist der inverse Kopfhörerfrequenzgang. Die Implementation zusätzlicher Schallquellen erfordert ein weiteres Zeitverzögerungsglied und 5 Multiplikatoren pro Ohr (in der Abbildung strichliert eingefasst)

4 AMBISONIC

Wie in Kapitel 3.1.3 gezeigt wurde, führt die Realisierung bewegter Schallquellen mittels direkter Filterung zum Problem der Implementierung zeitvarianter FIR- bzw. IIR-Filter. Dies kann im ungünstigsten Fall durch Unstetigkeiten in der Impulsantwort zu hörbaren Artefakten führen. Als eine Lösungsmöglichkeit wurde in Kapitel 3.2 die Hauptkomponentenmethode vorgestellt.

Eine weitere Lösungsmöglichkeit bietet die Kodierung der Signale in die Ambisonic-Domäne mittels zeitinvarianter Filter, wobei die Quellenbewegung mittels einer zeitvarianten Rotationsmatrix berücksichtigt wird. Die Erzeugung der binauralen Signale erfolgt durch zeitinvariante HRTF- bzw. HRIR-Kurven. Ein wesentlicher Unterschied des Ambisonicansatzes zur Hauptkomponentenmethode besteht darin, dass bei Ambisonic das Schallfeld modelliert wird, nicht die HRTF- bzw. HRIR-Kurven.

4.1 Ambisonic Grundlagen

Ambisonic bietet die Möglichkeit ein Schallfeld in einem Raumpunkt durch entlang einer Kugeloberfläche angeordnete Lautsprecher zu reproduzieren. Der grundlegende Ambisonicansatz beruht auf dem Vergleich einer aufgenommenen Referenzwelle mit der durch die Lautsprecher synthetisierten Welle.

Voraussetzungen für die Gültigkeit des Ambisonicansatzes:

- Die aufgenommene Schallwelle (Referenzwelle) ist eine ebene Welle.
- Die durch einen Lautsprecher erzeugte Schallwelle ist ebenfalls eine ebene Welle.

Diese Annahmen sind gültig, wenn die Schallquelle sowie auch die Lautsprecher weit genug von der Abhörposition entfernt sind. Der Ambisonic-Ansatz geht weiters von einer Freifeldsituation aus, d.h. es erfolgt keine explizite Behandlung von Reflexionen.

Um den Ambisonic-Ansatz auch in binauralen Wiedergabesystemen nutzen zu können, werden virtuelle Lautsprecher positioniert und die jeweiligen Lautsprechersignale mit den entsprechenden Außenohrübertragungsfunktionen gefaltet. Danach erfolgt die Superposition der Teilwellen (siehe Kap. 4.2).

4.1.1 Verwendetes Koordinatensystem

Die mathematische Beschreibung eines Ambisonic-Systems erfolgt in einem sphärischen Koordinatensystem. Dabei gelten folgende Konventionen zur Winkelbezeichnung (siehe Abb. 4.2).

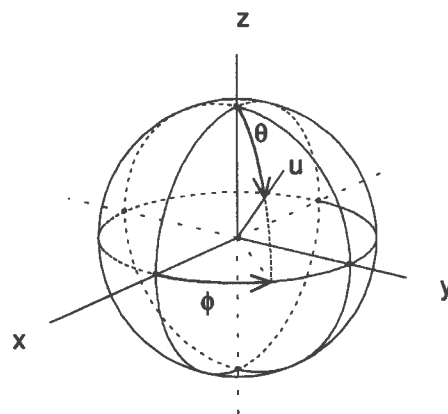


Abb. 4.2: Koordinatensystem zur Berechnung eines Ambisonic Systems [15]

4.1.2 Ambisonic Kodierung - Allgemein

Der Ansatz zur Ableitung der Kodierungsvorschriften eines Ambisonic-Systems besteht in der Lösung der verlustlosen homogenen Wellengleichung in sphärischen Koordinaten [52]:

$$\Delta p(t, \mathbf{r}) - \frac{1}{c^2} \frac{\partial^2 p(t, \mathbf{r})}{\partial t^2} = 0 \quad (4.1)$$

Dabei stellt $p(t, \mathbf{r})$ den Schalldruck, Δ den Laplaceoperator (zweite räumlich Ableitung) in sphärischen Koordinaten (r, ϑ, φ) und c die Ausbreitungsgeschwindigkeit dar. Im homogenen Medium gilt $c = \text{konst.}$.

Als Voraussetzung zur Lösung der Gl. 4.1 nehmen wir an, dass $p(t, \mathbf{r})$ eine zeitlich harmonische Schwingung darstellt:

$$p(t, \mathbf{r}) = p_f(\mathbf{r}) e^{j2\pi f t} \quad (4.2)$$

Die Substitution von Gl. 4.2 in die zeitvariante Wellengleichung (Gl. 4.1) führt zur zeitinvarianten verlustlosen *Helmholtz Wellengleichung*,

$$(\Delta + k^2) p(\mathbf{r}) = 0 \quad (4.3)$$

wobei $p(\mathbf{r})$ der Schalldruck und $k = 2\pi/\lambda$ die Wellenzahl ist.

Die Helmholtz Gleichung lässt sich durch Variablenseparation lösen (vgl. [52], S.269 ff).

$$p(\mathbf{r}) = R(r) \Theta(\vartheta) \Phi(\varphi) \quad (4.4)$$

Für die radial abhängige Variable $R(r)$ ergibt sich folgende Lösung:

$$R(r) = A_{m,\eta}^\sigma \cdot J_m(kr) + B_{m,\eta}^\sigma \cdot N_m(kr) \quad \text{mit: } m=0, 1, 2, \dots \quad (4.5)$$

Dabei bezeichnet J_m die sphärischen Bessel-Funktionen und N_m die sphärischen Neumann-Funktionen jeweils m-ter Ordnung. Die sphärische Neumannfunktion weist eine Singularität bei $kr=0$ auf. Dadurch werden die Koeffizienten $B_{m,\eta}^\sigma$ gleich Null.

Die Lösung für die Elevation ergibt:

$$\Theta(\vartheta) = C_{m,\eta} P_m^\eta(\cos \vartheta) + D_{m,\eta} Q_m^\eta(\cos \vartheta) \quad \text{mit: } m, \eta = 0, 1, 2, \dots \quad (4.6)$$

$P_m^\eta(\cos \vartheta)$ repräsentiert die Legendre Polynome 1. Art, $Q_m^\eta(\cos \vartheta)$ jene 2. Art. Die Koeffizienten $D_{m,\eta}$ werden auch hier gleich Null, da $Q_m^\eta(\cos \vartheta)$ Singularitäten bei $\vartheta=0$ und $\vartheta=\pi$ aufweist.

Der letzte Term der Variablenseparation ergibt die Lösung für die Horizontalebene.

$$\Phi(\varphi) = E_\eta \cos(\eta \varphi) + F_\eta \sin(\eta \varphi) \quad \text{mit: } \eta = 0, 1, 2, \dots \quad (4.7)$$

Beim allgemeinen Ansatz für Ambisonic wurde von ebenen Wellen ausgegangen. Dadurch ergibt sich die Lösung von $R(r)$ zu eins, da keine radiale Abhängigkeit besteht. Somit vereinfacht sich die Lösung mittels Variablenseparation näherungsweise zu:

$$p(\mathbf{r}) \simeq \Theta(\vartheta) \cdot \Phi(\varphi) = Y_{m,\eta}^\sigma(\mathbf{r}) \quad \text{mit: } m, \eta = 0, 1, 2, \dots \quad (4.8)$$

$Y_{m,\eta}^\sigma$ stellt dabei die Kugelflächenfunktion (*spherical harmonics*) dar und bildet ein Set von orthonormalen Basisvektoren.

Mit Hilfe der Kugelflächenfunktionen kann jede beliebige Funktion an einer Kugeloberfläche in eine Reihe von Kugelflächenfunktionen entwickelt werden.

Betrachtet man nun eine ebene Welle mit der Ausbreitungsrichtung \mathbf{u}_p in einem Raumpunkt $\mathbf{r} = r \mathbf{u}_r$, so kann diese wie folgt beschrieben werden:

$$S_k(\mathbf{r}) = p_k e^{j k \mathbf{r}^T \cdot \mathbf{u}_p} \quad (4.9)$$

Mit obigen Betrachtungen kann diese Funktion nun in folgende Reihe entwickelt werden:

$$S_k(\mathbf{x}; k) = \sum_{m=0}^{\infty} \sum_{\eta=-m}^m G_{m,\eta}(\mathbf{x}; k) \cdot Y_{m,\eta}^{\sigma}(\hat{\mathbf{x}}) \quad (4.10)$$

In Gl. 4.10 ist sowohl die Vorschrift für die Kodierung, wie auch die Vorschrift für die Dekodierung eines Ambisonic-Systems implizit enthalten. Die Koeffizienten $G_{m,\eta}$ stellen die Gewichtung der einzelnen Kugelflächenfunktionen bei der Superposition dar und sind vom Einfallswinkel unabhängig. Die Kugelflächenfunktionen stellen somit die Richtungsinformation bei der Synthetisierung der ursprünglichen Schallwelle dar. Für eine einfallende Schallwelle aus der Richtung $\mathbf{u}_p = [\Phi, \Theta]^T$ gilt:

$$B_{m,\eta}^{\sigma} = Y_{m,\eta}^{\sigma}(\Phi, \Theta) p_{\Phi, \Theta} \quad (4.11)$$

wobei folgende Lösungsschar entsteht:

$$Y_{m,\eta}^{\sigma}(\mathbf{r}) = \begin{cases} A_{m,\eta} \cdot P_m^{\eta}(\cos \Theta) \cdot \cos(m\Phi) & \text{für } \sigma = 1 \\ A_{m,\eta} \cdot P_m^{\eta}(\cos \Theta) \cdot \sin(m\Phi) & \text{für } \sigma = -1 \end{cases} \quad (4.12)$$

Da bei der praktischen Umsetzung eine Einschränkung der Zahl der Übertragungskanäle erforderlich ist, tritt bei der Synthese ein Informationsverlust auf, die ebene Welle wird im Betrachtungspunkt nur angenähert.

$$\mathbf{B}_{\Phi, \Theta} = \mathbf{Y}_{\Phi, \Theta} \mathbf{p}_{\Phi, \Theta} \quad (4.13)$$

Die Ambisonic-Kanäle werden wie folgt bezeichnet:

$$\mathbf{B} = \begin{pmatrix} B_{0,0} \\ B_{1,1}^1 \\ B_{1,1}^{-1} \\ B_{1,0}^1 \\ \vdots \end{pmatrix} = \begin{pmatrix} W \\ X^{(1)} \\ Y^{(1)} \\ Z^{(1)} \\ \vdots \end{pmatrix} \quad (4.14)$$

Die Kugelflächenfunktionen berechnen sich somit aus den Legendre-Polynomen wie folgt:

M	m	η	$Y_{m,\eta}^{+1}$	$Y_{m,\eta}^{-1}$
0	0	0	1	0
1	1	0	$\sin \Theta$	0
	1	1	$\cos \Theta \cos \Phi$	$\cos \Theta \sin \Phi$
2	2	0	$\frac{1}{2}(3\sin^2 \Theta - 1)$	0
	2	1	$\frac{3}{2}\sin 2\Theta \cos \Phi$	$\frac{3}{2}\sin 2\Theta \sin \Phi$
	2	2	$\frac{3}{2}\cos^2 \Theta \cos 2\Phi$	$\frac{3}{2}\cos^2 \Theta \sin 2\Phi$

Ambisonic B-Format [18]: Aus Gründen der besseren Energieverteilung für jeden Raumpunkt und für die einzelnen Ambisonic-Kanäle wird der W-Kanal mit $1/\sqrt{2}$ gewichtet.

Da die Kugelflächenfunktionen $Y_{m,0}^{-1} \equiv 0 \quad \forall \eta=1 \wedge \sigma=-1$ sind, ergibt sich die Anzahl L der benötigten Übertragungskanäle für ein Ambisonic-System der Ordnung M

- für den dreidimensionalen Fall

$$L^{3D} = (M+1)^2 \quad (4.15)$$

- für den zweidimensionalen Fall ($\Theta=0$)

$$L^{2D} = 2M+1 \quad (4.16)$$

4.1.3 Ambisonic Dekodierung – Allgemein

Bei der Dekodierung erfolgt eine Verteilung der Ambisonic-Signale auf N Lautsprecher. Die Gewichtung der einzelnen Signale bei der Dekodierung wird nur von der Aufstellung der Lautsprecher und der Anzahl der Ordnung des Ambisonic-Systems (Anzahl der Übertragungskanäle) bestimmt. Somit besteht eine Entkopplung der Kodierung von der Dekodierung.

Eine ebene Welle aus der Richtung $\mathbf{u}_p = [\Phi, \Theta]^T$ kann im Betrachtungspunkt $[r, \varphi, \vartheta]$ in folgende Reihe entwickelt werden [15]:

$$S_{Analyse} = S_{\Phi, \Theta}(r, \varphi, \vartheta) = s \cdot \sum_{m=0}^{\infty} \sum_{\eta=0}^m \sum_{\sigma=\pm 1} (2m+1) j^m J_m(kr) Y_{m,\eta}^{\sigma}(\Phi, \Theta) Y_{m,\eta}^{\sigma}(\varphi, \vartheta) \quad (4.17)$$

Diese ebene Welle wird nun unter Verwendung von N Lautsprechern durch Superposition der abgestrahlten Teilwellen synthetisiert. Bei ausreichender Entfernung der Lautsprecher kann jede Teilwelle als ebene Welle betrachtet werden, die Superposition ergibt somit ebenfalls eine ebene Welle. Jede abgestrahlte Teilwelle eines Lautsprechers der Richtung $[\varphi_n, \vartheta_n]^T$ lässt sich wiederum durch Kugelflächenfunktionen beschreiben (vgl. Gl. 4.17).

$$S_n(r, \varphi, \vartheta) = \rho_n \cdot \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y_{m,\eta}^{\sigma}(\varphi_n, \vartheta_n) Y_{m,\eta}^{\sigma}(\varphi, \vartheta) \quad (4.18)$$

Wobei ρ_n das Signal der jeweiligen Lautsprecher darstellt.

Die einzelnen Teilwellen werden nun überlagert und bilden somit das Synthese-Schallfeld im Betrachtungspunkt.

$$S_{Synthese} = \sum_{n=1}^N S_n(r, \varphi, \vartheta) = \sum_{n=1}^N \rho_n \cdot \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y_{m,\eta}^{\sigma}(\varphi_n, \vartheta_n) Y_{m,\eta}^{\sigma}(\varphi, \vartheta) \quad (4.19)$$

Im Betrachtungspunkt soll nun das ursprüngliche Schallfeld durch das mittels Lautsprecher wiedergegebene Schallfeld resynthetisiert werden.

$$S_{Analyse} \equiv S_{Synthese} \quad (4.20)$$

Die Randbedingungen (*matching conditions*) für die Gültigkeit von Gl. 4.20 erhält man durch Koeffizientenvergleich von Gl. 4.17 und Gl. 4.19,

$$\mathbf{s} \cdot Y_{m,\eta}^\sigma(\Phi, \Theta) = \sum_{n=1}^N \rho_n Y_{m,\eta}^\sigma(\varphi_n, \vartheta_n) \quad (4.21)$$

wobei s den Schalldruck der ursprünglichen Welle darstellt.

In ausgeschriebener Form ergeben sich die sogenannten *matching conditions*:

$$\begin{aligned} \mathbf{s} &= \sum_{n=1}^N \rho_n \\ \mathbf{s} \cdot P_{m,\eta}(\cos \Theta) \cdot \cos(\eta \Phi) &= \sum_{n=1}^N \rho_n \cdot P_{m,\eta}(\cos \vartheta_n) \cdot \cos(\eta \varphi_n) \\ \mathbf{s} \cdot P_{m,\eta}(\cos \Theta) \cdot \sin(\eta \Phi) &= \sum_{n=1}^N \rho_n \cdot P_{m,\eta}(\cos \vartheta_n) \cdot \sin(\eta \varphi_n) \end{aligned} \quad (4.22)$$

wobei für $m=1, \dots, \infty$ und $\eta=1, \dots, m$ gilt.

Es sei nun \mathbf{p} die Menge aller Lautsprechersignale ρ_n und \mathbf{B} die Beschreibung einer ebenen Welle aus der Richtung $[\Phi, \Theta]^T$ in der Ambisonic-Domäne für ein System der Ordnung M .

$$\begin{aligned} \mathbf{p} &= [\rho_1, \rho_2, \dots, \rho_N]^T \\ \mathbf{B} &= [Y_{0,0}^1(\Phi, \Theta), Y_{1,0}^1(\Phi, \Theta), \dots, Y_{M,M}^{-1}(\Phi, \Theta)]^T \cdot \mathbf{s} \end{aligned} \quad (4.23)$$

Somit lassen sich die *matching conditions* (Gl. 4.21) wie folgt darstellen:

$$\mathbf{B} = \mathbf{C} \cdot \mathbf{p} \quad (4.24)$$

mit

$$\mathbf{C} = \begin{bmatrix} Y_{0,0}^1(\varphi_1, \vartheta_1) & Y_{0,0}^1(\varphi_2, \vartheta_2) & \dots & Y_{0,0}^1(\varphi_N, \vartheta_N) \\ Y_{1,0}^1(\varphi_1, \vartheta_1) & Y_{1,0}^1(\varphi_2, \vartheta_2) & \dots & Y_{1,0}^1(\varphi_N, \vartheta_N) \\ \vdots & \vdots & \ddots & \vdots \\ Y_{M,M}^{-1}(\varphi_1, \vartheta_1) & Y_{M,M}^{-1}(\varphi_2, \vartheta_2) & \dots & Y_{M,M}^{-1}(\varphi_N, \vartheta_N) \end{bmatrix} \quad (4.25)$$

Die allgemeine Form der Dekodierung in Matrizenschreibweise lautet

$$\mathbf{p} = \mathbf{D} \cdot \mathbf{B} = \mathbf{C}^{-1} \cdot \mathbf{B} \quad (4.26)$$

Da im Allgemeinen $M \neq N$ ist, kann die Dekodiermatrix nur mit Hilfe der Pseudoinversen berechnet werden.

$$\mathbf{D} = \text{pinv}(\mathbf{C}) = \mathbf{C}^T \cdot (\mathbf{C} \cdot \mathbf{C}^T)^{-1} \quad (4.27)$$

In [18] wird gezeigt, dass für eine eindeutige Abbildung die notwendige Anzahl N der Lautsprecher bei Dekodierung der L Ambisonic-Kanäle.

$$N \geq L \quad (4.28)$$

Die Dekodierung ist bei Ambisonic-Systemen von der Kodierung entkoppelt und wird von der Wahl der Lautsprecheraufstellung bestimmt. Eine Einschränkung bildet die eindeutige Lösbarkeit der Pseudoinversen für die gewählte Lautsprecheraufstellung. Eine äquidistante Aufstellung der Lautsprecher entlang eines Kreises (regelmäßige Polyeder) vereinfacht die Lösung der Dekodiergleichung zu:

$$\mathbf{D}_{\text{Projektion}} = \frac{1}{N} \mathbf{C}^T \quad (4.29)$$

Die Anwendung der vereinfachten Dekodiermatrix auf ein unregelmäßiges Lautsprecherlayout wird *Projektion* der Ambisonic-Kanäle auf die jeweiligen Lautsprecher genannt.

4.1.4 Ambisonic Kodierung und Dekodierung im 2D – Fall

Bei ausschließlicher Betrachtung des zweidimensionalen Falles, lassen sich obige Beziehungen wie folgt vereinfachen (vgl. [3]). Folgende Winkelbezeichnungen werden verwendet:

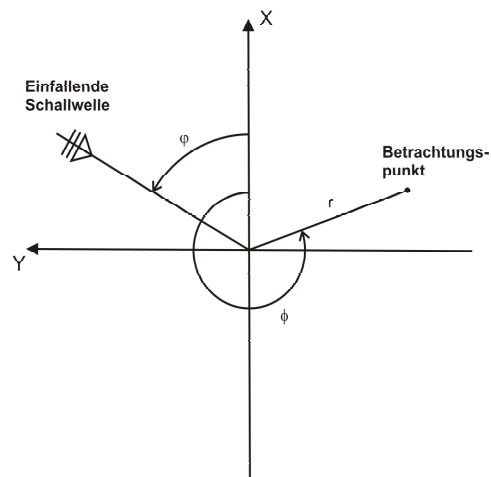


Abb. 4.3: Winkelbezeichnung [3]

Ambisonic Kodierung

Die Reihenentwicklung der Referenz – Schallwelle vereinfacht sich zu:

$$\begin{aligned}
 S_{\text{Ref}}(r, \phi) &= P_{\varphi} e^{jk_r r} = P_{\varphi} J_0(kr) + 2P_{\varphi} \sum_{m=1}^{\infty} j^m J_m(kr) \cos[m(\phi - \varphi)] = \\
 &= P_{\varphi} \left(J_0(kr) + 2 \sum_{m=1}^{\infty} j^m J_m(kr) [\cos(m\varphi) \cos(m\phi) + \sin(m\varphi) \sin(m\phi)] \right) \quad (4.30)
 \end{aligned}$$

Die abgestrahlte Welle eines Lautsprechers läßt sich durch folgende Beziehung ausdrücken, wobei ϕ_n den Winkel des jeweiligen Lautsprechers darstellt.

$$\begin{aligned}
 S_n(r, \phi) &= P_n e^{jk_n r} = \\
 &= P_n \left(J_0(kr) + 2 \sum_{m=1}^{\infty} j^m J_m(kr) [\cos(m\phi_n) \cos(m\phi) + \sin(m\phi_n) \sin(m\phi)] \right) \quad (4.31)
 \end{aligned}$$

Die Reproduktion der Referenzwelle erfolgt durch Superposition der durch die N Lautsprecher abgestrahlten Teilwellen.

$$\begin{aligned}
S_{Ambi}(r, \phi) &= \sum_{n=1}^N S_n(r, \phi) = \sum_{n=1}^N P_n e^{jk_n r} = \\
&= \sum_{n=1}^N P_n \left(J_0(kr) + 2 \sum_{m=1}^{\infty} j^m J_m(kr) [\cos(m\phi_n) \cos(m\phi) + \sin(m\phi_n) \sin(m\phi)] \right)
\end{aligned} \tag{4.32}$$

Da eine Gleichheit zwischen dem Referenzfeld S_{Ref} und dem reproduzierten Schallfeld S_{Ambi} angestrebt wird, d.h.

$$S_{Ref} \equiv S_{Ambi} \quad \forall (r, \varphi) \tag{4.33}$$

erfolgt aus dem Koeffizientenvergleich der beiden Reihenentwicklungen (Gl. 4.30 und Gl. 4.32).

$$\begin{aligned}
P_\varphi &= \sum_{n=1}^N P_n \\
P_\varphi \cos(m\varphi) &= \sum_{n=1}^N P_n \cos(m\phi_n) \\
P_\varphi \sin(m\varphi) &= \sum_{n=1}^N P_n \sin(m\phi_n)
\end{aligned} \tag{4.34}$$

Ein Ambisonic-System der Ordnung M bringt somit die reduzierten⁶ Kugelflächenfunktionen der Ordnung $2M+1$ zur Übereinstimmung. Die Ambisonic-Kanäle lassen sich wie folgt darstellen:

$$\begin{pmatrix} W \\ X^{(1)} \\ Y^{(1)} \\ X^{(2)} \\ Y^{(2)} \\ \vdots \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \cdot P_\varphi \\ P_\varphi \cos(\varphi) \\ P_\varphi \sin(\varphi) \\ P_\varphi \cos(2\varphi) \\ P_\varphi \sin(2\varphi) \\ \vdots \end{pmatrix} \tag{4.35}$$

Wobei W , $X^{(1)}$, $Y^{(1)}$, ... die jeweiligen Ambisonic-Kanäle darstellen. Der Faktor $1/\sqrt{2}$ ergibt sich durch die Konvention des B-Formats (vgl. Kap.4.1.2, sowie [18]).

⁶ Da hier lediglich der 2D-Fall betrachtet wird, d.h. $\vartheta = 90^\circ$ (vgl. Abb. 4.2), kommt es zu einer Eliminierung der Abhängigkeit der Kugelflächenfunktionen von der Elevation.

Ambisonic Dekodierung

Bei einer Anordnung von N Lautsprechern entlang eines Kreises, berechnen sich die Ansteuersignale für den n -ten Lautsprecher mit Winkel ϕ_n aus Gl. 4.27 wie folgt:

$$P_n = \frac{1}{N} (\alpha W + 2\beta^{(1)} X^{(1)} \cos \phi_n + 2\gamma^{(1)} Y^{(1)} \sin \phi_n + \dots + 2\beta^{(N)} X^{(N)} \cos N\phi_n + 2\gamma^{(N)} Y^{(N)} \sin N\phi_n) \quad (4.36)$$

Die Faktoren α, β, γ dienen als Skalierungsparameter bei nichtsymmetrischem Lautsprecherlayout (vgl. [23]) sowie der Optimierung.

4.2 Binaurales Wiedergabesystem (2D)

Um in binauralen Wiedergabesystemen die Position einer virtuellen Schallquelle abbilden zu können, muss die einfallende Schallwelle mit einer der Quellposition entsprechenden Außenohrübertragungsfunktion (HRTF bzw. HRIR) gefiltert werden. Bei direkter Implementierung der HRTF-Kurven ist eine ausreichende Winkelauflösung nur mit sehr hohem Speicheraufwand zu erreichen. Der Ambisonic-Ansatz löst dieses Problem durch die Reproduktion der Schallwelle durch eine endliche Anzahl an virtuellen Lautsprechern (vgl. Kap. 4.1). Die Signale der virtuellen Lautsprecher werden mit den entsprechenden Impulsantworten (HRIRs) gefiltert, wobei durch Superposition der Teilsignale ein binaurales Kopfhörersignal erzeugt wird.

Die Erzeugung des Kopfhörersignals L/R (linker/rechter Kanal) kann für ein Ambisonic-System 4. Ordnung wie folgt beschrieben werden:

$$\begin{pmatrix} L \\ R \end{pmatrix} = \begin{pmatrix} h_{1,L}(n) & h_{2,L}(n) & \dots & h_{9,L}(n) \\ h_{1,R}(n) & h_{2,R}(n) & \dots & h_{9,R}(n) \end{pmatrix} * \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_9 \end{pmatrix} = \mathbf{F} * \mathbf{P} \quad (4.37)$$

$h_{i,L}(n)$ sind die Impulsantworten für das linke, $h_{i,R}(n)$ für das rechte Ohr. p_i sind die Signale der virtuellen Lautsprecher. Der Operator $*$ entspricht einer Faltung. Obige Matrixschreibweise stellt somit die Faltung von $h_{i,j}(n)$ mit p_i dar. Dies ergibt für das linke Ohr folgende Summation:

$$L = \sum_{i=1}^9 h_{i,L} * p_i \quad (4.38)$$

Der Operator $*$ wird im Folgenden nicht mehr explizit angeführt bzw. in den mathematischen Formulierungen erwähnt.

Im beschriebenen System werden $N = (2M+1)$ Lautsprecher verwendet, wobei laut [35] im tieffrequenten Bereich somit der minimale Fehler in der Reproduktion gegeben ist (vgl. dazu auch Kap. 4.3.1).

Für den 2D – Fall berechnen sich die Lautsprechersignale aus den Ambisonic-Signalen wie folgt (vgl. Kap. 4.1.4):

$$\mathbf{P} = \mathbf{D} \mathbf{A} \quad (4.39)$$

wobei der Vektor $\mathbf{P} = [p_1, p_2, \dots, p_9]^T$ die Lautsprechersignale darstellt, $\mathbf{A} = [W, X^{(1)}, \dots, Y^{(N)}]^T$ die Ambisonic-Kanäle und \mathbf{D} die Dekodermatrix. In ausgeschriebener Form ergibt sich:

$$\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_9 \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \alpha_1 & 2\beta_1^{(1)} \cos \phi_{n,1} & 2\gamma_1^{(1)} \sin \phi_{n,1} & \dots & 2\beta_1^{(N)} \cos(N \phi_{n,1}) & 2\gamma_1^{(N)} \sin(N \phi_{n,1}) \\ \alpha_2 & 2\beta_2^{(1)} \cos \phi_{n,2} & 2\gamma_2^{(1)} \sin \phi_{n,2} & \dots & 2\beta_2^{(N)} \cos(N \phi_{n,2}) & 2\gamma_2^{(N)} \sin(N \phi_{n,2}) \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \alpha_N & 2\beta_N^{(1)} \cos \phi_{n,N} & 2\gamma_N^{(1)} \sin \phi_{n,N} & \dots & 2\beta_N^{(N)} \cos(N \phi_{n,N}) & 2\gamma_N^{(N)} \sin(N \phi_{n,N}) \end{pmatrix} \begin{pmatrix} W \\ X^{(1)} \\ Y^{(1)} \\ \vdots \\ X^{(N)} \\ Y^{(N)} \end{pmatrix} \quad (4.40)$$

Für regelmäßige Lautsprecheranordnungen ergeben sich die Faktoren α, β, γ zu eins, können jedoch zum Zweck der Optimierung variiert werden (vgl. dazu Kap. 4.3.3: Fensterung der Ambisonic-Kanäle höherer Ordnung).

Rotation des Schallfeldes:

Die Abbildung bewegter Schallquellen, sowie die Einbindung eines Headtrackers (vgl. Kap. 2.2.2: Auswirkung spontaner Kopfbewegungen) erfordert die Möglichkeit der Drehung des erzeugten Schallfeldes. In der Ambisonic-Domäne läßt sich dies durch die Implementation einer

Rotationsmatrix \mathbf{R} sehr einfach erzeugen. Mit den Sumsätzen der Sinus- und Cosinusfunktionen folgt für die Rotation um den Winkel ρ (Drehung in der Horizontalebene):

$$\mathbf{A}' = \mathbf{R} \cdot \mathbf{A} \tag{4.41}$$

$$\begin{pmatrix} W' \\ X^{i(1)} \\ Y^{i(1)} \\ \vdots \\ X^{i(N)} \\ Y^{i(N)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & \cos \rho & -\sin \rho & \dots & 0 & 0 \\ 0 & \sin \rho & \cos \rho & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \cos N \rho & -\sin N \rho \\ 0 & 0 & 0 & \dots & \sin N \rho & \cos N \rho \end{pmatrix} \cdot \begin{pmatrix} W \\ X^{(1)} \\ Y^{(1)} \\ \vdots \\ X^{(N)} \\ Y^{(N)} \end{pmatrix} \tag{4.42}$$

Die binauralen Kopfhörersignale L bzw. R berechnen sich somit aus den obigen Beziehungen wie folgt:

$$\begin{pmatrix} L \\ R \end{pmatrix} = \mathbf{F} \cdot \mathbf{P} = \mathbf{F} \cdot \mathbf{D} \cdot \mathbf{A}' = \mathbf{F} \cdot \mathbf{D} \cdot \mathbf{R} \cdot \mathbf{A} \tag{4.43}$$

Es ergibt sich somit folgende Gesamtstruktur des binauralen Wiedergabesystems:

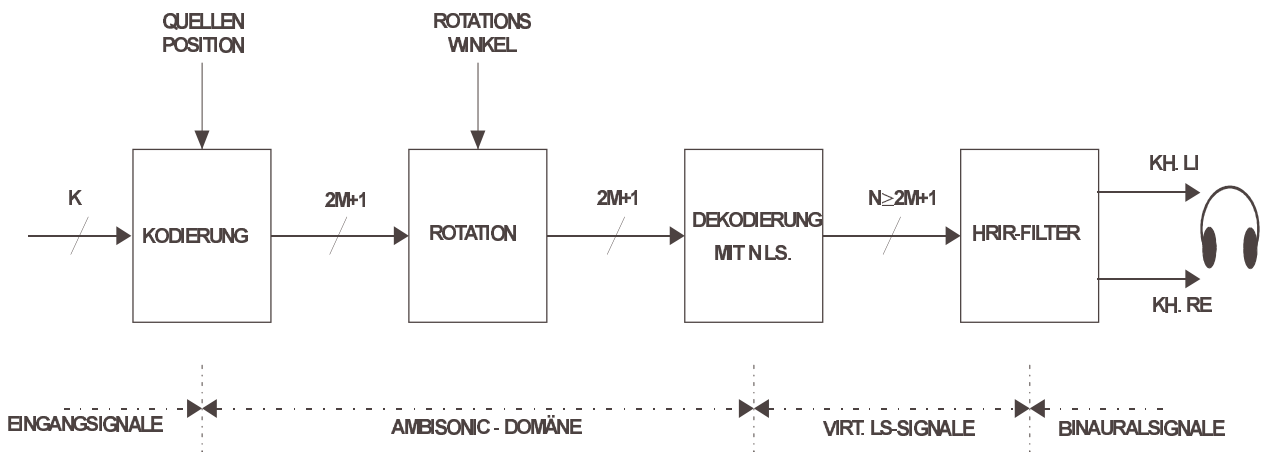


Abb. 4.4: Strukturschaltbild eines binauralen Wiedergabesystems virtueller Schallquellen

4.3 Optimierungen bezüglich der DSP - Implementation eines 2D Binauralsystems

Das folgende Kapitel behandelt die Optimierung des in Kap. 4.2 beschriebenen binauralen Wiedergabesystems, hinsichtlich der Implementierung auf Digitalen Signalprozessoren (DSP). Ziel ist die Entwicklung eines effizienten Algorithmus, welcher durch Optimierung folgender Parameter des Ambisonic-Systems

- Anzahl der Übertragungskanäle
- Anzahl und Aufstellung der virtuellen Lautsprecher
- Gewichtungen der Ambisonic-Kanäle bei der Dekodierung

die zur Verfügung stehenden Ressourcen (Rechenleistung, Speicher, Architektur) optimal nutzt. Die Qualität der Abbildung einer virtuellen Schallquelle steht dabei als Designparameter im Vordergrund. Weiters erfolgt in Kapitel 5 und 6 ein Vergleich der verschiedenen Möglichkeiten der Implementierung anhand eines objektiven Fehlermaßes, sowie anhand der Ergebnisse aus Hörversuchen.

4.3.1 Anzahl der virtuellen Lautsprecher

Zur Erzeugung der binauralen Signale müssen die Signale der virtuellen Lautsprecher mit den zugehörigen Außenohrübertragungsfunktionen gefiltert werden. Für die Berechnung der Filterung steht nur eine durch das DSP-System begrenzte Rechenleistung zur Verfügung, d.h. die Wahl der Anzahl N der Lautsprecher wird ebenfalls durch das DSP-System begrenzt. Untersuchungen von Gerzon [18], Poletti [35] und Bamford [3] haben gezeigt, dass die Dekodierung eines 2D Ambisonic-Systems der Ordnung M auf $N=2M+1$ Lautsprecher, den Fehler bei der Abbildung der ebene Welle minimiert. In [14] wird gezeigt, dass diese Optimierung nur im tieffrequenten Bereich Gültigkeit hat. Eine Optimierung im hochfrequenten Bereich führt zur Implementation von $N \geq 2M+2$ Lautsprechern. Die Trennung der Signale wird mittels phasenkompensierter '*Shelving Filter*' durchgeführt, wobei eine Korrektur der Dekodermatrix mittels frequenzabhängiger Verstärkungsfaktoren durchgeführt wird (vgl. auch [19]). Da dies jedoch zusätzliche Rechenleistung erfordert wird in der weiteren Betrachtung von einer Optimierung mit $N=2M+1$ Lautsprechern ausgegangen. Wie Abb. 4.5 zu entnehmen ist, steigt die Abbildungsgenauigkeit bei Erhöhung der

Ordnung des Ambisonic-Systems. Für ein System 4. Ordnung ergibt sich für $kd < 6.5$ ein Fehler kleiner -20 dB. Da hier der Faktor $d = 8.25$ cm durch den Abstand der beiden Ohren fixiert ist, ergibt sich somit für den Frequenzbereich $f < 4.2$ kHz ein vernachlässigbar kleiner Fehler.

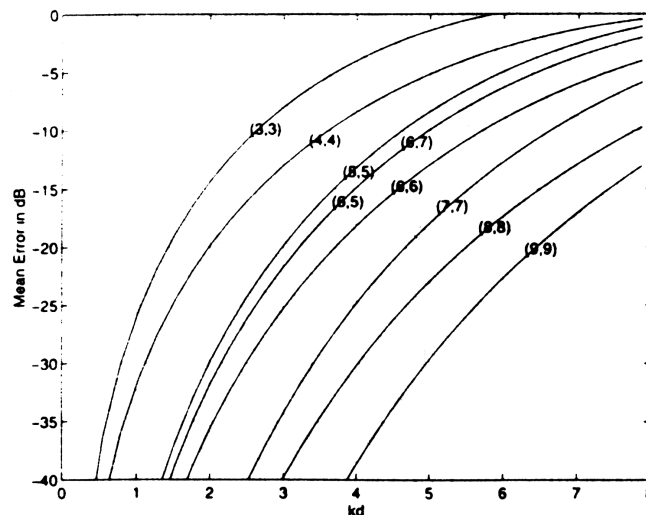


Abb. 4.5: Gemittelter Fehler über die Einfallrichtungen für unterschiedliche Systemlösungen (N, M) mit N Lautsprechern und M Übertragungskanälen [35].

4.3.2 Positionierung der virtuellen Lautsprecher

Die Dekodierung eines Ambisonic-Systems wird im wesentlichen durch die Aufstellung der virtuellen Lautsprecher bestimmt. Für eine ungerade Anzahl N an Lautsprechern weist das System den geringsten Abbildungsfehler auf. Eine spiegelsymmetrische Anordnung der N Lautsprecher minimiert die Anzahl der benötigten HRIR-Filter [42]. Die Einsparung an Rechenleistung kann der Erhöhung der Abbildungsgenauigkeit dienen, z.B. durch Implementation längerer HRIR-Kurven bzw. der Verwendung von Systemen höherer Ordnung. Für ein Ambisonic-System 4. Ordnung ergibt sich folgende Anordnung für $N = 2M + 1 = 9$ virtuelle Lautsprecher:

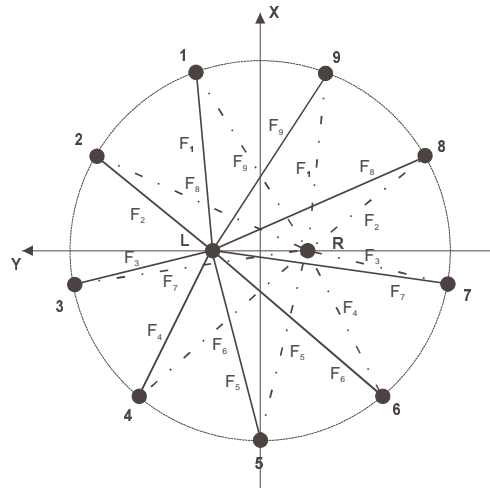


Abb. 4.6: Spiegelsymmetrisches Lautsprecherlayout zur Optimierung der benötigten HRIR-Filter F_i [42]

Ausgehend von der vereinfachenden Annahme einer Symmetrie des Kopfes ergeben sich die in Abb. 4.6 dargestellten Übereinstimmungen der HRIR-Kurven (Filter F_i). Es ergibt sich somit folgende Berechnung der Binauralsignale aus den Lautsprechersignalen:

$$\begin{pmatrix} L \\ R \end{pmatrix} = \mathbf{F} \cdot \mathbf{P} = \begin{pmatrix} F_1 & F_2 & \dots & F_8 & F_9 \\ F_9 & F_8 & \dots & F_2 & F_1 \end{pmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_8 \\ p_9 \end{bmatrix} \quad (4.44)$$

Unter Berücksichtigung der entstehenden Winkelsymmetrie

$$\begin{aligned} \varphi_i &= -\varphi_{N-i} && \text{für gerades } N (i=1, 2, \dots, N/2) \\ \varphi_i &= -\varphi_{N+1-i} && \text{für ungerades } N (i=1, 2, \dots, (N-1)/2) \\ \varphi_{(N+1)/2} &= \pi && \text{bei 1 Lautsprecher hinten, zentriert} \end{aligned} \quad (4.45)$$

und der Eigenschaften der Dekodermatrix kann die Anzahl der benötigten Filter auf die Hälfte reduziert werden [42]:

$$\mathbf{F}_{\text{reduziert}} = \begin{bmatrix} F_{1,1} & F_{1,2} & F_{1,3} & F_{1,4} & F_{1,5} & F_{1,6} & F_{1,7} & F_{1,8} & F_{1,9} \\ F_{1,1} & F_{1,2} & -F_{1,3} & F_{1,4} & -F_{1,5} & F_{1,6} & -F_{1,7} & F_{1,8} & -F_{1,9} \end{bmatrix} \quad (4.46)$$

Das binaurale Signal berechnet sich somit aus den Ambisonic – Kanälen durch Dekodierung bei spiegelsymmetrischem Lautsprecherlayout wie folgt (vgl. auch Abb. 4.7).

$$\begin{pmatrix} L \\ R \end{pmatrix} = \mathbf{F}_{\text{reduziert}} \cdot \mathbf{A} \quad (4.47)$$

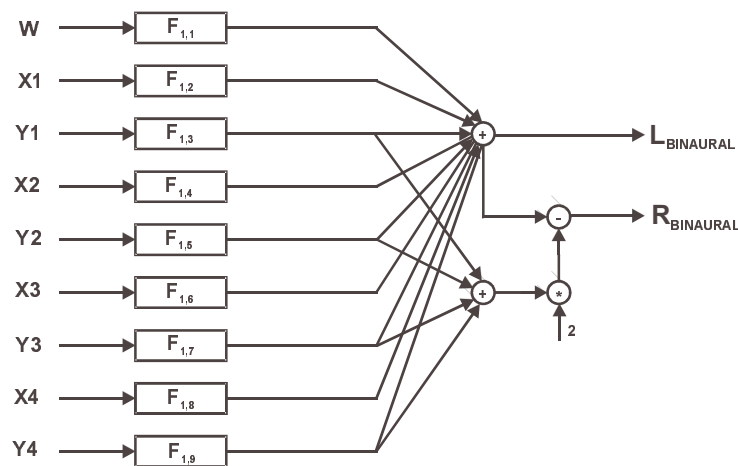


Abb. 4.7: Dekodierung und HRIR Filterung der Ambisonic-Signale 4.Ordnung zur Erzeugung binauraler Signale unter Verwendung eines reduzierten Filtersets

4.3.3 Fensterung der Ambisonic-Kanäle bei der Dekodierung

In [35] wird anhand des Abbildungsfehlers eines Ambisonic-Systems gezeigt, dass für N Lautsprecher die optimalen Gewichte bei der Dekodierung winkelabhängige Sinc-Funktionen der Ordnung N ($asinc_N$, *angular sinc functions*) darstellen. Dies stimmt auch mit den Untersuchungen in [49] und [3] überein. Weiters wird gezeigt, dass bei Wahl der Lautsprecheranzahl N gleich der Anzahl der benötigten Übertragungskanäle, der Abbildungsfehler minimiert wird. Eine

signaltheoretische Betrachtung der Entstehung der asinc_N -Funktionen führt zu folgendem Optimierungsansatz:

Da die Ableitung des Schallfeldes an den Kugelflächenfunktionen eine periodische Funktion bezüglich des Winkels darstellt, kann auf die Reihe eine Fouriertransformation angewendet werden. Für ein Ambisonic-System der Ordnung M wird die Reihenentwicklung nach dem M -ten Glied abgebrochen. Dies entspricht einer Multiplikation der Reihe mit einem Rechteckfenster. Dadurch ergeben sich die sogenannten Gewichtungsfunktionen in Abhängigkeit vom Winkel, die genau den asinc_N -Funktionen entsprechen. Die Amplitude der asinc_N -Funktionen entspricht bei der Dekodierung der Gewichtung des Signals für einen Lautsprecher an der jeweiligen Position. Aufgrund dieser Betrachtung kann eine Optimierung der Dekodierung durch Verwendung verschiedener Fensterfunktionen durchgeführt werden.

Kaiserfenster

Betrachtet man ein Ambisonic-System 3.Ordnung mit 7 Lautsprechern, so ergibt sich bei der Darstellung einer virtuellen Schallquelle aus der 0° -Richtung folgende Aufteilung des Signals auf die jeweiligen Lautsprecher. Abb. 4.8 stellt nur den Betrag der Gewichtung dar. Bei einer sinc-Funktion ist jedoch jede 2.Keule gegenphasig. Dies kann zu Verwirrungen in der Lokalisation führen, d.h. es kommt zu einer „nicht Ortbarkeit“ durch Verräumlichung des Signals. Die Lokalisation wird erschwert, da die Laufzeitunterschiede zu kurz sind – es entsteht eine „Im Kopf Lokalisation“ [18].

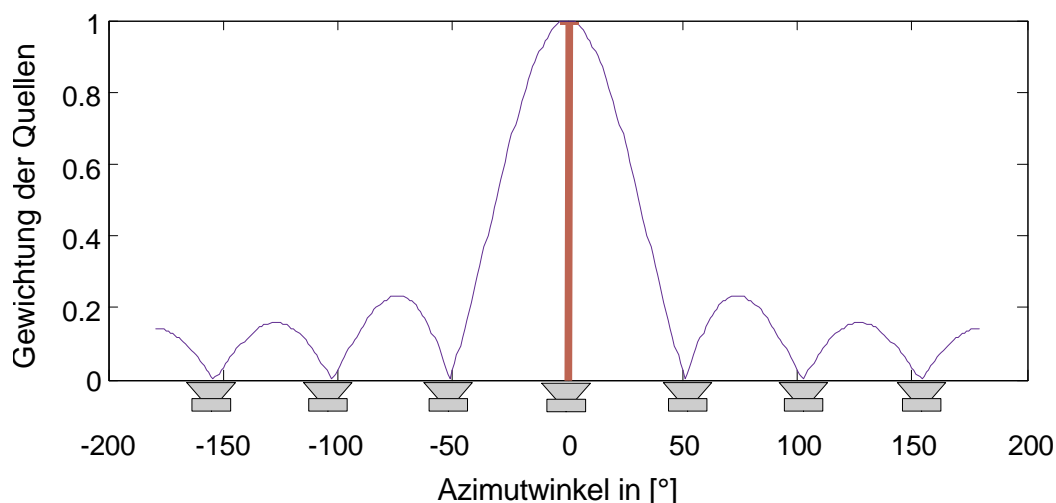


Abb. 4.8: Aufteilung des Signals einer virtuellen Schallquelle aus 0° -Richtung auf die jeweiligen Lautsprecher, bei einem Ambisonic-System 3.Ordnung mit 7 Lautsprechern.

Stimmt die Position einer virtuellen Quelle mit der Position eines Lautsprechers überein, so wird dieser voll gewichtet. Bei äquidistanter Anordnung der Lautsprecher, liegen diese genau in den Nullpunkten der $asinc_7$ Funktion, werden also mit null gewichtet (vgl. Abb. 4.8).

Soll jedoch eine virtuelle Schallquelle zwischen zwei Lautsprechern abgebildet werden, ergibt sich folgende Gewichtung des Signals für die jeweiligen Lautsprecher:

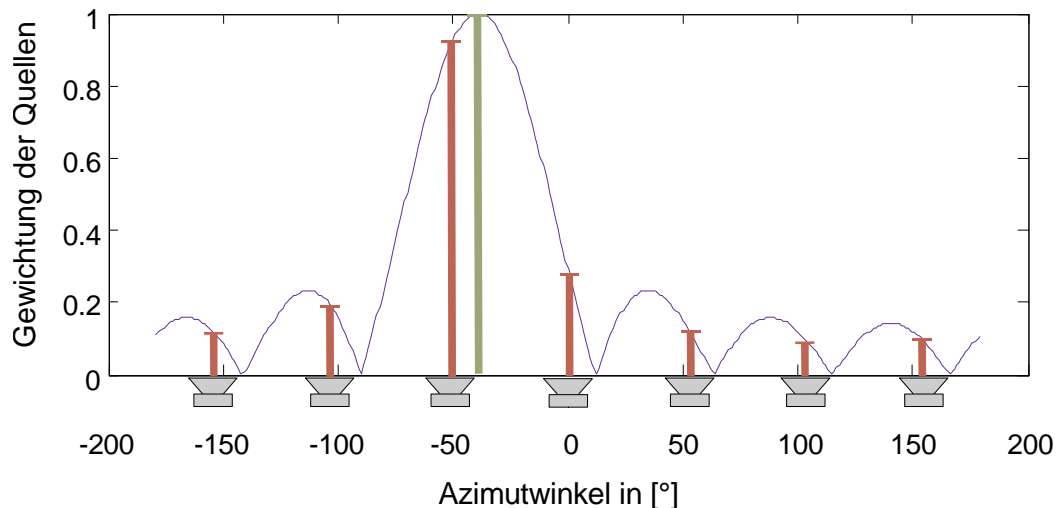


Abb. 4.9: Aufteilung des Signals einer virtuellen Schallquelle aus -40° Richtung auf die jeweiligen Lautsprecher, bei einem Ambisonic-System 3. Ordnung mit 7 Lautsprechern. (grün – virtuelle Schallquelle, rot – Gewichtungen für die Lautsprecher)

Wie Abb. 4.9 zeigt, werden die benachbarten Lautsprecher am stärksten gewichtet. Durch die Nebenkeulen der $asinc_7$ Funktion wird das Signal jedoch auch auf weiter entfernte Lautsprecher verteilt. Bei der Darstellung einer Schallquelle aus $\pm 90^\circ$ werden somit auch aus kontralateraler Richtung Signalanteile abgestrahlt. Der Vergleich der HRTF Kurven ergibt einen maximalen Pegelunterschied von 20 dB zwischen ipsi- und kontralateralem Ohr. Durch Verwendung verschiedener Fenstertechniken kann die Dämpfung der Nebenkeulen der $asinc$ -Funktion erhöht werden (siehe Abb. 4.10). Wesentlich dabei ist die Reduktion der gegenphasigen Anteile, die Lokalisation wird verbessert.

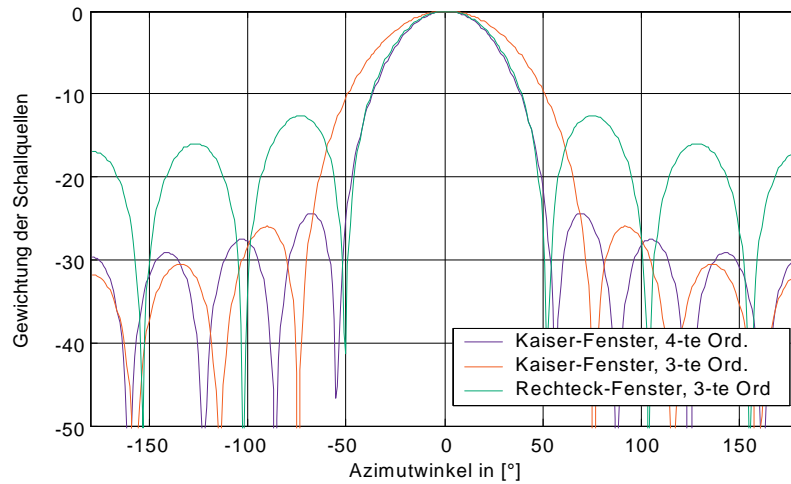


Abb. 4.10: Vergleich der Dämpfung der Nebenkeulen bei Rechteckfenster und Kaiserfenster (Ambisonic 3. und 4. Ordnung)

Die Verwendung eines Kaiserfensters führt zu folgender Verteilung des Signals, bei Abbildung der Schallquelle aus 0° (Abb. 4.11) und -40° (Abb. 4.12):

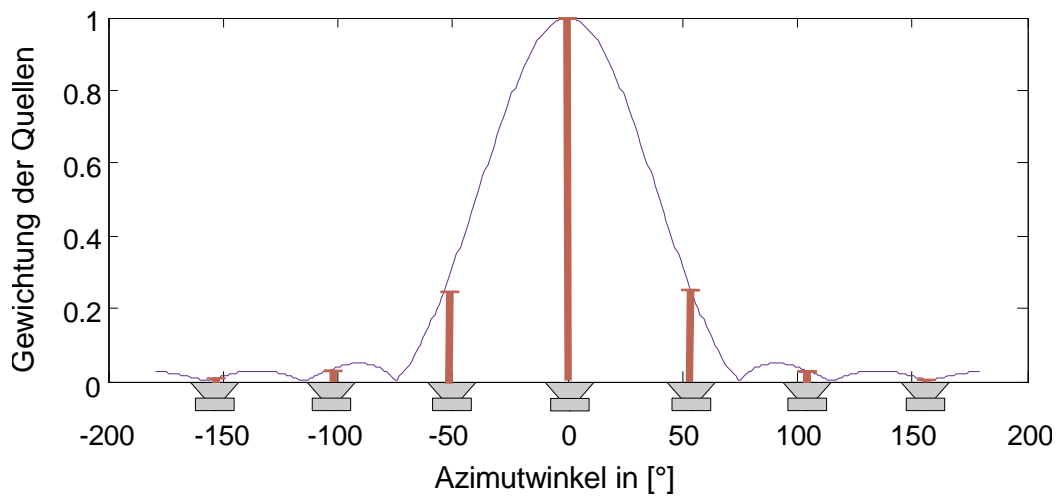


Abb. 4.11: Aufteilung des Signals einer virtuellen Schallquelle aus 0° Richtung auf die jeweiligen Lautsprecher bei Verwendung eines Kaiserfensters (Ambisonic-System 3. Ordnung mit 7 Lautsprechern).

Wie den Abb. 4.11 und 4.12 zu entnehmen ist, erhöht die Verwendung eines Kaiserfensters die Dämpfung der Nebenkeulen deutlich, erzeugt jedoch eine wesentlich breitere Hauptkeule. Das Kaiserfenster wurde so berechnet, dass die Dämpfung der Nebenkeulen > 20dB ist (vgl. Abb. 4.10).

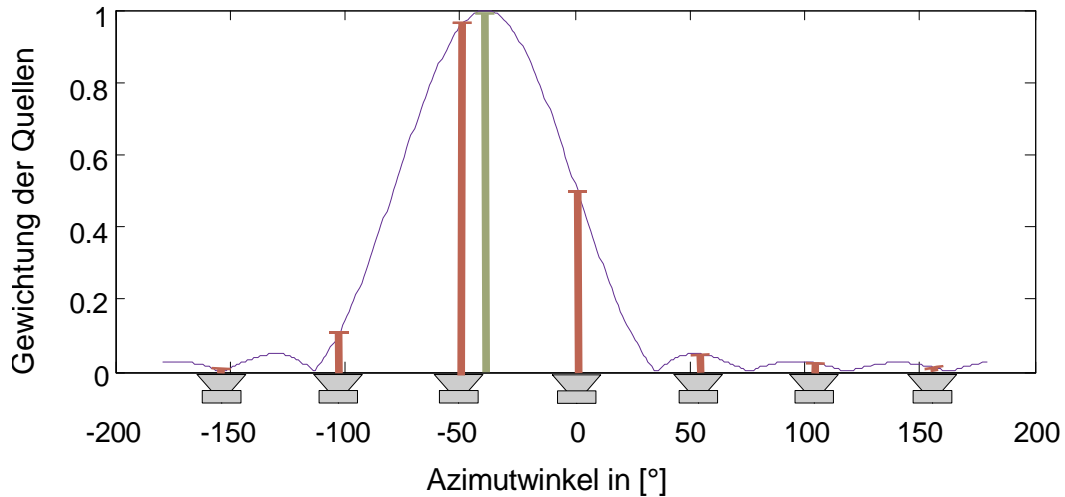


Abb. 4.12: Aufteilung des Signals einer virtuellen Schallquelle aus -40° Richtung auf die jeweiligen Lautsprecher bei Verwendung eines Kaiserfensters (Ambisonic-System 3.Ordnung mit 7 Lautsprechern; grün – virtuelle Schallquelle, rot – Gewichtungen für die Lautsprecher)

Dies führt bei der Abbildung einer Quelle bei $\pm 90^\circ$ zur Vermeidung der störenden Signalanteile aus kontralateraler Richtung. Durch die breitere Hauptkeule kommt es jedoch zu einer Aufweitung der Abbildung der virtuellen Schallquelle (vgl. Abb. 4.11), die Quelle wird weniger punktuell wahrgenommen (*localisation blur* vergrößert sich). Um diesem Effekt entgegenzuwirken kann die Ordnung des Ambisonic-Systems erhöht werden. Abb. 4.13 zeigt ein System 4.Ordnung, mit 9 Lautsprechern.

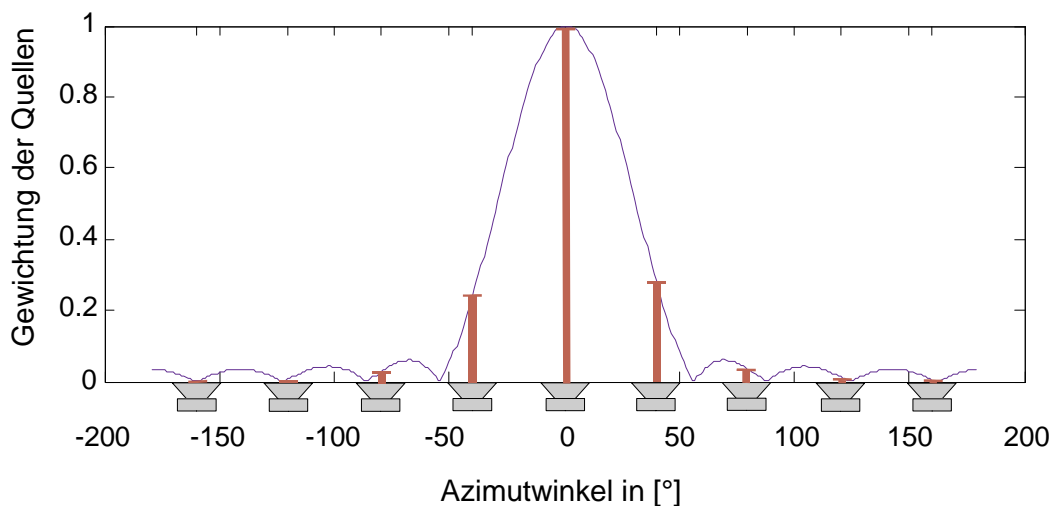


Abb. 4.13: Aufteilung des Signals einer virtuellen Schallquelle aus 0° Richtung auf die jeweiligen Lautsprecher bei Verwendung eines Kaiserfensters (Ambisonic-System 4.Ordnung mit 9 Lautsprechern)

Die Erhöhung der Ordnung ergibt eine schmalere Hauptkeule, da signaltheoretisch betrachtet die Fensterlänge erhöht wird. Für eine optimale Dekodierung (laut [35]) werden $N=2M+1$ Lautsprecher äquidistant angeordnet. Dadurch wird der Abstand zwischen den Lautsprechern verringert, d.h. die virtuelle Schallquelle wird weniger aufgeweitet, der *localisation blur* wird kleiner.

Zusammenfassend können aus der signaltheoretischen Betrachtung folgende Designparameter zur Optimierung des Ambisonic-Systems herangezogen werden:

- Variation der Ambisonic-Ordnung
Bedeutet eine Änderung der Fensterlänge und bestimmt somit die Breite der Hauptkeule der asinc-Funktion. Es kann die Aufweitung der wahrgenommenen Schallquelle variiert werden (*localisation blur*).
- Gewichtung der Ambisonic-Ordnungen
Bestimmt den Typ des verwendeten Fensters und somit die Dämpfung der Nebenkeulen der entstehenden asinc-Funktion. Dadurch können störende Signalanteile unterdrückt werden.

Es ergibt sich folgende Beschreibung in Matrizenform, wobei **HP** die binauralen Signale und **WND** die Fensterfunktion darstellt.

$$\mathbf{HP} = \mathbf{F}_{\text{reduziert}} \cdot \mathbf{WND} \cdot \mathbf{R} \cdot \mathbf{A} \quad (4.48)$$

wobei sich für ein Ambisonic-System 4.Ordnung folgende Fenstermatrix ergibt (W ist voll gewichtet, g_i entspricht den jeweiligen Koeffizienten der Fensterfunktion):

$$\mathbf{WND} = \begin{bmatrix} 1 & 0 & \dots & & \dots & 0 & 0 \\ 0 & g_{x1} & \ddots & & & & 0 \\ \vdots & \ddots & g_{y1} & & & \ddots & \vdots \\ & & & g_{x2} & 0 & & \\ & & & & g_{y2} & & \\ & & & 0 & & g_{x3} & \\ \vdots & & \ddots & & & & g_{y3} & \ddots & \vdots \\ 0 & & & & & & \ddots & g_{x4} & 0 \\ 0 & 0 & \dots & & \dots & 0 & & g_{y4} \end{bmatrix} \quad (4.49)$$

4.3.4 Länge der HRIR-Filter

Die Reduktion der Filterlänge stellt einen wesentlichen Parameter zur Optimierung der Recheneffizienz dar. Untersuchungen der Referenz-HRIRs (512 Samples) zeigen ab dem 250-ten Sample bereits einen Abfall des Signals auf -90dB. Bei der Implementierung erfolgt eine Kodierung der Filter mit 16 Bit Auflösung. Dies ergibt ein SNR (*signal to noise ratio*) von 96dB. Somit genügt eine maximale Filterlänge von 256 Samples den Anforderungen des Systems.

Eine weitere Verkürzung der Filterlänge bewirkt eine stärkere Glättung der Frequenzantwort des Filters, d.h. wichtige Details bezüglich der Lokalisation können verloren gehen. Weiters kann eine starke Klangverfärbungen auftreten.

4.4 Erweiterungen eines binauralen 2D Wiedergabesystems

4.4.1 Akustische Linse

Der Begriff „akustische Linse“ beschreibt die Möglichkeit einer gezielten Betrachtung eines begrenzten Ausschnittes der Grundgesamtheit und wird in Analogie zum Begriff der „optischen Linse“ gewählt [42]. Betrachtet man ein binaurales Wiedergabesystem (vgl. Kap. 4.2), so bedeutet die Implementierung einer „akustischen Linse“ eine Unterdrückung aller virtuellen Schallquellen außerhalb eines definierten Bereiches. Die Position der Quellen bleibt dabei unverändert. Somit wird der sogenannte „Cocktail Party Effekt“ binauralen Wiedergabesystemen zugänglich gemacht. Dieser beschreibt die Fähigkeit der menschlichen Hörwahrnehmung sich in einem Stimmen- und Geräuschgewirr auf eine spezielle Schallquelle konzentrieren zu können.

Eine Implementierung der akustischen Linse in das in Kap. 4.2 beschriebene binaurale Wiedergabesystem erfolgt durch eine entsprechende Gewichtung der virtuellen Lautsprechersignale.

$$\tilde{\mathbf{P}} = \mathbf{W}(\alpha) \cdot \mathbf{P} \quad (4.50)$$

Wobei \mathbf{P} die virtuellen Lautsprechersignale und $\tilde{\mathbf{P}}$ die modifizierten Lautsprechersignale darstellt. Die Linsenwirkung kann durch den Parameter α wie folgt eingestellt werden:

$$\mathbf{W}(\alpha) = (1 - \alpha) \cdot \mathbf{V} + \alpha \cdot \mathbf{I} \quad (4.51)$$

Es ergibt sich folgender einstellbarer Bereich:

- Für $\alpha=0$ ist $\mathbf{W}(\alpha) = \mathbf{V}$:

Die akustische Linse ist vollständig wirksam.

- Für $\alpha=1$ ist $\mathbf{W}(\alpha) = \mathbf{I}$:

Die akustische Linse hat keinen Einfluss mehr.

Bestimmung der Filtermatrix \mathbf{V} :

Die Erzeugung des Effektes der akustischen Linse basiert auf einer räumlichen Filterung der Lautsprechersignale, d.h. die jeweiligen Lautsprechersignale werden unterschiedlich gewichtet. Betrachtet man die räumliche Anordnung der Lautsprecher, so ist die Folge der Teilsignale periodisch in 2π . Somit kann diese Folge in eine Fourier Reihe entwickelt werden. Die Filterwirkung kann dadurch besser beurteilt werden. Die Filtermatrix \mathbf{V} zur Gewichtung der einzelnen Lautsprechersignale kann für ein Ambisonic System 4. Ordnung mit 9 Lautsprechern, mit einer Linsenwirkung um die Position des 3. Lautsprechers, wie folgt dargestellt werden,

$$\mathbf{V} = \begin{bmatrix} \alpha_2 & 0 & \dots & & \dots & 0 & 0 \\ 0 & \alpha_1 & \ddots & & & & 0 \\ \vdots & \ddots & 1 & & \ddots & & \vdots \\ & & & \alpha_1 & 0 & & \\ & & & & \alpha_2 & & \\ & & & 0 & \alpha_3 & & \\ \vdots & & \ddots & & & \alpha_4 & \ddots & \vdots \\ 0 & & & & & \ddots & \alpha_4 & 0 \\ 0 & 0 & \dots & & \dots & 0 & \alpha_3 \end{bmatrix} \quad (4.52)$$

Die Gewichtungsfaktoren α_i dienen der Implementierung verschiedener Fenstertechniken, um eine höhere Dämpfung unerwünschter Richtungen bezüglich der selektierten Richtung zu erreichen.

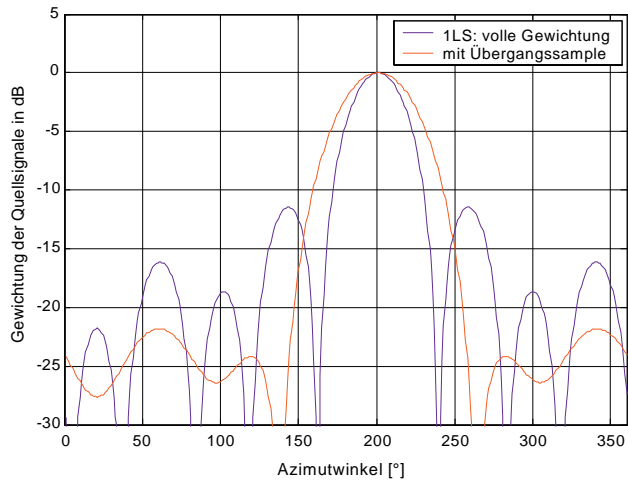


Abb. 4.14: Ergebnisse der Filtermatrix V ohne und mit Übergangssample (0.39) für virtuellen Lautsprecher bei 200° (Ambisonic-System 4. Ordnung)

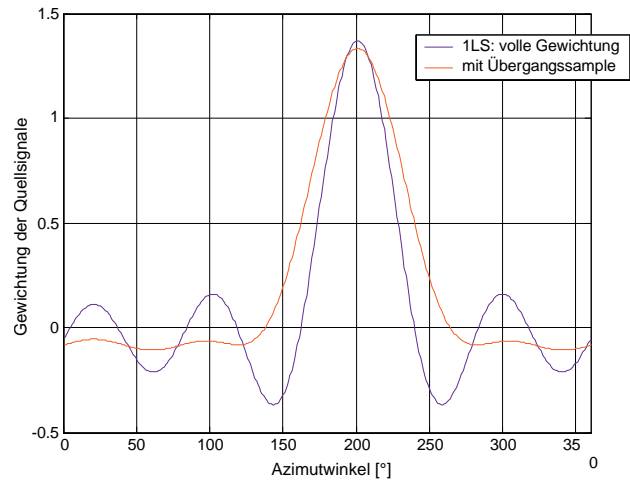


Abb. 4.15: Ergebnisse der Filtermatrix V ohne und mit Übergangssample (0.39) für virtuellen Lautsprecher bei 200° (Ambisonic-System 4. Ordnung)

Die minimale Breite der akustischen Linse resultiert aus der Verwendung eines Rechteckfensters (schmälste Hauptkeule) und beträgt

$$\Delta \varphi_{\text{Lens, min}} = 2 \cdot \Delta \varphi_{\text{Speaker}} = 2 \cdot \frac{2\pi}{N} \quad (4.53)$$

wobei $\Delta \varphi_{\text{Speaker}}$ den Abstand zweier Lautsprecher bei äquidistanter Aufstellung bezeichnet. Um ein geringeres Übersprechen zu unerwünschten Richtungen zu erreichen, kann mittels verschiedener Fensterfunktionen die Dämpfung der Nebenkeulen erhöht werden. Dies führt jedoch zu einer Verbreiterung der Hauptkeule und somit zu einer Vergrößerung der Breite der Linse $\Delta \varphi_{\text{Lens}}$. Abb. 4.14 und Abb. 4.15 zeigen die Verbreiterung der Hauptkeule bei Verwendung eines Übergangssamples ([34], p. 253-255). Eine Erhöhung der Anzahl N der Lautsprecher wirkt der Verbreiterung entgegen, die Ambisonic-Signale werden jedoch nicht mehr optimal dekodiert. Um trotzdem eine optimale Dekodierung zu erreichen, muss die Ambisonic-Ordnung M erhöht werden, damit weiterhin $N=2M+1$ gilt [35].

In Kap. 4.3.2 wurde zur Erhöhung der Recheneffizienz ein optimierter Algorithmus vorgestellt. Dabei wird die Symmetrie der Lautsprecheranordnung ausgenutzt, um ein reduziertes Filterset $F_{\text{reduziert}}$ berechnen zu können. Die Filtermatrix F und die Dekodermatrix D sollen dabei nicht getrennt werden. $W(\alpha)$ muss in der Signalkette vor die Dekodermatrix gezogen werden [42].

Somit kann mit nur einem Gewichtungsparemeter α und einer fixen Gewichtungsmatrix V die Wirkung der akustischen Linse gesteuert werden.

$$\mathbf{W}(\alpha) \cdot \mathbf{D} \equiv \mathbf{D} \cdot \tilde{\mathbf{W}}(\alpha) \quad (4.54)$$

Aus Gl. 4.54 ergibt sich für die akustische Linse L folgende Beziehung, die während der Laufzeit nicht verändert wird.

$$\mathbf{L} = \mathbf{D}^{-1} \cdot \mathbf{V} \cdot \mathbf{D} \quad (4.55)$$

Die Wirkungsweise zur Laufzeit kann wie folgt beschrieben werden:

$$\tilde{\mathbf{W}}(\alpha) = (1 - \alpha) \cdot \mathbf{L} + \alpha \cdot \mathbf{I} \quad (4.56)$$

Somit kann die optimierte Dekodierung unverändert verwendet werden, was zu folgender Signalkette des Gesamtsystems führt:

$$\mathbf{HP} = \mathbf{F}_{\text{reduziert}} \cdot \tilde{\mathbf{W}}(\alpha) \cdot \mathbf{W} \mathbf{N} \mathbf{D} \cdot \mathbf{R} \cdot \mathbf{A} \quad (4.57)$$

4.4.2 Translatorische Bewegung

Bei translatorischer Bewegung ändern sich Winkel und Entfernung der jeweiligen virtuellen Lautsprecher relativ zum Betrachter (siehe Abb. 4.16: Dabei stellen die strichlierten Pfeile die Änderung des Winkels dar, die fett gedruckten Pfeile symbolisieren eine Zunahme bzw. Abnahme des Pegels).

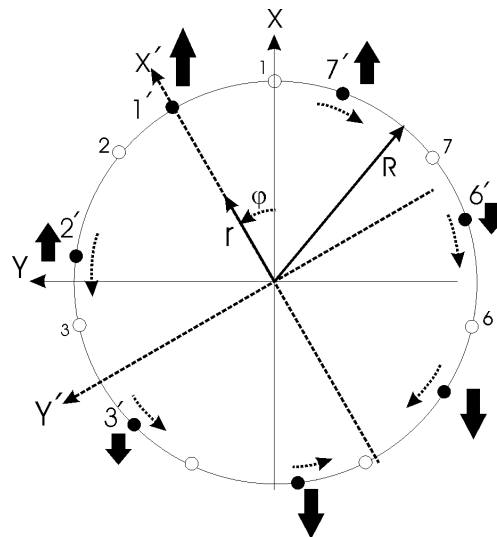


Abb. 4.16: Winkelverzerrung und Änderung des Abstandes relativ zum Betrachter bei translatorischer Bewegung [42]

Die Berechnung erfolgt in zwei Schritten [42]:

- Veränderung des wahrgenommenen Pegels
- Veränderung des relativen Winkels

Entfernungsabhängige Pegeländerung

Da der Ambisonic-Ansatz von Freifeldbedingungen ausgeht, kann diese Abhängigkeit als Grundlage der Berechnung verwendet werden, wobei $R \gg r$ gewählt wird, d.h. r niemals R (Referenzposition, von allen Lautsprechern gleich weit entfernt) erreicht. Es ergibt sich somit ein die translatorische Bewegung begrenzender Bereich (in Abb. 4.17 für $r \leq 0.5R$ strichliert dargestellt).

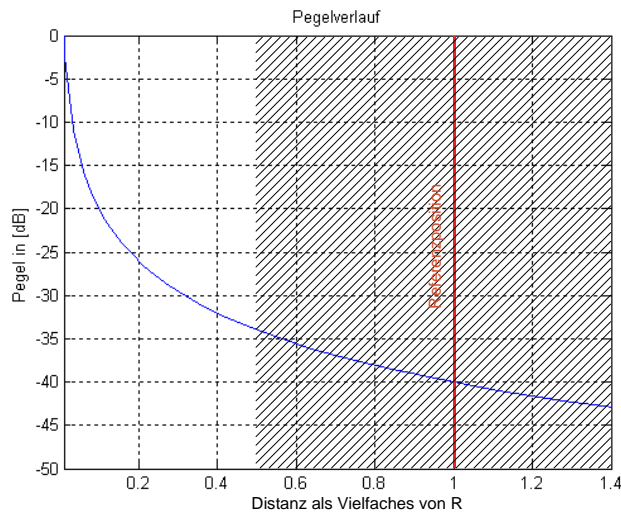


Abb. 4.17: Pegeländerung in Abhängigkeit von der Distanz, wobei der Bewegungsfreiraum (für $r \leq 0.5 R$) strichliert dargestellt ist.

Die veränderte Distanz R_i' der virtuellen Lautsprecher kann wie folgt berechnet werden (vgl. auch Abb. 4.18):

$$R_i' = \sqrt{R^2 + r^2 - 2 \cdot r \cdot R \cdot \cos \theta_i} \tag{4.58}$$

Wobei sich für das Gesamtsystem folgende Matrix ergibt:

$$\tilde{\mathbf{P}} = \begin{bmatrix} R/R_1' & 0 & 0 & \dots & 0 & 0 \\ 0 & R/R_2' & 0 & \dots & & 0 \\ 0 & 0 & \ddots & & \ddots & \vdots \\ \vdots & \ddots & & \ddots & & \vdots \\ 0 & & & & R/R_8' & 0 \\ 0 & 0 & \dots & \dots & 0 & R/R_9' \end{bmatrix} \tag{4.59}$$

Die Gewichtungsfaktoren R/R_i zur Distanzkodierung wirken auf die Signale der virtuellen Lautsprecher. Um die Multiplikation durchführen zu können, muss zuerst eine Dekodierung

erfolgen. Nach der Gewichtung der Lautsprechersignale wird wieder in die Ambisonic-Domäne kodiert.

$$P = D^{-1} \cdot \tilde{P} \cdot D \quad (4.60)$$

Entfernungsabhängige Winkelverzerrung

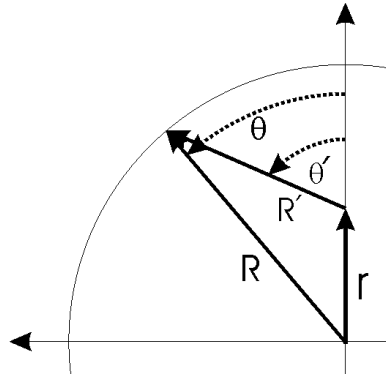


Abb. 4.18: Winkelverzerrung bei translatorischer Bewegung, relativ zum Betrachter [42]

Wie Abb. 4.18 zeigt, kann die Winkelverzerrung über folgenden Ansatz berechnet werden

$$R \cdot \sin \theta = R' \cdot \sin \theta' \quad (4.61)$$

wobei für $a = r/R$ gilt und R' mit Gl. 4.58 berechnet werden kann. Nach einigen mathematischen Umformungen ergibt sich die Matrix $WV(a)$ zur Modifikation des Ambisonic-Systems [42]:

$$WV(a) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sqrt{2}\left(\frac{a}{2} + \frac{a^3}{16}\right) & 1 - \frac{3a^2}{8} & 0 & \frac{a}{2} + \frac{a^3}{8} & 0 & \frac{3a^2}{8} & 0 \\ 0 & 0 & 1 - \frac{a^2}{8} & 0 & \frac{a}{2} - \frac{a^3}{8} & 0 & \frac{3a^2}{8} \\ -\sqrt{2}\left(\frac{a^2}{8}\right) & -a & 0 & 1 - \frac{5a^2}{8} & 0 & a - \frac{3a^3}{8} & 0 \\ 0 & 0 & -a + \frac{3a^3}{16} & 0 & 1 - \frac{11a^2}{8} & 0 & a - a^3 \\ -\sqrt{2}\left(\frac{a}{2} - \frac{9a^3}{16}\right) & \frac{a^2}{2} & 0 & \frac{-3a}{2} - \frac{5a^3}{16} & 0 & 1 - \frac{23a^2}{8} & 0 \\ 0 & 0 & \frac{-a^2}{4} & 0 & \frac{-3a}{2} & 0 & -1 + \frac{5a^2}{8} \end{bmatrix} \quad (4.62)$$

Die Auswirkung der Winkelverzerrung auf die Ambisonic-Signale kann wie folgt dargestellt werden:

$$\mathbf{A}' = \mathbf{WV}(a) \cdot \mathbf{A} \quad (4.63)$$

4.4.3 Raumsimulation

Begault und Wenzel haben in [4] anhand von Hörversuchen gezeigt, dass die Implementierung einer Raumsimulation (erste Reflexionen und Diffusanteil) in binauralen Wiedergabesystemen zu

- einer Verringerung des Azimut-Fehlers bei der Lokalisation (vgl. Abb. 4.19), sowie
- einer verbesserten Außerkopf-Lokalisation bzw. Externität (vgl. Abb. 4.20)

führt.

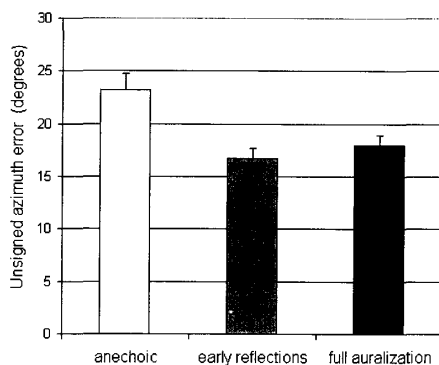


Abb. 4.19: Mittelwert des Azimut-Fehlers der Lokalisation ohne (anechoic) und mit Raumsimulation (early reflections – ersten 80ms; full auralization – mit Diffusanteil) [4];

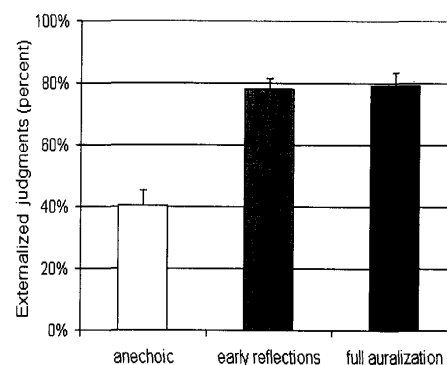


Abb. 4.20: Externität einer virtuellen Schallquelle ohne (anechoic) und mit Raumsimulation (early reflections – ersten 80ms; full auralization – mit Diffusanteil) [4]

Weiters konnte in [4] bei der Außerkopf-Lokalisation kein signifikanter Unterschied zwischen der Verwendung lediglich der 1.Reflexionen (bis 80ms) und der 1.Reflexionen mit zusätzlichem Diffusanteil festgestellt werden. Durch die Verwendung der 1.Reflexionen ohne Diffusanteil, kann somit sehr recheneffizient Räumlichkeit implementiert werden.

Algorithmen zur Raumsimulation können in zwei große Gruppen unterteilt werden [41]:

1. Numerische Methoden

- FEM (finite elements methode)
- BEM (boundary elements methode)
- FDM (finite differences methode)

2. Geometrische Methoden

- Spiegelquellenmethode (image source)
- Strahlenmodelle (path tracing, ray tracing, beam tracing)
- Schallteilchenmodelle (radiosity)

Einheitliche Streutheorie (*Uniform Theory of Diffraction*):

Obige Methoden berücksichtigen keine Beugungseffekte bei tiefen Frequenzen. Die Betrachtung der Beugung spielt bei konvexen Raumstrukturen eine wesentliche Rolle, kann bei konkaven Räumen jedoch vernachlässigt werden. Die *Uniform Theory of Diffraction (UTD)* ([29], [48]) stellt eine Methode zur Berechnung der Beugungseffekte dar. Dabei wird die Beugung an einer Kante als eigene Schallquelle betrachtet (vgl. Abb. 4.21).

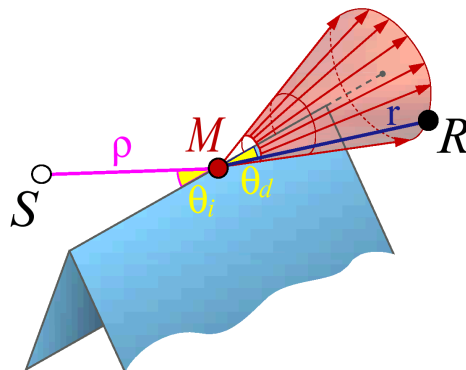


Abb. 4.21: Simulation der Beugung einer Schallwelle an einer Kante, anhand der *Uniform Theory of Diffraction (UTD)* [48].

Prinzip:

Bei der UTD wird eine unter dem Winkel θ_i einfallende Schallwelle ρ in mehrere reflektierte Schallwellen aufgespalten. Dabei wird der Öffnungswinkel θ_d des Strahlenkegels gleich dem Winkel θ_i , wobei die Kante die Achse des Kegels bildet. Für einen einzelnen Betrachtungspunkt R kann die Beugung nun mittels eines einzelnen Schallstrahls r beschrieben werden. Bei der

Überlagerung kann der Beugungsstrahl noch durch einen komplexen Beugungskoeffizienten abgeschwächt werden. Die Ausbreitung des Schallstrahls r erfolgt dabei nach dem Fermatschen Prinzip. Dies besagt, dass im homogenen Medium die Schallausbreitung auf kürzestem Weg zwischen Quelle und Beobachtungspunkt erfolgt.

In den folgenden Betrachtungen werden Beugungseffekte nicht berücksichtigt.

Effizienter Algorithmus zur Raumsimulation

In [41] wird ein recheneffizienter Algorithmus vorgestellt. Dabei werden die Erstreflexionen mit Hilfe eines Ringbuffers (*tapped delay*), die diffusen Reflexionen durch ein *Waveguide Digital Network (WDN)* implementiert.

Bei Verwendung des Spiegelquellenprinzips zur Berechnung der Struktur des Ringbuffers steigt die Anzahl der virtuellen Spiegelquellen exponential mit der Ordnung ϵ des Systems an.

$$N_{vir} = \frac{n_w}{n_w - 2} [(n_w - 1)^\epsilon - 1] \quad (4.64)$$

wobei n_w die Anzahl der reflektierenden Wände darstellt. Aufgrund der Komplexität der Berechnung kann ein Ringbuffer lediglich zur Darstellung der Erstreflexionen verwendet werden. Um die freie Bewegung der Abhörposition zu ermöglichen, wird der Ringbuffer dynamisch ausgeführt (*dynamical tapped delay*).

Die Modellierung der späten Reflexionen basiert auf einem numerischen Algorithmus, dem sog. *Waveguide Digital Network (WDN)*. Dabei erfolgt eine Zerlegung der reflektierenden Wände in Knotenpunkte (*wall nodes*). Für jeden dieser Knotenpunkte wird nun eine Streumatrix berechnet, wobei eine Berücksichtigung der akustischen Gegebenheiten (Absorptionskoeffizienten, usw.) erfolgt. Die Streumatrix \mathbf{A} beschreibt somit das Verhältnisse der einfallenden Schallwelle \mathbf{P}^- zur reflektierten Schallwelle \mathbf{P}^+ .

$$\mathbf{P}^+ = \mathbf{A} \cdot \mathbf{P}^- \quad (4.65)$$

Die Generierung des WDN erfolgt durch Zuordnung der 1. Reflexionen der Ordnung ϵ zu den jeweiligen Knotenpunkten (d.h. die jeweiligen *samples* am Ausgang des Ringbuffers werden der

zugehörigen Streumatrix zugeordnet). An den jeweiligen Knotenpunkten wird durch Multiplikation mit der Streumatrix das Diffusfeld berechnet. Die Implementierung der Streuung erfolgt mittels rekursiver Filtern, auf deren Design hier nicht näher eingegangen wird.

Es ergibt sich folgende Gesamtstruktur:

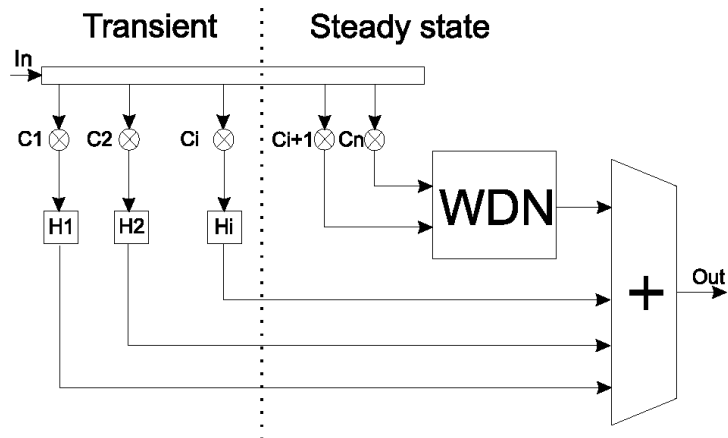


Abb. 4.22: Raumsimulation mittels Ringbuffer für die 1. Reflexionen und einem „Waveguide Digital Network“ für den Diffushall [41].

wobei die Koeffizienten c_i der distanzabhängigen Dämpfung der Signale dienen und die Filter H_i eine distanzabhängige Klangfärbung bewirken.

5 Objektives Fehlermaß

Im folgenden Kapitel wird ein Fehlermaß zur objektiven Beurteilung binauraler Wiedergabesysteme vorgestellt. Die Definition dieses Fehlermaßes basiert auf der Abweichung der durch das Wiedergabesystem verfälschten HRIRs zu den direkt gemessenen HRIRs (Referenzkurven), unter Berücksichtigung psychoakustischer Effekte. Somit besteht ein wesentlicher Punkt bei der Betrachtung des Fehlermaßes in der Annahme, dass die Referenz-HRIRs optimale Lokalisation bieten.

Hinsichtlich des in Kap. 4 entwickelten Systems wird eine Fehlerberechnung durchgeführt. Aus den Ergebnissen wird die Auswirkung verschiedener Optimierungsverfahren (vgl. Kap. 4.3) auf die Abbildungsgenauigkeit virtueller Schallquellen untersucht. Weiters wird in Kap. 6 anhand von Hörversuchen die Aussagekraft des entwickelten Fehlermaßes untersucht. Ziel ist die Optimierung des Gesamtsystems hinsichtlich der Abbildungsgenauigkeit, bei gleichzeitiger Dezipierung des benötigten Rechenaufwandes.

5.1 Berechnung von Lokalisationsfehlern

Die Optimierung eines Systems bedarf eines Maßes, mit dem sich das Verhalten des Systems beschreiben lässt. Normalerweise handelt es sich um eine skalare Größe, deren Wert Rückschlüsse auf die Qualität des Systems hinsichtlich der untersuchten Parameter zulässt. Die Untersuchung der Lokalisation lässt sich mittels zweier Funktionen beschreiben:

- Lokalisationsfunktion
- Lokalisationsunschärfe

Beide Funktionen sind abhängig vom Azimutwinkel θ . Bei der Fehlerberechnung hängt die Auflösung der betrachteten Azimutwinkel von der Anzahl der Referenz-HRIRs ab, stellt somit eine diskrete Größe dar. Die Berechnung der Funktionen erfolgt somit an diskreten Punkten, die Darstellung erfolgt in Form von Vektoren.

Definitionen:

- Lokalisationsfunktion $L(\theta)$,
gibt den Azimutwinkel an, aus dem die virtuelle Schallquelle wahrgenommen wird
- Lokalisationsunschärfe (localisation blur) $Bl(\theta)$,
gibt die Breite der wahrnehmbaren Schallquelle in Grad an

Um im Sinne der Optimierung die vektoriell dargestellten Größen auf ein Skalar reduzieren zu können, erfolgt eine Mittelung über den Azimutwinkel θ . Dies führt zur Definition folgender Größen:

- Mittlerer Lokalisationsfehler: $\bar{L} = \sqrt{\frac{1}{N} \sum_{i=1}^N [L(\theta_i) - \theta_i]^2}$

Stellt den Effektivwert der Abweichung der Lokalisationsfunktion von der Soll-Lokalisation dar.

- Mittlere Lokalisationsunschärfe \bar{Bl} :

Stellt den Mittelwert der Lokalisationsunschärfe dar und wird als Vielfaches des *MAA* (*Minimum Audible Angle*) angegeben.

5.1.1 Berechnung der Interauralen Laufzeitdifferenz

Die Berechnung der Gruppenlaufzeit durch Fouriertransformation und anschließendem Ableiten des Phasenspektrums nach der Frequenz, führt zu gravierenden Artefakten. Durch die Fensterung der Impulsantwort im Zeitbereich sind jene Frequenzen im Spektrum nicht vorhanden, deren Gruppenlaufzeiten außerhalb des Fensters liegen. Die Faltung im Frequenzbereich führt durch die Nebenkeulen der Fensterfunktion an den Stellen der fehlenden Frequenzen zu Phasenlagen, deren Ableitung zu physikalisch nicht vorhandenen negativen Gruppenlaufzeiten führt.

Aus psychoakustischer Sicht ist die Berechnung der mittleren Gruppenlaufzeit innerhalb der gehörspezifischen Frequenzbereiche (Bark-Bereiche) ausreichend. Dies führt zu folgendem Ansatz:

Die Impulsantworten werden mittels frequenzgruppenbreiter, nullphasiger Bandpaßfilter gefiltert. Von dem gefilterten Signal wird der Energieschwerpunkt ermittelt, welcher den Mittelwert der Gruppenlaufzeit der in der jeweiligen Frequenzgruppe enthaltenen Frequenzen darstellt.

Die nullphasigen Filter werden durch die Filterung des Signals mit der kausalen und anschließend antikausalen Impulsantwort eines nicht nullphasigen Filters erreicht. Das Filterdesign erfolgt mittels *Least-Square-Error-Minimalization*, wobei die Ordnung der Filter von 10 bis 400 (beginnend mit der höchsten hin zu tiefen Mittenfrequenzen) variiert. Somit wird eine Anpassung der Bandbreite der jeweiligen Filter an die Frequenzgruppen der menschlichen Hörwahrnehmung gewährleistet.

$$h_F(n, z, \theta) = h(n, \theta) * f(n, z) * f(-n, z) \quad (5.1)$$

Wobei $h_F(n, z, \theta)$ die gefilterten Impulsantworten, $h(n, \theta)$ die Impulsantworten des Systems und $f(n, z)$ die frequenzgruppenbreiten Filter darstellt.

Die Berechnung des Energieschwerpunktes und somit der Gruppenlaufzeit innerhalb der jeweiligen Frequenzgruppe, kann wie folgt durchgeführt werden:

$$T_g(z, \theta) = \frac{\sum_n n \cdot h_F^2(n, z, \theta)}{\sum_n h_F^2(n, z, \theta)} \quad (5.2)$$

Mit diesem Verfahren werden nun folgende Gruppenlaufzeiten berechnet:

- $T_{g,REF,links}(z, \theta)$... Referenz-HRIRs, für das linke Ohr
- $T_{g,REF,rechts}(z, \theta)$... Referenz-HRIRs, für das rechte Ohr
- $T_{g,AMBI,links}(z, \theta)$... Ambisonic-System, für das linke Ohr
- $T_{g,AMBI,rechts}(z, \theta)$... Ambisonic-System, für das rechte Ohr

Die Interaurale Laufzeitdifferenz (*ITD*, *Interaural Time Difference*) wird durch die Berechnung der Laufzeitdifferenz der einzelnen Gruppenlaufzeiten ermittelt. Dies erfolgt sowohl für die Referenz-HRIRs,

$$ITD_{REF}(z, \theta) = T_{g,REF,links}(z, \theta) - T_{g,REF,rechts}(z, \theta) \quad (5.3)$$

als auch für die durch das Ambisonic-System verzerrten HRIRs:

$$ITD_{AMBI}(z, \theta) = T_{g, AMBI, links}(z, \theta) - T_{g, AMBI, rechts}(z, \theta) \quad (5.4)$$

5.1.2 Lokalisation aufgrund der Interauralen Laufzeitdifferenzen

Die Laufzeitdifferenzen (ITDs) sind vorwiegend für die Lokalisation im tieffrequenten Bereich verantwortlich. Bei gegenüber den Referenz-HRIRs größerer ITD, wird die Schallquelle bis zu einer bestimmten Grenze weiter auslateralisiert empfunden. Um die Verfälschung der Lokalisation durch das System ermitteln zu können, wird die Änderung der Laufzeitdifferenz betrachtet. Dieser allgemeine Ansatz geht davon aus, dass die Lokalisation der Schallquellen sich aus den Referenz-HRIRs ableiten läßt und Verzerrungen der ITD eine Veränderung der Lokalisation hervorrufen. Die durch das Ambisonic-System erzeugten HRIRs werden mit den Referenz-HRIRs verglichen.

Das Ambisonic-System erzeugt für jeden Azimutwinkel und jede Frequenzgruppe eine bestimmte ITD, welche durch die Größe $ITD_{AMBI}(z, \theta_0)$ dargestellt wird. $ITD_{REF}(z, \theta)$ folgt aus den Referenz-HRIRs. Dabei wird für jede Frequenzgruppe ein Zusammenhang zwischen dem Azimutwinkel und der wahren Laufzeitdifferenz berechnet.

Um die Systemlokalisierung θ_{ITD} berechnen zu können, wird jener Azimutwinkel θ ermittelt, bei dem die Referenz-HRIRs und die HRIRs am Ausgang des Systems denselben Wert für die ITD aufweisen.

$$\theta_{ITD}(z, \theta_0) = \arg \min_{\theta} |ITD_{REF}(z, \theta) - ITD_{AMBI}(z, \theta_0)| \quad (5.5)$$

Die Suche nach der passenden ITD erfolgt allerdings nur in der Umgebung des gegebenen Azimutwinkels θ_0 :

$$\forall \theta \in \left[\theta_0 - \frac{\theta_{WND}}{2}; \theta_0 + \frac{\theta_{WND}}{2} \right] \quad (5.6)$$

wobei θ_{WND} die Breite des Suchfensters darstellt und mit 180° angenommen wurde.

Ist der gesuchte Wert für die ITD jedoch größer als die maximale ITD der Referenz-HRIRs, terminiert die Suche nicht. Für die Lokalisation bedeutet eine größere ITD, dass die Schallquelle weiter auslateralisiert wahrgenommen wird. Ein ITD $\gg 1\text{ms}$ zerstört die Konsistenz des Binauralsignals. Auf dieser Tatsache beruhend wird die Systemlokalisierung in solchen Fällen auf $\pm 90^\circ$ gesetzt:

$$\forall \theta: ITD_{REF}(z, \theta) < ITD_{AMBI}(z, \theta_0) \Rightarrow \theta_{ITD}(z, \theta_0) = \begin{cases} 90^\circ \text{ für } \theta_0 \in [0^\circ; 180^\circ] \\ -90^\circ \text{ für } \theta_0 \in (0^\circ; 180^\circ) \end{cases} \quad (5.7)$$

Die Berechnung wird für folgende Bereiche durchgeführt: $\theta_0 \in [0^\circ; 360^\circ]$ und $z \in [1; 24]$

Dies ergibt den auf der ITD basierenden Datensatz an Lokalisationspunkten $\theta_{ITD}(z, \theta_0)$, welcher auch als ITDA (*Interaural Time Difference Angle*) bezeichnet wird ([36], [37]).

5.1.3 Berechnung der Interauralen Pegeldifferenz

Die Differenz der Pegel am linken und rechten Ohr wird als ILD (*Interaural Level Difference*) bezeichnet und in dB dargestellt. Für die Berechnung der ILD werden die Impulsantworten mittels Fouriertransformation in den Frequenzbereich transformiert und logarithmisch dargestellt. Um die Amplituden der auralen Signale an die psychoakustischen Gegebenheiten anzupassen, werden die Differenzen in die Bark-Skala transformiert (vgl. [6]). Somit erhält man die Pegeldifferenzen der Referenz-HRIRs

$$ILD_{REF}(z, \theta) = 10 \log \sum_{f(z-1)}^{f(z)-1} [FFT h_{REF,L}(n, \theta)]^2 - 10 \log \sum_{f(z-1)}^{f(z)-1} [FFT h_{REF,R}(n, \theta)]^2 \quad (5.8)$$

sowie des Ambisonic-Systems:

$$ILD_{AMBI}(z, \theta) = 10 \log \sum_{f(z-1)}^{f(z)-1} [FFT h_{AMBI,L}(n, \theta)]^2 - 10 \log \sum_{f(z-1)}^{f(z)-1} [FFT h_{AMBI,R}(n, \theta)]^2 \quad (5.9)$$

5.1.4 Lokalisation aufgrund der Interauralen Pegeldifferenzen

Die Berechnung der Lokalisation aufgrund der Pegeldifferenzen erfolgt mittels des in Kap. 5.1.2 beschriebenen Suchalgorithmus. Dies führt auf die Lokalisationsfunktion $\theta_{ILD}(z, \theta_0)$:

$$\theta_{ILD}(z, \theta_0) = \arg \min_{\theta} |ILD_{REF}(z, \theta) - ILD_{AMBI}(z, \theta_0)| \quad (5.10)$$

Auch hier wird die Suche auf den Bereich $\theta_0 \pm \frac{\theta_{WND}}{2}$ eingeschränkt.

Bei Pegeldifferenzen größer als die maximale ILD der Referenz-HRIRs kann kein passender Azimutwinkel gefunden werden. Für die Lokalisation bedeuten größere Pegeldifferenzen stärkere Auslateralisierung der wahrgenommenen Schallquelle. Die Systemlokalisierung wird in solchen Fällen auf $\pm 90^\circ$ gesetzt, was einer monauralen Wahrnehmung entspricht.

$$\forall \theta: ILD_{REF}(z, \theta) < ILD_{AMBI}(z, \theta_0) \Rightarrow \theta_{ILD}(z, \theta_0) = \begin{cases} 90^\circ & \text{für } \theta_0 \in [0^\circ; 180^\circ] \\ -90^\circ & \text{für } \theta_0 \in (0^\circ; 180^\circ) \end{cases} \quad (5.11)$$

Dies ergibt den auf der ILD basierenden Datensatz an Lokalisationspunkten $\theta_{ILD}(z, \theta_0)$, welcher auch als ILDA (*Interaural Level Difference Angle*) bezeichnet wird ([36], [37]).

5.1.5 Zusammenführung der Lokalisationsparameter

Sowohl die Lokalisation aufgrund der Pegeldifferenzen, wie auch die Lokalisation aufgrund der Laufzeitdifferenzen sind für die Richtungswahrnehmung der virtuellen Schallquelle verantwortlich. Bei der Zusammenführung der beiden Größen müssen diese je nach Frequenzbereich unterschiedlich gewichtet werden.

Die Lokalisation aufgrund der Laufzeitdifferenz $\theta_{ITD}(z, \theta)$ wirkt vorwiegend im tieffrequenten Bereich [6].

Folgende Aufbereitung des Datensatzes der ITD wurde durchgeführt:

- Ausblenden der 1. Frequenzgruppe ($f < 100\text{Hz}$), da diese unter der Grenzfrequenz des Gesamtübertragungssystems (DSP, Kopfhörer, ...) liegt
- Ausblenden ab Bark 7, da oberhalb von 800Hz der Einfluss der ITD auf die Lokalisation geringer wird.

Somit ergeben sich die in Abb. 5.1 dargestellten Gewichtungen $w_{ITD}(z)$.

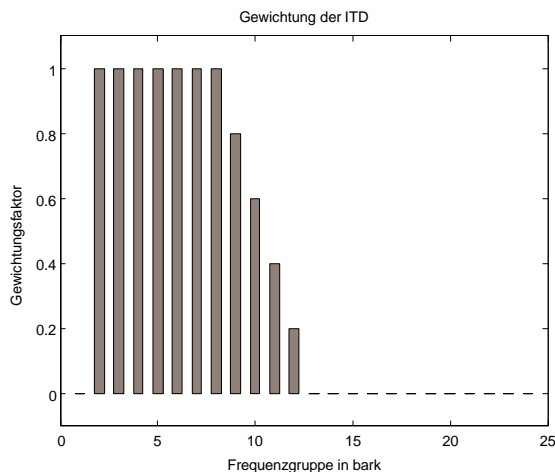


Abb. 5.1: Gewichtung $w_{ITD}(z)$ der ITD über die Bark-Bereiche

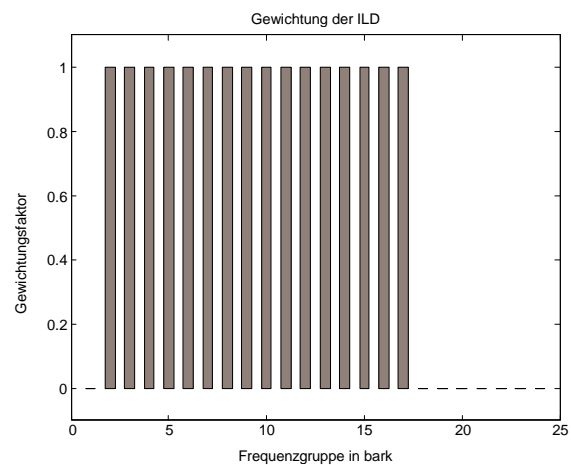


Abb. 5.2: Gewichtung $w_{ILD}(z)$ der ILD über die Bark-Bereiche

Die Auswirkung der Pegeldifferenzen auf die Lokalisation von Schallquellen ist eigentlich im gesamten Frequenzbereich vorhanden [6]. Auch hier wird die erste Frequenzgruppe aus den oben genannten Gründen ausgeblendet. Das in Kap. 4.2 beschriebene zweidimensionale binaurale Wiedergabesystem liefert eine ausreichende Abbildungsgenauigkeit bis annähernd 4 kHz. Aus diesem Grund werden alle Frequenzgruppen oberhalb Bark 17 ($f > 3700\text{ Hz}$) ausgeblendet (vgl. Kap. 4.3.1). Die Gewichtungen für die ILD $w_{ILD}(z)$ zeigt Abb. 5.2. Die Grenzfrequenz kann durch Erhöhung der Ambisonic-Ordnung erhöht werden (vgl. Kap. 4.3).

Nun wird die zusammengesetzte Lokalisationsfunktion berechnet, wobei über die Frequenzgruppen gewichtet gemittelt wird:

$$L(\theta) = \frac{1}{2} \cdot \left(\frac{1}{\sum w_{ITD}(z)} \cdot \sum_{z=1}^{24} w_{ITD}(z) \cdot \theta_{ITD}(z, \theta) + \frac{1}{\sum w_{ILD}(z)} \cdot \sum_{z=1}^{24} w_{ILD}(z) \cdot \theta_{ILD}(z, \theta) \right) \quad (5.12)$$

5.1.6 Lokalisationsunschärfe

Die Lokalisationsunschärfe (*localisation blur*) gibt die Breite des Hörereignisses an und entspricht der Standardabweichung der Lokalisationsfunktion über die Frequenzgruppen, bei gegebenem Azimutwinkel. Da der Frequenzgruppenmittelwert der Lokalisationsfunktion gewichtet berechnet wurde, sollten auch hier die Gewichte berücksichtigt werden:

$$BI(\theta) = \sqrt{\frac{1}{2} \cdot \frac{\sum_{i=ILD}^{ITD} 1}{\sum w_i(z)} \cdot \sum_{z=1}^{24} w_i(z) \cdot [\theta_i(z, \theta) - L(\theta)]^2} \quad (5.13)$$

wobei für $i = ILD$ gilt $\theta_i \equiv \theta_{ILD}$, $w_i \equiv w_{ILD}$ und für $i = ITD$ gilt $\theta_i \equiv \theta_{ITD}$, $w_i \equiv w_{ITD}$

5.1.7 Mittlere Fehler

Mit der berechneten Lokalisationsfunktion, sowie der Lokalisationsunschärfe kann nun eine Aussage über die Lokalisation von Schallquellen bezüglich der Referenz-HRIRs getroffen werden. Um das Ambisonic-System einfacher beurteilen zu können, wird eine Mittelwertsberechnung gemäß Kap. 5.1 durchgeführt.

Aufgrund der symmetrischen Eigenschaften der ITD und ILD stellt die Lokalisationsfunktion eine schiefsymmetrische, die Lokalisationsunschärfe eine symmetrische Funktion bezüglich der X-Achse dar. Die Berechnung entlang eines Halbkreises (in den ersten beiden Quadranten) beinhaltet somit alle relevanten Daten.

Der Lokalisationsfehler \bar{L} berechnet sich somit folgendermaßen:

$$\bar{L} = \sqrt{\frac{1}{N} \sum_{i=1}^N [L(\theta_i) - \theta_i]^2} \quad (5.14)$$

wobei N der Anzahl der Referenz-Aufnahmen zwischen 0° und 180° entspricht.

Um die mittlere Lokalisationsunschärfe zu berechnen, wird der jeweilige Wert der Lokalisationsunschärfe auf den $MAA(\theta)$ bezogen. Zwischen den einzelnen der Literatur entnommenen Werten wird linear interpoliert.

Die mittlere Lokalisationsunschärfe berechnet sich folgendermaßen:

$$\overline{Bl} = \frac{1}{N} \cdot \sum_{\theta=0^{\circ}}^{180^{\circ}} \frac{Bl(\theta)}{MAA(\theta)} \quad (5.15)$$

wobei N wiederum der Anzahl der Referenz-Aufnahmen zwischen 0° und 180° entspricht.

5.1.8 Zusammenfassung der Berechnungen

Die Güte des untersuchten Systems läßt sich nun mittels folgender 2 Parameter beschreiben:

- Mittlerer Lokalisationsfehler \overline{L}
- Mittlere Lokalisationsunschärfe \overline{Bl}

Es ist leicht ersichtlich, dass im Idealfall der Lokalisationsfehler \overline{L} gegen Null und die Lokalisationsunschärfe \overline{Bl} gegen 1 gehen sollte. Somit besteht die Möglichkeit Optimierungen der Ambisonic-Parameter hinsichtlich eines vorgegebenen Ziels durchzuführen – d.h. im Fall einer konkreten DSP-Implementierung, bei gleichbleibendem Lokalisationsfehler den notwendigen Rechenleistungsbedarf zu minimieren.

Die optimierten Parameter-Settings müssen dann jedoch anhand von Hörversuchen auf deren Relevanz verifiziert werden, da nicht die gesamte Komplexität der menschlichen Hörwahrnehmung in die Berechnung einfließt. Es ist mathematisch somit lediglich eine Dezimierung der Anzahl der zu untersuchenden Parameter-Settings möglich.

5.2 Betrachtung und Analyse ausgesuchter Systeme

Im Folgenden Kapitel erfolgt die Analyse unterschiedlicher Parameter-Settings des binauralen Wiedergabesystems. Die genaue Beschreibung der jeweiligen Settings kann Anhang A entnommen

werden. Folgende Kurzbezeichnung wird bei den Abbildungen verwendet: #xxx, wobei xxx für die Nummer des jeweiligen Settings steht.

Bei der folgenden Untersuchung der verschiedenen Parameter-Settings wird mehr Gewicht auf die Optimierung hinsichtlich der Lokalisationsunschärfe gelegt. Der Lokalisationsfehler kann mittels einer Mapping-Funktion (durch spezielle Vorverzerrung bei der Kodierung) korrigiert werden.

5.2.1 Ordnung des Ambisonic-Systems und Anzahl der Lautsprecher

Änderung der Ambisonic-Ordnung bei jeweils optimaler Dekodierung

Wie in Kap. 4.3.1 beschrieben, erfolgt eine optimale Dekodierung eines 2D-Ambisonic-Systems bei $N=2M+1$ Lautsprechern. Dabei tritt laut [35] der geringste Abbildungsfehler auf. Die Fehlerfunktionen von Systemen unterschiedlicher Ordnung bei optimaler Lautsprecheranzahl werden im Folgenden dargestellt und analysiert.

Analyse:

Wie Abb. 5.3 und Abb. 5.4 zu entnehmen ist, tritt bei den Ordnungen 3 und 4 eine wesentliche Verbesserung der Lokalisationseigenschaften gegenüber Systemen der Ordnung 2 auf. Durch Erhöhung der Ordnung des Systems von 3 auf 4 kann jedoch keine wesentliche Verbesserung erzielt werden. Die wesentlichen Vorteile die den erhöhten Rechenaufwand bei einer Implementierung von Systemen 4.Ordnung rechtfertigen, können Abb. 5.6 und Abb. 5.5 entnommen werden. Der Lokalisationsfehler wie auch die Lokalisationsunschärfe bei Systemen 4.Ordnung sind im vorderen Bereich wesentlich besser als bei Systemen 3.Ordnung. Im hinteren Bereich verschlechtern sich jedoch die Lokalisationseigenschaften gegenüber der 3.Ordnung. Für binaurale Wiedergabesysteme ist die genaue Lokalisation in den vorderen Quadranten primär wichtig. Da Abb. 5.3 und Abb. 5.4 den Mittelwert über den Winkel darstellen, ist hier keine wesentliche Verbesserung bei Erhöhung der Ordnung ersichtlich.

Weiters ist aus der Abb. 5.3 ersichtlich, dass bei einem System 2.Ordnung ein Anstieg des Lokalisationsfehlers bei einer Filterlänge von 256 Samples auftritt. Dieser Sachverhalt ist durch den theoretischen Ambisonic-Ansatz nicht erklärbar. Ein mögliche Fehlerquelle liegt bei den implementierten HRIRs selbst. Es wurden keine näheren Untersuchungen diesbezüglich

durchgeführt, da Systeme 2.Ordnung nicht zur Implementierung vorgesehen sind. Die Tendenz, dass mit Erhöhung der System-Ordnung der Lokalisationsfehler abnimmt ist jedoch gut sichtbar.

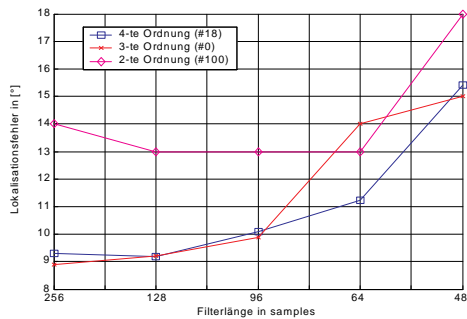


Abb. 5.3: Lokalisationsfehler verschiedener Filterlängen für unterschiedliche Ambisonic-Ordnungen bei optimaler Dekodierung

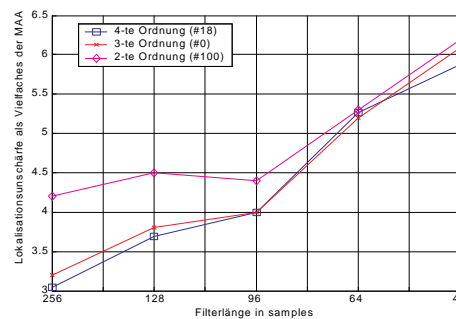


Abb. 5.4: Lokalisationsunschärfe verschiedener Filterlängen für unterschiedliche Ambisonic-Ordnungen bei optimaler Dekodierung

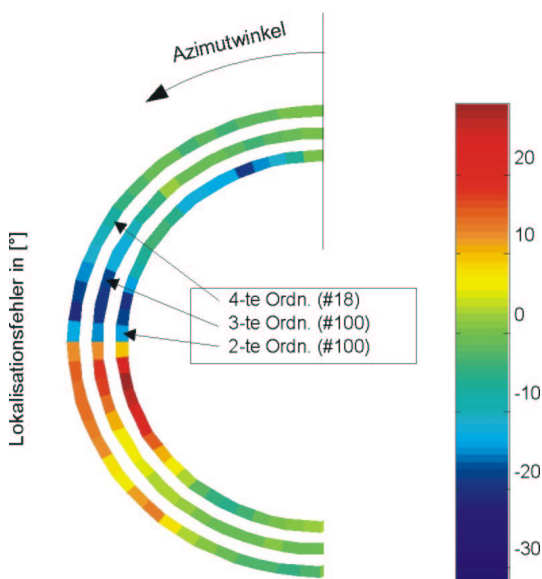


Abb. 5.5: Lokalisationsfehler verschiedener Ordnungen bei Filterlängen von 128 Samples und optimaler Dekodierung

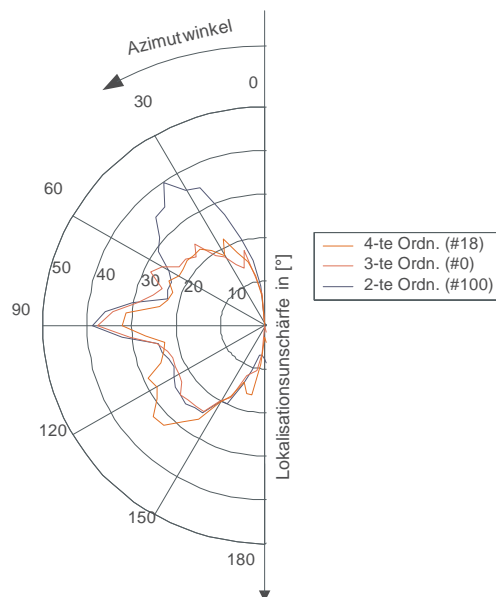


Abb. 5.6: Lokalisationsunschärfe verschiedener Ordnungen bei Filterlängen von 128 Samples und optimaler Dekodierung

Änderung der Ambisonic-Ordnung bei gleichbleibender Lautsprecheranzahl

Bei den vorangegangenen Analysen wurde bei Veränderung der Ambisonic-Ordnung, aufgrund der Bedingung der optimalen Dekodierung, die Anzahl der Lautsprecher mit verändert. Die binauralen Signale werden durch Filterung der Lautsprechersignale mit den zu den jeweiligen Richtungen

gehörenden HRIR-Kurven erzeugt. Aus diesem Grund wirken in obige Untersuchungen verschiedene HRIR-Kurven ein. Die folgenden Beispiele zeigen die Auswirkungen der Änderung der Ambisonic-Ordnung bei gleichbleibender Anzahl der Lautsprecher. Somit wird das Ergebnis immer durch die selben HRIR-Kurven beeinflusst.

Analyse:

Aus den Abbildungen 5.7 und 5.8 ist gut ersichtlich, dass durch Erhöhung der Ambisonic-Ordnung eine Verbesserung des Lokalisationsfehlers und der Lokalisationsunschärfe erreicht wird. Diese Verbesserung wird jedoch mit steigender Ordnung kleiner. Weiters ist den Abbildungen zu entnehmen, dass ein System 3.Ordnung bessere Lokalisationseigenschaften als ein System 4.Ordnung aufweist. Dies lässt sich wiederum durch betrachten der Abbildung 5.9 und 5.10 erklären. Diese Abbildungen zeigen wieder, dass ein Ambisonic-System 4.Ordnung vorne gute Abbildungseigenschaften aufweist. Seitlich und hinten ist der Lokalisationsfehler jedoch größer als bei Systemen 3.Ordnung. Bei Betrachtung der Abb. 5.10 zeigt sich jedoch, dass im Unterschied zu den vorhergehenden Betrachtungen die Lokalisationsunschärfe bei Systemen 3.Ordnung im gesamten Bereich schlechter als bei Systemen 4.Ordnung ist. Dies ist darauf zurückzuführen, dass durch die erhöhte Lautsprecheranzahl keine optimale Dekodierung bei Systemen 3.Ordnung stattfindet. Bei $N \geq 2M+1$ verringert sich der Abstand der Lautsprecher. Wird eine virtuelle Schallquelle genau aus der Richtung eines Lautsprechers abgebildet, liegen die anderen Lautsprecher nicht mehr in den Nullpunkten der asinc-Funktionen (vgl. Kap. 4.3.1). Dadurch erfolgt ein Übersprechen auf nebenliegende Lautsprecher, die virtuelle Schallquelle wird aufgeweitet.

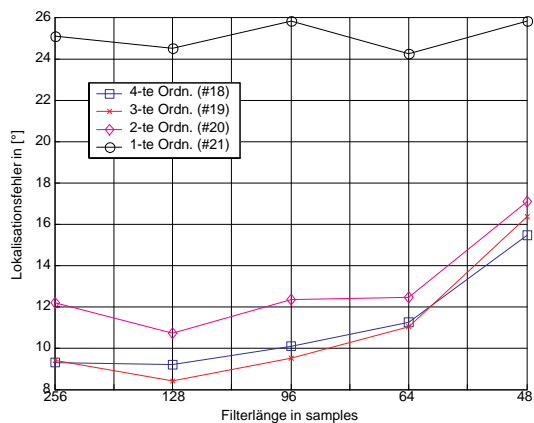


Abb. 5.7: Lokalisationsfehler verschiedener Filterlängen für unterschiedliche Ambisonic-Ordnungen bei gleichbleibender Lautsprecheranzahl

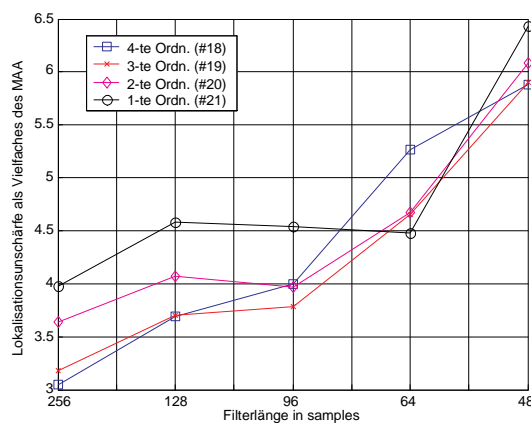


Abb. 5.8: Lokalisationsunschärfe verschiedener Filterlängen für unterschiedliche Ambisonic-Ordnungen bei gleichbleibender Lautsprecheranzahl

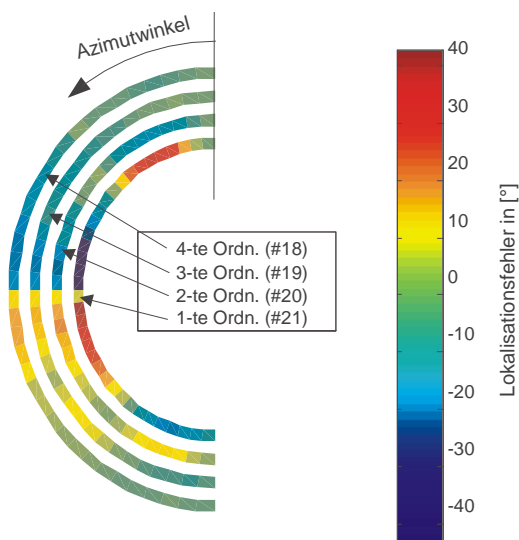


Abb. 5.9: Lokalisationsfehler verschiedener Ordnungen bei Filterlängen von 128 Samples und gleichbleibender Lautsprecheranzahl

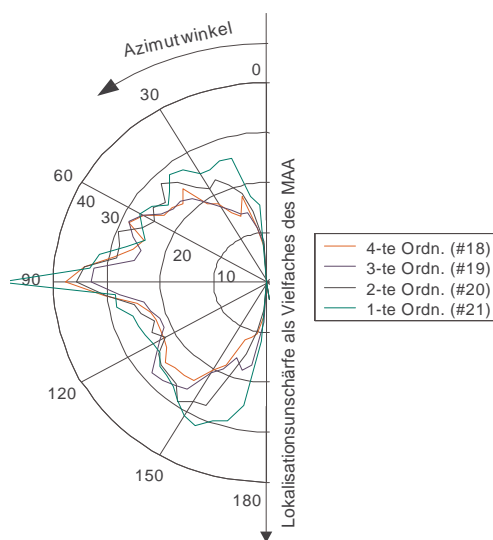


Abb. 5.10: Lokalisationsunschärfe bei verschiedenen Ordnungen für Filterlängen von 128 Samples und gleichbleibender Lautsprecheranzahl

5.2.2 Vergleich der Referenz-HRIRs (Kemar vs. AKG)

Bei der Implementierung des binauralen Wiedergabesystems standen verschiedene Referenz HRIRs zur Verfügung (Kemar⁷, AKG⁸). Im Folgenden wird die Auswirkung bei Verwendung der jeweiligen Kurven auf die Lokalisationseigenschaften untersucht.

Allgemein:

Den Abbildungen 5.12 und 5.11 ist zu entnehmen, dass die HRIR-Kurven von AKG gegenüber der Kemar-Kurven geringere Lokalisationsfehler und eine geringere Lokalisationsunschärfe bei Systemen unterschiedlicher Ordnung aufweisen.

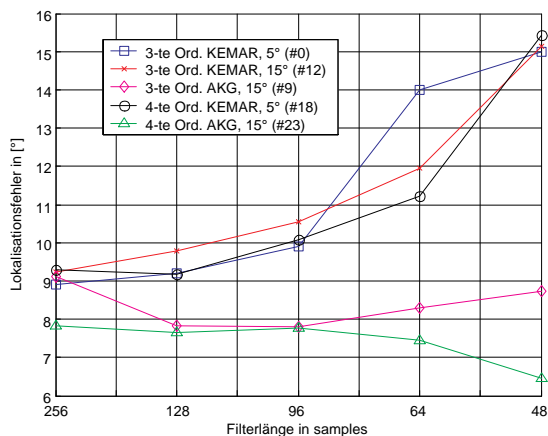


Abb. 5.11: Lokalisationsfehler verschiedener Ordnungen bei unterschiedlichen Referenz-HRIRs (AKG, Kemar) über der Filterlänge

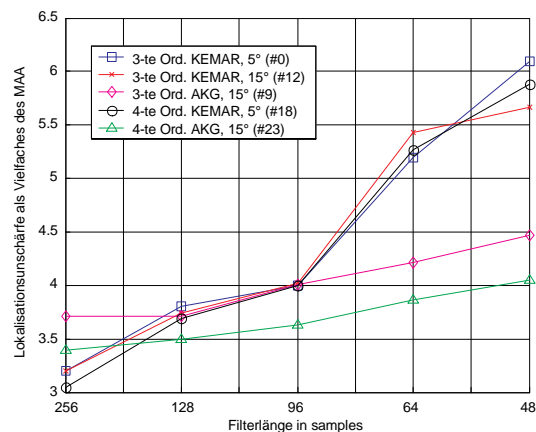


Abb. 5.12: Lokalisationsunschärfe bei verschiedenen Ordnungen bei unterschiedlichen Referenz-HRIRs (AKG, Kemar) über der Filterlänge

7 HRIR Kurven, gemessen am Kemar Kunstkopf von Gardner und Martin [<http://sound.media.mit.edu/KEMAR.html>]

8 HRIR Kurven, zur Verfügung gestellt von der Fa. AKG Acoustics, Wien (Kooperationspartner)

Auswirkung der räumlichen Auflösung

Die Anzahl der vorliegenden Referenz-HRIRs bestimmt die Auflösung des Azimutwinkels. Ein Ambisonic-System der Ordnung M wird dann optimal dekodiert, wenn der Abstand der Lautsprecher den Abständen der Nullpunkte der $asinc_N$ -Funktion entspricht (vgl. Kap. 4.3.1). Die Kemar-Kurven liegen in 5° Auflösung, die AKG-Kurven in 15° Auflösung vor. Dadurch können die Lautsprecher nur mit begrenzter Genauigkeit positioniert werden. Im Folgenden soll die Auswirkung dieser räumlichen Diskretisierung der Lautsprecherpositionen untersucht werden. Dazu werden Kemar-Kurven bei einer auf 5° und 15° diskretisierten Aufstellung der Lautsprecher implementiert und die Lokalisation bewertet.

Analyse:

Wie Abb. 5.13 und 5.14 zu entnehmen ist, sind weder beim Lokalisationsfehler, noch bei der Lokalisationsunschärfe signifikante Unterschiede bei der räumlichen Diskretisierung der Lautsprecheranordnung auf 5° bzw. 15° festzustellen. Deshalb wird bei den weiteren Untersuchungen keine Rücksicht auf die räumliche Diskretisierung der Lautsprecherpositionen genommen.

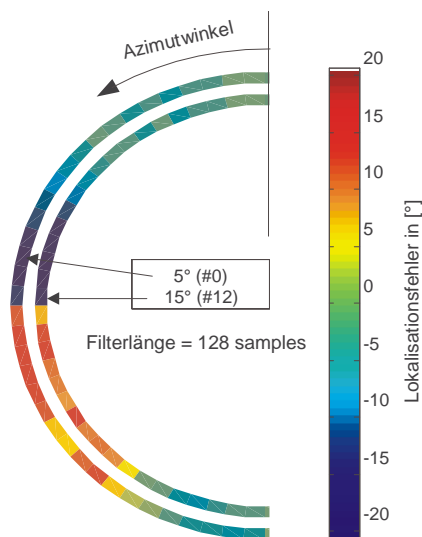


Abb.5.13: Lokalisationsfehler unterschiedlicher Referenz-HRIRs mit 128 Samples Filterlänge

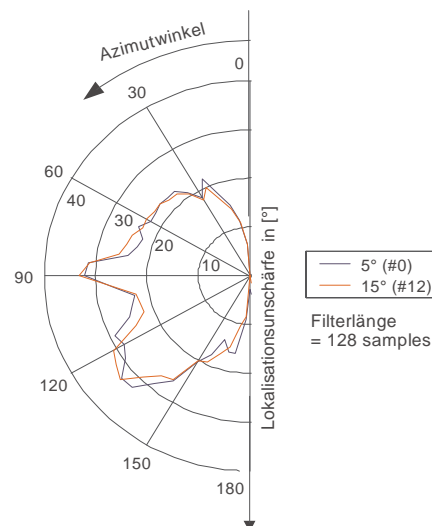


Abb. 5.14: Lokalisationsunschärfe bei unterschiedlichen Referenz-HRIRs mit 128 Samples Filterlänge

Vergleich des Lokalisationsfehlers sowie der Lokalisationsunschärfe bei Kemar und AKG Kurven

Der Vergleich erfolgt für Ambisonic-Systeme 3.Ordnung (siehe Abbildungen 5.15 und 5.16) sowie für Systeme 4.Ordnung (siehe Abbildungen 5.17 und 5.18) bei einer Filterlänge von 128 Samples. Für die Kemar-Kurven wurde trotz vorliegender 5° Auflösung eine räumliche Diskretisierung der Lautsprecherpositionen auf 15° vorgenommen. Somit kann die Einwirkung der räumlichen Diskretisierung auf das Ergebnis des Vergleiches ausgeschlossen werden.

Analyse:

Die AKG-Kurven liefern einen relativ gleichmäßigen Lokalisationsfehler über den gesamten Winkelbereich. Bei den Kemar-Kurven ist der Lokalisationsfehler in 0° Richtung sehr klein, nimmt jedoch zur Seite hin stark zu. Es ergibt sich somit bei den AKG-Kurven ein im Mittel geringerer Lokalisationsfehler, obwohl die Kemar-Kurven im vorderen Bereich eine bessere Lokalisation bieten.

Die Lokalisationsunschärfe weist bei AKG- und Kemar-Kurven die gleiche Tendenz auf. Vorne ist die Lokalisationsunschärfe sehr klein, nimmt jedoch zur Seite hin sehr stark zu.

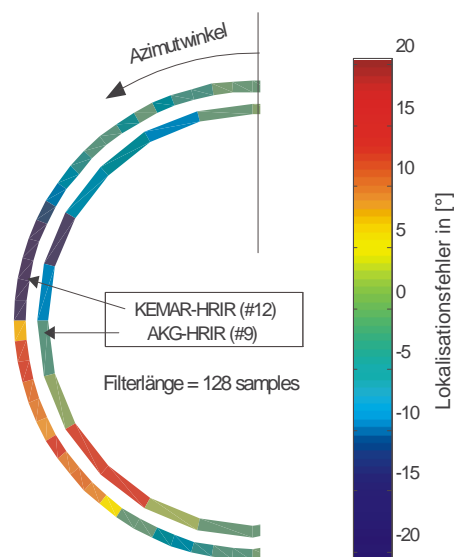


Abb. 5.15: Lokalisationsfehler bei einem System 3.Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 128 Samples

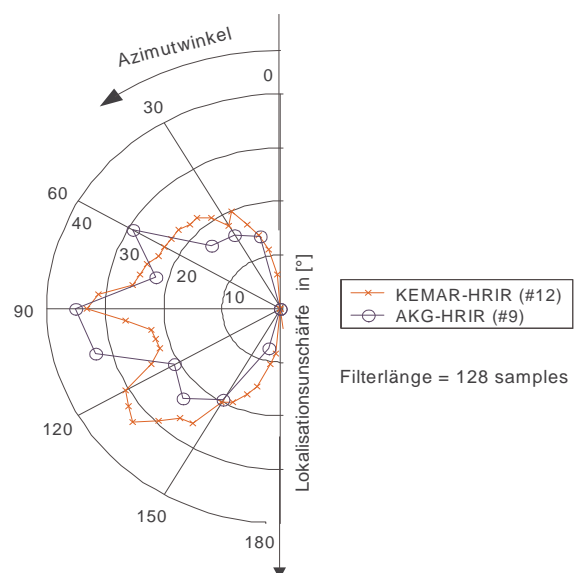


Abb. 5.16: Lokalisationsunschärfe bei einem System 3.Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 128 Samples

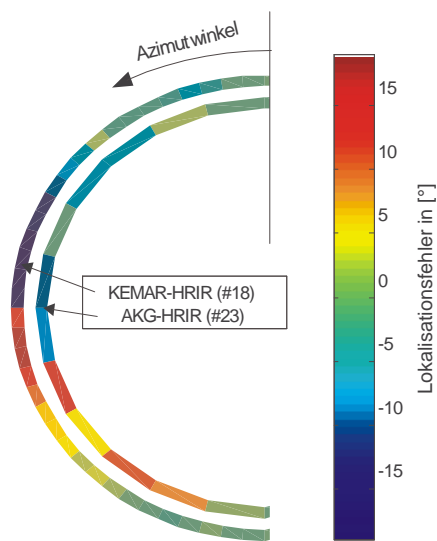


Abb. 5.17: Lokalisationsfehler bei einem System 4.Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 128 Samples

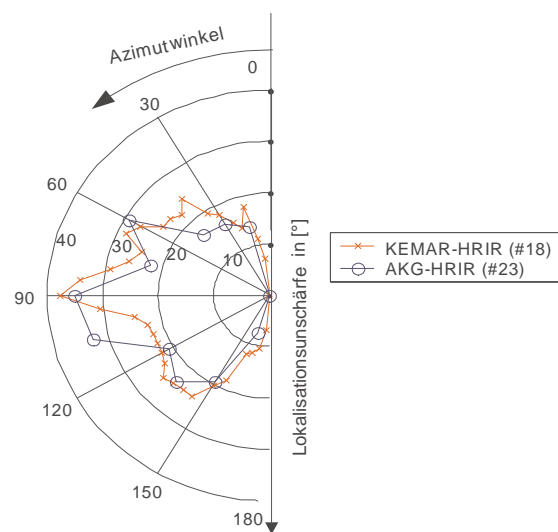


Abb. 5.18: Lokalisationsfehler bei einem System 4.Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 128 Samples

Untersuchung der Verbesserung des Lokalisationsfehlers bei Ambisonic-Systemen 4.Ordnung mit AKG-Kurven der Filterlänge von 48 Samples

In Abb. 5.11 nimmt der Lokalisationsfehler für Ambisonic-Systeme 4.Ordnung bei einer Filterlänge von 48 Samples stark ab. Dies widerspricht jedoch den theoretischen Überlegungen. Die Betrachtung von den Abbildungen 5.19 und 5.20 läßt folgende Schlussfolgerung zu:

Der in Abb. 5.11 dargestellte Lokalisationsfehler wird über den absoluten Mittelwert der Einzelfehler berechnet, d.h. die Vorzeichen der Einzelfehler gehen nicht ins Ergebnis mit ein. Bei Betrachtung der Abb. 5.19 ist ersichtlich, dass für Azimutwinkel $< 90^\circ$ negative Lokalisationsfehler entstehen. Für Azimutwinkel $> 90^\circ$ ergeben sich positive Lokalisationsfehler. In beiden Fällen bedeutet dies für die Wahrnehmung der Position einer virtuellen Schallquelle, eine zu geringe Auslateralisierung. Somit ist die Position von 90° nicht lokalisierbar. Trotz des geringen mittleren Lokalisationsfehlers liefert dieses Setting keine zufriedenstellenden Ergebnisse.

Der Sachverhalt, warum das Fehlermaß obige Werte liefert, kann wie folgt erklärt werden. Die Berechnung der ILD bzw. ITD erfolgt aus der Differenz der Referenz-HRIRs, sowie der durch das Ambisonic-System verzerrten HRIRs, zwischen linkem und rechtem Ohr. Liefern die verzerrten

HRIRs für die ILD bzw. ITD größere Werte als die Referenz HRIRs, wird die virtuelle Schallquelle voll auslateralisiert wahrgenommen. Bei einer Länge von 48 Samples enthalten die Filter wenig räumliche Information, d.h. es gibt nur geringe Unterschiede zwischen den HRIRs des ipsi- und kontralateralen Ohrs. Die Berechnung der ITD bzw. ILD für derartig kurze HRIRs liefert somit stets zu kleine Werte, was eine zu geringe Auslateralisierung der virtuellen Schallquelle zur Folge hat (vgl. auch Kap. 5.1). Bei der Optimierung der Filterlänge darf dieses Problem nicht außer acht gelassen werden.

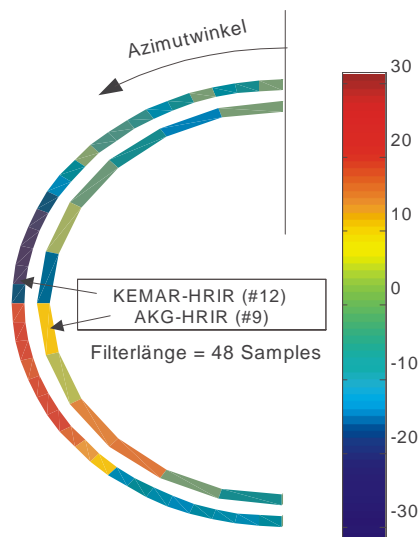


Abb. 5.19: Lokalisationsfehler bei einem System 4. Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 48 Samples

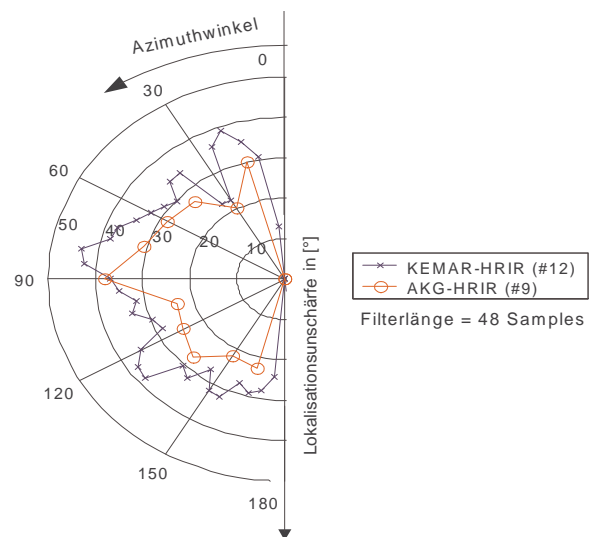


Abb. 5.20: Lokalisationsfehler bei einem System 4. Ordnung für verschiedene Referenz-HRIRs bei einer Filterlänge von 48 Samples

5.2.3 Auswirkung unterschiedlicher Lautsprecheranordnung

Wie in Kap. 4.3.2 gezeigt wurde, kann durch symmetrische Aufstellung von $N=2M+1$ Lautsprechern die Anzahl der benötigten HRIR-Filter halbiert werden. Durch die ungerade Anzahl an Lautsprechern ergeben sich nun folgende Möglichkeiten der Aufstellung (Bezeichnung wie in den Grafiken):

- 3 LS vorne: Drehung des Systems äquidistanter Lautsprecher, so dass in 0° Richtung genau ein Lautsprecher steht (Mittenslautsprecher)

- 2 LS vorne: Anordnung des Systems äquidistanter Lautsprecher, so dass links und rechts von der 0° Richtung genau ein Lautsprecher steht (kein Mittenlautsprecher)

Im Folgenden wird die Auswirkung der Lautsprecheranordnung auf die Lokalisationseigenschaften bei unterschiedlichen Referenzkurven untersucht.

Analyse:

Aus Abbildung 5.21 ist bei Vergleich der AKG-Kurven ersichtlich, dass eine Aufstellung mit 3 Lautsprechern (LS) vorne wesentlich geringere Lokalisationsfehler als mit 2 LS vorne liefert. Der Vergleich anhand von Abbildung 5.23 zeigt jedoch, dass eine Aufstellung mit 2LS vorne in 0° Richtung wesentlich bessere Werte liefert. Der Lokalisationsfehler ist bis 90° sehr konsistent. Im Bereich $> 90^\circ$ ist der Lokalisationsfehler sehr inkonsistent und die Abbildungseigenschaften sind schlechter als bei 3 LS vorne. Aus diesem Grund liefert der mittlere Lokalisationsfehler (Abb. 5.21) schlechtere Werte bei 2 LS vorne.

Mit der Anordnung von 2 LS vorne kann im vorderen Bereich ein sehr geringer und zur Seite hin konsistenter Lokalisationsfehler erreicht werden. Im hinteren Bereich weist diese Anordnung jedoch größere Abbildungsfehler als die Anordnung mit 3 LS vorne auf.

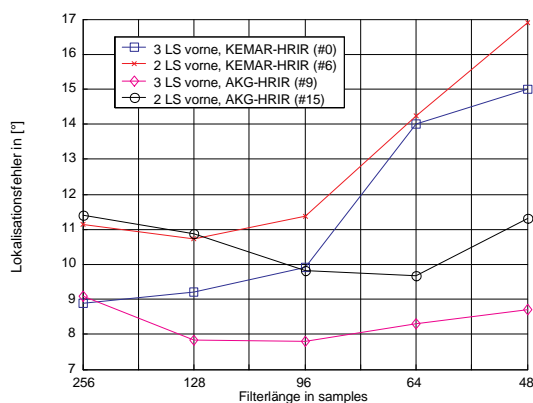


Abb. 5.21: Lokalisationsfehler verschiedener Lautsprecheranordnungen bei unterschiedlichen Referenz HRIRs

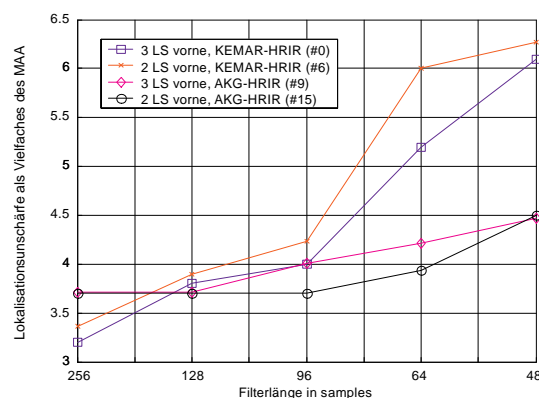


Abb. 5.22: Lokalisationsunschärfe verschiedener Lautsprecheranordnungen bei unterschiedlichen Referenz HRIRs

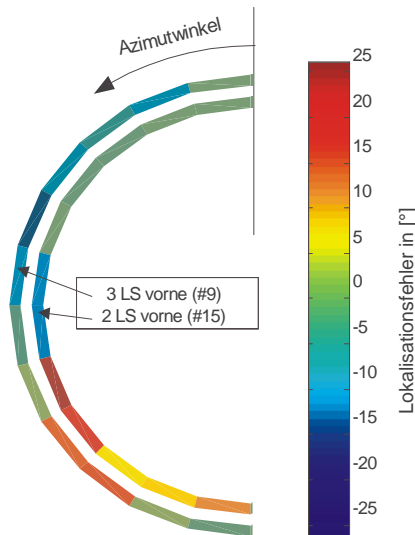


Abb. 5.23: Lokalisationsfehler bei verschiedenen Lautsprecheranordnungen für unterschiedliche Referenz HRIRs mit 128 Samples Länge

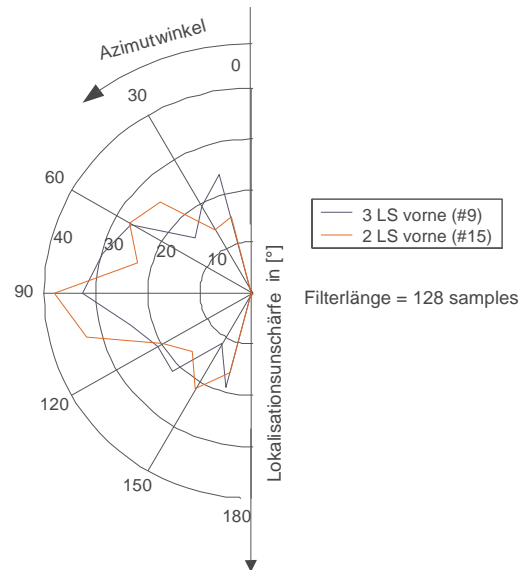


Abb. 5.24: Lokalisationsunschärfe bei verschiedenen Lautsprecheranordnungen für unterschiedliche Referenz HRIRs mit 128 Samples Länge

5.2.4 Auswirkung der Gewichtung der Ambisonic-Ordnungen

Die Auswirkung der in Kap. 4.3.3 beschriebenen Gewichtung der Ambisonic-Kanäle bei der Dekodierung auf die Lokalisationseigenschaften wird untersucht.

Gewichtung der höchsten Ordnung

Bei Gewichtung der letzten Ordnung bleibt die grundlegende Tendenz erhalten, dass kürzere Filter größere Lokalisationsfehler produzieren (vgl. Abb. 5.25 und 5.26).

Bei längeren Filtern wird der Effekt der Verbreiterung der Hauptkeule durch Gewichtung der höchsten Ordnung gut sichtbar (vgl. Kap. 4.3.3). Die Verkleinerung der Gewichtung der höchsten Ordnung führt durch die breitere Hauptkeule, zu einem Anstieg der Lokalisationsunschärfe (vgl. Abb. 5.28). Durch die damit verbundene größere Dämpfung der Nebenkeulen, verringert sich der Lokalisationsfehler (vgl. Abb. 5.27). Bei einer Gewichtung größer 0.8 steigt die Lokalisationsunschärfe wieder an. Die höheren Nebenkeulen wirken der schmälere Hauptkeule entgegen.

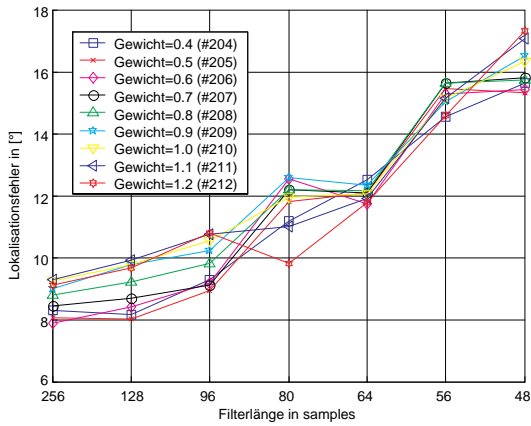


Abb. 5.25: Lokalisationsfehler bei unterschiedlicher Gewichtung der 3.Ordnung für verschiedene Filterlängen

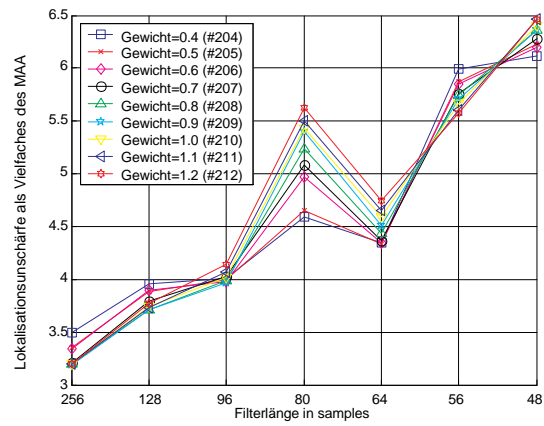


Abb. 5.26: Lokalisationsunschärfe bei unterschiedlicher Gewichtung der 3.Ordnung für verschiedene Filterlängen

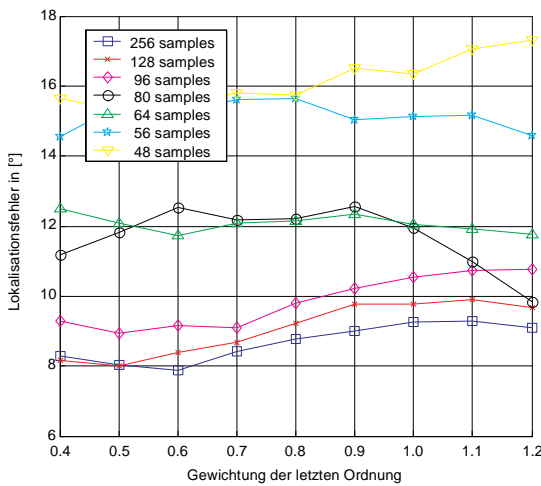


Abb. 5.27: Lokalisationsfehler bei verschiedener Gewichtung der 3.Ordnung und unterschiedlichen Filterlängen

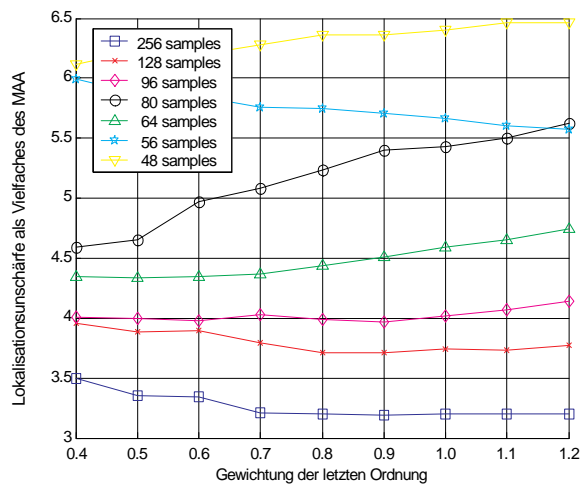


Abb. 5.28: Lokalisationsunschärfe bei verschiedener Gewichtung der 3.Ordnung und unterschiedlichen Filterlängen

Systeme 3. und 4.Ordnung mit Kaiserfenster

Durch Verwendung verschiedener Fenstertechniken, lässt sich eine weitere Verbesserung der Lokalisation des Systems erreichen (vgl. Kap. 4.3.3).

In Abb. 5.29 ist ersichtlich, dass die Gewichtung der 3.Ordnung mit 0.4 zu einer Verschlechterung des Lokalisationsfehlers führt. Eine geringe Gewichtung der 3.Ordnung kommt annähernd einem voll gewichteten System 2.Ordnung gleich. Um dieses Problem zu umgehen, wird nicht nur die

3.Ordnung gewichtet, sondern eine Fensterfunktion implementiert. Aus der signaltheoretischen Betrachtung (vgl. Kap. 4.3.3) liefert die Verwendung eines Kaiserfensters eine wesentliche Verbesserung. Dies ist auch Abb. 5.29 zu entnehmen, wo ein System 3.Ordnung, gewichtet mit einem Kaiserfenster, sogar bessere Werte als ein voll gewichtetes System 4.Ordnung bringt. Weiters ist ersichtlich, dass ein System 4.Ordnung, gewichtet mit einem Kaiserfenster, eine wesentliche Verbesserung des Lokalisationsfehlers darstellt. Abb. 5.31 zeigt den Lokalisationsfehler bei einer Filterlänge von 128 Samples. Es ist gut ersichtlich, dass der Lokalisationsfehler eines Systems 3.Ordnung im vorderen Bereich annähernd einem System 4.Ordnung gleichkommt. Bei 90° tritt eine Verkleinerung des schwer lokalisierbaren Bereiches ein. Die Gewichtung des Systems 4.Ordnung mit einem Kaiserfenster, führt zu einem kleineren Lokalisationsfehler über den gesamten Bereich. Bei Winkeln > 90° tritt gegenüber einem System 3.Ordnung mit Kaiserfenster eine wesentliche Verbesserung auf. Abb. 5.32 ist zu entnehmen, dass keine wesentliche Verschlechterung der Lokalisationsunschärfe bei Systemen 4.Ordnung mit Kaiserfenster entsteht.

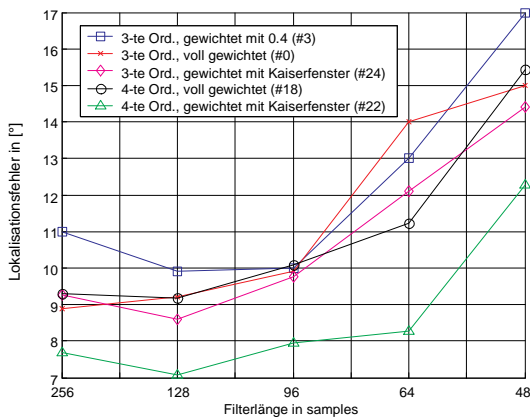


Abb. 5.29: Lokalisationsfehler bei Gewichtung der höchsten Ordnung und Kaiserfenster für verschiedene Filterlängen

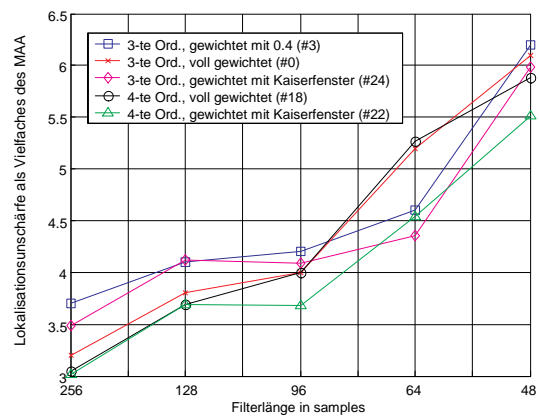


Abb. 5.30: Lokalisationsunschärfe bei Gewichtung der höchsten Ordnung und Kaiserfenster für verschiedene Filterlängen

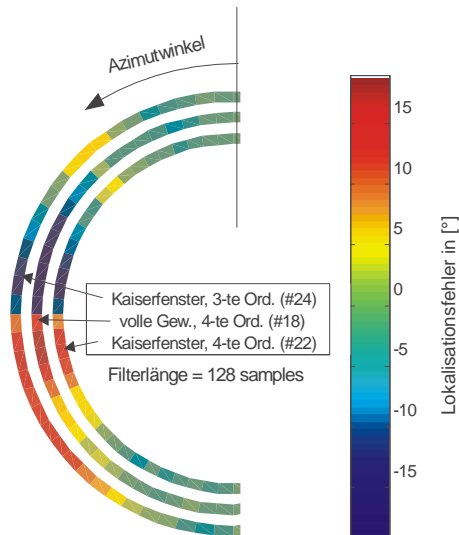


Abb. 5.31: Lokalisationsfehler bei Gewichtung der höchsten Ordnung und Kaiserfenster für eine Filterlänge von 128 Samples

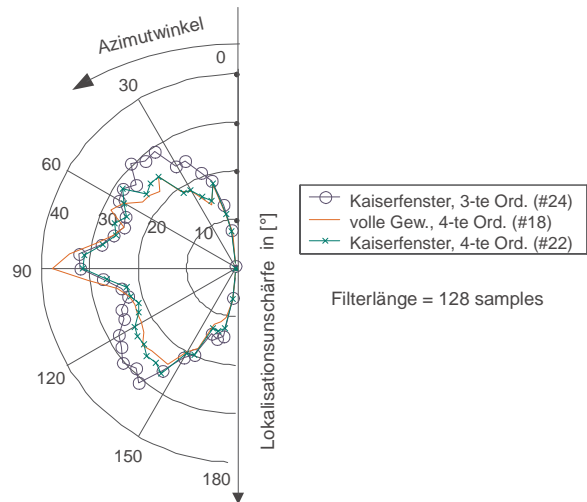


Abb. 5.32: Lokalisationsunschärfe bei Gewichtung der höchsten Ordnung und Kaiserfenster für eine Filterlänge von 128 Samples

6 Hörversuche

6.1 Einleitung

Die durchgeführten Hörversuche dienen der Verifizierung der theoretischen Betrachtungen. Die Auswirkung der einzelnen Parameter des binauralen Wiedergabesystems (Ambisonic-Ordnung, HRIR-Filterlänge, Aufstellung der virtuellen Lautsprecher, usw.) auf die Wahrnehmungsparameter (Lokalisationsfehler, Lokalisationsunschärfe, Klangfarbe, usw.) wird anhand von Hörversuchen ermittelt. Zur Reduzierung der zu untersuchenden Parameter erfolgt eine Simulation des binauralen Wiedergabesystems in MATLAB und eine Bewertung der Ergebnisse anhand des Fehlermaßes (vgl. Kap. 5.2, sowie Anhang B). Aus diesen Ergebnissen werden verschiedene Parameter-Settings generiert.

Zur Durchführung der Hörversuche, wurde das Gesamtsystem auf einem DSP implementiert. Dabei erfolgt die Steuerung der Parameter über ein Benutzerinterface (GUI, *graphical user interface*) durch Einbindung eines PCs (siehe Abb. 6.1).

6.2 Hörversuchsdesign

Beim Design von Versuchsreihen sind folgende Aspekte bezüglich der Aussagekraft einer auf statistischen Methoden basierenden Untersuchung zu berücksichtigen ([7], [9]):

- Objektivität
- Reliabilität (Konstanz der Urteile)
- Validität (Abschätzung ob durch Versuch gewünschtes Ziel erreicht wird)

Um die wahrgenommene Position der virtuellen Schallquelle bestimmen zu können, müssen die Versuchspersonen (Vpn⁹) diese mit einer über Lautsprecher zugespielten „realen“ Schallquelle zur Deckung bringen. Die Zuspiegelung der realen Signale erfolgt direkt, d.h. es werden keine Phantomschallquellen gebildet. Da die Vpn über diesen Sachverhalt instruiert werden, kann die Wahrnehmung der Position der realen Schallquelle auf einen Punkt konzentriert werden (vgl. „Räumliches Sampling“ in [12]). Die virtuelle Schallquelle kann mit einem Joystick bewegt werden, wobei die Startposition mindestens 30° von der Position der realen Quelle entfernt ist. Ist die Wahrnehmung der Position der virtuellen Schallquelle deckungsgleich mit der realen Schallquelle, erfolgt die Bestätigung durch Betätigen des Joystick-Buttons. Die so ermittelte Position wird ausgelesen und automatisch protokolliert.

Um der Vorne/Hinten Ambiguität entgegen wirken zu können werden die Kopfdrehungen mit einem Headtracker erfasst. Während des gesamten Versuchs dürfen die Vpn den Kopf nur in einem Bereich von $\pm 5^\circ$ bewegen, wobei zu große Bewegungen der Vpn durch Anzeige auf einem Display signalisiert werden (vgl. Anhang C). Weiters wird eine Protokollierung der Kopfbewegungen durchgeführt. Um bei der realen Schallquelle eine Vorne/Hinten Verwechslung zu vermeiden, erfolgt die Zuspiegelung nur über Lautsprecher aus den beiden vorderen Quadranten ($0^\circ \pm 90^\circ$). Die Vpn werden auf diesen Sachverhalt explizit hingewiesen. Damit kann der Einfluss der Vorne/Hinten Verwechslung realer Quellen auf das Ergebnis ausgeschlossen werden.

Die Bewertung der Parameter

- Klangfarbe der virtuellen Schallquelle
- Externität der Schallquelle (Außerkopflokalisation)
- Lokalisationsunschärfe
- Konsistenz der Bewegung der virtuellen Schallquellen

erfolgt durch verbale Befragung. Dabei werden den Vpn die verwendeten Begriffe und deren Bedeutung erläutert. Bei der Bewertung obiger Parameter können die Vpn die virtuelle Schallquelle frei bewegen, wobei mittels des GUIs die Schallquellenposition angezeigt wird. Weiters werden die zu bewertenden Begriffe und eine dazugehörige Bewertungsskala am Bildschirm angezeigt. Während dieser Versuchsphase erfolgte keine Zuspiegelung über die Lautsprecher.

⁹ Im Folgenden wird Vpn als Abkürzung für den Begriff Versuchsperson(en) verwendet

6.3 Versuchspersonen

Der Versuch wurde an 10 freiwilligen Versuchspersonen (4 weiblich, 6 männlich, Alter zwischen 20 und 40 Jahren) durchgeführt. Von den 10 Personen hatten 6 bereits Erfahrung mit Hörversuchen. Vor der Durchführung der Versuchsreihe wurden die Vpn mit der Steuerung der virtuellen Schallquelle, sowie dem Softwareinterface (GUI, *Graphical User Interface*) vertraut gemacht. Weiters erfolgte eine Instruierung der Vpn, Kopfbewegungen nur in dem durch das GUI vorgegebenen Bereich durchzuführen (vgl. Anhang C). Dabei erfolgte kein Training der Vpn, den Kopf in einer bestimmten Weise zu bewegen. Um Fehlern durch Ermüdung vorzubeugen, wurden die Versuchsreihen auf mehrere Tage verteilt durchgeführt und zwischen den jeweiligen Settings¹⁰ Pausen eingehalten.

6.4 Verwendete Testsignale

Mögliche Applikationen des binauralen Wiedergabesystems umfassen hauptsächlich die Spatialisation von Sprach- und Musiksignalen (virtuelles Surround, Telekonferenzen, usw.). Ergebnisse von Hörversuchen, welche künstlich generierte Signale wie Rauschen¹¹ und Klicks verwenden, können laut Begault in [4] nicht mit Studien verglichen werden, die natürliche Signale verwenden. Laut Blauert ([5], S 32 ff.) eignen sich bekannte Signale gut zur Untersuchung natürlicher Lokalisation, da monaurale Faktoren der Lokalisation (vgl. Kap. 2.2.1) nicht ausgeschlossen werden. Aus diesen Gründen wurden für die Durchführung des Hörversuchs Sprachsignale gewählt. Dabei wurden jeweils eine männliche und eine weibliche Stimme als reales bzw. virtuelles Signal zugespielt, um eine gute Unterscheidbarkeit zu gewährleisten.

6.5 Versuchsaufbau und Hardware

Der grundlegende Versuchsaufbau kann Abb. 6.1 entnommen werden. Die zu untersuchenden Algorithmen zur Berechnung der Binauralsignale wurden auf einem DSP (Texas Instruments TMS320C6711) in Verbindung mit einem Audio-Codec (Texas Instruments TLV320AIC27)

¹⁰ Der Begriff *Setting* beschreibt eine gewisse Parametrisierung des Ambisonic-Systems (vgl. Kap. 5 und Anhang C)

¹¹ Mit Rauschsignalen können auch die hohen Frequenzbereiche der Hörwahrnehmung erfasst werden. Zum Vergleich: Bei Sprachsignalen liegt die maximale Energie im Oktavband bei 500Hz wobei zu höheren und tieferen Frequenzen ein Abfall von 6dB/Oktave stattfindet [4]

implementiert. Den Versuchspersonen wurden die binauralen Signale über einen elektrostatischen Kopfhörer (Sennheiser HEV70) zugespielt, welcher fix mit einem Headtracker (Ascension Technology, Flock of Birds) verbunden war. Die gesamte Ablaufsteuerung, sowie die benötigten Benutzeroberflächen (GUIs, *graphical user interfaces*) wurden in Visual Basic unter Einbindung eines handelsüblichen PCs realisiert (vgl. Anhang C).

Bei der Aufstellung der Lautsprecher (Tannoy System 800, Verstärker: t-amp S150) wurde eine Winkelauflösung von 10° gewählt. Unter Ausnutzung der Symmetrieeigenschaften des zu untersuchenden Systems (vgl. Kap. 4.1.4) wurden die Lautsprecher wie folgt angeordnet (Abb. 6.2).

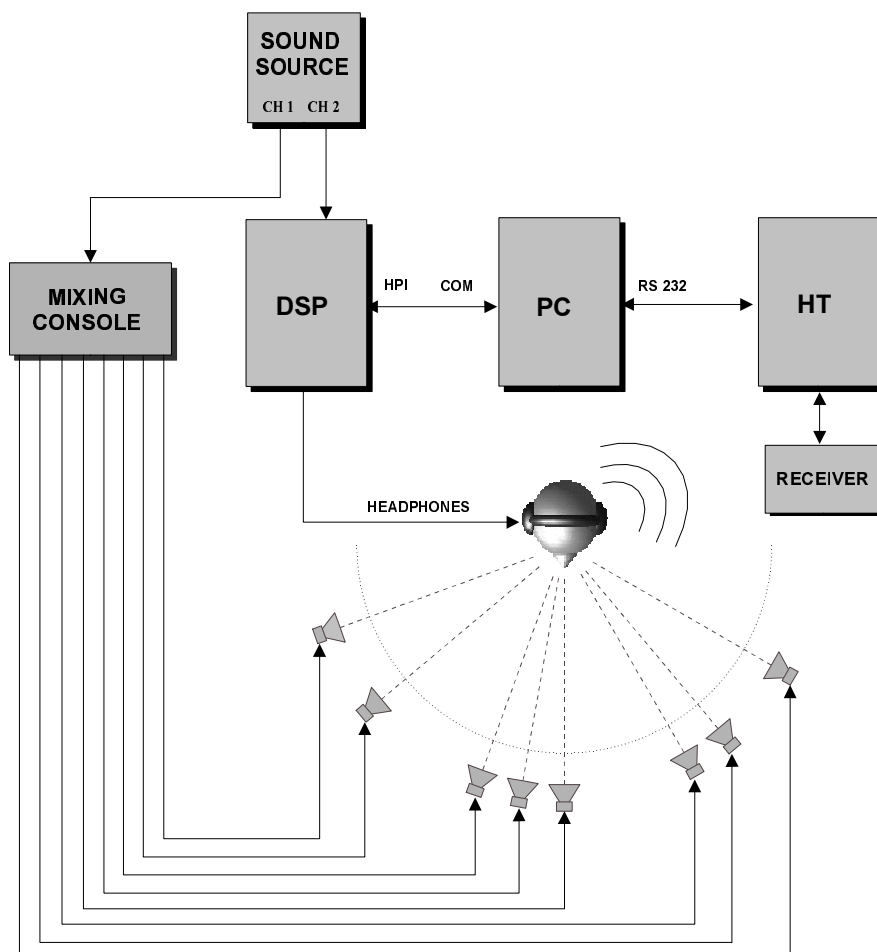


Abb. 6.1: Gesamtaufbau zur Durchführung der Hörversuche

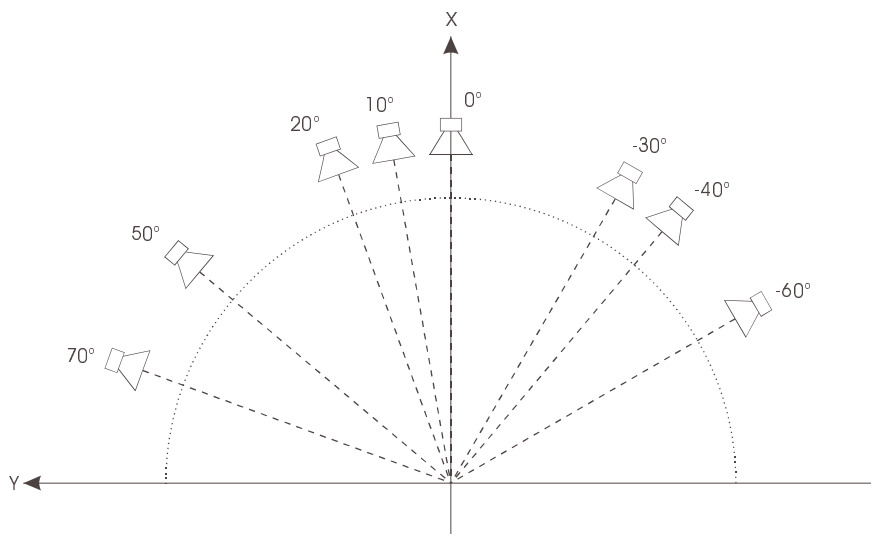


Abb. 6.2: Aufstellung der Lautsprecher mit einer Winkelauflösung von 10° unter Ausnutzung der Symmetrieeigenschaften des Systems

Die Durchführung erfolgte im Experimentalstudio des IEM¹², wobei durch geeignete Aufstellung variabler Schallabsorberwände (siehe Anhang C) eine mittels MLS¹³ gemessene Nachhallzeit von 0,27s erreicht wurde.

6.6 Versuchsablauf (chronologisch)

1. Instruierung der Versuchspersonen (Vpn)
2. Trainings- und Eingewöhnungsphase
3. Kalibrieren des Gesamtsystems auf die 0° Richtung
4. Befragen: Klangfarbe, Lokalisationsunschärfe, Externität, Konsistenz der Bewegung
Dieser Punkt wird für jedes Setting nur einmal ausgeführt.

5. Offsetmessung

Für ein Setting werden je Lautsprecherposition n Messungen durchgeführt, wobei die Ansteuerung der jeweiligen Lautsprecher nach einer Zufallsfolge erfolgt. Zwischen den einzelnen Messpunkten erfolgt eine kurze Stummschaltung des Gesamtsystems. Beim nächsten Messpunkt liegt die Startposition der virtuellen Schallquelle mindestens 30° von der Position der realen Quelle entfernt. Nach Abschluss eines Settings, erfolgt eine kurze Pause.

¹² Institut für Elektronische Musik und Akustik der Universität für Musik und Darstellende Kunst, Graz

¹³ Maximum Length Sequences [40]

6.7 Mathematische Auswertung der Hörversuche

6.7.1 Aufbereiten der Daten

Offsetmessung:

Es erfolgt ein automatisiertes Einlesen der einzelnen Werte der Offsetmessung in MATLAB, wobei eine Klassifizierung der Daten nach folgenden Kriterien durchgeführt wird:

- Versuchsperson (Vpn)
- Nummer des Settings ($SetNr$, ist dem Anhang C zu entnehmen)
- Position der realen Quelle ($QPos$)
- Nummer der Messung ($VersNr$: (1 .. n) da pro Position n Messungen durchgeführt wurden)

Es entsteht die vierdimensionale Matrix der Offsetmessung

$$OS(Vpn, SetNr, VersNr, Qpos) \quad (6.1)$$

deren numerische Einträge die jeweiligen Messwerte darstellen. Auf diese Matrix können nun diverse statistische und numerische Methoden angewendet werden.

Verbale Befragung:

Um die Ergebnisse der verbalen Befragung mathematisch weiterverarbeiten zu können, müssen diese in ein numerisches Format transformiert werden. Den Aussagen der jeweiligen Versuchspersonen werden Zahlenwerte zugewiesen (Schlüssel zur Kodierung siehe Anhang C). Zur statistischen Weiterverarbeitung liegen die Ergebnisse dann in zweidimensionalen Matrizen, in Abhängigkeit von Vpn und Setting Nummer vor (vgl. auch Kap. 6.7.6).

6.7.2 Vorne/Hinten Ambiguität

Die Matrix der Offsetmessung wird auf Werte durchsucht, die in den hinteren beiden Quadranten liegen. Da die realen Schallquellen nur aus den vorderen Quadranten zugespielt wurden, liegt bei diesen Werten eine Vorne/Hinten Verwechslung vor. Für jeden gefundenen Wert wird ein Marker (*Flag*) gesetzt.

Es entsteht eine vierdimensionale Ambiguitätsmatrix **AG** :

$$\mathbf{AG}(Vpn, SetNr, VersNr, QPos) \quad (6.2)$$

Folgende Untersuchungen werden hinsichtlich der Vorne/Hinten Ambiguität durchgeführt:

1. Vorne/Hinten Ambiguität der jeweiligen Versuchspersonen:

Es kann eine Aussage darüber getroffen werden, wie oft bei der jeweiligen Vpn Vorne/Hinten Ambiguitäten aufgetreten sind. Der Querstrich kennzeichnet jene Parameter über die gemittelt wurde (vgl. Abb. 6.8).

$$\mathbf{AG}(Vpn) = \mathbf{AG}(Vpn, \overline{SetNr}, \overline{VersNr}, \overline{QPos}) \quad (6.3)$$

2. Vorne/Hinten Ambiguität für die jeweiligen Quellenpositionen:

Es kann eine Aussage darüber getroffen werden, bei welchen Schallquellenpositionen und welcher Vpn vermehrt Vorne/Hinten Ambiguitäten aufgetreten sind (vgl. Abb. 6.9).

$$\mathbf{AG}(Vpn, QPos) = \mathbf{AG}(Vpn, \overline{SetNr}, \overline{VersNr}, QPos) \quad (6.4)$$

3. Vorne/Hinten Ambiguität für die jeweiligen Settings:

Es kann eine Aussage darüber getroffen werden, bei welchen Settings (vgl. auch Anhang C) vermehrt Vorne/Hinten Ambiguitäten aufgetreten sind. Hinsichtlich der Optimierung des Systems, stellt dies die wichtigste Untersuchung der Ambiguität dar (vgl. Abb. 6.10).

$$\mathbf{AG}(SetNr) = \mathbf{AG}(\overline{Vpn}, SetNr, \overline{VersNr}, \overline{QPos}) \quad (6.5)$$

6.7.3 Ambiguitätsbefreiung der Daten

Bei dem in Kap. 6.7.2 beschriebenen Vorgang zur Erzeugung der Ambiguitätsmatrix \mathbf{AG} wird gleichzeitig die ambiguitätsbefreite Offsetmatrix $\hat{\mathbf{OS}}(Vpn, SetNr, VersNr, QPos)$ berechnet. Dabei werden alle Datenpunkte in die vorderen beiden Quadranten ($\varphi = \pm 90^\circ$) transformiert, da die Zuspaltung der „realen“ Schallquelle über Lautsprecher nur von vorne durchgeführt wurde.

6.7.4 Berechnen der Streuparametermatrix

Bei der Durchführung des Hörversuches wurden für jede Schallquellenposition n Messpunkte pro Versuchsperson und Setting aufgenommen. Wird nun für jede Position aus den n Messpunkten die Standardabweichung σ berechnet, entsteht aus der vierdimensionalen ambiguitätsbefreiten Matrix der Offsetmessung \hat{OS} die dreidimensionale Matrix der Standardabweichung STD :

$$STD(Vpn, SetNr, Qpos) = std[\hat{OS}(Vpn, SetNr, VersNr, Qpos)] \quad (6.6)$$

d.h. je genauer eine Vpn eine virtuelle Schallquelle lokalisieren konnte, desto kleiner wird die Standardabweichung.

Je weiter eine Schallquelle auslateralisiert wird, desto schlechter kann durch die menschliche Hörwahrnehmung die Position lokalisiert werden [6]. Die Standardabweichung stellt somit kein repräsentatives Maß dar, da sich die Einzelwerte nicht auf ein Maß reduzieren lassen. Es wird für jede Quellposition der Mittelwert der n gemessenen Werte berechnet und der dem Azimutwinkel zugehörige MAA^{14} bestimmt. Die Standardabweichung wird nun auf diesen MAA bezogen. Es ergeben sich die jeweiligen Streuparameter, bzw. die Streuparametermatrix $S(Vpn, SetNr, QPos)$. Die einzelnen Werte der Streuparametermatrix können somit gemittelt werden.

6.7.5 Bewertung der Offsetmessung

Ausgangspunkt für die weiteren Berechnungen ist die dreidimensionale Streuparametermatrix $S(Vpn, SetNr, QPos)$. Ein wesentlicher Punkt bei der Weiterverarbeitung ist die Feststellung signifikanter Unterschiede zwischen den Datensätzen. Die Berechnung der Signifikanz erfolgt mittels der *Einfaktoriellen Varianzanalyse* (ANOVA, *Uni-variate Analysis of Variance*) in MATLAB.

Folgende Berechnungen werden durchgeführt:

1. Bewertung der Lokalisationsfähigkeit der jeweiligen Versuchspersonen:

Die Berechnung der ANOVA über die Vpn ergibt, dass signifikante Unterschiede bei der

¹⁴ *MAA* (*minimum audible angle*), vgl. Kap. 2.2.3

Lokalisation der Quellpositionen durch die einzelnen Vpn bestehen. Somit kann die Streuparametermatrix über alle Quellpositionen und Settings gemittelt werden. Es ergibt sich für jede Vpn ein Wert, der eine Aussage ermöglicht, wie sicher eine Vpn im Durchschnitt lokalisiert (vgl. Abb. 6.3).

2. Bewertung der Lokalisation der jeweiligen Parameter-Settings:

Die ANOVA über die SetNr ergibt, dass kein signifikanter Unterschied zwischen den jeweiligen Parameter Settings besteht. Um trotzdem Vergleiche der Settings zueinander durchführen zu können, wurde wie folgt vorgegangen:

Die Mittelung der Streuparametermatrix in Pkt. 1 ergibt eine mittlere Lokalisationsunsicherheit je Vpn, d.h. die Unsicherheit mit der eine Vpn lokalisiert. Diese kann als Gleichanteil (bias) der Streuparametermatrix aufgefasst werden. Durch Subtraktion des Gleichanteils von den zur jeweiligen Vpn gehörenden Streuparametern erhält man die vom Gleichanteil befreite Streuparametermatrix $\mathbf{S}_{unbias}(Vpn, SetNr, QPos)$. Eine Mittelung über die Vpn und QPos ergibt die Matrix

$$\bar{\mathbf{S}}_{unbias} = \mathbf{S}_{unbias}(\overline{Vpn}, \overline{SetNr}, \overline{QPos}) \quad (6.7)$$

Die Darstellung von $\bar{\mathbf{S}}_{unbias}$ in einem zweidimensionalen Diagramm erlaubt nun einen Vergleich von jeweils zwei Settings zueinander (vgl. Abb. 6.4).

Ergibt die ANOVA zwischen zwei Settings signifikante Unterschiede, können die Mittelwerte der Streuparameter dieser Settings miteinander verglichen werden.

6.7.6 Auswertung der verbalen Befragung

Nach der in Kap. 6.7.1 erläuterten Aufbereitung der verbalen Daten, liegen folgende Datensätze in Matrizenform vor:

- **Lokalisationsunschärfematrix $\mathbf{LU}(SetNr, Vpn)$:**

Aus der Aufbereitung der Daten für die Lokalisationsunschärfe liegen diese für vorne (0° Richtung) und seitlich ($\pm 90^\circ$ Richtung) vor. Die Matrix \mathbf{LU} wird durch Mittelwertbildung der jeweiligen Daten erzeugt (siehe Abb. 6.11).

- **Externitätsmatrix EX** ($SetNr, Vpn$) :

Die Daten liegen wiederum für vorne und seitlich vor, die Matrix wird durch Mittelwertbildung erzeugt (siehe Abb. 6.13, 6.14).

- **Klangmerkmalsmatrizen:**

Die Berechnung erfolgt wiederum wie oben, wobei für

- Hohe Frequenzen: $KM_H(SetNr, Vpn)$

- Mittenfrequenzen: $KM_M(SetNr, Vpn)$

- Tiefe Frequenzen (Bässe): $KM_B(SetNr, Vpn)$

eigene Matrizen gebildet werden. Bei der Auswertung der Datensätze müssen diese im Kontext zueinander betrachtet werden (siehe Abb. 6.15).

- **Konsistenzmatrix KO** ($SetNr, Vpn$) :

Es muss keine Mittelwertbildung erfolgen, da die Daten für den gesamten Bereich ($0^\circ - 360^\circ$) global erfasst wurden (siehe Abb. 6.12).

6.8 Ergebnisse

6.8.1 Beurteilung der Versuchspersonen

Der Mittelwert der Streuparameter gibt Auskunft wie sicher eine Vpn lokalisiert. Abb. 6.3 ist zu entnehmen, dass Vpn 7 große Probleme bei der Lokalisierung virtueller Schallquellen hat. Vpn 2 weist eine sehr sichere Lokalisation auf. Dies ist wahrscheinlich darauf zurückzuführen, dass diese Vpn sehr große Erfahrung bei der Lokalisation virtueller Schallquellen mittels binauraler Wiedergabesysteme aufweist. Vpn 6 lokalisiert ebenfalls sehr gut, ist jedoch im Vergleich zu Vpn 2 völlig unerfahren im Umgang mit binauralen Systemen ist. Die Auswirkung der Hörerfahrung auf die Lokalisation müsste getrennt untersucht werden. Um in weiterer Folge Vergleiche zwischen den Settings durchführen zu können, dürfen die Streuparameter nicht von der Lokalisationsfähigkeit der jeweiligen Vpn beeinflusst werden. Es erfolgt eine Mittelwertbefreiung wie in Kap. 6.7.5 beschrieben.

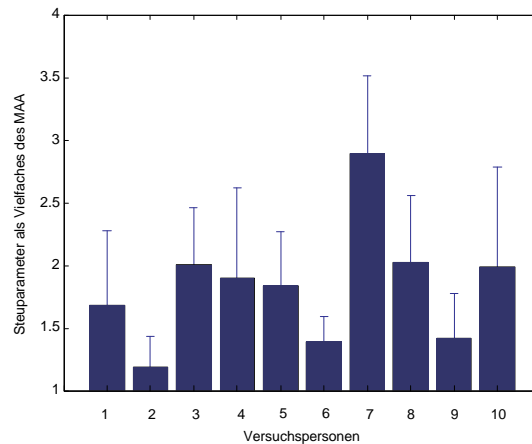


Abb. 6.3: Mittelwert des Streuparameters für die jeweiligen Versuchspersonen (Mittelwert und Standardabweichung)

6.8.2 Unterschiede zwischen den Settings

Aus Abb. 6.4 lassen sich keine globalen Aussagen über den Unterschied der Settings treffen. Tendenzen sind auch nicht erkennbar. Ein geringer Mittelwert über alle Vpn. (kleiner Streuparameter) bei einem Setting bedeutet sicheres Lokalisieren durch die Vpn. Die Standardabweichung in der Darstellung ist ein Maß für die Personenabhängigkeit des jeweiligen Settings, wobei eine höhere Standardabweichung größere Personenabhängigkeit eines Settings bedeutet. Es ist somit deutlich zu erkennen, dass die Kemar HRIRs eine geringere Personenabhängigkeit aufweisen als die AKG-HRIRs. Über die Lokalisationssicherheit kann anhand dieser Abbildung keine eindeutige Aussage getroffen werden. Um Aussagen treffen zu können erfolgt ein Vergleich der Settings mittels ANOVA.

Bei der Untersuchung des Einflusses der jeweiligen HRIR-Kurven¹⁵ auf die Lokalisationsunsicherheit liefert ANOVA $F_{(1,98)} = 4.94$, $p = 0.0285$. Dies bedeutet, dass ein signifikanter Unterschied besteht. Eine Unterteilung der Streuparameter in zwei Gruppen (AKG, Kemar) und Mittelung über alle Settings, ergibt folgende Darstellung (vgl. Abb. 6.5). Es ist ersichtlich, dass die Kemar-HRIRs eine höhere Lokalisationssicherheit aufweisen.

¹⁵ Vergleich der Settings (7, 8, 9, 10, 11) mit (1, 2, 3, 4, 5, 6)

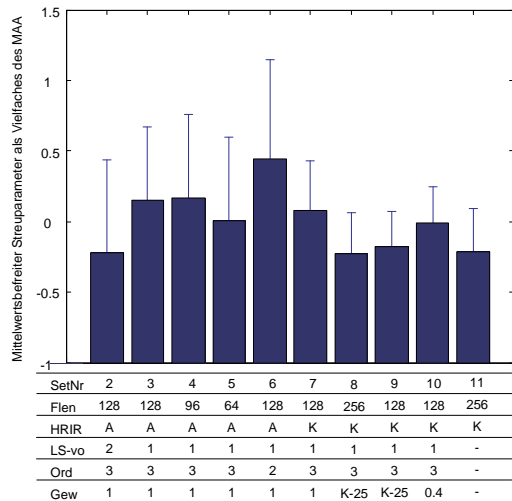


Abb. 6.4: Mittelwertsbefreiter Streuparameter für die jeweiligen Settings (Mittelwert und Standardabweichung)

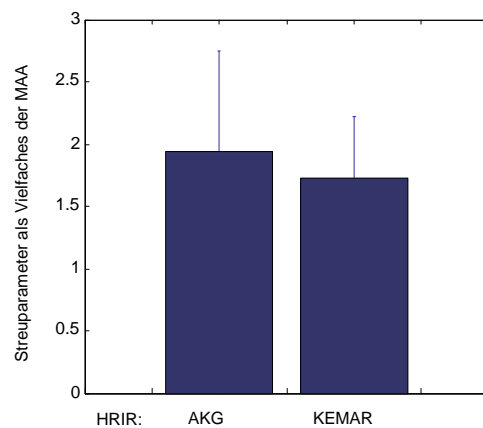


Abb. 6.5: Lokalisationsunsicherheit für versch. HRIR-Kurven als Mittelwert der Streuparameter (Mittelwert und Standardabweichung)

Weiters wird der Einfluss der Ambisonic-Ordnung auf die Lokalisationsunsicherheit untersucht. Für den Vergleich der Settings (2, 3, 4, 5) mit (6) liefert ANOVA $F_{(1,48)} = 3.72$, $p = 0.0598$. Es ergibt sich ein geringer, jedoch nicht signifikanter Unterschied. Die Darstellung der Lokalisationsunsicherheit für diese zwei Gruppen bestätigt die theoretischen Betrachtungen, dass eine Ambisonic-System geringerer Ordnung eine höhere Lokalisationsunsicherheit aufweist (vgl. Abb. 6.6).

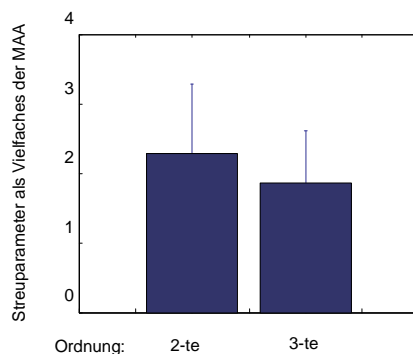


Abb. 6.6: Lokalisationsunsicherheit für verschiedene Ambisonic-Ordnungen als Mittelwert der Streuparameter (Mittelwert und Standardabweichung)

Eine weitere Abhängigkeit konnte mittels ANOVA gefunden werden. Der Einfluss der Gewichtung der Ambisonic-Ordnungen (Fensterung) auf die Lokalisationsunsicherheit weist signifikante Unterschiede bei Betrachtung der Settings (8, 9) gegen (7, 10) auf. Setting (8, 9) sind mit einem Kaiserfenster gewichtet, Setting 7 ist ungewichtet und bei Setting 10 ist die letzte Ordnung mit 0.4

gewichtet. Die ANOVA für diese Settings liefert $F_{(1,38)} = 7.61$, $p = 0.0117$. Stellt man die Lokalisationsunsicherheit dar (Abb. 6.7), lässt sich eine Verbesserung bei Verwendung eines Kaiserfensters feststellen.

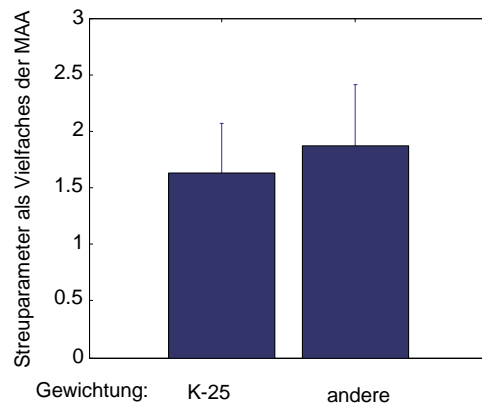


Abb. 6.7: Lokalisationsunsicherheit für verschiedene Gewichtungen der Ambisonic-Ordnungen (Fensterung) als Mittelwert der Streuparameter (Mittelwert und Standardabweichung)

6.8.3 Auswertung der Ambiguität

Wie Abb. 6.8 zu entnehmen ist, weist die Vorne/Hinten Ambiguität eine starke Personenabhängigkeit auf. Vpn 7 sowie Vpn 3 hatten eindeutige Probleme bei der Unterscheidung zwischen vorne und hinten. Es trat bei allen Vpn, bis auf Vpn 6, mindestens eine Fehllokalisierung auf.

Die Vorne/Hinten Ambiguität für die verschiedenen Lautsprecherpositionen ist in Abb. 6.9 dargestellt. Je größer die Auslateralisierung der virtuellen Schallquelle ist, desto schwieriger wird es eine Vorne/Hinten Unterscheidung zu treffen. Dies lässt sich auf die zur Seite hin steigende Lokalisationsunschärfe zurückführen.

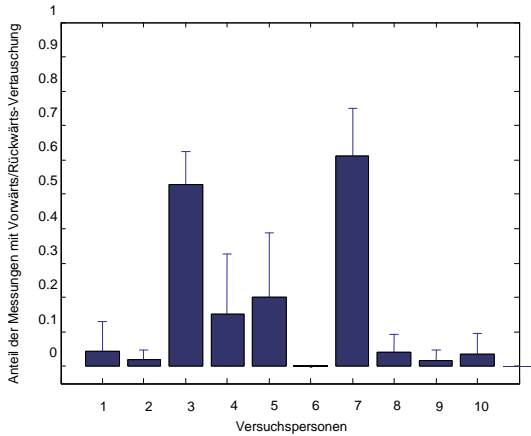


Abb. 6.8: Vorne/Hinten Ambiguität der Versuchspersonen

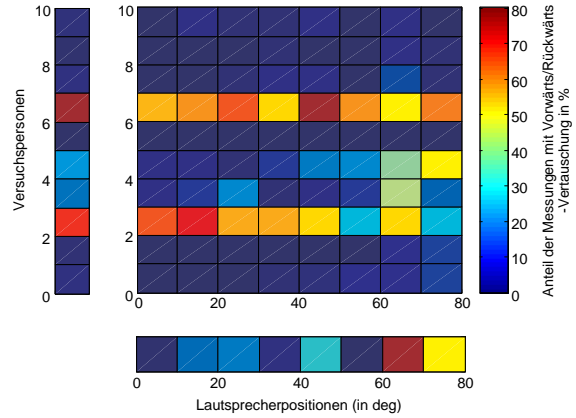


Abb. 6.9: Vorne/Hinten Ambiguität für die Lautsprecherposition sowie die Versuchspersonen.

Die Untersuchung der Vorne/Hinten Ambiguität bezüglich der Settings, ergibt eine Erhöhung der Wahrscheinlichkeit der Vertauschung bei Systemen 2.Ordnung. Bei Gewichtung der letzten Ordnung eines Ambisonic-Systems mit 0.4 ergibt sich ebenfalls eine erhöhte Wahrscheinlichkeit. Dies lässt sich darauf zurückführen, dass eine geringe Gewichtung der letzten Ordnung annähernd einer Reduzierung der Ordnung bedeutet.

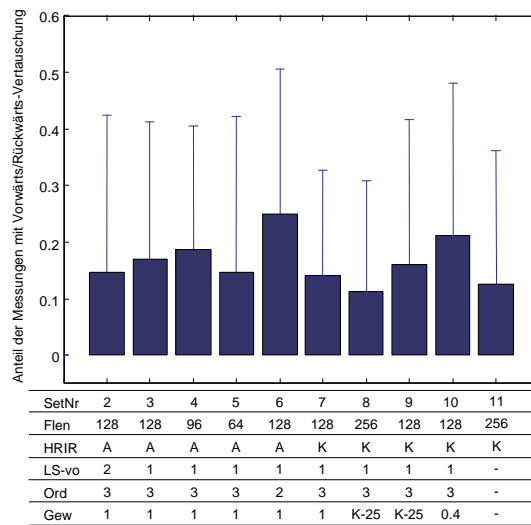


Abb. 6.10: Vorne/Hinten Ambiguität für verschiedene Settings (Mittelwerte und Standardabweichung).

6.8.4 Auswertung der verbalen Beschreibung

Lokalisationsunschärfe:

Abb. 6.11 zeigt die Lokalisationsunschärfe für verschiedene Settings. Deutlich erkennbar ist die Vergrößerung der Lokalisationsunschärfe bei Verwendung eines Kaiserfensters. Dies lässt sich dadurch erklären, dass durch die Fensterung die Hauptkeule der sinc-Funktion verbreitert wird, es kommt zu einer Quellaufweitung (vgl. Kap. 4.3.3). Die geringste Lokalisationsunschärfe entsteht bei direkter Implementation der Kemar HRIRs, da keine Verwischung der virtuellen Schallquelle durch das Ambisonic-System entsteht. Erwartungsgemäß liegt die Lokalisationsunschärfe von Ambisonic-Systemen 3.Ordnung zwischen der direkten Implementation und Systemen 2.Ordnung. Die hohe Lokalisationsunschärfe bei direkter Implementation der AKG-HRIRs ist darauf zurückzuführen, dass zwischen den HRIRs nicht interpoliert wird¹⁶ und diese in 15° Auflösung vorliegen. Es entstehen sprunghafte Bewegungen, was zu einer Verwirrung der Hörwahrnehmung führt.

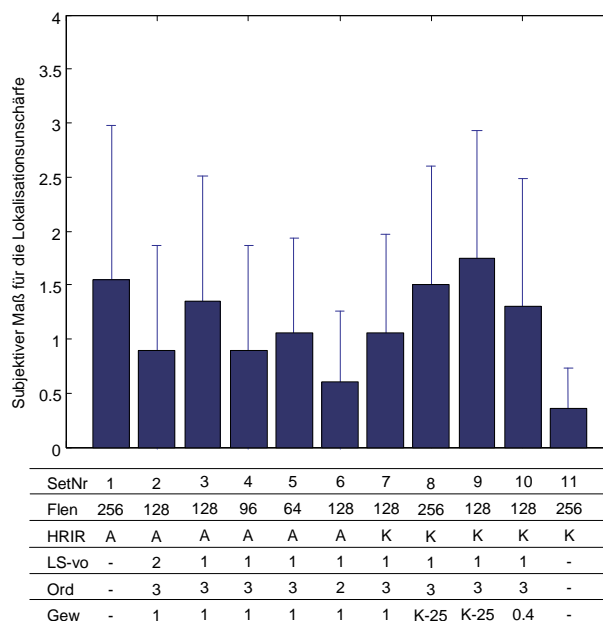


Abb. 6.11: Lokalisationsunschärfe verschiedener Settings (Mittelwert und Standardabweichung)

¹⁶ Die direkte Implementierung erfolgte lediglich zur Klangbewertung, aus diesem Grund wurde auf eine Interpolation verzichtet.

Konsistenz:

Die größte Inkonsistenz (kleinste Konsistenz) weist die direkte Implementation der AKG-HRIRs auf, was sich auf die 15° Auflösung zurückführen lässt (vgl. Lokalisationsunschärfe). Dieses Setting wurde bewusst in die Versuchsreihe mit aufgenommen, um die Aufmerksamkeit der Vpn zu testen. Die Inkonsistenz der Bewegung wurde von allen Vpn richtig erkannt. Aus Abb. 6.12 lässt sich erkennen, dass die Verkürzung der Filterlänge zu einer signifikanten Verschlechterung der Konsistenz der Bewegung führen. Die direkte Implementation der Kemar-HRIRs ergibt eine konsistente Bewegung, d.h. die Winkelauflösung von 5° führt zu keiner signifikanten Verschlechterung. Weiters ist ersichtlich, dass durch Gewichtung der Ordnungen (Fensterung) die Konsistenz der Bewegung erhöht werden kann. Die breitere Hauptkeule führt zu einer Aufweitung der Quelle und somit zu einer als „weich“ wahrgenommenen Bewegung.

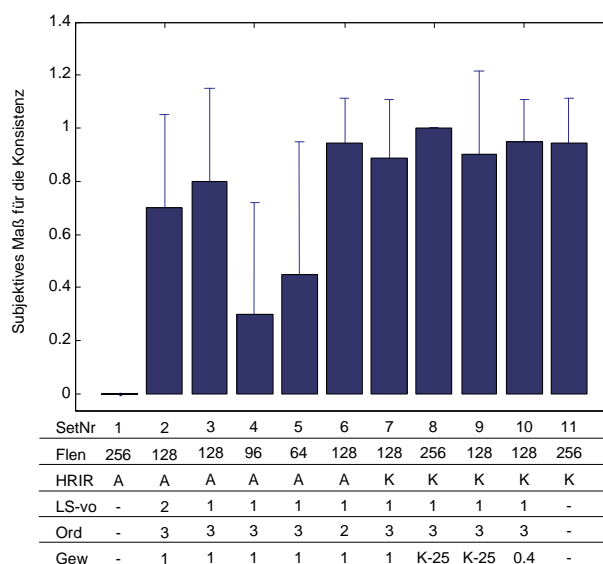


Abb. 6.12: Konsistenz verschiedener Settings
(Mittelwert und Standardabweichung)

Externität:

Aus Abb. 6.13 ist gut ersichtlich dass bei geringer werdender Auslateralisierung die virtuelle Schallquelle in den Kopf wandert. Es entsteht eine elliptische Bahn der Bewegung. Bei Gewichtung der Ordnungen (Fensterung) wird der Unterschied der Wahrnehmung der Externität zwischen vorne und seitlich geringer. Die bewegte Quelle beschreibt eine kreisförmige Bahn. Weiters ist erkennbar,

das bei der direkten Implementation der Kemar HRIRs die größte Externität erzielt werden kann. Das Ambisonic-System verschlechtert die Außerkopflokalisation der virtuellen Quelle.

Aus Abb. 6.14 lässt sich gut die Personenabhängigkeit der verschiedenen Settings bei Bewertung der Externität feststellen. Setting 9 und 10 liefern in Abb. 6.13 annähernd gleiche Werte. Es ist jedoch gut erkennbar, dass Setting 9 wesentlich personenunabhängiger wie Setting 10 ist. Weiters ist ersichtlich dass Vpn 9 große Probleme bei der Außerkopflokalisation unabhängig vom Setting hatte.

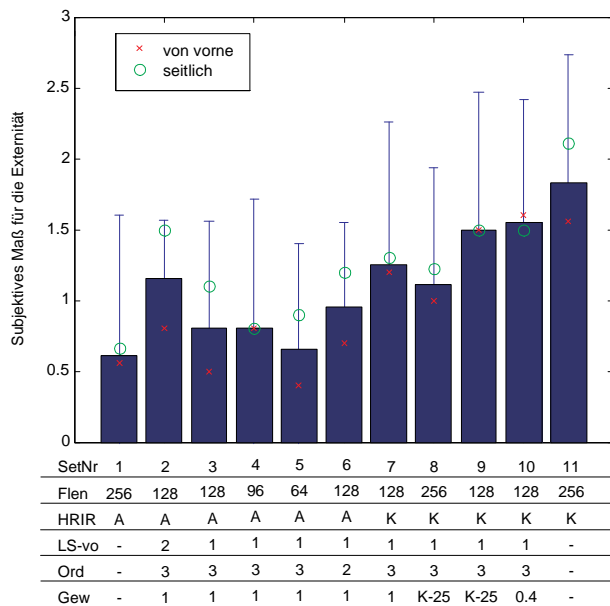


Abb. 6.13: Wahrgenommene Externität bei verschiedenen Settings (Mittelwert, Standardabweichung, getrennte Darstellung der seitlichen und frontalen Externität)

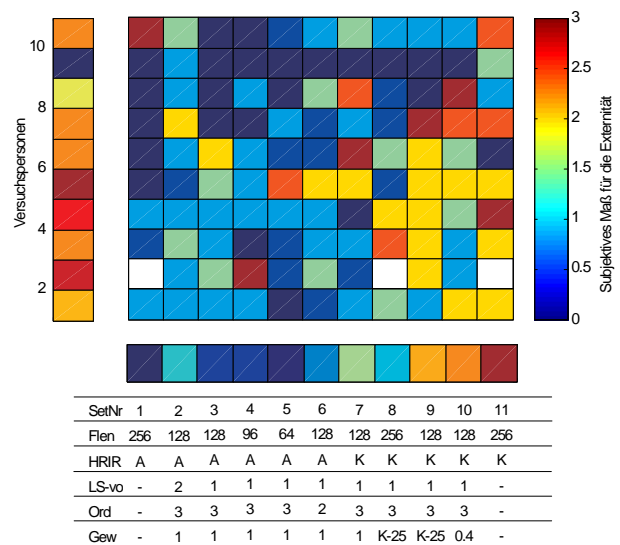


Abb. 6.14: Wahrgenommene Externität für verschiedene Settings bei verschiedenen Vpn

Klangfarbe:

Wichtig bei der Analyse von Abb. 6.15 ist die Betrachtung der Klangmerkmale (Bässe, Mitten, Höhen) im Kontext zueinander. Es ist gut ersichtlich, dass die AKG-HRIRs wesentlich mehr Bässe aufweisen als die KEMAR-HRIRs. Bei einer Kürzung der Filterlänge ergibt sich eine Dämpfung des Bassbereiches. Die AKG-HRIRs klingen grundsätzlich wesentlich realistischer als die Kemar-HRIRs.

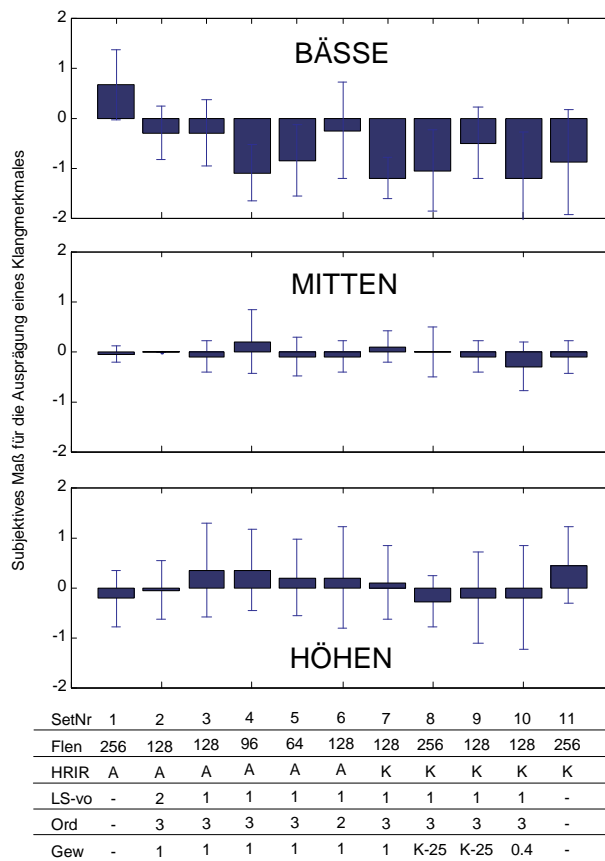


Abb. 6.15: Klangmerkmale (Bässe, Mitten, Höhen) für verschiedene Settings (Mittelwert und Standardabweichung)

7 Implementation des Ambisonic-Algorithmus

Eines der Ziele dieser Diplomarbeit war die Implementierung eines binauralen Systems in ein Echtzeitssystem. Die Implementierung sollte als Vorentwicklung eines kommerziellen Gerätes in Zusammenarbeit mit der Firma AKG-Acoustics dienen. Aufgrund der Anforderungen an das Gesamtsystem hat sich AKG dazu entschieden, einen DSP¹⁷ der 56300-Familie von Motorola zu verwenden. Es handelt sich um den DSP56366-120, einen 24-Bit-DSP der auf Audioanwendungen optimiert wurde. Dieser DSP wurde uns zusammen mit einem *Evaluation Module*¹⁸ DSP56367EVM zur Verfügung gestellt.

7.1 Motorola DSP56366

Der Motorola DSP56366 baut auf einem Kern der DSP56300-Familie auf. Dieser Teil des DSPs stellt einen leistungsfähigen 24-bit-Fixkomma-Kern dar, der es erlaubt, eine Instruktion innerhalb eines Taktzyklus zu verarbeiten. Mit einer Taktrate von bis zu 120MHz und geringem Leistungsverbrauch eignet sich der DSP56366 für die meisten Anwendungen im Bereich der Telekommunikation und Multimedia inklusive High-End-Audio und Videokonferenzsysteme.

Motorolas DSP56366 beinhaltet weiters diverse Peripheriebausteine, wie serielle Schnittstellen, parallele Ports, *Timers*, verschiedene Speicherkonfigurationen und *Debugging-Interfaces* [31], [32].

¹⁷ *Digital Signal Processor*

¹⁸ *Alle aus dem Englischen übernommenen Begriffe werden kursiv formatiert*

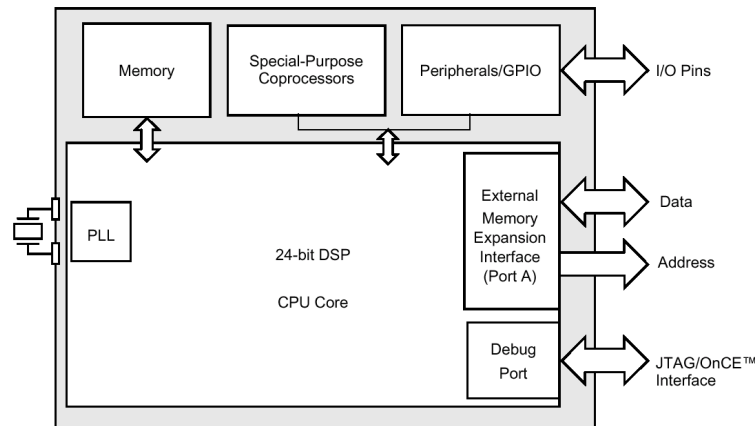


Abb. 7.1: Aufbau der Familie DSP56300

7.1.1 Der Prozessorkern

Hier sollen nur einige wichtige Eigenschaften des Kerns des DSP vorgestellt werden, weitere Detailbeschreibungen finden sich unter [31] und [32].

- ◆ Rechenleistung 1Million Instruktionen pro Sekunde (MIPS) pro MHz-Taktrate, bis 120 MHz taktbar
- ◆ *Data Arithmetic Logic Unit (ALU)*: zwei 56-Bit Akkumulatoren, Multiplikation und Addition in einem Taktzyklus
- ◆ *Address Generation Unit (AGU)*: eigene Register zur Adressierung des Speichers, Unterstützung für *modulo-buffering* und *reverse-carry*-Adressierung zur Implementierung einer FFT.
- ◆ *Program Control Unit (PCU)*: mehrere Adressen- und Datenbusse für paralleles Laden und Ausführen der Befehle
- ◆ *On-chip Cache Controller*
- ◆ Schnittstelle zu einem externen Speicher (Port A)
- ◆ *Phase Locked Loop (PLL)* für die Erzeugung des Systemtaktes
- ◆ *Hardware Debugging (JTAG, OnCE)*
- ◆ *Direct Memory Access (DMA) Controller* mit 6 unabhängigen Kanälen
- ◆ statischer Aufbau – Möglichkeit der Anpassung des Leistungsverbrauchs an die Anforderungen.

7.1.2 Die Speicherkonfiguration

Der Motorola DSP56366 wurde in erweiterter Harvard-Architektur konzipiert. In dieser Architektur werden nicht nur der Programm- und Datenspeicher getrennt verwaltet, es wird sogar der Datenspeicher selbst in 2 Blöcke getrennt. Dieser Aufbau ermöglicht einen gleichzeitigen Zugriff auf die Instruktion aus dem Programmspeicher und das Laden von 2 verschiedenen Datenwörtern.

Der Kern des Motorola DSP56366 kann sowohl auf einen internen als auch auf externe Speicher zugreifen, wobei der interne Speicher zum Teil aus ROM¹⁹ und aus RAM²⁰ besteht. Da bei unserer Implementierung nur auf RAM zugegriffen werden kann, wird im Folgenden nur die Konfiguration des RAM beschrieben:

- ◆ 7Kx24 Bit Y-Data RAM
- ◆ 13Kx24 Bit X-Data RAM
- ◆ 3Kx24 Bit Programmspeicher. 1K des Programmspeichers kann als *instruction cache*²¹ oder als *patch*²² vom Programmspeicher benutzt werden.
- ◆ 2Kx24 Bit von Y-Data RAM und 5Kx24 Bit von X-Data RAM können als Programmspeicher umgeschaltet werden um insgesamt 10Kx24 Bit Programmspeicher zu erhalten

Der Speicher des DSP wird in der Standardkonfiguration benutzt²³ – Speicher mit folgender Größe und Lage im Adressenbereich steht nun zur Verfügung:

	<i>Prog. RAM</i>	<i>X-Data RAM</i>	<i>Y-Data RAM</i>
Größe	3K	13K	7K
Lage	\$0000 ²⁴ bis \$0BFF	\$0000 bis \$33FF	\$0000 bis \$1BFF

7.1.3 Peripheriebausteine

Folgende Peripheriebausteine sind direkt im DSP enthalten:

¹⁹ Read Only Memory

²⁰ Random Access Memory

²¹ Schneller Zwischenspeicher für häufig benutzte Befehle.

²² Programm zur nachträglichen Verbesserung von bereits implementierten Programmteilen.

²³ Im OMR werden die Bits CE, MS und SC gelöscht, der Zustand der Bits MS ist unwichtig. Siehe [31], Table 3-1

²⁴ Hexadezimale Schreibweise. Alle Zahlen mit vorangestelltem '\$'-Zeichen sind als Hexadezimalzahlen zu verstehen

- ◆ *Serial Audio Interface* (ESAI): bis zu 4 Empfangs- und 6 Sendeeinheiten, sowohl im Master- oder Slavebetrieb. I²C²⁵, Sony und AC97²⁶ werden von der Hardware unterstützt.
- ◆ *Serial Audio Interface 1* (ESAI_1): zweites Audio-Interface mit gleichem Aufbau wie EASI
- ◆ *Serial Host Interface* (SHI): SPI²⁷ und I²C Protokolle in Hardware implementiert, Multi-Master-fähig, inkl. FIFO²⁸ mit der Tiefe von 10 Worten.
- ◆ *Host Data Interface* (HDI08): byte-breite Schnittstelle für Multiprozessor-Anbindungen mit DMA²⁹-Unterstützung
- ◆ *Digital Audio Transmitter* (DAX): serielle Sendeeinheit für Unterstützung von S/P-DIF³⁰ und AES/EBU³¹

Weiters besteht die Möglichkeit, alle unbenutzten Ein- und Ausgänge als frei programmierbare Pins einzusetzen.

25 *Langsamer serieller Bus für den Austausch von Steuerungsdaten zwischen mehreren Einheiten [22]*

26 *Audio Codec '97 – Protokoll für die Verbindung von Audio-Bauteilen [1]*

27 *Serial Port Interface, schnelle serielle synchrone Schnittstelle, proprietär*

28 *First In First Out, Speicher für sichere Datenübertragung*

29 *Direct Memory Access*

30 *Sony/Philips Digital Interface Format, digitales Protokoll für Übertragung von Audiodaten im Consumer-Bereich, genormt als IEC 60958-3, [39]*

31 *Audio Engineering Society / European Broadcasting Union, digitales Protokoll für Übertragung von Audiodaten im professionellen Bereich, normiert als ANSI S4.40-1992 oder IEC 60958-4 [39]*

7.1.4 Blockschaltbild

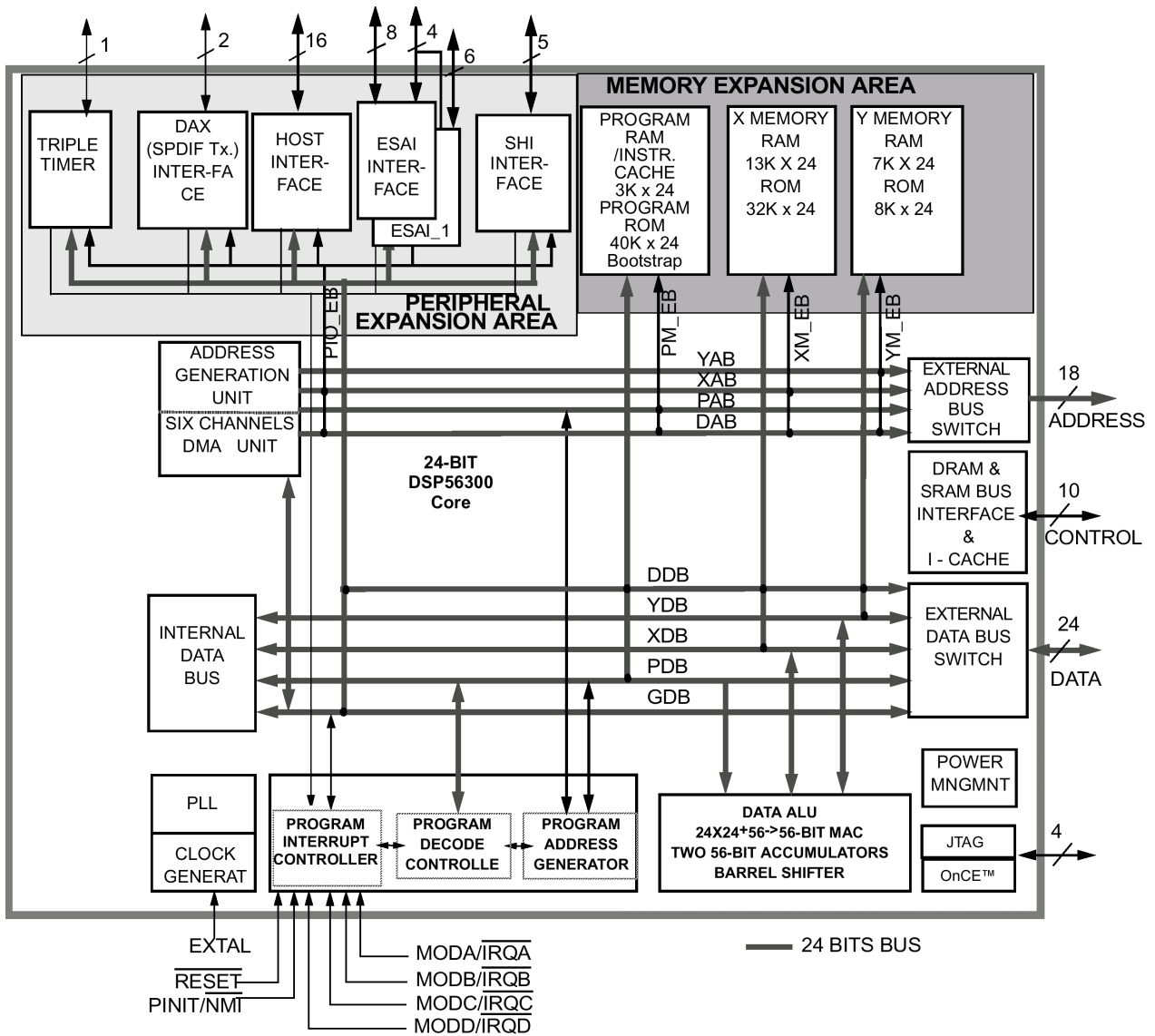


Abb. 7.2: Blockschaltbild des Motorola DSP56366

Weitere Details über den Aufbau und die Funktionsweise des Motorola DSP56366 können [31] entnommen werden.

7.2 Evaluation Module DSP56367EVM

Für die Implementierung des Algorithmus wurde uns von der Firma AKG-Acoustics das *Evaluation Module* DSP56367EVM mit dem DSP XCB56366P zur Verfügung gestellt.

Das *Evaluation Module* (EVM) bietet eine universelle Hardware-Plattform zur Entwicklung von Prototypen und der Erforschung der Umgebung des DSP. Folgende Komponenten befinden sich auf diesem EVM:

- DSP56366, taktbar bis 120MHz
- 256Kx24 Bits externer SRAM³²
- 128Kx8 Bits externer Flash-ROM³³
- 20-bit Stereo ADC³⁴
- Zwei DACs³⁵ mit je 6 Kanälen
- optische und elektrische Ein- und Ausgänge für digitale Audiodaten im S/P-DIF
- 8-Bit *microcontroller*³⁶ mit einem 4-zeiligen LCD³⁷ und 4 Tasten.

Der funktionelle Aufbau des EVM ist in 3.1 dargestellt.

³² *Static Random Access Memory*

³³ *Reprogrammierbarer ROM*

³⁴ *Analog Digital Converter, notwendig, um analoge Audiosignale mit dem DSP zu verarbeiten.*

³⁵ *Digital Analog Converter, setzt die digitalen Audiodaten in analoge Audiosignale um.*

³⁶ *hier: eigenständiger Prozessor für diverse Initialisierungen und Steuerungsaufgaben*

³⁷ *Liquid Crystal Display, hier: alphanumerische Anzeige für etwaige Kommunikation mit dem Benutzer.*

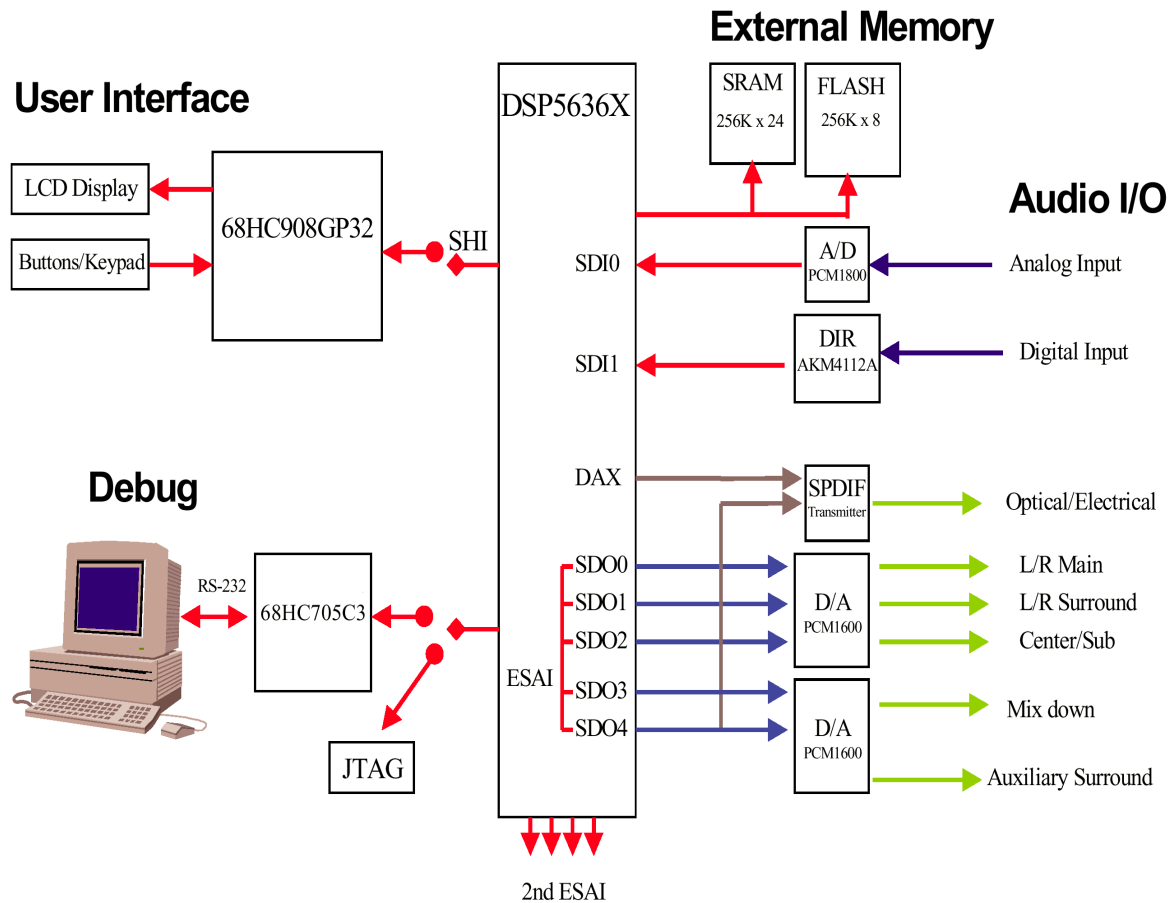


Abb. 7.3: Blockschaltbild von EVM DSP56367EVM

7.3 Die Benutzerschnittstelle

7.3.1 Hardware

Vor der eigentlichen Implementierung eines Algorithmus sollte auf der vorhandenen Hardware-Plattform bereits eine Benutzerschnittstelle (user interface³⁸) zur Verfügung stehen. Zusätzlich zu ihrer endgültigen Bestimmung, der Kommunikation mit dem Benutzer, muss sie auch als Ein- und Ausgabemedium für eventuelle Debugging-Aufgaben während der Implementierung dienen.

Die am EVM vorhandene Anzeige und Tasten sind nur zur Initialisierung der Audio-Kanäle und nicht zur freien Programmierung verwendbar. Die Anzeige und Tasten sind direkt an den 8-Bit-

³⁸ Das Wort Benutzerschnittstelle wird in der Informatikbranche selten benutzt – es wird stattdessen die englische Bezeichnung *User-Interface* verwendet

Mikrokontroller³⁹ angebunden, wobei das Protokoll zur Verbindung des Mikrokontrollers mit dem DSP seitens Motorola nicht freigegeben wurde. Nach eingehender Korrespondenz mit dem Hersteller wurde uns der Quell-Code des Mikrokontrollers zur Verfügung gestellt – dessen Analyse bzw. eine Eigenentwicklung hat sich jedoch als zu umfangreich herausgestellt.

Somit wurde die Entwicklung eines eigenen User-Interfaces notwendig. Folgende Kriterien mussten dabei erfüllt werden:

- ◆ Möglichkeit zur Eingabe und Anzeige der momentanen Position der virtuellen Schallquelle
- ◆ Rücksetzen der Position auf 0°
- ◆ Ein- und Ausschalten des implementierten Algorithmus
- ◆ Diverse Statusanzeigen für eventuelle Debugging-Aufgaben
- ◆ möglichst einfache Anbindung an den DSP mit möglichst wenigen zusätzlichen Bauteilen

Die unbenutzten Pins des DSPs können frei programmiert und als GPIO⁴⁰ genutzt werden. Für die Anbindung der Benutzerschnittstelle kann also jede Leitung, die nicht vom EVM benutzt wird, verwendet werden. Die Wahl fiel auf die am Steckplatz JP11⁴¹ ausgeführten Leitungen, die am DSP als HDI08⁴² vorliegen. Da die Schnittstelle HDI08 in unserer Anwendung frei bleibt, können diese Pins wahlweise als Ein- oder Ausgänge genutzt werden – sie werden in diesem Fall als Port B bezeichnet. 2 Leitungen des Ports B werden vom EVM benutzt, so dass 14 Pins für die Anbindung des User-Interfaces zur Verfügung stehen, die folgendermaßen eingeteilt werden:

Bezeichnung		Richtung	Verwendung
als HDI08	als Port B		
H0 bis H4	PB0 bis PB4	Output ⁴³	Schallquellenposition / Pegelanzeige
H5, H6, H7, HA0	PB5 bis PB8	Output	diverse Statusanzeigen
HA1, HA2, HDS, HACK, HRW ⁴⁴	PB9 bis PB11	Input ⁴⁵	Taster für diverse Steuerungsaufgaben

³⁹ U13, MC68HC908, siehe Schaltplan [33], sheet 3

⁴⁰ General Purpose Input / Output, siehe [31], section 1.4.2

⁴¹ 20-poliger Steckplatz, siehe Schaltplan [33], sheet 1

⁴² Host Interface, siehe [31], section 6 ff.

⁴³ aus dem Engl.: Ausgang. Der Zustand dieses Pins kann vom DSP direkt verändert werden.

⁴⁴ Signalbezeichnungen des HDI08, werden hier nicht verwendet, siehe [33]

⁴⁵ aus dem Engl.: Eingang. Der Zustand dieses Pins kann vom DSP abgefragt werden.

Weiters ist am EVM eine LED⁴⁶ vorhanden, deren Zustand vom DSP direkt verändert werden kann. Es handelt sich um den Ausgang TIO0⁴⁷, der vom Timer 0 des DSPs gesteuert wird. Der Zustand dieses Pins zeigt in der vorliegenden Programmierung die DSP-Belastung an und kann mittels eines Oszilloskops zur Anzeige gebracht werden.

Alle weiteren technischen Details befinden sich im Anhang D.

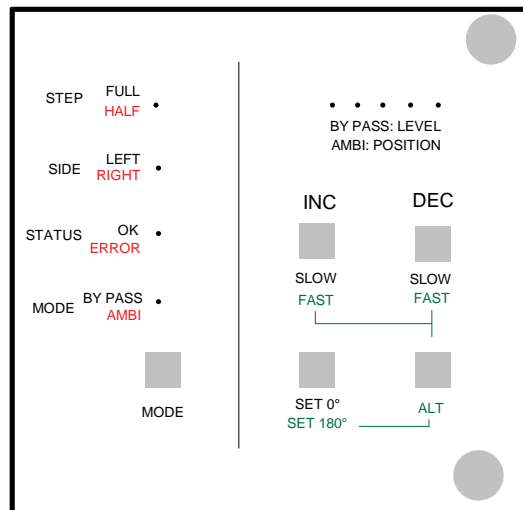


Abb. 7.4: Frontplattenschema des User-Interface, nicht maßstabsgetreu

7.3.2 Bedienung

Die folgenden Bezeichnungen beziehen sich auf die in Abb. 7.4 dargestellte Frontplatte.

Die „MODE“-Taste schaltet zwischen dem „BY PASS“ und Ambisonic-Modus um. Im „BY PASS“-Modus wird das Audio-Signal direkt durchgeschaltet, im Ambisonic-Modus erfolgt stattdessen die Berechnung des Ambisonic-Algorithmus. Die „MODE“-LED zeigt den gewählten Modus an (ausgeschaltet im „BY PASS“-Modus).

Die „ALT“-Taste bietet eine Umschaltung der Funktionen anderer Tasten auf eine zweite Ebene an, indem bei festgehaltener „ALT“-Taste eine weitere Taste betätigt wird. Die Funktionen in der zweiten Ebene sind auf der Frontplatte mit einer anderen Farbe formatiert.

⁴⁶ Light Emitted Diode, Leuchte auf Halbleiter-Basis

⁴⁷ Timer Input Output, siehe Schaltplan von EVM, [33], sheet 1.

Die Taste „SET 0°“ setzt die Position der Schallquelle auf 0° - die Kombination „ALT“ und „SET 0°“ auf 180°.

Mit den Tasten „INC“ bzw. „DEC“ kann die Position der Schallquelle um eine Einheit gegen bzw. mit dem Uhrzeigersinn verändert werden. Beim gleichzeitigen Halten der Taste „ALT“ wird die Position um 5 Einheiten verändert. Die virtuelle Schallquelle kann somit in zwei verschiedenen Geschwindigkeiten um den Kopf bewegt werden. Eine Einheit bedeutet die Bewegung der Schallquelle um 1.4025° und ergibt sich aus der Länge der implementierten Cosinus-Tabelle⁴⁸ (siehe Gl. 7.2).

Die Tasten „SET“, „INC“ und „DEC“ werden im „BY PASS“-Modus nicht verwendet.

Die LED („STATUS: OK/ERROR“) leuchtet im Fall einer Übersteuerung des Audiosignals am Eingang oder beim Überlauf während der Berechnung auf und bleibt in diesem Zustand. Sie kann mit jeder Taste (auch „ALT“) ausgeschaltet werden.

Der horizontale Balken mit 5 LEDs („AMBI: POSITION“) wird verwendet, um den Winkel der Schallquellenposition grafisch anzuzeigen. Eine weitere LED („SIDE: LEFT/RIGHT“) leuchtet auf, wenn die Schallquelle sich in der rechten Hälfte der Horizontalebene befindet, und erlischt, wenn die Schallquelle sich in der linken Hälfte befindet. Aufgrund der groben Anzeige des horizontalen Balkens zeigt eine zusätzliche LED („STEP: HALF/FULL“) den Halbschritt der Bewegung der Quelle an. Die Zuordnung der Schallquellenposition und der Balkenanzeige zeigt die Tabelle 7.1, wobei nur die wichtigsten Positionen angegeben werden sollen.

⁴⁸ in momentanen Implementierung ist die Cosinus-Tabelle mit 256 Punkten gespeichert

<i>Position</i> <i>in Grad</i>	<i>LEDs</i>		
	<i>STEP</i>	<i>SIDE</i>	<i>Balkenanzeige</i>
0°, (Vorne-Mitte)	■	■	■ ■ ■ ■ ■
9°	■	■	■ ■ ■ ■ ■
18°	■	■	■ ■ ■ ■ ■
27°	■	■	■ ■ ■ ■ ■
36°	■	■	■ ■ ■ ■ ■
54°	■	■	■ ■ ■ ■ ■
72°	■	■	■ ■ ■ ■ ■
90° (Links)	■	■	■ ■ ■ ■ ■
108°	■	■	■ ■ ■ ■ ■
126°	■	■	■ ■ ■ ■ ■
144°	■	■	■ ■ ■ ■ ■
162°	■	■	■ ■ ■ ■ ■
180°, (Hinten-Mitte)	■	■	■ ■ ■ ■ ■
198°	■	■	■ ■ ■ ■ ■
216°	■	■	■ ■ ■ ■ ■
270° (Rechts)	■	■	■ ■ ■ ■ ■
342° (-18°)	■	■	■ ■ ■ ■ ■
351° (-9°)	■	■	■ ■ ■ ■ ■

Tabelle 7.1: Zuordnung: Schallquellenposition - LED-Anzeige

7.4 Audio-Routing

Für die Anbindung der Audio-Signale an den DSP wurden die im EVM integrierten Analog/Digital⁴⁹ und Digital/Analog-Konverter⁵⁰ verwendet. Beim ADC handelt es sich um einen 20-Bit Stereo Konverter, der einen Sigma-Delta-Modulator mit 64-fachen Oversampling beinhaltet und Taktraten bis zur 48kHz bietet⁵¹. Der verwendete DAC ist ein 24-Bit, 6-Kanal, Konverter mit Taktraten bis zu 96kHz⁵².

⁴⁹ kurz: ADC

⁵⁰ kurz: DAC

⁵¹ PCM1800 [10], am Schaltplan mit U19 bezeichnet, siehe [33], sheet 6

⁵² PCM1600 [11], am Schaltplan mit U21 bezeichnet, siehe [33], sheet 7

Die Initialisierung der beiden Konverter erfolgt beim Reset und wird vom EVM durchgeführt. Dieser Vorgang ist für den DSP völlig transparent – in der Software muss lediglich die Initialisierung der Audio-Schnittstellen des DSP berücksichtigt werden.

Da in der vorliegenden Software-Version nur eine Schallquelle im Ambisonic-Algorithmus kodiert wurde, ist nur ein Mono-Eingang notwendig. Der Audio-Eingang ist auf dem EVM am Steckplatz J8 zu finden, es handelt sich um die Buchse Lin.

Für die Ausgabe der binauralen Audiosignale werden die im Steckplatz J7 eingebauten Buchsen R und L verwendet.

7.5 Software

7.5.1 Allgemeiner Aufbau

Die Software wurde in 3 funktionelle Gruppen zusammengefasst:

<i>Funktion</i>	<i>Kennzeichnung</i>
Audio-Treiber	alle Dateien audio_-Präfix
Treiber für „user interface“	Dateien mit uim_ im Namen
Hauptprogramm, Algorithmen und diverse Tabellen und	Dateien ohne jeglichen Präfix

Die Datei `main.c` ist die übergeordnete Datei, die auch vom Compiler⁵³ aufgerufen werden sollte. Der Aufruf des Compilers und Linkers⁵⁴ erfolgt aus der Datei `make.bat`, sie enthält alle notwendigen Anweisungen um eine ausführbare Datei erstellen zu können, sofern die Entwicklungsumgebung von Motorola ordnungsgemäß installiert wurde.

Weiters befindet sich in der Datei `regusage.asm` die Belegung der DSP-Register, sie sollte bei etwaigen Änderungen der Software berücksichtigt werden.

Allen in der Software verwendeten Labels⁵⁵ wurde ein '_' nachgestellt, um eine deutliche Unterscheidung zu den Variablennamen zu ermöglichen.

⁵³ übersetzt die Quell-Codes in Module, ohne Berücksichtigung der endgültigen Lage der übersetzten Bereiche im Speicher.

⁵⁴ Fügt einzelne, vom Compiler übersetzte, Module zu einer ausführbaren Datei zusammen unter Berücksichtigung der Lage der Module im Speicher

⁵⁵ Namen der Funktionen, Einsprungadressen im Programm aber **keine** Variablennamen

7.5.2 Audio-Treiber

Die gesamte Verarbeitung der Audio-Signale geschieht in folgenden Dateien:

<code>audio_main:</code>	Enthält alle Definitionen und Deklarationen, sowie die Initialisierungsfunktionen
<code>audio_vectors:</code>	Enthält Einträge in die Interruptvektortabelle
<code>audio_isr:</code>	Enthält die Interrupt Service Routine für die Erzeugung der Datenströme

Die Routine `AUDIO_INIT_` initialisiert das `ESAI`, über dieses Interface wird die Kommunikation zu den ADC und DAC abgewickelt. Weiters werden die notwendigen Interrupts freigegeben.

Die Interrupt-Routinen `TX_ISR_` bzw. `RX_ISR_` werden vom DSP aufgerufen sobald ein Datenwort gesendet bzw. empfangen wird. Obwohl das Hauptprogramm nur die Daten des linken Kanals benutzt wurden die Routinen so konzipiert, dass 2 Kanäle (1 Stereo-Kanal) empfangen bzw. gesendet werden können. Die Audiodaten werden in Shadow-Registern verwaltet um einen asynchronen Zugriff seitens des Hauptprogramms zu ermöglichen. Die zu sendenden Daten werden in den Registern `DAC_L` und `DAC_R` (`DAC_Ls` und `DAC_Rs` bei Shadow-Register), die empfangenen Daten in den Register `ADC_L` und `ADC_R` (Shadow-Register: `ADC_Ls` und `ADC_Rs`), gespeichert. Nachdem die Audiodaten für den linken und den rechten Kanal empfangen und gespeichert wurden, wird die Bereitschaft der Daten dem Hauptprogramm mit einem Flag gemeldet. Das Hauptprogramm wartet bis dieses Flag gesetzt wird und fängt unverzüglich die neuen Audiodaten zu verarbeiten an. Da das Hauptprogramm so konzipiert wurde, dass die empfangenen Daten sofort nach dem Empfang verarbeitet werden, wird auf der Empfangsseite auf die Shadow-Register verzichtet – die Variablennamen `ADC_L` und `ADC_Ls` (für den rechten Kanal: `ADC_R` und `ADC_Rs`) bezeichnen somit die selben Variablen. Die Abb. 7.5 zeigt den zeitlichen Ablauf der Kommunikation der ADC, DAC mit dem Hauptprogramm.

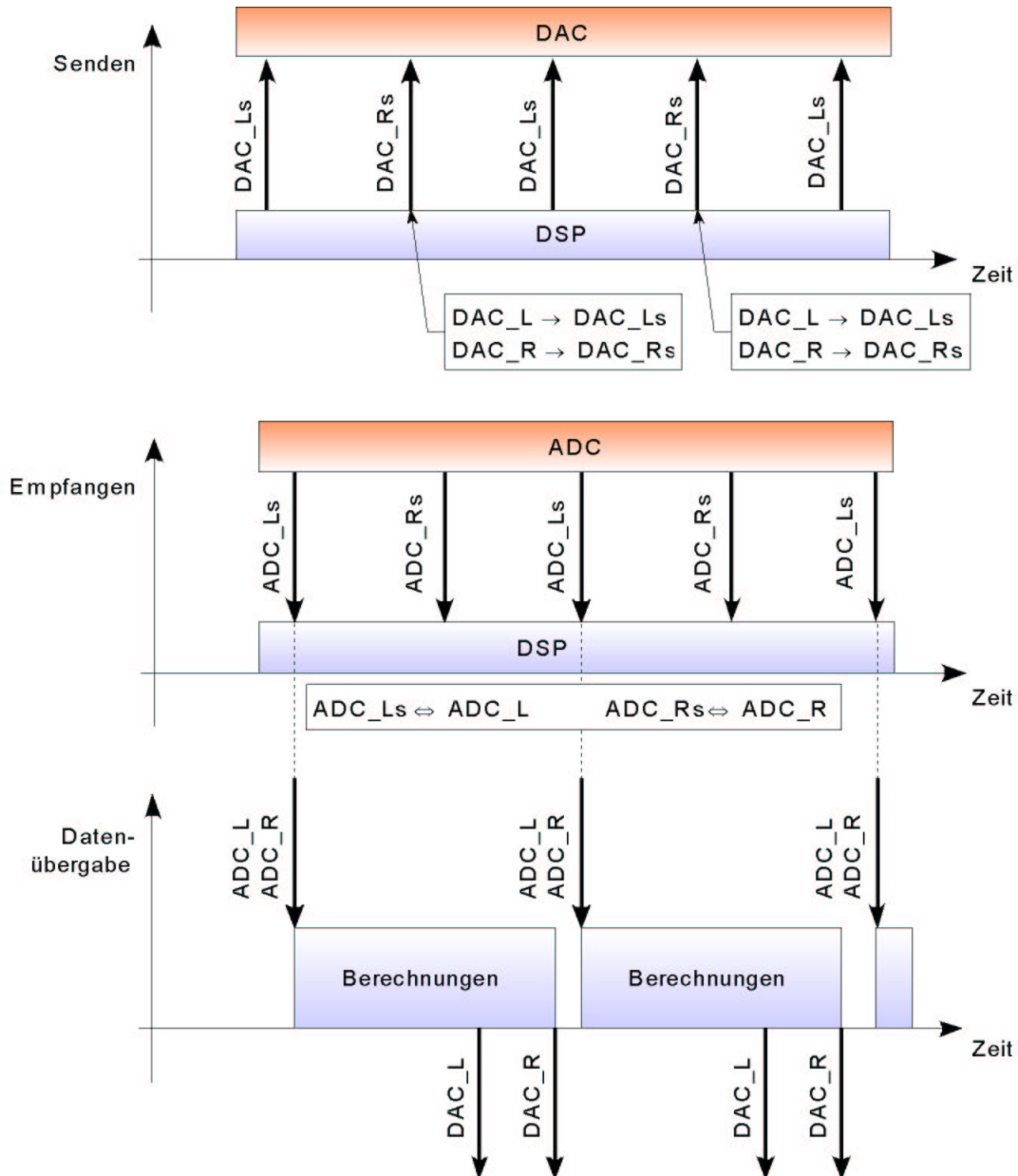


Abb. 7.5: Zeitlicher Ablauf der Datenübergabe

7.5.3 User Interface

Es wurden 3 Dateien für das User-Interface erzeugt:

- `uim_main`: Enthält alle Definitionen und Deklarationen, sowie die Initialisierungsfunktionen
- `uim_vectors`: Enthält die Einträge in die Interruptvektortabelle
- `uim_isr`: Enthält die Interrupt-Service-Routine zur Tastenabfrage

Der Treiber wird mit dem Aufruf von `USERINTERFACE_INIT_` initialisiert. Nach der Freigabe aller Interrupts wird die Funktion `TIMER0_ISR_` periodisch aufgerufen. Dort findet die Umschaltung zwischen Ambisonic- und „BY PASS“-Modus und die Aktualisierung des Azimutwinkels statt. Dieser steht im Register `r5` anderen Programmteilen zur Verfügung.

Die Aktualisierung der Anzeige findet in den Programmteilen statt, die über die jeweiligen Informationen zur Anzeige verfügen. Die Treiber des User-Interface enthalten lediglich die Definitionen der Zuordnung der einzelnen LEDs zu den verwendeten Ports.

7.5.4 Ambisonic-Algorithmus

Der Ambisonic-Algorithmus besteht, wie in Kap. 4.2 beschrieben, aus 3 Abschnitten: Kodierung der Schallquelle in die Ambisonic-Domäne, Filterung der Ambisonic-Kanäle mit reduzierten Filtern und anschließender Dekodierung mit binauralen Signalen als Ergebnis (siehe Gleichung 4.43).

Die Implementation des Ambisonic-Algorithmus liegt in Datei `ambisonic.asm` vor und besteht aus zwei Funktionen: `AMBISONIC_INIT_` und `AMBISONIC_`. `AMBISONIC_INIT_` initialisiert am Anfang des Programms die verwendeten Variablen und löscht alle Ringbuffer für Ambisonic-Daten. Die Funktion `AMBISONIC_` berechnet das binaurale Signal aus dem Audiosignal unter Berücksichtigung des Azimutwinkels, sie wird bei jedem ankommenden Datenwort vom Hauptprogramm periodisch aufgerufen.

Alle im Algorithmus verwendeten Parameter wurden am Anfang dieser Datei vordefiniert, um diverse Veränderungen und Anpassungen des Algorithmus leicht vornehmen zu können. Sowohl diese Definitionen als auch alle Deklarationen der Variablen und Buffer werden hier nicht aufgelistet, sie sind dem Quell-Code selbst zu entnehmen.

Bei der Beschreibung des Quell-Codes wurde auf die zeitliche Abarbeitung der Befehle seitens des DSPs keine Rücksicht genommen – der Quell-Code selbst enthält mehrere NOPs⁵⁶, die wegen der Pipeline-Architektur⁵⁷ des Prozessors notwendig sind. Weiters wurden zur Erhöhung der Übersichtlichkeit die Zeilennummern der Datei `ambisonic.asm` angegeben.

⁵⁶ *No Operation, Prozessorbefehl, es wird keine Operation ausgeführt.*

⁵⁷ *Für die Verarbeitung eines Befehls am DSP muss dieser mehrere Stationen nacheinander passieren. Diese Station arbeiten gemäß einer Pipeline-Struktur – erst nach der letzten Station liegt ein Ergebnis der Operation vor. Es sind deswegen immer mehrere Befehle gleichzeitig in der Pipeline, die von Station zu Station weitergegeben werden. Wird an einer Station das Ergebnis des vorangegangenen, aber noch nicht fertigen Befehls benötigt, muss diese Station eine Pause einlegen. Diese Pause wird in Form eines NOPs bewerkstelligt.*

Berechnungsvorgang - Allgemein:

Vor der eigentlichen Verarbeitung wird das für das Audiosignal repräsentative Datenwort in das Register x0 geladen. Weiters wird der Zeiger auf den freien Platz aller Ringbuffer in das Register r6 geladen:

```
203:    move    x:ADC_L,x0
204:    move    x:AMBBufferPTR,r6
```

Die Werte dieses Registers stehen anschließend während der gesamten Berechnung zur Verfügung.

Berechnungsvorgang – Kodierung:

Es wird das Audiosignal x_n in die Ambisonic-Domäne kodiert, wobei die Ambisonic-Kanäle nacheinander berechnet und die Ergebnisse in die dafür vorgesehenen Ringbuffer abgelegt werden. Die Kodierung erfolgt nach folgender Formel (vgl. Kap 4.1.4):

$$\begin{pmatrix} W \\ X1 \\ Y1 \\ X2 \\ Y2 \\ X3 \\ Y3 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \cdot x_n \\ \cos(\varphi) \cdot x_n \\ \sin(\varphi) \cdot x_n \\ \cos(2\varphi) \cdot x_n \\ \sin(2\varphi) \cdot x_n \\ \cos(3\varphi) \cdot x_n \\ \sin(3\varphi) \cdot x_n \end{pmatrix} \quad (7.1)$$

Für den W-Kanal wird der Azimutwinkel nicht benötigt, weil er die richtungsunabhängige Kugelflächenfunktion 0-ter Ordnung darstellt. Die Kodierung des Audiosignals für den W-Kanal geschieht durch eine Multiplikation mit einem Skalar, der als AMBFACTOR_W definiert wurde. Nach der Kodierung wird der Wert im Ringbuffer AMBBufferW abgespeichert und steht dort für die weitere Verarbeitung zur Verfügung:

```
219:    move    #>AMBFACTOR_W,x1
220:    mpy    x0,x1,a
221:    move    #>AMBBufferW,n6
225:    move    a,x:(r6+n6)
```

Die Kodierung weiterer Ambisonic-Kanäle erfordert die Einbeziehung des Azimutwinkels, der über die Cosinus- bzw. Sinus-Funktionen in die Berechnung eingeht. Die Berechnung dieser Funktionen

wird mittels vorgefertigter Look-Up-Table⁵⁸ durchgeführt. Eine über den Winkel diskretisierte Tabelle mit den Werten der Cosinusfunktion im Bereich von 0° bis 360° wurde im Speicher abgelegt, wobei die Länge dieser Tabelle in `COSTABLELEN` definiert wurde. Der Azimutwinkel dient nun als Index für diese Tabelle. Für die Berechnung der Sinusfunktion wird der Azimutwinkel um -90° verschoben und auf die Cosinustabelle zugegriffen. Der Wert der Verschiebung des Azimutwinkels wurde in `SINOFFSET` definiert. Bei der vorliegenden Version wurde eine Tabelle mit 256 Einträgen implementiert, die Auflösung des Azimutwinkels wird somit auf

$$\Delta\varphi = \frac{360^\circ}{256} = 1.40625^\circ \quad (7.2)$$

reduziert.

Für die weiteren Betrachtungen gilt: der Azimutwinkel steht im `r5`, Register `n1` enthält die Basisadresse der Cosinus-Tabelle und das Audiosignal ist weiterhin im Register `x0` enthalten. Nun erfolgt die Berechnung des $\cos(\varphi)$ -Terms und Kodierung des Audiosignals in den X1-Kanal:

```

240:    move    r5,r1
244:    move    y:(r1+n1),x1
245:    mpy     x0,x1,a
246:    move    #>AMBBufferX1,n6
249:    move    a,x:(r6+n6)

```

Da die Struktur zur Kodierung weiterer Kanäle gleich bleibt, wird in weiterer Folge nur die Berechnung der Cosinus- bzw. Sinus-Terme beschrieben.

Für die Berechnung des $\sin(\varphi)$ bei der Kodierung des Y1-Kanals ergibt sich:

```

266:    move    r5,b
267:    add     #SINOFFSET,b
268:    and     #COSTABLELEN-1,b
269:    move    b,r1

```

Die Berechnung des $\cos(2\varphi)$ -Terms für die Kodierung des X2-Kanals wurde folgendermaßen implementiert. Dabei sollte die Modulo-Berechnung des Index beachtet werden:

⁵⁸ Look-Up-Table, Verfahren für die Durchführung komplizierter Berechnungen in Echtzeit trotz geringer Rechenleistung

```

281:    move    r5,b
282:    asl     b
283:    and     #COSTABLELEN-1,b
285:    move    b,r1

```

Die Berechnung des $\sin(2\varphi)$ -Terms ergibt sich aus der Multiplikation des Azimutwinkels mit dem Faktor 2, einer Verschiebung des so entstandenen Index um `SINOFFSET` und anschließendem Auslesen des entsprechenden Wertes aus der Cosinus-Tabelle. Bei der Kodierung der Kanäle 3-er Ordnung X3 und Y3 ist lediglich die Erstellung der Terme $\cos(3\varphi)$ und $\sin(3\varphi)$, insbesondere die Multiplikation des Azimutwinkels mit dem Faktor 3, erwähnenswert:

```

311:    move    r5,b
312:    move    r5,y0
313:    asl     b
314:    add     y0,b
315:    and     #COSTABLELEN-1,b
317:    move    y0,r1

```

Die Berechnung des $\sin(3\varphi)$ -Terms erfolgt bis auf die Verschiebung des Winkels in gleicher Weise.

Nun liegen die Daten in der Ambisonic-Domäne vor – in Ringbuffern gespeichert um mittels FIR-Filter gefiltert zu werden.

Berechnungsvorgang – Filterung:

Die Ambisonic-Signale werden nun mit den in `filterW.asm`, `filterX1.asm` ... `filterY3.asm` vorliegenden Filterkoeffizienten gefiltert. Bei den Filtern handelt es sich um FIR⁵⁹-Filter – die Berechnungen können somit im Zeitbereich mittels diskreter Faltung durchgeführt werden:

$$y(n) = \sum_{m=0}^{m=N} h(m) \cdot x(n-m) \quad (7.3)$$

wobei $h(m)$ die Impulsantwort mit der Länge N , $x(n-m)$ das vorliegende und $y(n)$ das gefilterte Signal darstellen. Die Impulsantwort des gewünschten Systems entspricht in unserem Fall der HRIRs für den jeweiligen Lautsprecher und hat die Länge von $N=128$.

Da ein Teil der anschließenden Dekodierung einer weiteren Filterung entspricht, können die HRIRs und die Dekoder-Matrix zusammengefasst werden. Die endgültigen Filterkoeffizienten können mit

⁵⁹ *Finite Impulse Response Filter, stabiler Filter mit endlicher Impulsantwort und linearem Phasengang*

der MATLAB-Funktion `mn_genfilt_asm56` (siehe Anhang B) generiert werden – man erhält als Ergebnis direkt die notwendigen Assembler-Dateien.

Für die Implementierung des FIR-Filter-Algorithmus werden die Filterkoeffizienten und Ambisonic-Signale in zwei verschiedenen Datenspeichern des DSPs abgelegt. Dadurch läßt sich die Fähigkeit des DSPs, eine Multiplikation und ein gleichzeitiges Nachladen der Koeffizienten, dazu nutzen, einen Schritt des FIR-Filters mittels eines Befehles und somit in einem Taktzyklus auszuführen. Dies führt zu einem äußerst effizienten Algorithmus für ein FIR-Filter:

```
353:      rep      #BUFFERLEN
354:      mac      x0,y0,a y:(r0)+,y0 x:(r4)-,x0
```

Der Befehl `mac` wird dabei öfters wiederholt, die Anzahl der Wiederholungen gibt der Befehl `rep` mit `BUFFERLEN` vor. `BUFFERLEN` entspricht der Länge der Impulsantwort des FIR-Filters und enthält in diesem Fall den Wert 128.

Voraussetzung für die korrekte Ausführung der Filterung ist die richtige Initialisierung aller Register und eine nach der Berechnung notwendige Skalierung und Rundung. All diese Schritte enthält der gesamte Filter-Algorithmus, der für jeden Ambisonic-Kanal wiederholt wird. Hier als Beispiel die Implementation eines FIR-Filters für den W-Kanal:

```
343:      move     r6,x0
344:      move     #>AMBBufferW,a
345:      add      x0,a
347:      move     a,r4
348:      move     #>FILTERTableW,r0
352:      clr      a      y:(r0)+,y0 x:(r4)-,x0
353:      rep      #BUFFERLEN
354:      mac      x0,y0,a y:(r0)+,y0 x:(r4)-,x0
355:      asr      #SCALE_FIR,a,a
356:      rnd      a
365:      move     a,x:AMBW
```

Der Befehl `asr` skaliert das Ergebnis auf einen 24-Bit-Wert. Durch die 128-fache Addition der Produkte kann das Ergebnis um bis zu 7-Bit anwachsen und müsste somit um diesen Faktor skaliert werden. Dieser Wertzuwachs kann allerdings nur im Fall der durchgehenden Addition von Maximalwerten entstehen, der bei einem Filter mit Koeffizienten kleiner 1 nie entstehen kann⁶⁰. Die Skalierung hängt also von den verwendeten Filterkoeffizienten stark ab und kann deswegen kleiner gewählt werden. In unserem Fall enthält `SCALE_FIR` den Wert 2.

⁶⁰ Alle Koeffizienten sind kleiner als 1, auch das Audiosignal kann den Wert 1 nicht überschreiten. Somit ergeben sich für die Produkte der Multiplikationen stets Werte kleiner 1.

Das Ergebnis der Filterung wird anschließend in der Variable `AMBW` abgespeichert. Für jeden Ambisonic-Kanal existieren ähnliche Variablen.

Berechnungsvorgang – Dekodierung:

Ein Teil der Dekodierung – die Berücksichtigung der Dekoder-Matrix – erfolgt während der Filterung der Ambisonic-Signale. Nun werden die Ergebnisse der Filterungen zur Erzeugung der binauralen Audiosignale zusammengefasst (siehe Kap 4.3.2, Gleichung 4.45, Abb. 4.6):

```
519:    move    x:AMBW, a
520:    move    x:AMBX1, x0
521:    add     x0, a
523:    move    a, b
524:    move    x:AMBY1, x0
525:    add     x0, a
526:    sub     x0, b
527:    move    x:AMBX2, x0
528:    add     x0, a
529:    add     x0, b
530:    move    x:AMBY2, x0
531:    add     x0, a
532:    sub     x0, b
533:    move    x:AMBX3, x0
534:    add     x0, a
535:    add     x0, b
536:    move    x:AMBY3, x0
537:    add     x0, a
538:    sub     x0, b
539:    asr     #SCALE_ADD, b, b
540:    asr     #SCALE_ADD, a, a
```

In den beiden Akkumulatoren `a` und `b` liegen nun die Audiosignale für das linke und das rechte Ohr vor. Auch hier erfolgt eine Skalierung der Werte. Nach der vorliegenden 7-fachen Addition kann das Ergebnis maximal um 3-Bit anwachsen – durch die Skalierung wird ein Überlauf verhindert.

Berechnungsvorgang – Ausgabe des binauralen Signals:

Die berechneten Audiodaten können nun dem Audio-Treiber übergeben werden. Als letzter Schritt wird noch der im Register `r6` abgelegte Zeiger aktualisiert und abgespeichert – beim nächsten Aufruf von `AMBISONIC_` werden somit die ältesten Werte in den Ambisonic-Ringpuffern mit den neu berechneten überschrieben

549:	move	a,x:DAC_L
550:	move	b,x:DAC_R
564:	lua	(r6)+,r6
568:	move	r6,x:AMBufferPTR

7.6 Rechenleistung

Die benötigte Rechenleistung stellt einen wesentlichen Parameter zur Beurteilung eines Algorithmus dar. Eine Abschätzung dieser vor der eigentlichen Implementierung ermöglicht eine gezielte Auswahl der Hardware-Plattform. Zu diesem Zweck wurde vor der Durchführung der Implementation eine Abschätzung der benötigten Leistung vorgenommen.

7.6.1 Abschätzung des voraussichtlichen Bedarfes

Für die Implementierung des Ambisonic-Algorithmus werden folgende Teil-Algorithmen benötigt:

- Kodierung einer Quelle in die Ambisonic-Domäne
- Filterung der Ambisonic-Kanäle mit der dazu notwendigen Bufferung der Daten
- Dekodierung der Daten zu einem binauralen Signal

Darüber hinaus wird zusätzliche Rechenleistung für das User-Interface, Audio-Treiber sowie allgemeine Verwaltung der Prozesse benötigt.

Die Kodierung einer Quelle in die Ambisonic-Domäne wird mit ca. 10 Befehlen (Speicherzugriffe, Multiplikationen und Additionen) pro Ambisonic-Kanal veranschlagt. Es werden also in diesem Fall (7 Ambisonic-Kanäle) $70T_C$ ⁶¹ benötigt.

Die Filterung wird mittels FIR-Filter durchgeführt, es ist ein Filter pro Ambisonic-Kanal notwendig. Laut „Benchmark Program“ [32] benötigt ein FIR-Filter $N+10T_C$, wobei N für die Länge der Impulsantworten steht, in unserem Fall 128. Es werden für die Filterung $966T_C$ benötigt.

Bei der Dekodierung der Daten werden die einzelnen Signale addiert bzw. subtrahiert. $10T_C$ pro Audio-Kanal sollte eine realistische Schätzung abgeben – es werden insgesamt $20T_C$ benötigt.

Der Rechenleistungsbedarf für die Verwaltung der Berechnungen kann nur sehr grob geschätzt werden – er hängt vom eingesetzten DSP-Typ, Programmierstil und Entwicklungsaufwand sehr stark ab. Unter Berücksichtigung der Häufigkeit der Interruptaufrufe, Anzahl der Unterrouinen und verwendeten Variablen sowie des Befehlssatzes des DPSs wird hier der Rechenleistungsbedarf für

⁶¹ Taktzyklen

die Verwaltung auf $100T_C$ zwischen zwei aufeinander folgenden Audiodaten geschätzt. Die Tabelle 7.2 fasst die geschätzten Werte nochmal zusammen und gibt Auskunft über den gesamten Rechenleistungsbedarf.

<i>Algorithmus</i>	<i>Bedarf</i>
Kodierung	70
Filterung	966
Dekodierung	20
Verwaltung	100
Insgesamt	1156

Tabelle 7.2: Rechenleistungsbedarf der Teilalgorithmen in Taktzyklen zwischen 2 aufeinander folgenden Samples

7.6.2 Abschätzung der DSP-Belastung

Es wurde ein DSP der 56300-Familie mit der Taktrate 120MHz angenommen. Wird die Taktrate der Audiodaten auf 44.1kHz festgelegt, stehen $2721 T_C$ zwischen zwei Audiodaten zur Verfügung. Das entspricht einer Belastung des DSPs von 42.5%.

Das uns zur Verfügung gestellte Modul weist eine etwas andere Konfiguration auf: Mit der Taktrate des DSPs von 98.304MHz und der festen Taktrate der Audiodaten von 48kHz stehen lediglich 2048 Taktzyklen zwischen zwei Audiodaten zur Verfügung. Umgerechnet auf die DSP-Belastung entspricht das einem Wert von 56.4%.

7.6.3 Messung der DSP-Belastung für das implementierte System

Wie im Kap 7.3.1 bereits erwähnt wird ein Ausgang des DSPs zur Anzeige der Belastung verwendet. Dieser Ausgang wird am Beginn der Berechnung des Algorithmus in den Zustand HIGH versetzt, um nach Abschluss der Berechnung auf den Zustand LOW gesetzt zu werden. Da die Berechnung des Algorithmus periodisch für jedes Datenwort statt findet, wird ein Rechtecksignal mit der Periodendauer von $1/48kHz = 20.83\mu s$ erzeugt. Über das Tastverhältnis kann die DSP-Belastung direkt gemessen werden.

Es wurde eine Belastung von ca. 64% gemessen, das entspricht ca. 1310 T_C .

Da die Implementation des Algorithmus in einer nicht optimierten Version vorliegt⁶², sind darin 82 NOPs enthalten. Diese könnten bei intensiver Optimierung entfallen und somit der Algorithmus in der endgültigen Version auf 1228 T_C gekürzt werden. Das entspricht ca. 60% der DSP-Belastung.

Die Speicherbelegung zeigt die nachfolgende Tabelle:

<i>Speichersegment</i>	<i>Belegung</i>	
	<i>Worte</i>	<i>%</i>
Programmspeicher P	850	27
Datenspeicher X	1032	8
Datenspeicher Y	1280	18

⁶² wegen der besseren Leserlichkeit der Quelldateien

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die Erkenntnisse aus

- den theoretischen Betrachtungen
- der Analyse verschiedener Systeme anhand des Fehlermaßes
- den Ergebnissen der Hörversuche

führen zu folgenden Vorschlägen zum Entwurf eines binauralen Wiedergabesystems.

Um eine ausreichende Genauigkeit der Abbildung virtueller Schallquellen zu erreichen, sollte ein Ambisonic-System mindestens 3. Ordnung konzipiert werden. Durch Gewichtung der Ordnungen kann die Stabilität der Lokalisation und somit die Konsistenz der Abbildung erhöht werden. Dies führt jedoch zu einer Quellaufweitung, was eine höhere Lokalisationsunschärfe bedeutet. Um diesem Effekt entgegenwirken zu können, empfiehlt sich die Erhöhung der Ambisonic-Ordnung, wenn die zur Verfügung stehende Rechenleistung ausreicht. Die Gewichtung der Kanäle eines Ambisonic-Systems 4. Ordnung gemäß einem Kaiserfenster lieferte sehr gute Ergebnisse (vgl. Kap. 4.3.3, Kap. 5.2.4, Kap. 6.8).

Poletti zeigt in [35], dass die Dekodierung eines zweidimensionalen Ambisonic-Systems der Ordnung M auf eine äquidistante Anordnung von $N=2M+1$ Lautsprechern den geringsten Abbildungsfehler ergibt. Durch eine symmetrische Anordnung der Lautsprecher bezüglich der Medianebene kann eine Halbierung der Anzahl der zur Erzeugung der binauralen Signale benötigten HRIR-Filter erreicht werden (vgl. Kap. 4.3.2). Bezüglich der räumlichen Anordnung des

Lautsprechersets, mit einem Lautsprecher bzw. zwei Lautsprechern vorne (0° -Richtung, vgl. auch Anhang B), kann keine allgemein gültige Aussage getroffen werden. Die Abbildungsgenauigkeit bei den verschiedenen Möglichkeiten der Anordnung ist für die jeweils implementierten HRIRs explizit zu untersuchen.

Die Betrachtung von Systemen mit verschiedenen Längen der HRIRs ergibt, dass Filterlängen kleiner 64 Samples⁶³ zu einer deutlichen Verringerung der wahrgenommenen Externität, sowie zu großen Lokalisationsfehlern bei bestimmten Azimutpositionen führen. Weiters ist ersichtlich, dass eine Filterlänge von 256 Samples bezüglich einer Länge von 128 Samples, keine signifikante Verbesserung der Lokalisationseigenschaften aufweist.

Die Verwendung von HRIRs verschiedener Herkunft zeigt, dass die Art der HRIRs eine wesentliche Rolle bezüglich der Lokalisationseigenschaften spielt. Bei den Kemar-HRIRs zeigen sich im vorderen Bereich ($\pm 45^\circ$) kleinere Lokalisationsfehler als bei den AKG-HRIRs. Im seitlichen Bereich ($\pm 90^\circ$) ergeben sich schlechte Lokalisationseigenschaften, d.h. großer Lokalisationsfehler und große Lokalisationsunschärfe. Die Kemar-HRIRs ermöglichen gegenüber den AKG-HRIRs eine bessere Außerkopflokalisierung (Externität). Hingegen ist die Klangqualität der abgebildeten Quelle über den gesamten Azimutbereich unzureichend. Steht die gleichmäßige Lokalisation über den gesamten Azimutbereich, sowie ausgewogener Klang im Vordergrund, sollten die HRIRs der Fa. AKG implementiert werden. Diese führen lediglich im vorderen Bereich zu einer leichten Verschlechterung der Lokalisationseigenschaften, sowie einer schlechteren Externität.

Eine Verbesserung der Lokalisationseigenschaften sowie der Externität, kann durch Hinzufügen von Räumlichkeit (erste Reflexionen und Diffushall) erreicht werden (vgl. Kap. 4.4.3).

8.2 Ausblick

Fehlermaß:

Folgende Vorschläge zur Verbesserung der Qualität, sowie der Erhöhung der Aussagekraft des Fehlermaßes sollten noch umgesetzt werden.

Es wurde der Einfluss der ITD auf die Lokalisation im hohen Frequenzbereich nicht berücksichtigt. Die menschliche Hörwahrnehmung wertet im höherfrequenten Bereich nicht den Phasenunterschied, sondern die zeitliche Differenz der Einhüllenden (IGD, *interaural group delay*) aus. Um dies bei der Zusammenführung von ITD und ILD berücksichtigen zu können, müssen die

⁶³ Bei einer Abtastfrequenz von 44.1kHz

relevanten Frequenzbereiche der ITD explizit berechnet werden. Danach kann wiederum eine gewichtete Summation der jeweiligen Lokalisationsparameter (ITD, IGD und ILD) erfolgen.

Weiters empfiehlt sich die Entwicklung eines Maßes zur objektiven Klangbewertung. Dabei ergibt sich die Problematik der Definition eines optimalen Klanges, sowie der Festlegung von Klangparametern. Eine Optimierung des Klanges kann auch zu einer Beeinflussung der Lokalisationseigenschaften des Systems führen.

Eine umfassende Systemanalyse des Fehlermaßes hinsichtlich verschiedener Eingangsgrößen (HRIRs, Ambisonic-Parameter, usw.) sollte durchgeführt werden, um weitere Optimierungsmöglichkeiten des mathematischen Modells ermitteln zu können.

Hörversuche:

Die Ergebnisse der Hörversuche zeigen eine sehr große Streuung der einzelnen Messwerte der jeweiligen Versuchspersonen. Eine Erhöhung der Anzahl der Messpunkte ist unbedingt erforderlich, um eine bessere statistische Auswertung zu erreichen. Da dies einen enormen Zeitaufwand für die Versuchspersonen bedeutet, sollte der Hörversuch über einen langen Zeitraum angelegt, sowie die Probanden finanziell entschädigt werden.

Um den Einfluss des Lerneffektes der Versuchspersonen auf die Ergebnisse minimieren zu können, sollte nicht nur die Reihenfolge der Messpositionen zufällig gewählt werden, sondern auch ein zufälliger Wechsel der Settings nach jedem Messpunkt erfolgen.

Weiters sollte die Trainingsphase zeitlich verlängert und mit mehreren verschiedenen Settings durchgeführt werden. Die verbale Befragung der Eigenschaften des Systems könnte nach dieser Trainingsphase, unabhängig von den Messungen der Lokalisationsfunktion erfolgen. Die Möglichkeit der Selektion der jeweiligen Settings durch die Vpn, könnte die Beurteilung der verbal abgefragten Merkmale erheblich erleichtern.

Eine technische Messung der Lokalisationsunschärfe wurde im Zuge dieser Arbeit nicht durchgeführt. Der Messvorgang könnte ähnlich der Messung des Lokalisationsfehlers mit frequenzgruppenbreiten Rauschsignalen durchgeführt werden. Dies führt jedoch zu einem enormen Anstieg der durchzuführenden Messungen. Aus diesem Grund sollten alternative Messmethoden entwickelt werden.

Implementation:

Die Implementierung des Algorithmus führt zu einer relativ hohen Auslastung der Motorola Hardware. Um die Abbildungseigenschaften des Systems zu verbessern, sollte versucht werden ein System 4.Ordnung mit Kaiserfenster zu implementieren. Die verbleibende Rechenleistung kann dann jedoch nur mehr für Zusatzfunktionen, wie z.B. die akustische Linse (vgl. Kap. 4.4) in geringem Ausmaß genutzt werden.

Bei der Hardware von Texas Instruments sind ausreichende Leistungsreserven vorhanden um Weiterentwicklungen durchführen zu können, diese stellt somit eine geeignete Entwicklungsplattform dar. Implementierungen z.B. eines 3D-Systems, Raumsimulation in Echtzeit oder die Anbindung einer ADAT-Schnittstelle sind durchaus denkbar.

Optimierung des Gesamtsystems:

Die Filterung der virtuellen Lautsprechersignale mit den zugehörigen HRIRs benötigt einen Großteil der zur Verfügung stehenden Rechenleistung. Um die Effizienz des Algorithmus zu steigern, könnte versucht werden die HRIRs gezielt zu modellieren. Dabei könnte auch eine Optimierung hinsichtlich der Klangqualität durchgeführt werden.

Ein parametrisierbares HRIR-Modell könnte die Problematik der Verwendung nicht-individualisierter HRIRs, durch individuelle Anpassung des Modells auf den jeweiligen Benutzer lösen.

The only sure way to avoid making mistakes is to have no new ideas.

Albert Einstein (1879-1955)

Anhang A – MAJNOI_Toolbox für MATLAB

Die gesamten theoretischen Berechnungen wurden mit MATLAB 6.0 (R12) unter Verwendung der Signal Processing Toolbox und der Statistical Toolbox von Mathworks [28] durchgeführt.

Da der Schwerpunkt in der Berechnung binauraler Wiedergabesysteme liegt, mussten einige Erweiterungen der Standardfunktionen von MATLAB erstellt werden. Die Verarbeitung von vielen unabhängigen Vektoren, die in einer Matrix vorlagen, führte zur iterativen Abarbeitung und somit zum häufigen Gebrauch von Schleifen. Um die Verarbeitungsgeschwindigkeit steigern zu können, wurden vorwiegend die Vektoroperationen von MATLAB angewendet. Es mussten auch diverse Funktionen zur Visualisierung der 3-dimensionalen Vektoren geschrieben werden, damit eine übersichtliche Darstellung erreicht werden konnte.

Es wurde somit eine Sammlung einiger Funktionen erstellt, die die Probleme zum Teil behoben haben, zum Teil die Handhabung der Algorithmen erleichterten. Die Fülle an Funktionen kann in folgende Gruppen eingeteilt werden:

- ◆ allgemeine Visualisierung von Matrizen/Vektoren
- ◆ audio-bezogene Signalverarbeitung
- ◆ Erstellung und Beurteilung von Ambisonic-Systemen
- ◆ allfällige Berechnung zur Auswertung der Hörversuche

Die ersten zwei Gruppen enthalten relativ allgemein gehaltene Funktionen, die ohne jeglicher Änderung außerhalb dieser Diplomarbeit verwendet werden können.

Die Funktionen der zwei letzten Gruppen sind sehr ziel-spezifisch aufgebaut und deswegen in der Toolbox nicht enthalten. Diese Gruppen werden im Anhang B und Anhang C beschrieben.

Installation der MAJNOI_Toolbox:

- ◆ Die Dateien der MAJNOI_Toolbox in ein zuvor erstelltes Verzeichnis kopieren.
- ◆ Dieses Verzeichnis als Suchpfad in MATLAB festlegen:
 - Das Fenster „Set Path“ im Menü „File“ → „Set Path...“ öffnen
 - Mit „Add folder“ den Namen des Verzeichnisses angeben.
 - Mit „Save“ und „Close“ die Änderungen speichern und zum Hauptfenster zurückkehren

Funktionen zur Visualisierung der Daten:

◆ <code>mn_colorring</code>	Darstellung der Werte eines Vektors in einem kreisförmigen Ring über eine Farbskala
◆ <code>mn_pcolor</code>	Darstellung der Werte in einer Matrix unter Benutzung einer Farbskala (wie <code>pcolor</code>), mit zusätzlichen Balken für die Mittelwerte der Reihen und Spalten.
◆ <code>mn_plot</code>	Dia-Show von <code>plot</code> -Diagrammen
◆ <code>mn_polar</code>	Dia-Show von <code>polar</code> -Diagrammen
◆ <code>mn_scatter</code>	Übersichtlichere Variante von <code>plot</code> bei Darstellung von Matrizen

Funktionen zur audio-spezifischen Signalverarbeitung:

◆ <code>d2r</code>	Rechnet Winkelangaben in Grad Winkel in Radiant um (degrees → radiant). Matrixoperation
◆ <code>mn_error</code>	Berechnet Fehler zwischen Ist- und Soll-Vektoren auf verschiedene Arten. Matrixoperation
◆ <code>mn_findzero</code>	Sucht einen Nulldurchgang in einem Vektor. Matrixoperation.
◆ <code>mn_grpdelay</code>	Rechnet aus Impulsantworten die dazugehörigen Gruppenlaufzeiten aus. Matrixoperation.
◆ <code>mn_ir2tf</code>	Rechnet Impulsantworten in Spektren um. Matrixoperation.
◆ <code>mn_normmatr</code>	Normalisiert die Vektoren auf den Energiegehalt der Referenzmatrix. Matrixoperation

◆ mn_normvect	Normalisiert die Vektoren auf die Energiegehalte der Referenzvektoren. Matrixoperation.
◆ r2d	Rechnet Winkelangaben in Radiant in Winkel in Grad um (radian → degrees). Matrixoperation
◆ mn_barkfilter	Berechnet Filterkoeffizienten für eine Filterbank der gehörspezifischen Frequenzgruppen.
◆ mn_f2bark	Transformiert ein Spektrum in die Bark-Skala. Matrixoperation

Die mit „Matrixoperation“ bezeichneten Funktionen können durch Benutzung einer Matrix mehrere Vektoren in einem Aufruf behandeln. Es kann somit ein ganzes Set an Vektoren zu einer Matrix zusammengefasst und mit einem einzigen Aufruf bearbeitet werden.

Nun erfolgt eine detaillierte Beschreibung aller angegebenen Funktionen in alphabetischer Reihenfolge.

MAJNOI_TOOLBOX: d2r

Syntax:

```
rad=d2r(deg)
```

Beschreibung:

Rechnet Winkelangaben in Grad (degrees) in Radiant um. Wegen der Einfachheit der Berechnung kann hier nicht von einer „Funktion“ der MAJNOI_Toolbox im eigentlichen Sinne gesprochen werden – diese Funktion wird des öfteren von anderen Teilen der Toolbox benutzt und nur wegen der Vollständigkeit hier erwähnt.

Parameter:

deg	Winkelangabe in Grad
rad	Winkelangabe in Radiant

MAJNOI_TOOLBOX: mn_barkfilter

Syntax:

```
[B,A,soll,ist] = mn_barkfilter(ordr)
```

Beschreibung:

Liefert Filterkoeffizienten einer gehörspezifischen Filterbank um Signale in Frequenzgruppen filtern zu können. Mit dem Parameter `ordr` kann die Ordnung und somit die Genauigkeit der Filter variiert werden. Die Filterkoeffizienten werden in Form zweier Vektor-Sets (`B` und `A`)⁶⁴ geliefert. Diese Sets werden in „cells“ eingebettet, die Verpackung ist wegen der ungleichmässigen Länge der Vektoren zwingend notwendig. Die Koeffizienten können in dieser Form direkt von `mn_grpdelay` verarbeitet werden. Zur Kontrolle der entworfenen Filter werden die Spektren der Filter in zwei Matrizen geliefert: der Amplitudengang des idealen Filter (`soll`) und die Spektren der errechneten Filter (`ist`). Für einen direkten Vergleich bietet sich `mn_plot` optimal an. Da die Filter mit dem MATLAB-Befehl `firls` berechnet werden, handelt es sich hier um FIR-Filter.

Parameter:

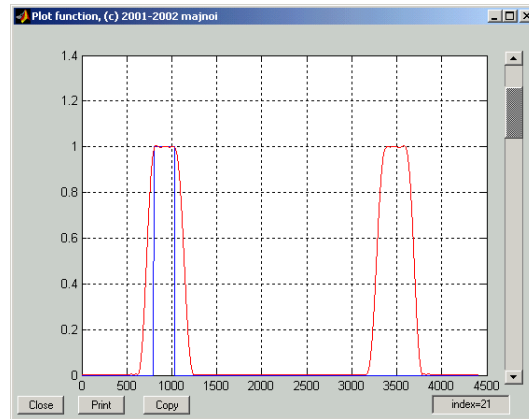
<code>ordr</code>	Vektor mit 2 Skalaren: die Ordnung des ersten Filters (am Anfang des Frequenzbereiches) und die Ordnung des letzten Filters (am Ende des Frequenzbereiches).
<code>B</code>	„Cell“ mit Filterkoeffizienten. Ist immer 1, da FIR-Filter erzeugt werden
<code>A</code>	„Cell“ mit Filterkoeffizienten, entspricht der Impulsantwort jedes einzelnen Filters
<code>soll</code>	Amplitudengänge der idealen Filter, für die Kontrolle der Ergebnisse
<code>ist</code>	Frequenzgänge der erzeugten Filter, auch für die Kontrolle

Beispiel:

```
[B,A,soll,ist] = mn_barkfilter([400 10]);
mn_plot(soll,abs(ist),'r')
```

Darstellung des Filters für das 22-te Frequenzband:

⁶⁴ Zähler- und Nennerpolynom der Übertragungsfunktion, siehe Beschreibung von `filter` in MATLAB [28]



Weitere Anwendung: siehe `mn_grpdelay`

MAJNOI_TOOLBOX: `mn_colorring`

Syntax:

```
mn_colorring(vec,os)
```

Beschreibung:

Die Werte des Vektors `vec` werden in einem Ring als Farbmuster angezeigt. Somit wird der Vektor als eine periodische Funktion in 2π und deren Werte als Farbinformationen interpretiert. Mit der Angabe von `os` kann der Radius des Ringes bestimmt werden und somit mehrere Ringe in einer Darstellung angezeigt werden. Die Breite des Ringes ist auf 0.05 fixiert, für eine ansprechende Darstellung wird die Verwendung von Radien im Bereich 0.7 und 1.2 empfohlen.

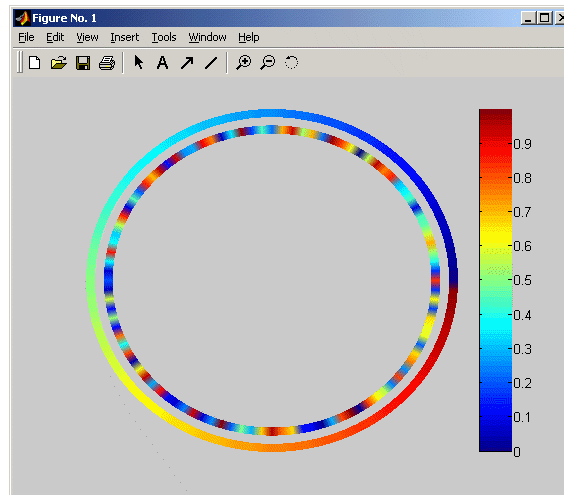
Parameter:

<code>vec</code>	Der darzustellende Vektor
<code>os</code>	Der Radius der Ringes, vom Mittelpunkt bis zur inneren Kante des Ringes.

Beispiel:

```
mn_colorring(0:0.01:0.99,1)
hold
mn_colorring(rand(1,100),0.9)
colorbar
```

Zeichnet zwei Farbringe und fügt die Farbskala hinzu.



MAJNOI_TOOLBOX: mn_error

Syntax:

```
res_m = mn_error(user_m, ref_m, mode)
```

Beschreibung:

Es wird die Abweichung der Matrix `user_m` im Bezug auf die Matrix `ref_m` berechnet. Der Parameter `mode` gibt an, in welcher Weise die Berechnung erfolgen sollte.

Parameter:

<code>user_m</code>	Zu prüfende Matrix
<code>ref_m</code>	Referenzmatrix
<code>mode</code>	Berechnungsmodus, als Zeichenkette anzugeben:
<code>'rel'</code>	relativer Fehler der Absolutwerte
<code>'%'</code>	relativer Fehler der Absolutwerte in %
<code>'log'</code>	relativer Fehler der Absolutwerte, logarithmisch
<code>'dB'</code>	relativer Fehler der Absolutwerte, in dB
<code>'ratio'</code>	Verhältnis der beiden Matrizen
<code>'abs'</code>	Absoluter Fehler
<code>'cpx'</code>	Fehler, bei komplexen Matrizen komplexer Fehler
<code>'cpxa'</code>	Absolutwert des komplexen Fehlers
<code>'cpxr'</code>	Relativfehler des Fehlers, komplex
<code>'cpxra'</code>	Absolutwert des relativen komplexen Fehlers

MAJNOI_TOOLBOX: mn_f2bark

Syntax:

```
[out,f] = mn_f2bark(vect, sr)
```

Beschreibung:

Transformiert einen oder mehrere Vektoren in der Matrix `vect` in den gehörspezifischen Frequenzbereich – die Bark-Skala. Weiters werden auch die jeweiligen Grenzen der Frequenzgruppen im Vektor `f` geliefert, damit eine sofortige Darstellung der Transformation möglich wird.

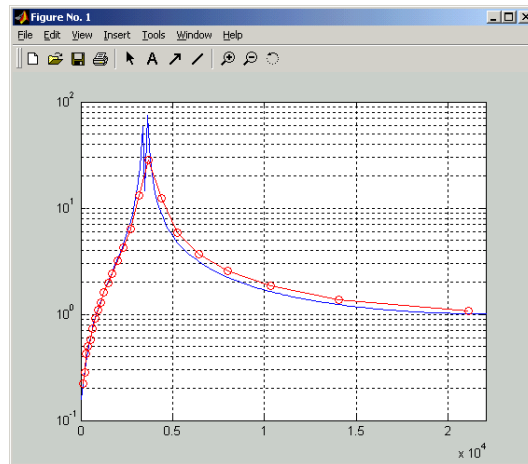
Parameter:

<code>vect</code>	Vektor oder Matrix mit Spektren
<code>sr</code>	Sampling Frequenz. Muss angegeben werden um eine korrekte Transformation durchführen zu können
<code>out</code>	Vektor oder Matrix mit transformierten Spektren in Bark-Skala
<code>f</code>	Vektor mit Frequenzen, die die Grenzen der Frequenzgruppen angeben

Beispiel:

```
fs=44100;          % Sampling Frequenz festlegen
                  % Signal generieren
x=abs(fft(cos(0:0.01:pi).*cos((0:0.5:50*pi))));
                  % In die Bark-Skala transformieren
[b,f]=mn_f2bark(x',fs);
                  % Darstellung, blau:Ausgangsspektrum, rot:transformierte
semilogy(0:fs/ss:fs-1/ss,x)
hold
semilogy(f,b,'ro-')
axis([0 fs/2 0.1 100])
grid
```

Es wird ein Spektrum in Hz-Skala in ein Spektrum in Bark-Skala transformiert. Das Ausgangsspektrum und das Ergebnis werden dargestellt.



MAJNOI_TOOLBOX: mn_findzero

Syntax:

```
res = mn_findzero(matr,near [,err])
```

Beschreibung:

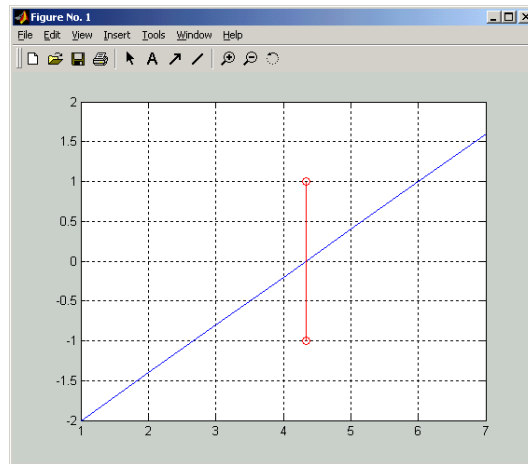
Sucht einen Nulldurchgang der einzelnen Vektoren in der Matrix `matr`, in der Nähe von `near`. Liefert ein Skalar pro Zeile der Matrix `matr`. Sollte kein Nulldurchgang gefunden werden, wird der Wert `err` zurückgegeben. Falls `err` nicht angegeben wurde wird der Wert `NaN` als Voreinstellung verwendet.

Parameter:

<code>matr</code>	Matrix mit Vektoren, innerhalb deren die Nulldurchgänge gesucht werden.
<code>near</code>	Gibt den Ausgangsposition der Suche an
<code>err</code>	Marker für den Fall, dass kein Nulldurchgang gefunden werden konnte. Optional, Voreinstellung: <code>NaN</code>

Beispiel:

```
x=-2:0.6:2      % erzeuge einen Vektor mit Nulldurchgang
plot(x)
hold
idx=mn_findzero(x',0); % finde einen Nulldurchgang
plot([idx idx],[-1 1], 'ro-')
grid
```



MAJNOI_TOOLBOX: mn_grpdelay

Syntax:

```
[out,grpdel] = mn_grpdelay(inp, B, A)
```

Beschreibung:

Berechnet die Gruppenlaufzeiten aus einer Impulsantwort mittels Bestimmung des Energieschwerpunktes einer Frequenzgruppe. Mehrere Impulsantworten können zu einer Matrix zusammengefasst werden. Die Filterkoeffizienten geben die gewünschten Frequenzgruppen an, pro Frequenzgruppe muss ein Vektor-Paar angegeben werden. Die Vektor-Paare müssen in „cells“ A und B verpackt werden – jeder Eintrag in den beiden „cells“ muss einen Vektor mit Filterkoeffizienten für eine Frequenzgruppe beinhalten. Dieses Vorgehen ist aufgrund der verschiedenen Längen der Filterkoeffizienten notwendig. Als Beispiel liefert `mn_barkfilter` die Filterkoeffizienten für die gehörspezifischen Frequenzgruppen in der Bark-Skala. Die Bedeutung der Filterkoeffizienten A und B ist MATLAB-konform, für Details siehe die Beschreibung zum Befehl `filter` in der Signal Processing Toolbox.

Parameter:

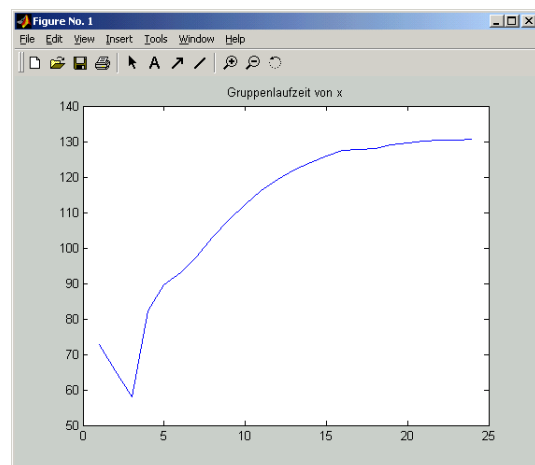
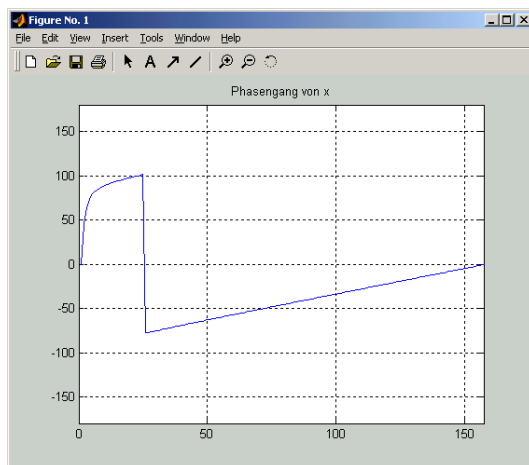
inp	Vektor oder Matrix mit einer oder mehreren Impulsantworten.
B	B-Filterkoeffizienten für einzelne Frequenzgruppen. Für jede Frequenzgruppe wird ein Vektor mit den Filterkoeffizienten angegeben, die Vektoren müssen zu einem „cell“ zusammengefasst werden.
A	A-Filterkoeffizienten für einzelne Frequenzgruppen. Für jede Frequenzgruppe wird ein Vektor mit den Filterkoeffizienten angegeben, die Vektoren müssen zu einem „cell“ zusammengefasst werden.
out	nicht benutzt, aus Kompatibilitätsgründen leider notwendig...
grpdel	Gruppenlaufzeiten, in Samples

Beispiel:

```

% filterkoeffizienten generieren
[B,A] = mn_barkfilter([400 10]);
% signal generieren
x=cos(0:0.01:pi).*cos((0:0.5:50*pi));
% phasengang anzeigen
plot(r2d(angle(fft(x))));
grid
axis([0 size(x,2)/2 -180 180])
title('Phasengang von x');
% gruppenlaufzeit berechnen
[nix,gd]=mn_grpdelay(x', B, A);
% und darstellen
plot(gd);
title('Gruppenlaufzeit von x');

```



MAJNOI_TOOLBOX: mn_ir2tf

Syntax:

```
sp = mn_ir2tf(imp, fade [,len])
```

Beschreibung:

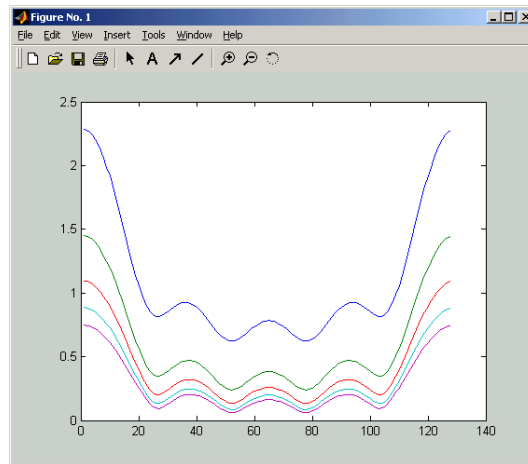
Berechnet Spektren aus gegebenen Impulsantworten. Die Impulsantworten können vor der Fourier-Transformation mit einem Hann-Fenster ausgeblendet (Parameter `fade`) bzw. auf eine bestimmte Länge erweitert werden (Parameter `len`) um eine höhere Auflösung im Frequenzbereich zu erzwingen.

Parameter:

<code>imp</code>	Vektor oder Matrix mit einer oder mehreren Impulsantworten
<code>fade</code>	Anzahl der Samples, die vor dem Ende jeder Impulsantwort ausgeblendet werden sollten. Falls die gesamte Länge der Impulsantworten verwendet werden sollte, muss <code>fade=0</code> angegeben werden
<code>len</code>	endgültige Länge der Impulsantworten vor der Transformation. Die Impulsantworten werden auf diese Länge erweitert. Die Angabe von <code>len</code> ist optional und muss größer als die ursprüngliche Länge der Impulsantworten sein

Beispiel:

```
x=zeros(128,5);  
x(1:5,:)=hilb(5);           % generiere 5 impulsantworten  
plot(abs(mn_ir2tf(x,0)))    % spektren anzeigen
```



MAJNOI_TOOLBOX: mn_normmatr

Syntax:

```
res_m = mn_normmatr(user_m, ref_m [,len])
```

Beschreibung:

Die Matrix `user_m` wird auf den Energiegehalt der gesamten Referenzmatrix `ref_m` normiert. Der Parameter `len` gibt die Möglichkeit die Matrix nur bis zu einer bestimmten Länge zu normieren und ist optional. Wichtiger Unterschied zur `mn_normvect`: hier wird die gesamte Energie der Matrix `ref_m` für die Normierung herangezogen, es werden NICHT einzelne Vektoren der beiden Matrizen aufeinander normiert.

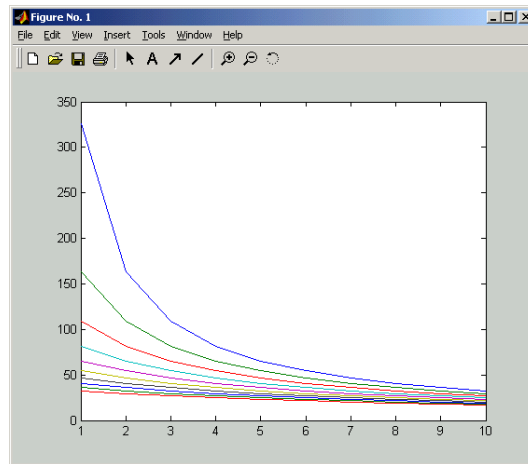
Parameter:

<code>user_m</code>	zu normierender Vektor oder Matrix
<code>ref_m</code>	Vektor oder Matrix mit Referenzdaten
<code>len</code>	Länge des Ergebnisses, optional

Beispiel:

```
ref=magic(10);
usr=hilb(10);
plot(mn_normmatr(usr,ref))
```

Vergleiche: `mn_normvect`



MAJNOI_TOOLBOX: mn_normvect

Syntax:

```
res_m = mn_normvect(user_m, ref_m [,len])
```

Beschreibung:

Die Matrix `user_m` wird auf den Energiegehalt der einzelnen Vektoren in der Referenzmatrix `ref_m` normiert. Der Parameter `len` gibt die Möglichkeit die Matrix nur bis zu einer bestimmten Länge zu normieren und ist optional. Wichtiger Unterschied zur `mn_normmatr`: es werden hier die einzelnen Vektoren über den Energiegehalt aufeinander normiert.

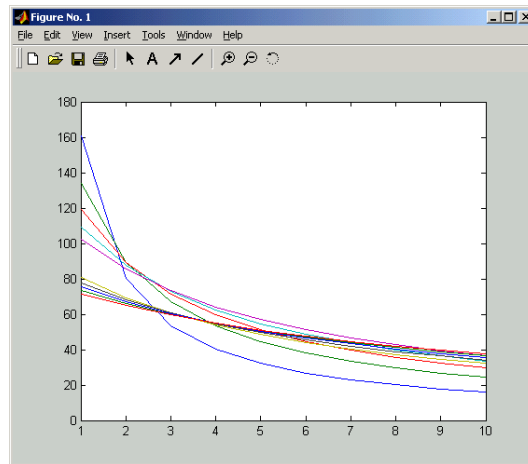
Parameter:

<code>user_m</code>	zu normierender Vektor oder Matrix
<code>ref_m</code>	Vektor oder Matrix mit Referenzdaten
<code>len</code>	Länge des Ergebnisses, optional

Beispiel:

```
ref=magic(10);
usr=hilb(10);
plot(mn_normvect(usr,ref))
```

Vergleiche: `mn_normmatr`



MAJNOI_TOOLBOX: mn_pcolor

Syntax:

```
mn_pcolor([x,y,] z [,tit,xlab,ylab])
```

Beschreibung:

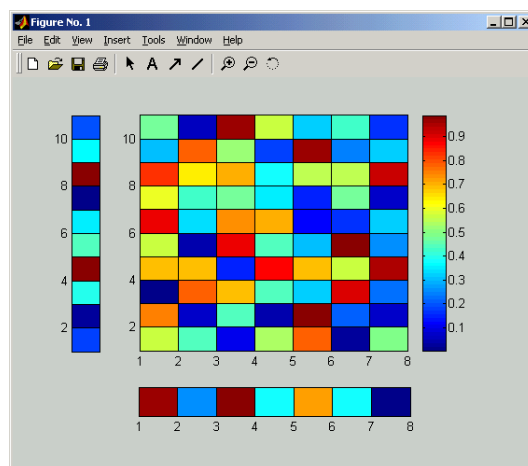
Die Werte der Matrix z werden in einer zweidimensionalen Tabelle auf einer Farbskala dargestellt. Die Reihen der Matrix befinden sich in der Horizontalen und die Spalten in der Vertikalen dieser Tabelle. Dasselbe Darstellung wird mit dem MATLAB-internen Befehl `pcolor` realisiert. Zusätzlich werden die Mittelwerte aller Spalten (für jede Reihe) in einem unter der Matrix liegendem Balken, ebenfalls mit einer Farbskala, dargestellt. In einem vertikalen, links neben der Matrix liegendem Balken werden die Mittelwerte aller Zeilen, Spalte für Spalte, dargestellt. Das Verhalten der X- bzw. Y-Achse kann durch die optionale Angabe der Vektoren x und y beeinflusst werden. Per Voreinstellung werden die Spalten und Zeilen linear dargestellt, beginnend bei 0. Für die Enddarstellung können zusätzliche Informationen wie Titel `tit`, Beschriftung der Spalten `xlab` und die Beschriftung der Zeilen `ylob` angegeben werden. Die Farbskala der Mittelwerte entspricht *nicht* der Farbskala der dargestellten Matrix!

Parameter:

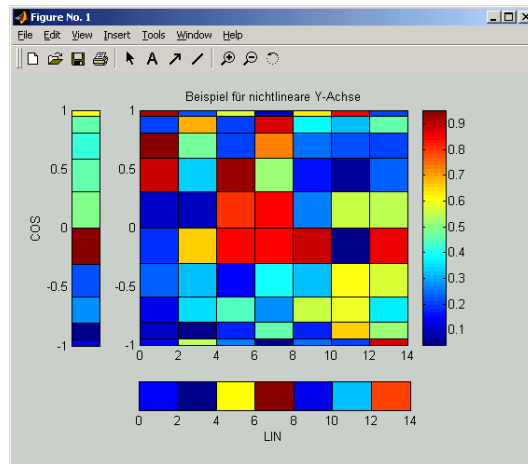
<code>z</code>	2-dim. Matrix mit der Größe $M \times N$
<code>x</code>	X-Vektor mit der Länge $M+1$, optional. Wenn <code>x</code> angegeben wurde muss auch <code>y</code> angegeben werden.
<code>y</code>	Y-Vektor mit der Länge $N+1$, optional. Muss zusammen mit <code>x</code> angegeben werden
<code>tit</code>	Titel des Fensters, optional. Bei Angabe von <code>tit</code> müssen auch <code>xlab</code> und <code>ylab</code> angegeben werden
<code>xlab</code>	Beschriftung der X-Achse, optional. Nur in Verbindung mit <code>tit</code> und <code>ylab</code>
<code>ylab</code>	Beschriftung der Y-Achse, optional. Nur in Verbindung mit <code>tit</code> und <code>xlab</code>

Beispiel 1:

```
mn_pcolor(rand(10,7))
```

**Beispiel 2:**

```
mn_pcolor(0:2:14,cos(0:pi/10:pi),rand(10,7),'Beispiel für nichtlineare  
Y-Achse', 'LIN', 'COS')
```



MAJNOI_TOOLBOX: mn_plot

Syntax:

```
mn_plot([xaxisD,] matrixD [,linetypeD] [,axis] [,step])
```

Beschreibung:

`mn_plot` zeichnet ein gewöhnliches X-Y-Diagramm mittels MATLAB-Funktion `plot`. Im Unterschied zur `plot`-Funktion kann eine gesamte Ansammlung von Vektoren in einer Matrix mit diesem Tools visualisiert werden. Es wird ein Slider auf der rechten Seite des Fensters dargestellt, welcher dem Benutzer die Möglichkeit gibt, zwischen den verschiedenen Vektoren umzuschalten hat. Es entsteht eine Art Dia-Show für eindimensionale Vektoren in kartesischer Darstellung. Diese Funktion ist vor allem bei 3-dimensionalen Matrizen, die in einer echten 3-dimensionalen Projektion (mittels `mesh` oder `surf`) nicht dargestellt werden können, sehr nützlich. Weiters können gleichzeitig mehrere Matrizen (bis zu 10 Stück) mit freigewählten Farben und Symbolen und sogar mit verschiedenen X-Achsen dargestellt werden. Auch die angezeigte Schrittweite bei der Umschaltung zwischen den einzelnen Vektoren kann frei gewählt werden, begonnen wird allerdings immer bei 0. Schaltflächen wie „copy“ und „print“ ermöglichen das Kopieren der momentanen Darstellung in die Zwischenablage oder das Ausdrucken des Fensters auf dem Standarddrucker.

Parameter:

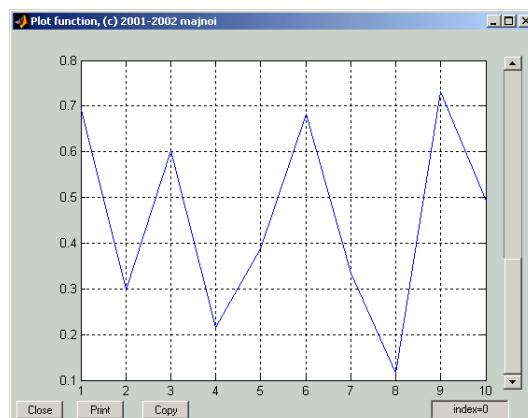
<code>matrixD</code>	darzustellende Matrix, zwei-dimensional
<code>xaxisD</code>	Bezugsvektor für die Abszisse, dient der Zuordnung der Werte in der Matrix <code>matrixD</code> der X-Achse und ist optional. Es kann für jede <code>matrixD</code> ein X-Achsenvektor angegeben werden
<code>linetypeD</code>	Farbe und Symbol für die darzustellende Matrix <code>matrixD</code> , gleiche Syntax wie bei <code>plot</code> . Ist optional und kann für jede <code>matrixD</code> separat angegeben werden
<code>axis</code>	Skalierung des Achsen, optional. Per Voreinstellung wird eine automatische Skalierung verwendet.
<code>step</code>	Angezeigte Schrittweite – stellt den logischen Offset zwischen den einzelnen Vektoren in den Matrizen <code>matrixD</code> . Es kann nur ein <code>step</code> für die gesamte Dia-Show angegeben werden.

Die Position der Parameter `step` und `axis` kann in der Syntax willkürlich gewählt werden, ein Vektor mit 4 Einträgen wird immer als `axis`, ein Skalar immer als `step` interpretiert. Die Parameter `matrixD`, `xaxisD` und `linetypeD` können bis zu 10 mal wiederholt werden, müssen allerdings in genau dieser Reihenfolge angegeben werden. Jede angegebene Matrix wird als `matrixD`, jede Zeichenkette als `linetypeD` und jedes einzeilige Vektor (außer 4x1) als `xaxisD` interpretiert.

Beispiel 1:

```
mn_plot(rand(10,8))
```

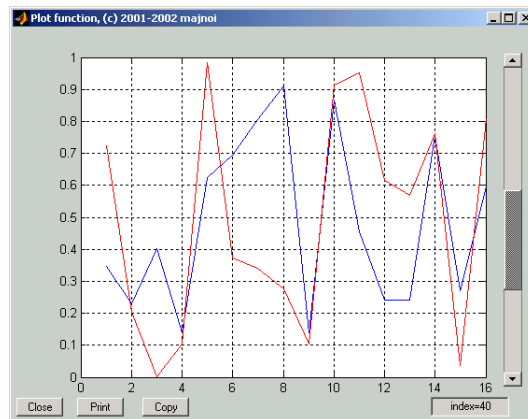
Dia-Show mit einer Matrix – es stehen 8 verschiedene Vektoren zur Auswahl.



Beispiel 2:

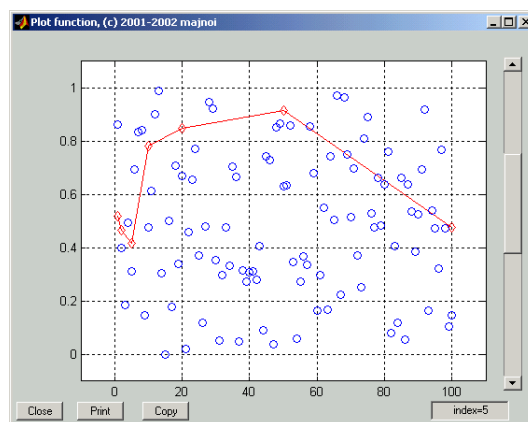
```
mn_plot(rand(36,10),rand(36,10),'r',10)
```

Zeichnet zwei unabhängige Matrizen, die zweite mit rote Farbe. Die Schrittweite `index` wird beim Umschalten der Vektoren um 10 erhöht bzw. erniedrig.

**Beispiel 3:**

```
mn_plot([1 2 5 10 20 50 100],rand(7,10),'rd-',rand(100,10),'bo',[-10 110  
-0.1 1.1])
```

Es werden zwei Matrizen angezeigt. Die erste wird in einer durchgezogenen roten Linie, wobei die Datenpunkte durch Diamanten hervorgehoben wurden, im logarithmischen Maßstab dargestellt. Der X-Achsenvektor `[1 2 5 10 20 50 100]` definiert die X-Achse. Die zweite Matrix wird im linearen Maßstab - ohne Angabe des X-Achsenvektors - mittels blauen Punkte dargestellt. Zusätzlich wird die Darstellung auf den Bereich `x=[-10..110]` und `Y=[-0.1..1.1]` festgelegt.



MAJNOI_TOOLBOX: mn_polar

Syntax:

```
mn_polar([xaxisD,] matrixD [,linetypeD] [,axis] [,step])
```

Beschreibung:

`mn_polar` zeichnet ein Diagramm in einem polaren Koordinatensystem mittels der MATLAB-Funktion `polar`. Im Unterschied zur `polar` kann eine gesamte Ansammlung von Vektoren in einer Matrix mit diesem Tools visualisiert werden. Mit einem Slider auf der rechten Seite des Fensters, hat der Benutzer die Möglichkeit, zwischen den verschiedenen Vektoren umzuschalten. Es entsteht eine Art Dia-Show für Vektoren, die in 2π periodisch sind. Dieses Tool ist vor allem bei 3-dimensionalen Vektoren, die in einer echten 3-dimensionalen Projektion (mittels `mesh` oder `surf`) nicht dargestellt werden können, sehr nützlich. Es können gleichzeitig mehrere Vektoren (bis zu 10 Stück) mit freigewählten Farben, Symbolen und verschiedener Winkel-Achsen dargestellt werden. Weiters kann die angezeigte Schrittweite bei der Umschaltung zwischen den einzelnen Vektoren frei gewählt werden, die Anzeige beginnt immer bei 0.

Schaltflächen wie „copy“ und „print“ ermöglichen das Kopieren der momentanen Darstellung in die Zwischenablage oder das Ausdrucken der Darstellung auf dem Standarddrucker.

Parameter:

<code>matrixD</code>	darzustellende Matrix, zwei-dimensional
<code>xaxisD</code>	Winkelvektor für die Zuordnung der Daten in der Matrix <code>matrixD</code> , optional. Es kann für jede <code>matrixD</code> ein Winkelvektor <code>xaxisD</code> separat angegeben werden.
<code>linetypeD</code>	Farbe und Symbol für die Werte der Matrix <code>matrixD</code> , gleiche Syntax wie bei <code>plot</code> . Ist optional und kann für jede <code>matrixD</code> separat angegeben werden
<code>axis</code>	Skalierung der Achsen, optional. Per Voreinstellung wird eine automatische Skalierung verwendet.
<code>step</code>	Angezeigte Schrittweite – stellt den logischen Offset zwischen den einzelnen Vektoren.

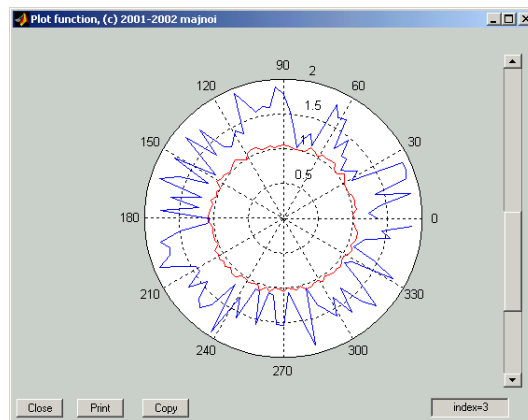
Die Position der Parameter `step` und `axis` kann in der Syntax willkürlich gewählt werden, ein Vektor mit 4 Einträgen wird immer als `axis` und ein Skalar immer als `step` interpretiert. Die Parameter `matrixD`, `xaxisD` und `linetypeD` können bis zu 10 mal wiederholt werden, müssen

allerdings genau in dieser Reihenfolge angegeben werden. Jede angegebene Matrix wird als `matrixD`, jede Zeichenkette als `linetypeD` und jeder einzeilige Vektor (außer `4x1`) als `xaxisD` interpretiert.

Beispiel:

```
mn_polar(rand(100,10)+1,rand(100,10)*0.1+1,'r')
```

Es wird eine Dia-Show mit 2 Matrizen erzeugt, die erste Matrix in blau, die zweite in rot.



MAJNOI_TOOLBOX: mn_scatter

Syntax:

```
mn_scatter(X)
```

Beschreibung:

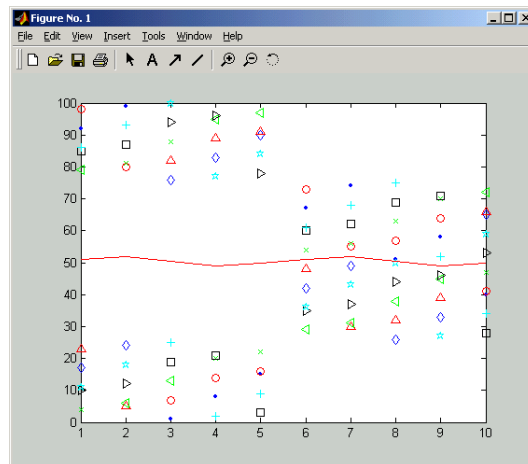
`mn_scatter` stellt die Werte der Matrix `X` in einem `plot` dar. Jede Zeile der Matrix wird entlang der Abszisse eingetragen, wobei die Elemente jeder Spalte farblich unterschiedlich markiert werden. Zusätzlich wird der Median der Spalten eingezeichnet. `mn_scatter` ist für Darstellung diverser stochastischer Daten geeignet, wenn der Befehl `plot(X)` keine übersichtliche Darstellung mehr liefern kann.

Parameter:

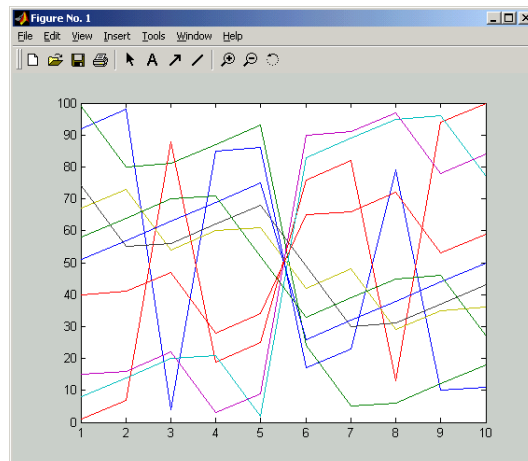
X	darzustellende Matrix, zwei-dimensional
---	---

Beispiel:

```
mn_scatter(magic(10))
```



Als Gegenbeispiel liefert `plot(magic(10))` für dieselbe Matrix eine etwas weniger übersichtliche Anzeige:

**MAJNOI_TOOLBOX: r2d****Syntax:**

```
deg=r2d(rad)
```

Beschreibung:

Die Winkelangabe in Radiant wird in Grad (degrees) umgerechnet. Nur vollständigkeithalber angegeben.

Parameter:

rad	Winkelangabe in Radiant
deg	Winkelangabe in Grad

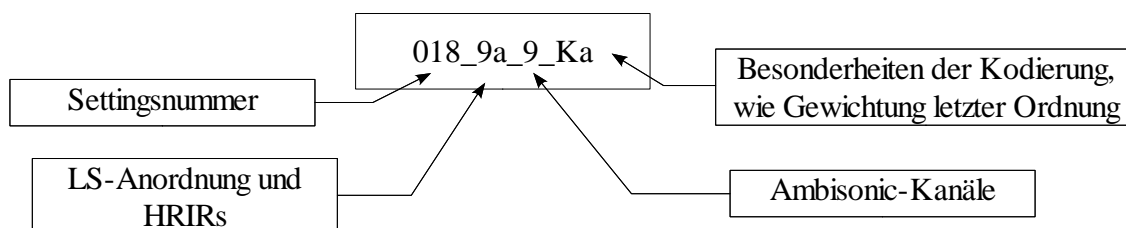
Anhang B – Berechnung der Ambisonic-Systeme

Organisation der Daten

Verzeichnisstruktur

Für die Berechnungen der verschiedenen Ambisonic-Systeme und deren Beurteilung durch ein Fehlermaß wurde das Verzeichnis `Fehlermaß` eingerichtet. Dort befinden sich alle berechneten Ambisonic-Systeme, deren verschiedene Parametrisierung hier als Settings bezeichnet wird. Zusätzlich wurde ein weiteres Unterverzeichnis namens `mfiles` eingerichtet, in dem die fürs Fehlermaß relevanten MATLAB-Funktionen abgelegt wurden.

Die Bezeichnung der berechneten Settings enthält eine Settingsnummer, die sich auf alle in diesem Dokument verwendeten Abbildungen bezieht. Weiters enthält sie auch die Anzahl der Ambisonic-Kanäle, die Anordnung der Lautsprecher, die Art der verwendeten HRIRs⁶⁵ und Hinweise auf Besonderheiten bei der Kodierung der Systeme:



In jedem Settings-Verzeichnis wurden weitere Verzeichnisse mit den PC- und DSP-Programmen, sowie aller für dieses Setting notwendigen Dateien, abgelegt. Dort werden auch alle automatisch generierten Dateien, wie Ambisonic-Filter und binäre DSP-Programme abgespeichert.

⁶⁵ kopfpositionbezogene Impulsantworten (*head related impulse response*)

Mittels diverser MATLAB-Funktionen im Verzeichnis `mfiles` ist es möglich Filterkoeffizienten für ein Ambisonic-System bis zur 4-ten Ordnung zu berechnen und dieses System mit Hilfe des theoretischen Fehlermaßes (siehe Kap. 5) zu erforschen. Es stehen diverse Simulations-, Berechnungs- und Visualisierungstools zur Verfügung. Da die Berechnungen in den jeweiligen Settings-Verzeichnissen stattfinden, sollte das Verzeichnis `mfiles` als globaler Pfad in MATLAB angebunden werden.

Die Tabellen B.2 und B.1 geben Auskunft über die simulierten Systeme und deren Zuordnung zu den Settingsnummern. Die verwendeten Anordnungen der Lautsprecher sind in der Tabelle B.3 angegeben.

Settings-Nr:	Bezeichnung des Datensets in der Datenbank	
HRIR:	verwendeten HRIRs:	
	KEMAR	frei verfügbare HRIRs, siehe [24]
	AKG	von der Firma AKG-Acoustics zur Verfügung gestellte HRIRs, siehe [2]
LA:	Lautsprecheranordnung, A1 bis A4, siehe Tabelle B.3	
LP:	Lautsprecherpositionen	
	7 LS, 1v, 5°:	0°-50°-105°-155°-205°-255°-310°
	7 LS, 1v, 15°:	0°-45°-105°-150°-210°-255°-315°
	7 LS, 2v, 15°:	30°-75°-135°-180°-225°-285°-330°
	5 LS, 1v, 5°:	0°-70°-140°-210°-280°
	9 LS, 1v, 5°:	0°-40°-80°-120°-160°-200°-240°-280°-320°
	9 LS, 1v, 15°:	0°-45°-75°-120°-165°-195°-240°-285°-315°
Ord:	Ordnung des Ambisonic-Systems	
GdO:	Gewichtung der Ambisonic-Ordnungen	
	1	alle Ordnungen voll gewichtet
	0.4	letzte Ordnung mit 0.4 gewichtet
	K-25	Kaiserfenster mit 25dB-Dämpfung der ersten Nebenkeulen
FLen:	Längen der verwendeten HRIR's in Samples	
	F1:	256, 128, 96, 64, 48
	F2:	256, 128, 96, 80, 64, 56, 48

Tabelle B.1: Legende zur Settings-Tabelle

Settings-Nr	HRIR	LA	LP	Ord	GdO	Flen
000_7a_7_1	KEMAR	A1	7 LS, 1v, 5°	3	1	F1
001_7a_5_1	KEMAR	A1	7 LS, 1v, 5°	2	1	F1
002_7a_3_1	KEMAR	A1	7 LS, 1v, 5°	1	1	F1
003_7a_7_1	KEMAR	A1	7 LS, 1v, 5°	3	0.4	F1
004_7a_5_1	KEMAR	A1	7 LS, 1v, 5°	2	0.4	F1
006_7b_7_1	KEMAR	A2	7 LS, 2v, 5°	3	1	F1
007_7b_5_1	KEMAR	A2	7 LS, 2v, 5°	2	1	F1
008_7b_3_1	KEMAR	A2	7 LS, 2v, 5°	1	1	F1
009_7c_7_1	AKG	A1	7 LS, 1v, 15°	3	1	F1
010_7c_5_1	AKG	A1	7 LS, 1v, 15°	2	1	F1
011_7c_3_1	AKG	A1	7 LS, 1v, 15°	1	1	F1
012_7d_7_1	KEMAR	A1	7 LS, 1v, 15°	3	1	F1
013_7d_5_1	KEMAR	A1	7 LS, 1v, 15°	2	1	F1
014_7d_3_1	KEMAR	A1	7 LS, 1v, 15°	1	1	F1
015_7e_7_1	AKG	A2	7 LS, 2v, 15°	3	1	F1
016_7e_5_1	AKG	A2	7 LS, 2v, 15°	2	1	F1
017_7e_3_1	AKG	A2	7 LS, 2v, 15°	1	1	F1
018_9a_9_1	KEMAR	A4	9 LS, 1v, 5°	4	1	F1
019_9a_7_1	KEMAR	A4	9 LS, 1v, 5°	3	1	F1
020_9a_5_1	KEMAR	A4	9 LS, 1v, 5°	2	1	F1
021_9a_3_1	KEMAR	A4	9 LS, 1v, 5°	1	1	F1
022_9a_9_Ka	KEMAR	A4	9 LS, 1v, 5°	4	K-25	F1
023_9c_9_1	AKG	A4	9 LS, 1v, 15°	4	1	F1
024_7a_7_Ka	KEMAR	A1	7 LS, 1v, 5°	3	K-25	F1
100_5a_5_1	KEMAR	A3	5 LS, 1v, 5°	2	1	F1
101_5a_3_1	KEMAR	A3	5 LS, 1v, 5°	1	1	F1
2xx_7d_7_x	KEMAR	A1	7 LS, 1v, 15°	3	letzte Ordnung = 0.4 bis 1.2	F2

Tabelle B.2: Settingsnummern der Ambisonic-Systeme

Funktionen in MATLAB

Für die Berechnung des Fehlermaßes steht die Funktion `mn_simulate` zur Verfügung. Sie bereitet alle Daten vor und ruft nacheinander die zur Berechnung notwendigen Module auf. Zum Schluss liefert sie die Lokalisationseigenschaften des berechneten Systems. Die eigentliche Simulation eines Ambisonic-Systems findet in `mn_simulateambi` statt – es handelt sich um eine schlichte Implementierung des Algorithmus und wird nur im Zusammenhang mit `mn_simulate` verwendet. Weitere Funktionen wie `mn_simloop`, `mn_simloopord` oder `mn_simloopordlen` berechnen das Fehlermaß auf der Grundlage von `mn_simulate` für mehrere Ambisonic-Parameter und dienen der Automatisierung der Berechnungen.

Die Ergebnisse der Simulation in MATLAB können in 2 Dateien organisiert abgelegt werden.

- Die Lokalisationsfunktion und Lokalisationsunschärfe für ein System mit gegebener Filterlänge wurden im `workspace` von MATLAB als eindimensionale Vektoren mit den Namen `LocErr_xxx_yyy` und `Bl_xxx_yyy` gespeichert. `xxx` steht für die Settingsnummer und `yyy` für die verwendete Filterlänge. Falls keine Filterlänge angegeben wurde, handelt es sich um Ergebnisse der Berechnung mit 128 Samples, mit denen hauptsächlich gearbeitet wurde. Abschließend wurde der `workspace` unter dem Namen `data.mat` im Verzeichnis `mfiles` gespeichert (`save data`).
- Die gemittelten Fehler der Ambisonic-Systeme (Lokalisationsfehler und Lokalisationsunschärfe) wurden für die Filterlängen 256, 128, 96, 64 und 48 Samples berechnet und in der Datei `database.m` gesammelt.

Zu einer direkten Visualisierung der Ergebnisse dient `mn_visuloc`, es wird eine interaktive Darstellung des Fehlermaßes erzeugt. Für die Darstellung des Fehlermaßes in Form von Diagrammen steht `mn_eval` zur Verfügung. `mn_eval` greift direkt auf die „Datenbank“ `mn_database` zu und generiert automatisiert Auswertungen des Fehlermaßes in Abhängigkeit der Filterlänge.

Bei einer DSP-Implementierung des Systems helfen die Funktionen `mn_genfilt_asm56` und `mn_compile_TI` weiter. Beide Funktionen bieten eine komfortable Parametrisierung der eigentlichen Berechnung, die in `mn_genambifilt` erfolgt. Diese Funktion bietet eine allgemeine Möglichkeit Filterkoeffizienten für die Implementierung eines Ambisonic-Systems in verschiedenen Formaten zu generieren. Basierend auf dieser Funktion generiert `mn_genfilt_asm56` Filterkoeffizienten für den Motorola Assembler [30]. Die Funktion `mn_compile_TI` erstellt dagegen die Filterkoeffizienten in 'C' und startet die Kompilierung für den 6x-Compiler von Texas Instruments [44].

Nun werden all diese Funktionen in den unterstehenden Abschnitten detailliert beschrieben.

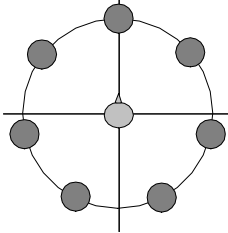
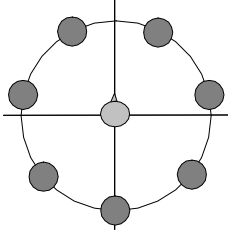
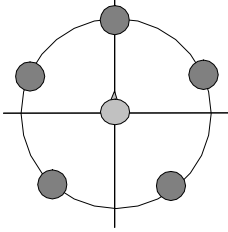
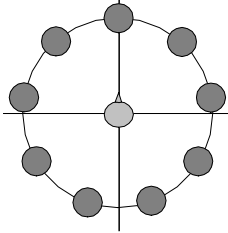
<i>Bezeichnung</i>	<i>Anordnung</i>	<i>Bemerkungen</i>
A1		7 Lautsprecher, 1 LS vorne, äquidistant, Lautsprecherpositionen auf die Anzahl der HRIRs diskretisiert
A2		7 Lautsprecher insgesamt, 2 vorne, äquidistant, Lautsprecherpositionen auf die Anzahl der HRIRs diskretisiert
A3		5 Lautsprecher, 1 LS vorne, äquidistant, Lautsprecherpositionen auf die Anzahl der HRIRs diskretisiert
A4		9 Lautsprecher, 1 LS vorne, äquidistant, Lautsprecherpositionen auf die Anzahl der HRIRs diskretisiert

Tabelle B.3: Anordnung der Lautsprecher

Fehlermaß: `mn_simulate`

Syntax:

```
[LocErr, Bl, LocAvr, BlAvr, LatAvr] = mn_simulate(len);
```

Beschreibung:

Berechnet das Fehlermaß für ein Setting. `mn_simulate` benötigt einen lokalen und mehrere globale Parameter. Bei den globalen Parametern handelt es sich um MATLAB-Variablen, die mit

dem Befehl `save` als `.mat`-Dateien abgespeichert wurden - die Namen der Variablen sind den Dateinamen gleich. Es wird dadurch gewisse Flexibilität bei Verwendung vorhandener Settings erreicht – ein bloßes Kopieren eines alten Setting-Verzeichnisses und das gezielte Überschreiben nicht benötigter Parameter ermöglicht leichte Handhabung bei Berechnung mehrerer Settings.

Die globalen Parameter werden vor den eigentlichen Berechnungen eingelesen und zur Kontrolle angezeigt. Die Dateien müssen sich im gewünschten Setting-Verzeichnis befinden, das als aktuelles Verzeichnis in MATLAB eingestellt werden muss. Die Beschreibung der notwendigen Dateien und globalen Parameter liegt in der Tabelle B.4 vor.

Parameter:

<code>len</code>	Länge der HRIRs ⁶⁶ für die Simulation. Bei <code>len=256</code> werden die HRIRs <code>ref_l</code> und <code>ref_r</code> unverändert übernommen, bei kleineren Angaben werden die HRIRs mit einem Hann-Fenster auf die angegebene Länge ausgeblendet, wobei die letzten 20% der Filter ausgeblendet werden. Zusätzlich werden die ersten 5 Samples der Filter eingeblendet (Hann-Fenster)
<code>LocErr</code>	Vektor mit der Lokalisationsfunktion über Azimutwinkel in Grad
<code>Bl</code>	Vektor mit der Lokalisationsunschärfe über Azimutwinkel als Vielfaches des MAA ⁶⁷
<code>LocAvr</code>	Absoluter Mittelwert des Lokalisationsfehlers in Grad
<code>BlAvr</code>	Mittelwert der Lokalisationsunschärfe als Vielfaches des MAA
<code>LatAvr</code>	Mittelwert der Auslateralisierung der Quelle, in Grad. Bei negativen Werten befindet sich die Schallquelle (im Mittelwert) zu nahe der 0°-Position. Dieses Ergebnis wird nicht weiter verwertet.

Vor der Berechnung sollten die in `data.mat` vorliegenden Daten in den `workspace` geladen werden, damit die Ergebnisse `LocErr` und `Bl` nach der Berechnung in entsprechende Variablen `LocErr_xxx_yyy` und `Bl_xxx_yyy` kopiert (siehe Datenorganisation) und das gesamte `workspace` in `data.mat` wieder gespeichert werden kann.

Arbeitsweise:

Die interne Struktur und somit die Arbeitsweise von `mn_simulate` wird in der Abb. B.1 beschrieben.

⁶⁶ Head Related Impulse Responses

⁶⁷ Minimum Audible Angle

<i>Datei</i>	<i>Variable</i>	<i>Beschreibung</i>
wvec.mat	wvec	Vektor mit den Gewichtungen der einzelnen Ambisonic-Kanäle. Muss diesselbe Länge wie lsvec haben.
ref_l.mat	ref_l	Matrix mit HRIR-Vektoren über den gesamten Azimutbereich von 0° bis 360° für das linke Ohr. Die Länge der HRIRs muss 256 Samples betragen, eine etwaige Kürzung wird direkt beim Aufruf von mn_simulate vorgenommen. Aus der Anzahl der HRIRs ergibt sich die räumliche Auflösung der Lautsprecherpositionen.
	beschreibung	Kurze Beschreibung der verwendeten HRIRs, wird zur Kontrolle angezeigt.
ref_r.mat	ref_r	Matrix mit HRIR-Vektoren für das rechte Ohr. Gleicher Aufbau wie ref_l.mat
lsvec.mat	lsvec	Vektor mit Lautsprecherpositionen, bezieht sich auf den Index der jeweiligen HRIR.

Tabelle B.4: Globale Parameter Berechnungen von Ambisonic-Systemen



Abb. B.1: Struktur von mn_simulate. (*: siehe Fußnote auf der nächsten Seite)

Fehlermaß: mn_simloop

Syntax:

```
[LocAvr, BlAvr, LatAvr] = mn_simloop
```

Beschreibung:

mn_simloop berechnet das Fehlermaß mittels mn_simulate für mehrere Filterlängen und sammelt die skalaren Einzelergebnisse in 3 Vektoren. Die Lokalisationsfunktion und Lokalisationsunschärfe in Abhängigkeit des Azimutwinkels werden verworfen – dafür sollte mn_simulate direkt verwendet werden.

LocAvr	Absoluter Mittelwert des Lokalisationsfehlers in Grad
BlAvr	Mittelwert der Lokalisationsunschärfe als Vielfaches des MAA ⁶⁸
LatAvr	Mittelwert der Auslateralisierung der Quelle, in Grad. Bei negativen Werten befindet sich die Schallquelle (im Mittelwert) zu nahe der 0°-Position.

Es werde Systeme mit folgenden Filterlängen simuliert:

- ◆ 256 Samples
- ◆ 128 Samples
- ◆ 96 Samples
- ◆ 64 Samples
- ◆ 48 Samples

Ergebnisse:

Nach der Berechnung sollten die Ergebnisse in die Datei database.m von Hand eingetragen werden. Es existiert für jede Settingsnummer ein eigener case-Abschnitt, in den die 3 Vektoren (mittels copy/paste) eingefügt werden sollen. Damit wird ein automatisierter Zugriff auf alle Settings möglich.

⁶⁸ ITD: Interaural Time Difference

ITDA: Interaural Time Difference Angle

ILD: Interaural Level Difference

ILDA: Interaural Level Difference Angle

MAA: Minimum Audible Angle, siehe Kap. 2.2.3

Fehlermaß: mn_simloopord

Syntax:

```
[LocAvr, BlAvr, LatAvr, LocErr, Bl] = mn_simloopord(len)
```

Beschreibung:

Simuliert mehrere Ambisonic-Systeme 3-er Ordnung, die Gewichtung der letzten Ordnung wird dabei von 0.4 bis 1.2 in 0.1-Schritten variiert. Die Länge der Filter kann dabei angegeben werden und wird bei allen Gewichtungen konstant gehalten. `mn_simloopord` benötigt dieselben globalen Parameter wie `mn_simulate`.

Parameter:

len	Länge der HRIRs für die Simulation. Bei len=256 werden die HRIRs <code>ref_1</code> und <code>ref_r</code> unverändert übernommen, bei kleineren Angaben werden die HRIRs mit Hann-Fenster auf die angegebene Länge ausgeblendet, wobei die letzten 20% der Filter ausgeblendet werden. Zusätzlich werden die ersten 5 Samples der Filter eingebledet (Hann-Fenster)
LocErr	Matrix mit Lokalisationsfunktionen über Azimutwinkel in Grad
Bl	Matrix mit Lokalisationsunschärfen über Azimutwinkel als Vielfaches des MAA
LocAvr	Vektor mit absoluten Mittelwerten des Lokalisationsfehlers in Grad
BlAvr	Vektor mit Mittelwerten der Lokalisationsunschärfe als Vielfaches des MAA
LatAvr	Vektor mit Mittelwerten der Auslateralisierung der Quelle, in Grad. Siehe <code>mn_simulate</code>

Fehlermaß: mn_simloopordlen

Syntax:

```
[LocAvr, BlAvr, LatAvr] = mn_simloopordlen
```

Beschreibung:

Simuliert mehrere Ambisonic-Systeme 3-er Ordnung, die Gewichtung der letzten Ordnung wird dabei von 0.4 bis 1.2 in 0.1-Schritten variiert. Weiters werden bei den Berechnungen verschiedene Filterlängen benutzt:

- ◆ 256 Samples
- ◆ 128 Samples
- ◆ 96 Samples
- ◆ 64 Samples
- ◆ 48 Samples

Die Funktion `mn_simloopordlen` wurde benutzt um das Fehlermaß für die Settings `2xx` zu erstellen. `mn_simloopordlen` benötigt dieselben globalen Parameter wie `mn_simulate`.

Parameter:

<code>len</code>	Länge der HRIRs für die Simulation. Bei <code>len=256</code> werden die HRIRs <code>ref_l</code> und <code>ref_r</code> unverändert übernommen, bei kleineren Angaben werden die HRIRs mit Hann-Fenster auf die angegebene Länge ausgeblendet, wobei die letzten 20% der Filter ausgeblendet werden. Zusätzlich werden die ersten 5 Samples der Filter eingeblendet (Hann-Fenster)
<code>LocAvr</code>	Matrix mit absoluten Mittelwerten des Lokalisationsfehlers in Grad. In den Zeilen stehen die Einträge für verschiedene Gewichtungen und in der Spalten für verschiedene Filterlängen.
<code>BlAvr</code>	Matrix mit Mittelwerten der Lokalisationsunschärfe als Vielfaches des MAA
<code>LatAvr</code>	Matrix mit Mittelwerten der Auslateralisierung der Quelle, in Grad. Siehe <code>mn_simulate</code>

Fehlermaß: `mn_simulateambi`

Syntax:

```
imp_resp=mn_simulateambi(sys_imp, ang, lsvec, wvec)
```

Beschreibung:

Simuliert ein Ambisonic-System bis zur vierten Ordnung bei gegebenen HRIRs, Lautsprecheranordnung und Gewichtung der einzelnen Ambisonic-Kanäle für eine Schallquellenposition und das linke Ohr. Die Ordnung des Ambisonic-Systems ergibt sich aus der Anzahl der Lautsprecher – es wird ein System mit optimaler Anzahl der Lautsprecher angenommen (siehe Kap. 4.3.1). Für ein überbestimmtes System (mehr Lautsprecher als Ambisonic-Kanäle) können die letzten Kanäle im Vektor `wvec` auf 0 gesetzt werden. Bei einem unterbestimmten

System ergibt die Berechnung der Dekodier-Matrix eine schlecht konditionierte Matrix und das Ergebnis wird unbrauchbar. Sollte die Simulation für das rechte Ohr erfolgen, sind die HRIRs um die Medianebene zu spiegeln.

Diese Funktion stellt eine schlichte Implementierung des im Kap. 4.2 beschriebenen Algorithmus. Für die weitere Benutzung sollte `mn_simulate` verwendet werden.

Parameter:

<code>sys_imp</code>	Matrix mit HRIRs für einzelne Azimutwinkel
<code>ang</code>	Position der Schallquelle in Grad
<code>lsvec</code>	Vektor mit Positionen der Lautsprecher, die Werte sind als Index zur Matrix <code>sys_imp</code> zu verstehen. Die Länge dieses Vektors ergibt die Anzahl der Ambisonic-Kanäle und somit die Ordnung des Systems (siehe Kap. 4.2)
<code>wvec</code>	Vektor mit Gewichtungen der einzelnen Ambisonic-Kanäle. Sollte die gleich Länge wie <code>lsvec</code> haben.

Implementierung: `mn_genfilt_asm56`

Syntax:

```
mn_genfilt_asm56(len);
```

Beschreibung:

Generiert Assembler-Dateien mit Filterkoeffizienten für die Implementierung eines Ambisonic-Systems. `mn_genfilt_asm56` dient lediglich der bequemen Parametrisierung. Die entsprechenden Berechnungen erfolgen in der Funktion `mn_genambifilt`. Die Parametrisierung des Ambisonic-Systems erfolgt über globale Parameter deren detaillierte Beschreibung in Tabelle B.4 vorliegt. Die Handhabung von `mn_genfilt_asm56` gleicht dem Aufruf von `mn_simulate` – hier sei auf die Beschreibung dieser Funktion verwiesen.

Parameter:

<code>len</code>	Länge der HRIRs für die Implementierung, maximal 128 Samples. Die HRIRs werden mit einem Hann-Fenster auf die angegebene Länge ausgeblendet, wobei die letzten 20% der Filter ausgeblendet werden. Zusätzlich werden die ersten 5 Samples der Filter eingeblendet (Hann-Fenster)
------------------	--

Als Ergebnis entsteht eine Schar von Assembler Dateien im aktuellen Verzeichnis. Jede Datei enthält immer 128 Koeffizienten (für ein Filter mit der Länge 128 Samples), bei kürzeren Filter werden die entsprechenden Koeffizienten auf 0 gesetzt. Folgende Dateinamen werden verwendet:

filterW.asm	Filterkoeffizienten des W-Kanals
filterXz.asm	Filterkoeffizienten des X-Kanals der Ordnung z (Bsp: filterX1.asm für Kanal $X^{(1)}$)
filterYz.asm	Filterkoeffizienten des Y-Kanals der Ordnung z (Bsp: filterY3.asm für Kanal $Y^{(3)}$)

Weiters werden in jeder Datei Hinweise auf die verwendete Gewichtung und Lautsprecheranordnung eingetragen.

Implementierung: mn_compile_TI

Syntax:

```
mn_compile_TI(len, prc);
```

Beschreibung:

Die Funktion `mn_compile_TI` ermöglicht je nach Parameter `prc` die Erstellung der Filterkoeffizienten für ein Ambisonic-System, Kompilierung des DSP-Programms mit den neuen Koeffizienten und den anschließenden Start des Virtual Sound Positioning Systems (*VSPS*) [26].

Für die korrekte Ausführung von `mn_compile_TI` muss die Compiler-Umgebung des DSP installiert werden. Dazu ist die Installation des Code Composer Studios [45] notwendig. Die Quelldateien des lauffähigen DSP-Programms müssen sich im Verzeichnis namens `DSP` befinden - ein Unterverzeichnis des aktuellen MATLAB-Verzeichnisses. Dort werden alle Dateien mit den erstellten Filterkoeffizienten abgelegt und die Kompilierung durch den Aufruf von `make.bat` begonnen. Falls erwünscht wird anschließend das *VSPS* gestartet. Dazu muss sich das gesamte *VSPS* im Verzeichnis `VB` befinden. Vor dem Start wird die kompilierte DSP-Datei `main.out` in dieses Verzeichnis kopiert.

Für die korrekte Ausführung benötigt `mn_compile_TI` die Dateien `callmake.bat` und `callvsps.bat` im aktuellen Verzeichnis.

Die Parametrisierung des Ambisonic-Systems erfolgt über globale Parameter deren detaillierte Beschreibung in Tabelle B.4 vorliegt, weitere Hinweise sind in der Beschreibung von `mn_simulate` enthalten.

Parameter:

<code>len</code>	Länge der HRIRs für die Implementierung, maximal 256 Samples. Die HRIRs werden mit Hann-Fenster auf die angegebene Länge ausgeblendet, wobei die letzten 20% der Filter ausgeblendet werden. Zusätzlich werden die ersten 5 Samples der Filter eingeblendet (Hann-Fenster)	
<code>prc</code>	Gibt die Tiefe der Automatisierung an:	
	1	Erstellung der Filterkoeffizienten und anschließende Kompilierung des DSP-Programmes
	2	Erstellung der Filterkoeffizienten, Kompilierung des DSP-Programms und anschließender Start von <i>VSPS</i>

Implementierung: `mn_genambifilt`

Syntax:

```
mn_genambifilt(sys_imp, lsvec, wvec [,output])
```

Beschreibung:

Berechnet Filterkoeffizienten für eine Implementierung des Ambisonic-System bis zur vierten Ordnung bei gegebenen HRIRs, Lautsprecheranordnung und Gewichtung der einzelnen Ambisonic-Kanäle. Die Ordnung des Ambisonic-Systems ergibt sich aus der Anzahl der Lautsprecher – es wird ein System mit optimaler Anzahl der Lautsprecher angenommen (siehe Kap. 2.2.2). Für ein überbestimmtes System (mehr Lautsprecher als Ambisonic-Kanäle) können die letzten Kanäle im Vektor `wvec` auf 0 gesetzt werden. Bei einem überbestimmten System ergibt die Berechnung der Dekoder-Matrix eine schlecht konditionierte Matrix und das Ergebnis wird unbrauchbar. Die Filterkoeffizienten werden je nach `output` in Dateien abgelegt und können direkt in den Quellcode eingefügt werden.

Diese Funktion stellt eine schlichte Implementierung des im Kap. 4.1.4 beschriebenen Algorithmus dar. Für die weitere Benutzung sollte `mn_compile_TI` oder `mn_genfilt_asm56` verwendet werden.

Parameter:

<code>sys_imp</code>	Matrix mit HRIRs für einzelne Azimutwinkel	
<code>lsvec</code>	Vektor mit Positionen der Lautsprecher, die Werte sind als Index der Matrix <code>sys_imp</code> zu verstehen. Die Länge dieses Vektors ergibt die Anzahl der Ambisonic-Kanäle und somit die Ordnung des Systems (siehe Kap. 4.3.1)	
<code>wvec</code>	Vektor mit Gewichtungen der einzelnen Ambisonic-Kanäle. Sollte die gleiche Länge wie <code>lsvec</code> haben.	
<code>output</code>	Gibt das Zielsystem an. Optional, die Voreinstellung ist <code>'c'</code>	
	<code>'c'</code> <code>'C'</code>	C-kompatible Dateien, vorzeichenbehaftete Koeffizienten, auf 2^{15} normiert und von 44.1kHz auf 48kHz interpoliert. Die Dateien haben die Endung <code>.h</code>
	<code>'asm56'</code> <code>'ASM56'</code>	ASM56-kompatible Dateien mit der Endung <code>.asm</code> für den Motorola Assembler der 56000-Serie [30]. Die vorzeichenbehaftete Koeffizienten werden auf 48kHz interpoliert und 2^{23} normiert.
	<code>'pd'</code> <code>'PD'</code>	Filterkoeffizienten im WAV-Format (16bit, 44.1kHz) für das Programm <code>pd</code> [38].

Visualisierung: mn_eval**Syntax:**

```
mn_eval(item, options [,flen])
```

Beschreibung:

Stellt grafisch die Ergebnisse der Fehlermaß-Berechnung für ausgewählte Settings dar. `mn_eval` greift dabei auf die Datei `database.m` zu, in der die Ergebnisse für verschiedene Settings abgespeichert wurden. Im erstellten Diagramm wird der gewünschte Parameter des Fehlermaßes auf der Ordinate eingezeichnet, wobei verschiedene Settings färbig markiert werden. Für die Unterscheidung der Settings wird ein Legende erstellt. Die für die Beschriftung der Abszisse verwendeten Filterlängen sind:

- ◆ 256 Samples
- ◆ 128 Samples
- ◆ 96 Samples

- ◆ 64 Samples
- ◆ 48 Samples

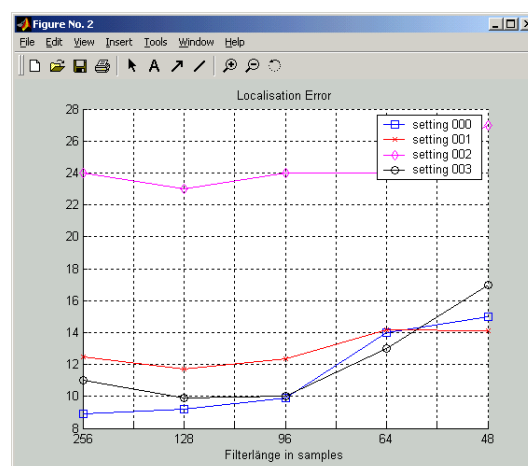
Falls andere Filterlängen verwendet werden sollten, kann das im Vektor `flen` angegeben werden. Die Filterlängen müssen natürlich mit den Einträgen in der Datei `database.m` übereinstimmen. Als Beispiel wurde das Fehlermaß für die Settings 200 bis 212 für zwei weitere Filterlängen (80 und 56 Samples) berechnet und als Settings 1200 bis 1212 abgespeichert. Für diese Settings sollte `mn_eval` mit `flen=[256 128 96 80 64 56 48]` aufgerufen werden. Für alle anderen Settings muß `flen` nicht angegeben werden.

Parameter:

item	Vektor mit den Settingsnummern	
options	Gibt die Art der Ausgabe an.	
	'bl'	Mittlere Lokalisationsunschärfe
	'loc'	Absoluter Mittlere Lokalisationsfehler
	'lat'	Mittlerer Lateralisationsfehler
flen	Vektor für die Beschriftung der Abszisse. Optional, wenn nicht angegeben wird <code>flen=[256 128 96 64 48]</code> verwendet. Für diese Filterlängen wurden auch alle Settings mit <code>Set# < 1000</code> in der <code>database.m</code> erstellt.	

Beispiel:

```
mn_eval([0 1 2 3 ], 'loc')
```



Evaluierung: mn_visuloc

Syntax:

```
mn_visuloc(offset,blur)
```

Beschreibung:

Bietet eine interaktive Darstellung der Lokalisationseigenschaften eines Ambisonic-Systems. Es wird die Lokalisationsfunktion und die Lokalisationsunschärfe des simulierten System im Vergleich zu einer idealen Lokalisation und der auf den MAA⁶⁹ bezogenen Lokalisationsunschärfe dargestellt.

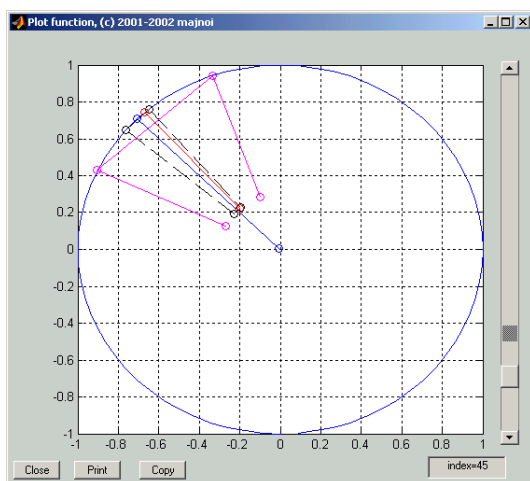
Parameter:

offset	Vektor mit der Lokalisationsfunktion
blur	Vektor mit der Lokalisationsunschärfe

Beispiel:

Für die Ausführung dieses Beispiels sollte das aktuelle Verzeichnis von MATLAB auf das Verzeichnis `mfiles` (mit den Fehlermaß-Funktionen) eingestellt werden.

```
load data
mn_visuloc(LocErr_000, Bl_000)
```



Legende:

blau:	Soll-Lokalisation
rot:	Ist-Lokalisation
schwarz:	Soll-Bereich der MAA
magenta:	Bereich der Lokalisationsunschärfe

⁶⁹ Minimum Audible Angle, siehe Kap 2.2.3

Anhang C – Hörversuche

Versuchsaufbau

Die Hörversuche wurden im Experimentalstudio des IEM⁷⁰ durchgeführt – dieser Raum wurde speziell für den Aufbau adaptiert.

Die Lautsprecher wurden kreisförmig mit einem Radius von 2m in einer Höhe von 1.4m auf Stativen montiert. Im Mittelpunkt der Anordnung wurde eine drehbare Sitzgelegenheit für die Versuchspersonen aufgestellt. Mittels variabler Absorberwände wurde die Nachhallzeit des Raumes auf 0.27s verringert.

Da für den Versuchsablauf 2 Computer⁷¹ und ein Mischpult notwendig waren, wurden diese hinter den Absorberwänden aufgestellt, wobei Sichtkontakt zur Versuchsperson gewährleistet war. Eine schematische Darstellung des Aufbau zeigt Abb C.3. Weiters werden in Abbildungen C.1 und C.2 Aufnahmen des fertiggestellten Aufbaus aus verschiedenen Blickwinkeln gezeigt.



Abb. C.1: Aufbau der Versuchsraumes (1)



⁷⁰ Institut für Elektronische Musik und Akustik

⁷¹ handelsübliche Personal Computer mit entsprechender Software

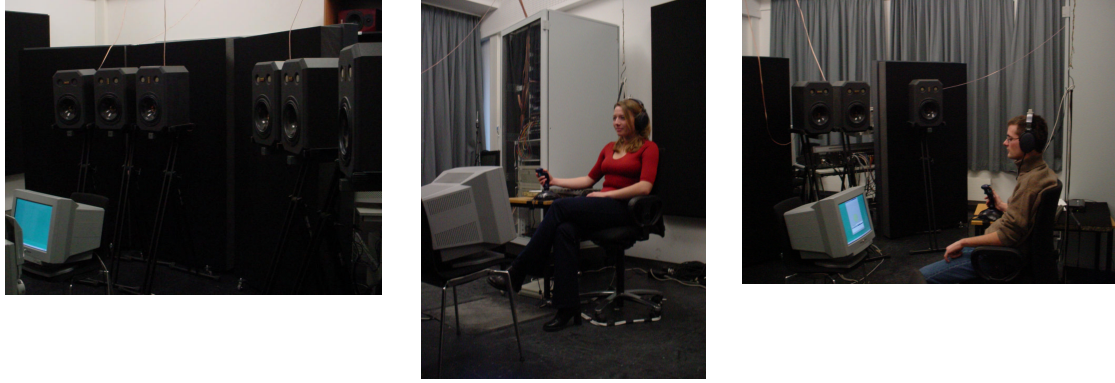


Abb. C.2: Aufnahmen des Aufbaus (2)

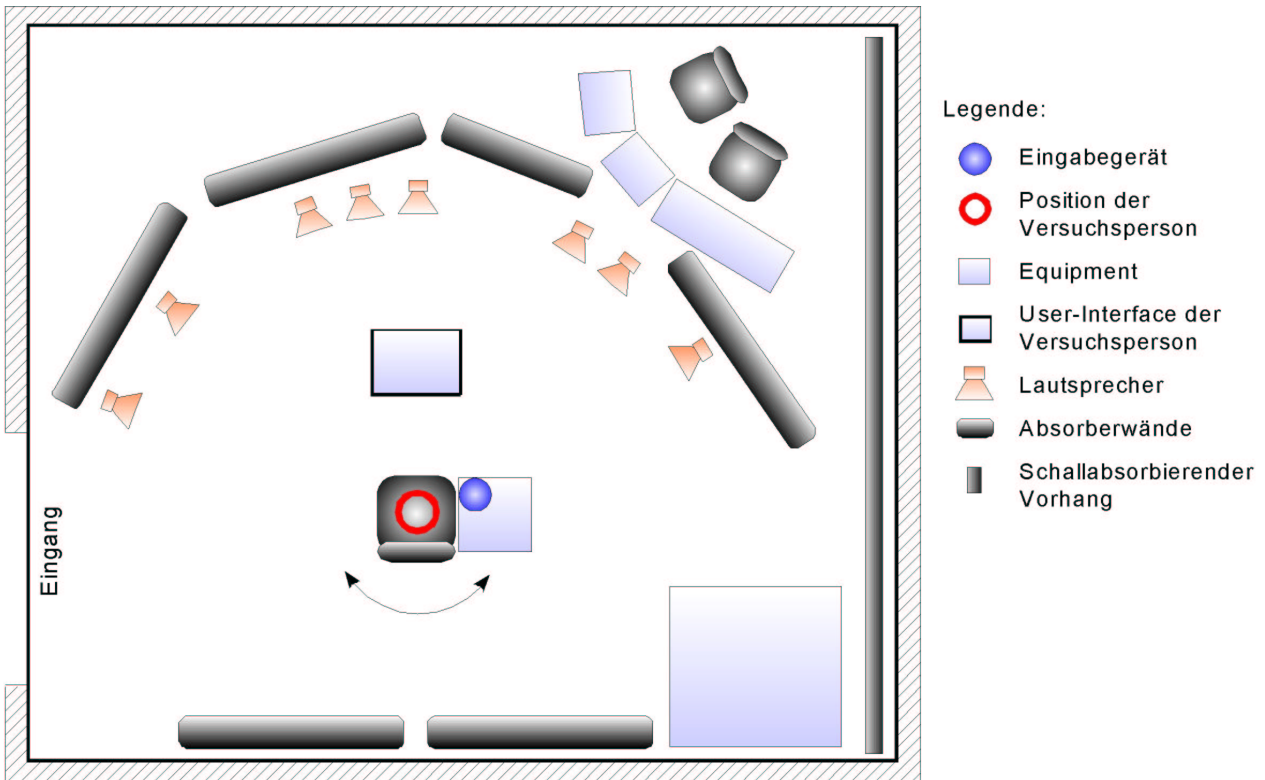


Abb. C.3: Schematische Darstellung des Aufbaus für die Durchführung der Hörversuche

Verwendete Settings

Aufgrund der hohen Anzahl an verschiedenen optimierten Ambisonic-Systemen, musste die zu untersuchende Anzahl an Settings dezimiert werden. Es wurden spezielle Settings erstellt, die sich nur in einem Parameter voneinander unterschieden. Dadurch können Rückschlüsse von den Versuchsergebnissen auf die jeweiligen Parameter gezogen werden.

Folgende Parameter wurden variiert um die Settings zu erstellen:

- Ordnung des Ambisonic-Systems
- Länge und Art der HRIR-Filter
- Gewichtung der einzelnen Ambisonic-Kanäle
- Anordnung der virtuellen Lautsprecher

Weiters erfolgte eine direkte Implementierung der unterschiedlichen HRIRs, wobei bei bewegter Schallquelle direkt zwischen den jeweiligen HRIRs umgeschaltet wurde. Es erfolgte keine Interpolation.

Zur Durchführung des Hörversuches erfolgte eine Nummerierung der Settings in der Reihenfolge ihrer Verwendung. Aufgrund der Übersichtlichkeit und besseren Darstellbarkeit bei der Auswertung erfolgte eine nachträgliche Änderung der Nummerierung der Settings. Die Tabelle C.1 gibt Auskunft über die verwendeten Settings und deren Nummerierungen, wobei beide Nummerierungsarten dargestellt werden.

SetNr	Set#	Flen	HRIR	LS-vorne	Ord	Gew	Exp-Setup
1	#301	256	AKG	-	-	-	11
2	#15	128	AKG	2	3	1	3
3	#9	128	AKG	1	3	1	2
4	#9	96	AKG	1	3	1	6
5	#9	64	AKG	1	3	1	4
6	#10	128	AKG	1	2	1	5
7	#0	128	KEMAR	1	3	1	1
8	#24	256	KEMAR	1	3	K-25	10
9	#24	128	KEMAR	1	3	K-25	8
10	#204	128	KEMAR	1	3	0.4	7
11	#300	256	KEMAR	-	-	-	9

Tabelle C.1: verwendete Settings, Legende siehe Tabelle C.2

SetNr	Nummer der verwendeten Settings, wird sowohl bei den Abbildungen als auch in den mathematischen Formulierung benutzt	
Set#	Nummer der entsprechenden Settings bei der Fehlermaßberechnung. Die Settings mit der direkten Implementation der HRIRs (Set#: #300 und #301) wurden nicht simuliert und sind deshalb bei der Fehlermaßberechnung nicht enthalten	
Flen	verwendete Länge der HRIRs in Samples	
LS-vorne	Anzahl der virtuellen Lautsprecher vorne, entspricht der jeweiligen Lautsprecheranordnung von Fehlermaß, siehe Tabelle B.3	
Ord	Ordnung des Ambisonic-Systems	
Gew	Gewichtung der jeweiligen Ordnungen	
	1	alle Ordnungen voll gewichtet
	0.4	die 3-te Ordnung mit dem Faktor 0.4 gewichtet
	K-25	Gewichtung aller Ordnungen mit einem Kaiserfenster mit 25dB-Dämpfung bei der ersten Nebenkeulen
Exp-Setup	Nummer der Settings während der Durchführung der Hörversuche. Entspricht dem Eintrag „Exp.Setup“ in der PC-Software, alle während der Hörversuche archivierten Daten sind ausschließlich nach dieser Nummer geordnet	

Tabelle C.2: Legende zur Tabelle C.1

Auswertung der Messdaten

Alle in den Hörversuchen gemessenen Daten wurden in der Datei `auswertungMATLAB.sdc`⁷² abgelegt. Über ein Text-Dokument können die dort vorliegenden Daten ins MATLAB importiert werden.

In MATLAB stehen mehrere Funktionen zur weiteren Auswertung zur Verfügung.

`mn_eval` stellt den ersten Schritt der Verarbeitung der Offset-Messungen dar. Es bereinigt die Daten von Ambiguitätsproblemen, berechnet den Fehler der Lokalisation und liefert das Ergebnis in Form einer Matrix $\hat{\mathbf{O}}\mathbf{S}(V_{pn}, SetNr, VersNr, Q_{pos})$ (siehe Kap 6.7.3 für weitere Erläuterungen). `mn_eval_scatter` berechnet die Streuparametermatrix $\mathbf{S}(V_{pn}, SetNr, Q_{Pos})$. In dieser Matrix werden alle ungültigen Werte mit `NaN`⁷³ markiert, so dass bei Benutzung der Befehle `nanmean`

⁷² erstellt mit Star-Office 5.2, [43]

⁷³ Not A Number, ein Wert der bei den Berechnungen nicht wie eine Zahl verarbeitet wird. Jede Operation mit NaN ergibt wiederum NaN.

(Mittelwertbildung ohne Berücksichtigung der NaN-Werte) oder `nanstd` (Standardabweichung) nur die gültigen Messungen im Ergebnis berücksichtigt werden.

Für die Berechnung der Ambiguitätsprobleme steht die Funktion `mn_ambig` zur Verfügung, sie berechnet den Anteil der Verwechslungen pro Person und Setting.

Für die Auswertung der verbalen Befragung wird `mn_eval_verbal` verwendet. Diese Funktion liefert die Vektoren $LU(SetNr, Vpn)$, $EX(SetNr, Vpn)$, $KM_H(SetNr, Vpn)$, $KM_M(SetNr, Vpn)$, $KM_B(SetNr, Vpn)$ sowie $KO(SetNr, Vpn)$ (siehe Kap 6.7.6 für weitere Erläuterungen).

Die meisten dieser Funktionen greifen auf weitere Routinen zur Verwaltung der Matrizen. `mn_get_qpos`, `mn_get_seti` und `mn_get_vp` extrahieren alle gewünschten Informationen aus der Matrix \hat{OS} bzw. S .

Die Darstellung aller Werte erfolgte mit `errorbar`, `bar`, `mn_scatter` bzw. `mn_pcolor`.

Schlüssel für verbale Beschreibungen

Da ein Teil der Ergebnisse in verbaler Form vorliegt (siehe Kap 6.7.6), wird ein Schlüssel verwendet um diese in numerische Werte umzuwandeln. Folgende Umrechnungsschlüssel wurden verwendet, wobei die idealen Werte kursiv und unterstrichen formatiert wurden:

Klangfarbe:

<i>Aussage</i>	<i>Bewertung</i>
keine Aussage	-
sehr wenig	-2
zu wenig	-1
<u><i>ausgewogen</i></u>	<u>0</u>
zu viel	+1
sehr viel	+2

Externität:

<i>Aussage</i>	<i>Bewertung</i>
keine Aussage	-
Im Kopf	0
Am Kopf	1
leicht Außerkopf, Radius von ca. 10cm	2

Außerkopf, Radius von ca. 30cm	3
<i>weit bzw. sehr <u>weit</u> Außerkopf, Radius >50cm</i>	<u>4</u>

Lokalisationsunschärfe:

<i>Aussage</i>	<i>Bewertung</i>
keine Aussage	-
Stark verschwommen	0
Verschwommen, Personenbreit	1
Leicht verschwommen	2
<u>Punktuell</u>	<u>3</u>

Konsistenz:

<i>Aussage</i>	<i>Bewertung</i>
keine Aussage	-
Sprunghaft	0
<u>Stetig</u>	<u>1</u>

Softwarebeschreibung

Für die Durchführung der Hörversuche wurde das Software-Paket aus dem Projekt VSPS⁷⁴ an die notwendigen Anforderungen adaptiert.

Am DSP waren bereits ein grundlegender Ambisonic-Algorithmus, die Anbindung des DSPs an die Benutzeroberfläche, sowie die Einbindung eines Headtrackers implementiert.

Folgende Funktionen mussten zusätzlich implementiert werden:

1. Direkte Auswahl der DSP-Programme mit verschiedenen Filterkoeffizienten. Es werden beim Programmstart alle binären DSP-Dateien⁷⁵ in eine Liste geladen und stehen dort zur Auswahl bereit.
2. Strukturierte Archivierung der Messdaten. Es werden Versuchsnummer, Setting, Nummer der Versuchsperson, Positionen der Schallquellen und die Bewegung des Kopfes in einer Datei gespeichert, wobei für jede Offset-Messung eine Datei erzeugt wird. Die

⁷⁴ Virtual Sound Positioning System, siehe [26]

⁷⁵ Dateien mit Endung .out

Schallquellenpositionen⁷⁶ der nacheinander folgenden Messungen werden automatisiert in eine Liste eingetragen. Nach einer vollständigen Messreihe können sie von dieser Liste in andere Anwendungen kopiert werden.

3. Kommunikation mit Versuchspersonen über eine grafische Benutzeroberfläche. Diese Anwendung wird im Zwei-Bildschirm-Modus betrieben. Der Versuchsleiter hat die Möglichkeit alle Parameter zu kontrollieren, sowie den Ablauf der Messung zu beeinflussen. Hinweise wie der Versuchsstatus (Training, Offsetmessung, usw.), der erlaubte Bereich der Kopfdrehung usw. werden der Versuchsperson am 2. Bildschirm angezeigt. Im Trainings-Modus wird zusätzlich die Position der virtuellen und realen Schallquelle eingeblendet. Während der verbalen Befragung werden die Fragen sowie die Antwortbereiche eingeblendet.
4. Automatisierung des Versuchablaufs. Während der Offset-Messung kann die Position der realen Schallquelle mit einem Maus-Klick eingegeben und die Messung gestartet werden. Dabei wird die Versuchsnummer automatisch erhöht.
5. Die DSP-Software wurde hinsichtlich der Ausgangslautstärke optimiert, d.h. um eine manuelle Eingabe des Skalierungsfaktors für die Berechnungen erweitert.

In der Abbildung C.4 erfolgt eine kurze Beschreibung der Benutzeroberfläche die dem Versuchsleiter zur Verfügung steht.

⁷⁶ *gegebene Soll-Position und befragte Ist-Position*

The screenshot shows the 'VSPS - Psychophysical Validation' software interface. It includes a central speaker position diagram, parameter input fields, a DSP program list, a history log, and a status window. Callouts provide detailed explanations for key features:

- Warnzeichen beim Verlassen des Bereiches der spontanen Kopfbewegung von einer Vpn:** A red triangle icon indicating a warning when leaving the spontaneous head movement area.
- Eintragung der vorgegebenen und der befragten Position:** Input fields for 'Phi' (7) and 'Soll' (-30), and a separate window showing 'Phi: 62' and 'Soll: 70'.
- Direkte Auswahl des DSP-Programmes:** A list of DSP programs such as '001_064.out', '001_096.out', etc., with '002_128.out' selected.
- Eingabe der Versuchsparameter und deren automatische Aktualisierung nach jedem Zyklus:** A parameter input window with fields for 'Exp. Setup' (02), 'Subject NR.' (05), and 'Exp. NR.' (07).
- Auswahl der Schallquellenposition:** A diagram showing speaker positions at various angles (0°, 10°, 20°, 30°, 40°, 50°, 60°, 70°).
- automatische Eintragung der Meßergebnisse nach Ablauf des Test-Zyklus:** A 'History' window displaying a table of results:

History:	
20	- 301-
30	471
50	301-
60	- 431
50	441
10	131
70	621

- Auswahl des Trainingsmodus und der Offset-Messung:** A 'Training' dropdown menu and a 'Status' window showing log messages like 'ABBRUCH: Der Testvorgang wurde abgebrochen!' and 'Connection to DSP started.'

Abb. C.4: Hauptfenster der PC-Software für Hörversuche

Anhang D – User Interface

Zur Vervollständigung der Dokumentation werden hier der Schalt-, Lageplan, das Layout der Frontplatte sowie die Stückliste des User-Interfaces angegeben.

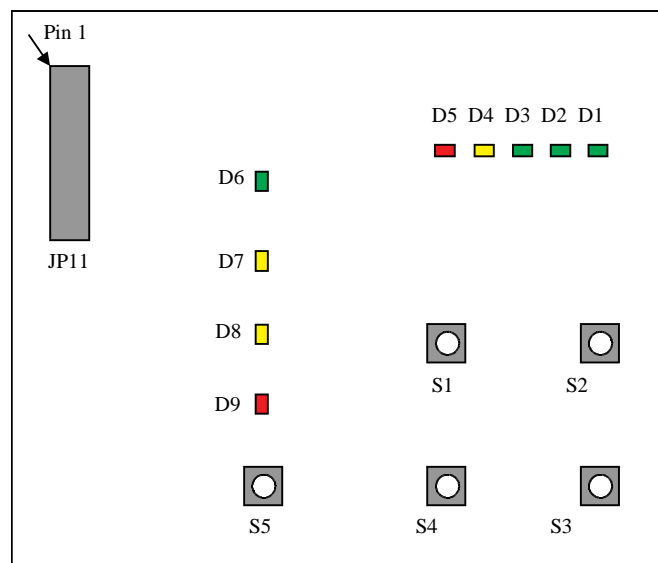


Abb.D.1: Lageplan der Bauteile im User-Interface

<i>Bauteil</i>	<i>Beschreibung</i>
D1, D2, D3, D6	LED SMD 1206 grün
D4, D7, D8	LED SMD 1208 gelb
D5, D9	LED SMD 1208 rot
R1 bis R9	680R, SMD 1208
R10 bis R14	10k, SMD 1208
S1, S2, S3, S4, S5	6mm Drucktaster, z.B.: Panasonic EVQ-QS205K
JP11	2x10 zweireihige Buchsenleiste
Befestigungsmaterial	2 Schrauben, Länge mind 40mm, mit je 2 Beilagscheiben und 2 passenden Mutter

Tabelle D.1: Bauteilenliste für das User-Interface

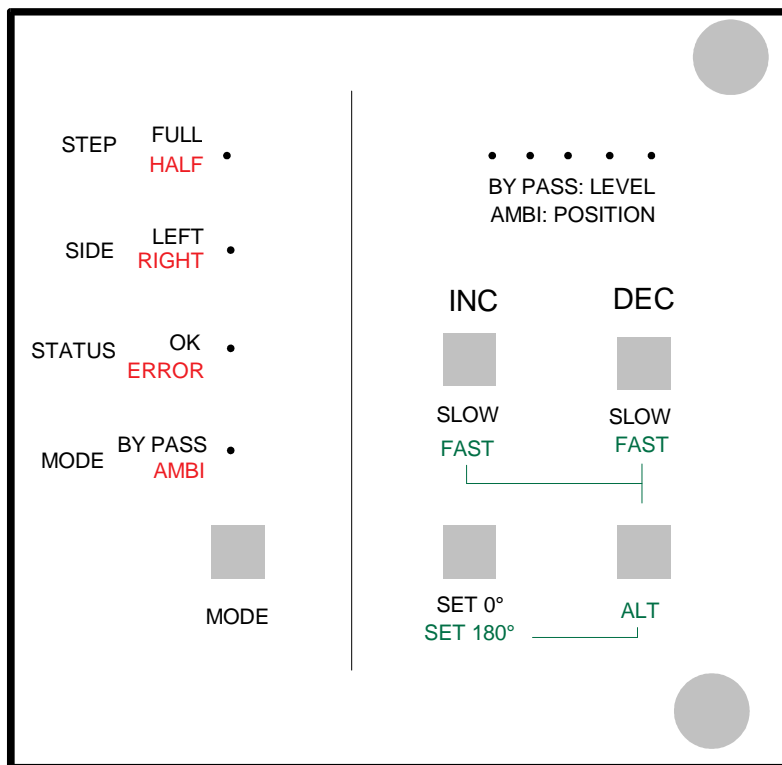


Abb. D.2: Frontplatte im Maßstab 1:1

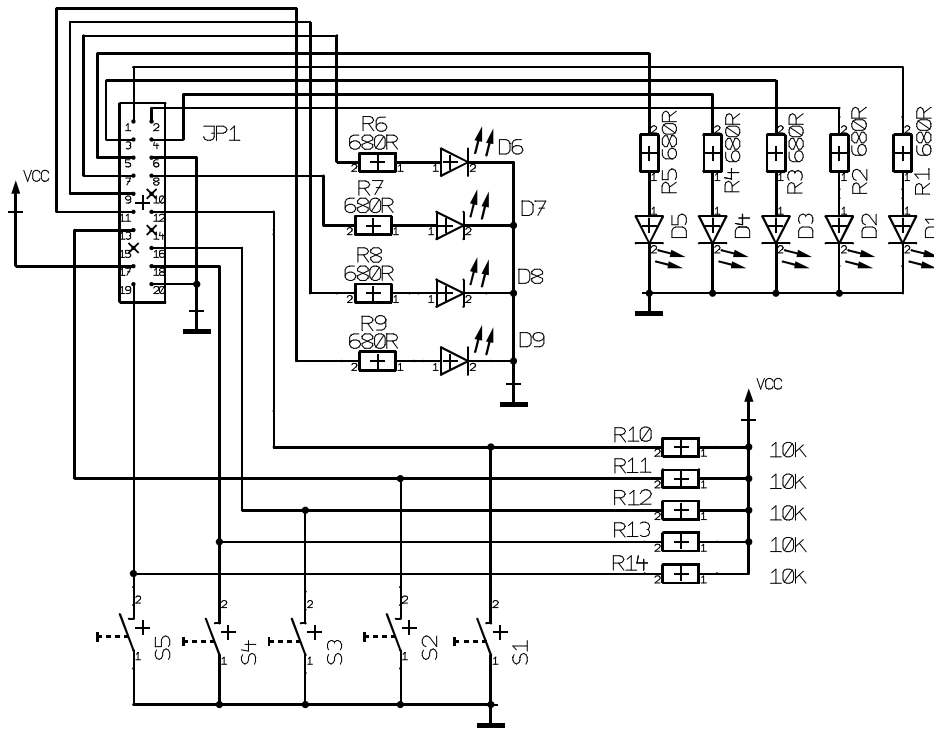


Abb. D.3: Schaltplan des User-Interfaces

Literaturverzeichnis

- [1]: Intel Corporation; Audio Codec 97; September 2000; <http://www.intel.com>
- [2]: AKG Acoustics; <http://www.akg-acoustics.com/>
- [3]: Bamford J.S.; An Analysis of Ambisonic Sound Systems of First and Second Order; 1995
- [4]: Begault D.R., Wenzel E.M.; Direct Comparison of the Impact of Head Tracking, Reverberation, and Individualized HRTFs on the Spatial Perception of a Virtual Speech Source; J. Audio Eng. Soc. Vol. 49, No.10; 2001
- [5]: Blauert J.; Räumliches Hören, Nachschrift; Hirzel Verlag, Stuttgart; 1985
- [6]: Blauert, Jens; Spatial Hearing 2nd ed.; MIT Press, Cambridge, MA; 1997
- [7]: Bortz, J.; Statistik; Springer Verlag; 1993
- [8]: Brandenstein H., Unbehauen R.; Least-Squares Approximation of FIR by IIR Digital Filters; IEEE Trans. on Sig. Proc. Vol. 46, No. 1, pp 21-30; 1992
- [9]: Braun, D.; Ein neues Hörversuchskonzept; DAGA 1996; 1996
- [10]: Burr-Brown Corporation; Single-Ended Analog Input 20-Bit Stereo, Analog-to-Digital Converter; August 1997; <http://www.burr-brown.com>
- [11]: Burr-Brown Corporation; 24-Bit, 96kHz Sampling, 6-Channel, Enhanced Multi-Level, Delta-Sigma Digital-to-Analog Converter; März 2000; <http://www.burr-brown.com>
- [12]: Carlile S., Delaney S., Corderoy A.; The localisation of spectrally restricted sounds by human listeners; Hear. Res. Vol. 128, pp 175-189; 1999
- [13]: Chen J., Van Veen B.D., Hecox K.E.; A spatial feature extraction and regularization model for HRTF; J. Acoust. Soc. Am. 97 (1), pp. 439 - 452; 1995
- [14]: Daniel J., Rault J.B., Polack J.D.; Ambisonics Encoding of Other Audio Formats for Multiple Listening Cond.; Audio Eng. Soc. 105th Conv., preprint 4795; 1998
- [15]: Daniel J.; Representation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia; 2000
- [16]: Fellner M.; Recheneffiziente Modellierung von Außenohrübertragungsfunktionen; 1998
- [17]: Gersho A., Gray R. M.; Vector Quantization and Signal Compression; Kluwer Academic Publishers ; 1992
- [18]: Gerzon M.A.; General Metatheory of Auditory Localisation; Audio Eng. Soc. 92nd Conv. Vienna, preprint 3306; 1992
- [19]: Gerzon M.A.; Decoders for HDTV; Audio Eng. Soc. ; 1992

- [20]: Gilkey R.H., Anderson T.R.; Binaural and spatial hearing in real and virtual environments; Lawrence Erlbaum Assoc., New Jersey ; 1997
- [21]: Griesinger D.; The Psychacoustics of Apparent Source Width, Spaciousness and Envelopment in Performance Spaces; *Acustica* Vol. 83, pp. 721 - 731; 1997
- [22]: PC Zone; <http://www.ping.be/~ping0751/i2czone.htm>
- [23]: Jot J.M., Larcher V., Warusfel O.; Digital Signal Processing Issues in the Context of Minaural and Transaural; *Audio. Eng. Soc. 98th Conv. Paris*, preprint 3980; 1995
- [24]: HRTF Measurements of a KEMAR Dummy-Head Microphone; <http://web.media.mit.edu/~kdm/hrtf.html>
- [25]: Kendall G.S.; A 3D sound primer: directional hearing and stereo reproduction; *Compute Music Journ.* Vol. 19, Nr. 4, pp. 23-46; 1995
- [26]: Piotr Majdak, Markus Noisternig; A Head Position Related Binaural Sound Reproduction System; Institut für Elektronische Musik und Akustik Version 1.0; 2001
- [27]: Marolt M.; A New Approach to HRTF Audio Spatialization; *ICMA Proc. ICMC 1997*, San Francisco, pp. 365-367; 1996
- [28]: The Mathworks; <http://www.mathworks.com>
- [29]: Mc Namara D.A., Pistoriua C.W.I., Malherbe J.A.G; Introduction to the Uniform Geometrical Theory of diffraction; Artech House ; 1990
- [30]: Motorola, Incorporated; Motorola DSP, Assembler Reference Manual; April 1998; <http://www.motorola-dsp.com>
- [31]: Motorola, Incorporated; DSP56366, 24-Bit Digital Signal Processor, User's Manual; September 2000; <http://www.motorola-dsp.com>
- [32]: Motorola, Incorporated; DSP56300 Family Manual; November 2000; <http://www.motorola-dsp.com>
- [33]: Motorola, Incorporated; DSP5636XEVM, User's Manual; Oktober 2001; <http://www.motorola-dsp.com>
- [34]: Oppenheim A.V., Schaffer R.W.; Digital signal processing; Englewood Cliffs, N.J.; 1975
- [35]: Poletti M.; The Design of Encoding Functions for Stereophonic and Polyphonic Sound Sys.; *J. Audio. Eng. Soc.* Vol. 44, No. 11; 1996
- [36]: Pulkki V., Karjalainen M.; Localization of Amplitude-Panned Virtual Sources I: Stereophonic Panning; *Audio Eng. Soc.* Vol. 49, No. 9; 2001
- [37]: Pulkki V.; Localization of Amplitude-Panned Virtual Sources II: 2D-3D Panning; *J. Audio Eng. Soc.* Vol. 49, No. 9; 2001

- [38]: Pure Data; <http://iem.kug.ac.at/pd/>
- [39]: Rane Corporation; Interfacing AES3 and S/P-DIF; August 2001; <http://www.rane.com>
- [40]: Rife D.D., Vanderkooy J.; Transfer Function Measurement with Maximum Length Sequences; J. Audio Eng. Soc. Vol. 37, No. 6; 1989
- [41]: Sarti A., Tubaro S.; Low-cost geometry-based acoustic rendering; DAFX Proc. of the COST G-6 Conference DAFX-01, Limerick; 2001
- [42]: Sontacchi A.; Neue Ansätze der Schallfeldreproduktion; 2002
- [43]: Desktop & Webtop, StarOffice 5.2 Software; <http://www.sun.com/products/staroffice/>
- [44]: Texas Instruments; TMS320C6000 Programmer's Guide; März 2000; <http://www.ti.com>
- [45]: Texas Instruments; Code Composer Studio, User's Guide; Februar 2000; <http://www.ti.com>
- [46]: Theile G., Felderhoff U., Mackensen P.; Stabilität der Lokalisation bei verfälschter Reproduktion verschiedener Merkmale der binauralen Signale; 20. Tonmeistertagung Karlsruhe pp. 229 - 237; 1998
- [47]: Theile G., Mackensen P., Reichenauer K.; Einfluss der spontanen Kopfdrehung auf d. Lokalisation b. Binauralen Hören; 20. Tonmeistertagung, Karlsruhe pp. 218 - 228; 1998
- [48]: Tsingos N., Funkhouser T., Ngan A., Carlbom I.; Modeling Acoustics in Virtual Environments using the UDT; Princeton www.cs.princeton.edu/~funk/sig01.pdf;
- [49]: Vanderkooy J., Lipshitz S.P.; Anomalies of Wavefront Reconstruction in Stereo and Surround Sound Repro...; J. Audio. Eng. Soc. Vol. 35, p. 1025, preprint 2554, 83rd Convention; 1987
- [50]: Wightman F.L., Kistler D.J.; Resolution of front-back ambiguity in spatial hearing by listener; J. Acoust. Soc. Am. 105 (5); 1999
- [51]: Wu Z., Chan F.H.Y., Lam F.K., Chan J.C.K; A time domain binaural model based on spatial feature extraction for HRTFs; J. Acoust. Soc. Am. 102 (4), pp. 2211-2218; 1997
- [52]: Ziomak L. J.; Fundamentals of Acoustic Field Theory and Space-Time Signal Processing; CRC Press, Inc.; 1995