Entwurf und Implementierung einer Mehrkanal-Beschallungsanlage

johannes m zmölnig

Diplomarbeit
an der *Erzherzog-Johann-Universität Graz*
und der *Kunstuniversität Graz*

durchgeführt am
**institut für elektronische musik und akustik,
kunstuniversität graz**

unter der Betreuung von *ao.Univ.Prof.DI. Winfried RITSCH*
beurteilt von *o.Univ.Prof.Mag.DI.Dr. Robert R. HÖLDRICH*

in Zusammenarbeit mit der

fakultät medien, bauhausuniversität weimar

Zusammenfassung

Ziel der Diplomarbeit ist es, eine Beschallungsanlage zur Wiedergabe von periphonen Schallfeldern zu schaffen.

Nach der Spezifizierung der Hardwarekomponenten erfolgt eine Software-Implementierung eines zur Reproduktion von dreidimensionalen Schallfeldern geeigneten Codecs, der vor allem für den "live"-Einsatz geeignet ist.

Es werden dabei Funktionen zur Verfügung gestellt, die es erlauben, herkömmliche Mehrkanalaufnahmen über "Virtuelle Lautsprecher" wiederzugeben, vorcodierte ambisonic-Aufnahmen abzuspielen, sowie bewegte diskrete Quellen in Echtzeit in den Zuhörerraum abzubilden.

Das Ergebnis sollte so modular und skalierbar sein, dass es sowohl für den Einsatz in "dedicated environments" (IEM-Cube) als auch in kleineren Umgebungen (temporäre Klanginstallationen, Entwicklungsumgebungen, "Heimgebrauch") zufriedenstellende Ergebnisse liefert.

Abstract

In this thesis the design and implementation of a multichannel audio-system for reproduction of periphonic soundfields is discussed.

After a specification of hardwarecomponents, a software implementation of a codec suitable for creation and recreation of threedimensional soundfields in “live”-environments has been written.

Functions are provided to play back traditional multi-channel recordings via “virtual loudspeakers”, to decode pre-encoded ambisonic-soundfields and to render moving discrete sources into the audience in real-time.

The presented system is both modular and scalable, to fulfill the needs of “dedicated environments” (IEM-Cube) as well as unforeseeable situations (temporary sound-installations, developing-environments, “home-use”) in a satisfactory way.

Notiz und Danksagung und Widmung

Dolby, *Dolby Stereo* und *Dolby Surround* sind eingetragene Warenzeichen der *Dolby Laboratories Licensing Company. Windows* ist eingetragenes Markenzeichen der Firma *Microsoft. MacOS* ist eingetragenes Markenzeichen der Firma *Apple. PostScript* ist eingetragenes Warenzeichen von *Adobe Systems Inc.*

Diese Diplomarbeit wurde zu 99% auf Basis von freier Software entwickelt. Mathematische Modelle und Simulationen wurden in *Octave* berechnet, die Implementierung erfolgte unter *pure data*. Visualisierungen wurden mit *Blender* und *Gem* realisiert. Der Text dieser Diplomarbeit wurde im *Xemacs* getippt und mit LaTeXgesetzt.

Danken möchte ich an dieser Stelle meinem Betreuer Winfried Ritsch, der mir die Idee der freien Software näher gebracht hat und bei dem ich bei Problemen hard- und vor allem softwaretechnischer Natur immer auf Beistand zählen konnte. Danken möchte ich auch Prof. Robert Höldrich, der durch den nötigen Druck die Fertigstellung dieser Diplomarbeit erst ermöglichte.

Weiters gedankt sei natürliche meinen Eltern, ohne die ich ja gar nicht hier wäre, sowie meinem Bruder Martin. Und Wolfi.

Gewidmet ist diese Arbeit aber meiner Lebensgefährtin Nicki, die in Glück und Not und Verzweiflung zu mir stand (und steht).

Inhaltsverzeichnis

Kapitel 1

Einführung

1.1 Virtuelle Realitäten

Der Technologieboom des letzten Jahrhunderts und im Besonderen der letzten zwanzig Jahre brachte der Menschheit der westlichen Hemisphäre in zunehmendem Maße eine “virtuelle Realität”.

Im Gegensatz zu traditionellen Pseudorealitäten, die sich erst individuell bei der neuronalen Reizverarbeitung bilden und dementsprechend schlecht kontrollierbar sind, basieren *virtuelle* Realitäten auf der reproduzierbaren und zielgerichteten Beeinflussung der Sinnesorgane, sodass es diesen nicht mehr möglich ist, einen künstlichen Reiz von seinem Pendant natürlichen Ursprungs zu unterscheiden. Die Schaffung künstlicher Welten passiert also nicht mehr individuell “im Kopf” sondern wird auf eine technische Ebene verlagert, die es den Rezipienten erlaubt, ein und dieselbe künstliche Umgebung als Realität individuell wahrzunehmen.

Um eine solche *virtuelle Realität* zu schaffen, ist zuvor eine genaue **Analyse** der tatsächlichen Realität von Nöten, um (beliebige) Wahrnehmungen eindeutig beschreiben zu können.

Eine Haupteigenschaft von *virtuellen Realitäten* ist, dass sie in einem – dem Analyseprozess reziproken – **Synthese**prozess durch allgemeine Projektoren in die “echte” Realität projiziert werden. Diese Projektoren sind (im

Allgemeinen) auf eine Klasse von Sensorien (Visus, Auditus,...) beschränkt, innerhalb dieser Klasse können sie aber (idealerweise) jede beliebige Wahrnehmung hervorrufen.[1] Im Gegensatz dazu, ermöglichen traditionelle Wiedergabeeinheiten keine allgemeine Produktion von *beliebigen* Wahrnehmungen. So kann zum Beispiel ein Musikinstrument zwar Klänge emittieren, allerdings sind diese Klänge sehr instrumentenspezifisch.

Eine weitere Haupteigenschaft neben der wahlfreien Produzierbarkeit ist die arbiträre *Reproduzierbarkeit*. Dies bedeutet, dass eine durch Projektoren hervorgerufene Wahrnehmung durch gleichartige Projektoren an beliebigen Orten zu beliebigen Zeiten wiederholt werden kann. Ein essentieller Bestandteil von Systemen zur Schaffung von virtuellen Realitäten ist daher die Möglichkeit, die abzubildende Wahrnehmung genau und eindeutig festzulegen und in weiterer Folge zu speichern.

Aus praktischen Gründen werden solche Projektionssysteme meist auf ein Sensorium beschränkt. Da in der "natürlichen" Umgebung selten Sinneseindrücke isoliert wahrgenommen werden, sondern meist im Verbund auftreten, werden oft mehrere solcher Systeme für getrennte Sensorien parallel verwendet, um komplexe mehrdimensionale (also sich über mehrere Sensorien erstreckende) *virtuelle Realitäten* zu projizieren.

In diesem Kontext wird in den letzten Jahren gerne das Schlagwort "Multimedia" gebraucht. Die "multiplen Medien" beschränken sich dabei zumeist auf Bild und Ton. Andere Ansätze in den 80er Jahren – wie die olfaktorische *SmellCard* als Zusatzmodul für den Heimcomputer – konnten sich – im Gegensatz zur *SoundCard* – nie am Markt etablieren.[2]

In den letzten Jahren haben vor allem die visuellen Medien von der rasanten Entwicklung am Halbleitersektor profitiert. Dies liegt zum einen daran, dass die hohe Komplexität und die großen auftretenden Datenmengen erst durch die Entwicklung von schneller Hardware eine effektive Generierung

[1] z.B. kann ein Videoprojektor jede beliebige Art von Bildern wiedergeben und ist nicht auf Typen wie Landschaftsaufnahmen beschränkt.

[2] Wohl deshalb, weil der verbreitete Geruch im Gegensatz zum verbreiteten Klang nicht sofort nach Abschalten des Rechners verschwand.

und Bearbeitung von visuellen Eindrücken erlaubte. Andererseits konnten die so gewonnen Erkenntnisse von der Filmindustrie sehr wirksam einem großen Publikum vorgeführt werden, wodurch wiederum viel in die weitere Verbesserung der Systeme investiert werden konnte. Vorläufige Höhepunkte dieser Entwicklung sind mit den ersten "100%-all-digital" Spielfilmen wie "Toy-Story" (USA 1996) sowie der photorealistischen Generierung von Menschen in Großaufnahmen ("Final Fantasy", USA 2001) erreicht.

Eine parallele Entwicklung ist auch im Homecomputer-Bereich auszumachen, wo in interaktiven Computerspielen immer realistischere visuelle Eindrücke erzeugt werden. Durch die so geschürte hohe Erwartungshaltung der Konsumentinnen steigt auch deren Anspruch auf eine adäquate Umsetzung in die akustische Ebene.

1.2 Historischer Überblick

Die Geschichte der Aufzeichnung und Wiedergabe von Klängen beginnt in der zweiten Hälfte des 19. Jahrhunderts mit dem (nie realisierten) "Pa(r)léophon" von Charles Cros (1877) [21]. Später im selben Jahr konstruiert Thomas Edison den "Phonographen" [42], das erste funktionsfähige System zur Aufzeichnung und der beliebigen Reproduktion von beliebigen Schallschwingungen. Diese wurden über eine Schallmembran auf einen Schreibstichel übertragen, der sie auf eine Zinnfolie in Tiefenschrift schrieb. Mit einer Wiedergabenadel und einem Schalltrichter konnten sie wiederholt wiedergegeben werden.

Die rein mechanischen Schallwandler wurden in den 1920er Jahren ersetzt durch elektrische Aufnahme- und Wiedergabesysteme wie Kondensatormikrophon und elektrodynamische Lautsprecher. Prinzipiell haben sich diese Schallwandler seit damals nicht mehr verändert, obwohl natürlich Qualitätsoptimierung stattgefunden hat.

Waren die ersten Systeme noch monophon, so wurden ab Anfang der 1930er Jahre mehrkanalige Systeme entwickelt. Deren Ziel war es, der zeitlichen Komponente des Schalles eine räumliche hinzuzufügen. Die ersten Stereoaufnahmen wurden 1931/32 von Alan Blumlein in Großbritannien durchgeführt. Etwa zeitgleich wurde in den Vereinigten Staaten von Amerika ein sehr ähnliches System von den *Bell Laboratories* entwickelt.

Blumlein als Ingenieur der BBC war in erster Linie an der Entwicklung eines idealen Wiedergabesystems für den "Heimgebrauch" interessiert. Es war daher möglich, das Wiedergabesystem für relativ kleine Beschallungsanlagen, bei denen sich ein zahlenmäßig geringes Publikum (im Extremfall ein einzelner Hörer) an einer gut definierten "idealen" Abhörposition oder um diese herum befindet, zu optimieren. Die so entstandenen Wiedergabesysteme liefern daher am "sweet spot" (der idealen Abhörposition) ein sehr natürliches Ergebnis, allerdings zerfällt das Stereobild sehr schnell, wenn man sich vom *sweet spot* entfernt.

Dagegen war die Entwicklung in den USA von Anfang an vom Medium Film dominiert [1]. Hier war es notwendig, einem vielköpfigen Publikum, dass sich notgedrungenermassen zu großen Teilen weit von der idealen Abhörposition befand, ein passables Klangbild zu liefern. Durch die Verwendung von mehreren Wiedergabekanälen wird der *sweet spot* zu einer relativ großen *sweet area* ausgedehnt. Allerdings ist die Abbildung innerhalb dieses Gebietes nirgends so gut, wie in Blumleins kleinem *sweet spot.*

Da ein Stereo-Wiedergabesystem prinzipiell aus 2 Lautsprechern vor dem Publikum besteht, können mit einer solchen Konfiguration keine virtuellen Schallquellen von der Seite oder von hinten wiedergegeben werden. In den frühen 1970er Jahren wurde daher versucht, das Stereo-Konzept zur *Quadrophonie*, wobei zwei zusätzliche Lautsprecher hinter oder neben dem Publikum angeordnet wurden, zu erweitern. Da man sich lange nicht auf einen Standard, der kompatibel zum bestehenden Stereo-Format war, einigen konnte und aufgrund einer verfehlten Verkaufspolitik wurde Quadrophonie jedoch zu einem marktwirtschaftlichen Desaster.

Wenige Jahre später wurde mit *ambisonic* ein anderes mehrkanaliges System von Gerzon vorgeschlagen. Obwohl dem Quadrophonie-System überlegen, wurde dieses System vom Markt, der den Vorläufer noch nicht vergessen hatte, ebenfalls nicht angenommen.

Auch die *Dolby Laboratories* entwickelten in den 1970er Jahren ein Mehrkanalsystem für den Heimgebrauch [1]. Als Weiterentwicklung des Kinostandards für Surround-Sound *Dolby Stereo* konnte sich *Dolby Surround* vor allem in Verbindung mit dem Homevideo-Markt durchsetzen. Eine Weiterentwicklung von *Dolby Surround* ist *Dolby pro logic*, das sich als Nachfolgemodell eines erfolgreichen Produktes ebenfalls am Markt durchsetzen konnte.

1.3 Räumliches Hören

Um akustische Wiedergabesysteme auf ihre Sinnhaftigkeit in Bezug auf die räumliche Abbildung beurteilen zu können, ist es notwendig, sich der Frage zu stellen, wie der menschliche Wahrnehmungsapparat von der Beobachtung eines Schallfeldes durch das Hörorgan Rückschlüsse auf die räumliche Verteilung der dieses Schallfeld emittierenden Schallquellen zieht.

1.3.1 Richtungswahrnehmung

Das menschliche Gehör evaluiert die Richtung einer Schallquelle auf mehrere, einander ergänzende Arten [31]:

Interaurale Pegeldifferenzen

Zwischen den beiden Ohren befindet sich der Schädel, der idealisiert als Kugel mit einem Durchmesser von etwa 20 Centimetern angenommen werden kann. Dadurch werden diejenigen Schallanteile, deren Wellenlänge $\lambda = \frac{c}{f}$ im Verhältnis zur Kopfgröße klein ist, abgeschattet. Durch diesen Abschattungseffekt ist die Schallintensität auf der der Quelle zugewandten Kopfseite höher als die auf der der Quelle abgewandten Seite.

Interaurale Phasendifferenzen

Tiefe Frequenzen werden um den Kopf herum gebeugt, weshalb es zu keiner wesentlichen Pegeldifferenz zwischen linkem und rechtem Ohr kommt. Da das der Quelle zugewandte Ohr näher bei der Schallquelle ist als das abgewandte Ohr, kommt es durch die endliche Schallgeschwindigkeit zu Laufzeitverschiebungen. Bei hohen Frequenzen, bei denen $\lambda < d_{\mathrm{K}opf}$ ist, liefert diese Methode jedoch keine eindeutigen Ergebnisse mehr. Sie ist daher nur für die Ortung von tieffrequenten Schallanteilen geeignet.

Außenohr-Übertragungsfunktion

Durch die komplexe unsymmetrische Form des Außenohrs, werden Quellen aus verschiedenen Richtungen unterschiedlich an das Innenohr weitergeleitet. Diese kopfbezogenen Außenohr-Übertragungsfunktionen (*HRTFs - Head Related Transfer Functions*) färben also das Spektrum des Schallsignales ganz spezifisch in Abhängigkeit von seiner Richtung. Da die Form des Außenohres (inklusive der Kopf- und Rumpfform) von Mensch zu Mensch verschieden ist, sind auch die Außenohr-Übertragungsfunktionen höchst individuell.

Kopfdrehung

Nur durch interaurale Phasen- und Pegeldifferenzen kann keine eindeutige Zuordnung zu einer Richtung vorgenommen werden. Die Schallquelle kann demnach aus einer begrenzten Menge von Richtungen abstrahlen (*cone of confusion*, siehe dazu Abb. 1.1).

Um diese Uneindeutigkeit zu überwinden, werden nach der Drehtheorie Kopfdrehungen durchgeführt. Eine an sich statische Schallquelle ändert durch diese Peilbewegung ihre relative Position zur Hörerin und liegt nun innerhalb eines anderen *cone of confusion*. Durch Überlagerung mehrerer *cones* kann dann die tatsächliche Abstrahlrichtung relativ einfach gefunden werden.

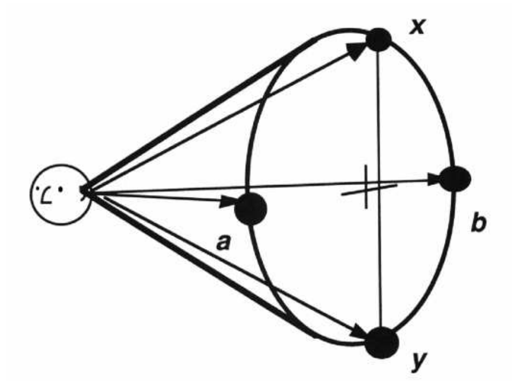


Abbildung 1.1: cone of confusion, nach [31]

Lokalisationsunschärfe

Die Lokalisationsunschärfe ist "...die kleinste Änderung eines bestimmten Merkmals oder bestimmter Merkmale des Schallereignisses oder eines anderen, mit dem Hörereignis korrelierten Ereignisses, die gerade zu einer Ortsänderung des Hörereignisses (z.B. bezüglich Richtung und/oder Entfernung) führt." [6]

Die Fähigkeit zur Lokalisierung ist nicht über das gesamte Hörfeld konstant. Im "Vorne"-Bereich ist es möglich, die Richtung bis auf 1° genau zu bestimmen. Die Unschärfe nimmt jedoch zu, je weiter man sich von der "vorne"-Richtung entfernt. "Oben" oder "seitlich" liegt die JND[3] um 10°.

1.3.2 Entfernungshören

Unabhängig von der Richtung, aus der eine Quelle den Schall abstrahlt, ist es möglich, eine Entfernungsbewertung durchzuführen.

Im (reflexions-)freien Schallfeld ändert sich der Schallpegel in Abhängigkeit von der Entfernung, und zwar nimmt er mit 6dB pro Entfernungsverdopplung ab.

Ist die Quelle "weit" (mehr als 15 Meter) entfernt, so wird zusätzlich

[3] *Just Notable Difference*: kleinster noch erkennbarer Unterschied, auch Schwelle

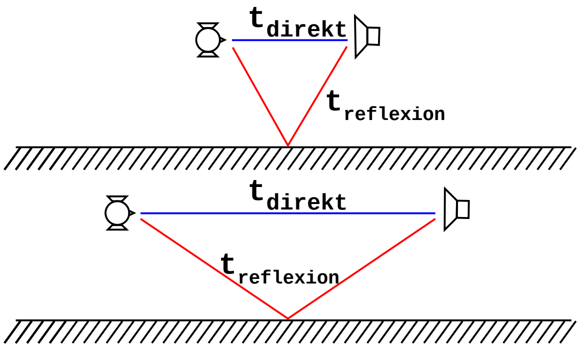


Abbildung 1.2: Entfernungsbewertung durch Erste Reflexion

noch die Dämpfungswirkung der Luft hörbar. Da hohe Frequenzen stärker gedämpft werden als tiefe, tritt eine, zur Entfernung proportionale, Klangfarbenänderung auf.

Ist die Quelle “nahe” (weniger als 3 Meter entfernt), so treten durch die Krümmung des Schallfeldes am Kopf und der Pinna weitere spektrale Färbungen auf, die allerdings durch die unsymmetrische Form des Kopfes und des Außenohres richtungsabhängig sind.

In einem reflexionsbehafteten Schallfeld entsteht außerdem noch eine Laufzeitdifferenz zwischen dem Direktschall und den ersten Reflexionen. Je näher die Quelle ist, desto später kommen die ersten Reflexionen ans Ohr (siehe Abb. 1.2)

In diesem Zusammenhang sei noch der Haas-Effekt erwähnt, der besagt, dass für die Ortung nur die erste Wellenfront eines Ereignisses ausschlaggebend ist. Danach eintreffende Wellenzüge (erste Reflexionen), können bis zu 10dB lauter sein, ohne die Lokalisation zu stören. (Gesetz der ersten Wellenfront)

1.3.3 Bewegungshören

Sehr langsame Bewegungen von Schallquellen werden nicht als solche wahrgenommen. Stattdessen wird eine solche sich langsam bewegende Schallquelle als stationär angenommen und ihre Position vom Gehör mit den obengenann-

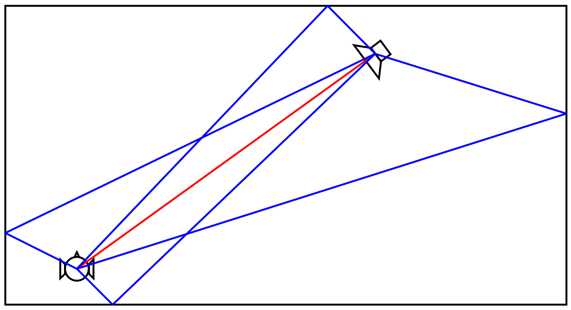


Abbildung 1.3: Erste Reflexionen

ten Methoden bestimmt. Die Bewegung entspricht dann der Positionsänderung der Quelle von einer Lokalisation zur anderen.

Bewegen sich Schallquelle und Kopf allerdings schnell zueinander, so tritt aus der Überlagerung von Relativgeschwindigkeit und der “normalen” Schallgeschwindigkeit der Doppler-Effekt auf. Bewegt sich die Quelle auf den Kopf zu, so treffen pro Zeiteinheit mehr Wellenzüge beim Ohr ein, wodurch die Frequenz steigt. Entfernt sich die Quelle, so sinkt ihre Frequenz.

1.3.4 Raumwahrnehmung

Im realen Raum wird der Schall an vielen Oberflächen reflektiert. Dadurch entsteht in Abhängigkeit von der Raumkonfiguration ein diffuses Schallfeld. Durch die Versatzzeiten der ersten Reflexionen ist es möglich, einen ungefähren Eindruck von der Größe eines Raumes zu erhalten.

Der Diffushall vermittelt einen Eindruck der Oberflächenstruktur des Raumes.[4]

[4]Die in diesem Zusammenhang vielzitierte Gotische Kathedrale etwa erlaubt durch die schallharten Wände ein dichtes lang anhaltendes Diffusfeld.

1.4 Wiedergabeansätze

Ein akustisches Wiedergabesystem hat die Aufgabe, im Publikum eine beliebige, gewünschte Wahrnehmung eines Schallfeldes zu evozieren. Im Normalfall wird das gewünschte "virtuelle" einem realen Schallfeld entsprechen, das aus irgendeinem Grund dem Auditorium nicht direkt zugänglich ist.

Prinzipiell ist es daher für ein solches Wiedergabesystem notwendig, eine genau Kopie des originalen Schallfeldes zu reproduzieren. Da der menschlichen Wahrnehmung Grenzen gesetzt sind, kann man diese Forderung einschränken auf die Reproduktion des für die erwünschte Wahrnehmung notwendigen Teiles des Schallfeldes. Es stellt sich dann die Frage, welche Eigenschaften des Schallfeldes für den gewünschten Höreindruck notwendig sind.

Im Allgemeinen lassen sich folgende wichtigen Eigenschaften festhalten:

- *zeitlicher Eindruck*: die "Klangqualität" des wiedergegebenen Schalles in Bezug auf Klangfarbe, Bandbreite, Dynamik,...

- *räumlicher Eindruck*: Es soll einerseits eine virtuelle Schallquelle des reproduzierten Schallfeldes korrekt lokalisiert werden, andererseits soll auch der gewünschte Raumeindruck (Raumgröße, ...) entstehen

- *Größe der* sweet area: Während es für Wiedergabesysteme für den "Heimgebrauch" genügt, wenn einer einzelnen Zuhörerin ein befriedigender Eindruck des Originalschallfeldes übermittelt wird, soll ein allgemeines System einer größeren Zuhörerschaft ein homogenes, überall gleich oder ähnlich gutes Hörerlebnis bieten.

Das Kriterium des *zeitlichen Eigenschaften* wird im Allgemeinen durch Verwendung von linearen Modulen und genügend großer Bandbreite / Dynamik ohne weiteres erfüllt (siehe Cap. 3)

Das räumliche Kriterium sowie die Größe der optimalen Abhörposition sind daher maßgebend für die Auswahl eines geeigneten Beschallungssystemes.

Zwar kein Kriterium bei der Reproduktion eines Schallfeldes aber dennoch von großer praktischer Bedeutung ist der Aufwand, der nötig ist, um mit einem bestimmten Beschallungssystem ein gewünschtes Schallfeld wiederzugeben. Dies ist vor allem im Hinblick auf eine mobile Beschallungseinheit wichtig.

1.4.1 Binauraltechnik

Die gesamte räumliche Information eines Schallfelder wird dem Gehör über die beiden Ohren übermittelt.

Binaurale Abbildungen versuchen daher das Schallfeld an den Trommelfellen der Zuhörerschaft zu resynthetisieren.

Da man nur das Schallfeld direkt am Ohr zu resynthetisieren braucht, genügen zwei Kanäle um eine exakte Kopie des Schallfeldes zu reproduzieren.

Durch Verwendung von Kopfhörern kann man den Einfluß des Wiedergaberaums minimieren.

Sollen reale Schallfelder wiedergegeben werden, so genügt es, mit zwei Mikrophonen das Schallfeld direkt in den beiden Gehörgängen aufzunehmen (Kunstkopftechnologie).

Werden binaurale Abbildungen künstlich erzeugt, so wird auf jede virtuelle Quelle die zu ihrer Position gehörige HRTF appliziert. Entsprechen die verwendeten HRTF-Kurven denen des Rezipienten, so entsteht ein sehr plastisches Bild.

Da allerdings die HRTF-Kurven stark individuell sind und aus praktischen Gründen oft nicht für jede einzelne Hörerin eine eigene Codierung erfolgen kann, liefert dieses System für die meisten Rezipientinnen den Eindruck des "Hörens mit fremden Ohren", wobei der gute Lokalisationseffekt wieder zunichte gemacht wird.

Bei der herkömmlichen Binaural-Technik wird zumeist keine Rücksicht auf die Peilbewegungen des Kopfes genommen. Das räumliche Abbild dreht sich also mit dem Kopf. Dies entspricht nicht der natürlichen Erfahrung.

Es ist nur schwer möglich, ein größeres Publikum gleichzeitig mit Binaur-

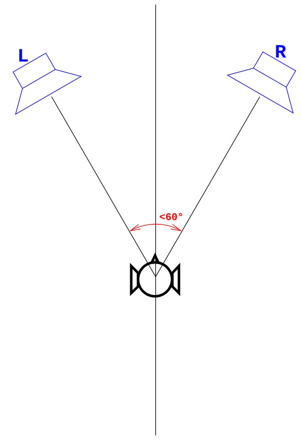


Abbildung 1.4: Stereophonie

altechnik zu beschallen. Die meisten Menschen empfinden es zudem störend, über Kopfhörer beschallt zu werden, vor allem im Rahmen eines öffentlichen Konzertes.

1.4.2 Stereophonie

Die Binauraltechnik ist ein Sonderfall der Stereophonie. (Lautsprecher-)Stereophonie besteht im Allgemeinen aus einem Lautsprecherpaar, das symmtrisch links und rechts vor der Hörerin positioniert ist. Mit Hilfe von Pegel- und auch Laufzeitunterschieden (siehe 1.3) ist es möglich, eine Phantomschallquelle zwischen den beiden Lautsprechern zu positionieren.

Da sich die beiden Lautsprecher "vorne" befinden, ist es allerdings nicht möglich, virtuelle Schallquellen "von hinten" abzubilden. Es ist nicht einmal möglich, eine virtuelle Quelle ausserhalb der relativ eng beieinanderstehenden Lautsprecher zu positioniern.

Andererseits ist Stereophonie sehr wohl geeignet, die akustischen Ereignisse zum Beispiel einer Bühne zu reproduzieren, da sich hier alle Quellen an einem relativ kleinen, frontal gelegenen Raum "zwischen den Lautsprechern" gruppieren.

Ein großer Nachteil der Stereophonie ist, dass die Rezipientin von bei-

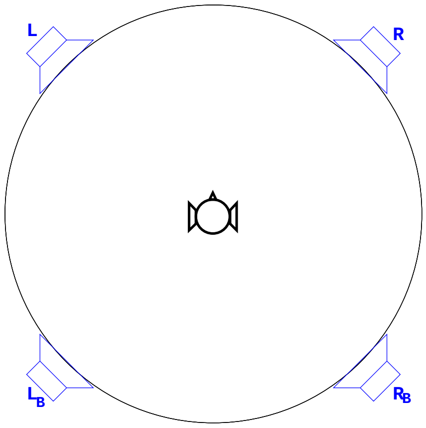


Abbildung 1.5: Quadrophonie

den Lautsprechern gleich weit entfernt sein muss, da anonsten die Phantomschallquelle scheinbar in den der Hörerin näheren Lautsprecher hineingezogen wird. Die ideale Abhörposition befindet sich also auf einer schmalen Symmetrieachse zwischen den Lautsprechern. Dieser kleine *sweet spot* macht die Stereowiedergabe eigentlich ungeeignet für ein mehrköpfiges Publikum.

1.4.3 Quadrophonie

Durch Erweiterung eines Stereo-Wiedergabesystems auf mehr als zwei Lautsprecher ist es prinzipiell möglich, den Schall auch von der Seite oder von hinten zu reproduzieren. In den 1970er Jahren wurde daher der Versuch unternommen, das Stereosystem auf vier Lautsprecher auszuweiten, wodurch man *Quadrophonie* erhielt.

Da Quadrophonie immer als Stereo-Erweiterung gedacht wurde, war man darauf bedacht, zum damals bereits etablierten Stereo-Format voll kompatibel zu bleiben. Es sollte daher möglich sein, eine herkömmliche Stereo-Aufnahme auf einem Quadrophonie-System abspielen zu können (und dabei idealerweise eine Verbesserung des räumlichen Eindrucks wahrzunehmen) aber auch neue quadrophone Aufnahmen über eine (gewöhnliche) Stereo-Anlage ohne Qualitätsverlust (gegenüber herkömmlichen Stereo-Aufnahmen) wiederzugeben.

Unter dieser Prämisse war es notwendig, die vier diskreten Quadrophonie-Signale in zwei stereo-kompatible Übertragungskanäle zu codieren und bei der Wiedergabe zu decodieren.

Aus informationstechnischen Gründen ist es aber nicht möglich vier vollwertige (also mit "gewohnter" Bandbreite und Dynamik) Kanäle verlustfrei in nur zwei Übertragungskanäle (die aus historischen Gründen nur eine beschränkte – und eben die war das Auditorium ja "gewohnt" – Bandbreite bzw. Dynamik aufweisen) zu kodieren.

Aus diesem Grund, und auch weil sich die Entwickler nicht auf notwendige Standards und Ziele einigen konnten, konnte sich das Quadrophoniesystem (oder besser: die Menge der Quadrophoniesysteme) nie durchsetzen.

1.4.4 Dolby-Surround

Das proprietäre *dolby-surround*-Format ist ein Sonderfall der Quadrophonie für die Wiedergabe von Kino- und TV-Ton.

Als solches ist es nicht notwendig, ein Schallfeld akkurat nachzubilden, da sich das Zentrum der Handlung durch die Beschränkungen der visuellen Projektion (Kino-Leinwand, Fernseh-Bildschirm) notwendigerweise immer vorne befindet.

Die Ziele von *dolby-surround* sind daher

- Stabilisierung der zentralen Phantomschallquelle (sprechende Schauspieler) im Zentrum der visuellen Projektion für eine relativ große *sweet area*

- nicht-lokalisierbare Spezialeffekte, die dem Zuseher das Gefühl geben, mitten im Handlungsraum zu sitzen

Dementsprechend gibt es neben dem Stereolautsprecherpaar einen Center-Kanal vorne in der Mitte sowie zwei Surround-Lautsprecher, die in der einfachsten Form von *dolby-surround* von einem gemeinsamen Surround-Kanal gespeist werden.

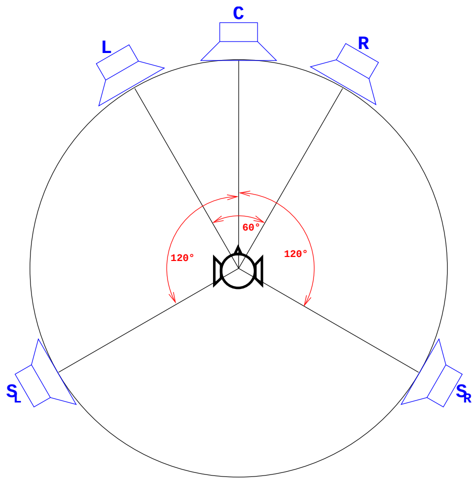


Abbildung 1.6: Dolby Surround

Die vier Kanäle werden für die Übertragung durch Matrizierung in ein zweikanaliges Übertragungsformat überführt [42]:

$$\begin{bmatrix} L_t \\ R_t \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{\sqrt{2}} & 0 & -\frac{j}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 1 & \frac{j}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} L \\ C \\ R \\ S \end{bmatrix} \tag{1.1}$$

Aus diesen können sie durch Matrizierung auch wieder zurückgewonnen werden:

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 1 \\ \frac{j}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \\ 1 & \frac{j}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} L_t \\ R_t \end{bmatrix} = \begin{bmatrix} L' \\ C' \\ R' \\ S' \end{bmatrix} \tag{1.2}$$

Durch die Codierung auf zwei Kanäle und die anschließende Dekodierung können die ursprünglichen Kanäle nicht mehr völlig getrennt werden. Das Übersprechen zwischen benachbarten Kanälen (-3dB), wurde daher durch das *dolby pro logic*-Verfahren mit einen aktiven DSP-Decoder verringert.

Dabei wird zusätzlich eine Separation in zwei Surroundkanäle sowie eine eigener Subwoofer-Kanal eingeführt.

Obwohl das *dolby-surround*-System nicht geeignet ist, einen beliebigen Raumeindruck zu schaffen, hat es in der Praxis durch die starke Verbreitung des Mediums Video eine große Bedeutung.

1.4.5 Holophone Techniken

Im Gegensatz zu *dolby surround* und *Quadrophonie*, die beide im Prinzip nur eine Erweiterung des Stereo-Systems darstellen, stellt sich die holophone Technik der Anforderung, ein komplexes Schallfeld bei der Wiedergabe mit Hilfe eines vielkanaligen Lautsprecherarrays so genau wie möglich zu resynthetisieren.

Prinzipiell können drei Untergruppen des Holophonie-Konzeptes unterschieden werden:

- echte Holophonie
- Wellenfeldsynthese (WFS)
- Ambisonic

Holophonie

Das Huygens'sche Prinzip besagt [7]:

> Jedes Element einer Wellenfront kann als Zentrum einer Störung betrachtet werden, die Ausgangspunkt einer sekundären kugelförmigen Wellenfront ist.

Fresnel hat dies zum *Huygens-Fresnel'schen Prinzip* ausgebaut:

> Jeder Punkt einer Wellenfront ist Zentrum einer neuen Elementarwelle, die mit allen übrigen interferiert. Die Wellenerregung in einem Raumpunkt ergibt sich durch Überlagerung aller dorthin gelangenden Wellen.

Nach diesem Prinzip kann das gesamte Schallfeld (an jedem Punkt $\vec{r}$ innerhalb eines quellenfreien Volumens durch den Druck P und den Druckgradienten (Schallschnelle) ∇P normal zur begrenzenden Oberfläche S dieses Volumens beschrieben werden. Dies kann geschlossen durch das Kirchhoff-Helmholtz-Integral ausgedrückt werden [22]:

$$P(\vec{r}_R) = \frac{1}{4\pi} \oint\limits_S \Big[P(\vec{r}_s) \cdot \nabla_s G(\vec{r}_R|\vec{r}_S) - \mathrm{G}(\vec{r}_R|\vec{r}_S) \cdot \nabla_s P(\vec{r}_s)\Big] \cdot \vec{n} dS \qquad (1.3)$$

mit den Punkten innerhalb des Volumens $\vec{r}_R \in R$, beziehungsweise auf der Volumsoberfläche $\vec{r}_S \in S$, und der Green'schen Funktion

$$\mathrm{G}(\vec{r}_R|\vec{r}_S) = \frac{e^{-jk|\vec{r}_R - \vec{r}_S|}}{|\vec{r}_R - \vec{r}_S|}$$

Daraus folgt, dass wenn man ein Schallfeld entlang einer geschlossenen Oberfläche aufzeichnen kann (zum Beispiel mit einem unendlich dichten Mikrophonarray) so kann man mit einer geeigneten Wiedergabeeinheit (einem unendlich dichten Lautsprecherarray) ein dem Original identes Schallfeld resynthetisieren.

Die Lautsprechersignale werden bei *Holophonie* durch einen adaptiven Prozess so eingestellt, dass bei gleicher Mikrophonanordnung wie bei der Aufnahme die empfangenen Signale der Mikrophone im Syntheseschallfeld ident denen im originalen Analyseschallfeld sind.

Dieser Ansatz bietet daher keine geschlossene Lösung für die (Re-)Produktion eines Schallfeldes.

Ambisonic

In einem realen Schallfeld treten Wellen mit beliebigen Wellenfronten auf.

Unter der Annahme, dass in einem Schallfeld nur Ebene Wellen auftreten, lässt sich der holophone Ansatz stark vereinfachen, sodass mit relativ wenigen, diskreten Lautsprechern (im Gegensatz zu Oberflächen/Konturen mit quasi unendlich vielen Schallgebern bei Holophonie/Wellenfeldsynthese) ein gegebenes Schallfeld bis zu einem gewissen Grad (“Ordnung”) resynthetisiert werden kann.

Das gesamte Schallfeld wird dabei in eine von der Ordnung abhängige, jedoch von der Anzahl der virtuellen Quellen (des Originalschallfeldes) und der Anzahl der realen Lautsprecher (des Syntheseschallfeldes) unabhängigen Menge von Übertragungskanälen codiert. Die Anzahl der Übertragungskanäle ist im Allgemeinen geringer als die Zahl der virtuellen und realen Quellen, wodurch sich dieses System auch gut zur Speicherung/Übertragung eignet.

Ambisonic ist zu sich selbst auf- und abwärtskompatibel. Dadurch kann der Encodierungsprozess völlig isoliert vom Resyntheseprozess behandelt werden.

Die Ordnung des resynthetisierten Schallfeldes resultiert aus der Anzahl der verwendeten Lautsprecher. Dabei gilt die Faustregel: *Je mehr Lautsprecher, desto höher die Ordnung, desto besser die Abbildung.*

Der ambisonic-Ansatz ist sowohl für zweidimensionale (pantophone) wie auch für dreidimensionale (periphone) Wiedergabe geeignet. Die Anzahl der Übertragungs- und Wiedergabekanäle steigt mit der Ordnung, im zweidimensionalen Falle *linear*, im dreidimensionalen Fall *quadratisch*

Wellenfeldsynthese

Im Gegensatz zur klassischen *Holophonie*, wird das Huygens’sche Prinzip, auf den zweidimensionalen Fall reduziert, bei der *Wellenfeldsynthese* geschlossen gelöst.

Das zweidimensionale Schallfeld wird mit Hilfe eines linearen (meist rechteckigen) Arrays von etwa 180 Lautsprechern[5] reproduziert.

Die große Anzahl an Wiedergabekanälen kann nur mit sehr kleinen Schallwandlern realisiert werden. Da diese im Allgemeinen schlechte Übertragungseigenschaften aufweisen, müssen Kompensationsfilter verwendet werden.

Für den hohen Rechenaufwand wird ein computergesteuertes DSP-System benötigt.

[5]oder mehr. Es geht hier nur darum, einen Eindruck vom Aufwand zu verschaffen

Die Erweiterung zu einem dreidimensionalen holophonen System ist mit heutiger DSP-Hardware noch nicht zu realisieren.

1.5 Conclusio

Trotz der bekannten Unzulänglichkeiten hat sich das Stereo-System in den letzten 70 Jahren einen wichtigen Platz unter den Wiedergabesystemen erhalten können. In den letzten Jahren hat sich auch das Surround-System der *Dolby Laboratories* behaupten können. Diese beiden Systeme dürfen daher von einer allgemeinen Beschallungsanlage nicht vernachlässigt werden.

Im Bereich der elektroakustischen Musik hat sich neben diesen Systemen vor allem die Mehrspur-Komposition mit diskreten Lautsprecher-Kanälen durchgesetzt. Die meisten Kompositionen bestehen aus vier Spuren, die (ähnlich der Quadrophonie) über eine quadratische Lautsprecheranordnung (siehe Abb. 1.5) wiedergegeben werden. Eine Erweiterung dieses horizontalen Systems in den 3D-Raum wird durch zwei Lautsprecherquadrate, die übereinander angeordnet sind (siehe Abb. 1.7), erreicht (8-Spur-Komposition).

Aus Kompatibilitätsgründen sollten solche 4- oder 8-Kanal-Aufzeichnungen über ein allgemeines Wiedergabesystem reproduzierbar sein.

Mit den holophonen Ansätzen (und dem binauralen Ansatz, der in weiterer Folge aufgrund seiner für die allgemeine Aufführungspraxis inpraktikablen Art vernachlässigt wird) lassen sich eine beliebige Anzahl von virtuellen Quellen an beliebigen Orten abbilden. Es ist daher leicht, mit einem solchen Meta-System, zum Beispiel ein virtuelles Stereo-Lautsprecherpaar abzubilden.

Die (eigentliche) Holophonie eignet sich für solche Anforderungen nicht, da sie für den Adaptionsprozess echte Holophonie-Aufnahmen benötigt.

Die Wellenfeldsynthese ist relativ aufwendig. Vor allem zur Reproduktion von dreidimensionalen (periphonen) Schallfeldern eignet sie sich mit heuti-

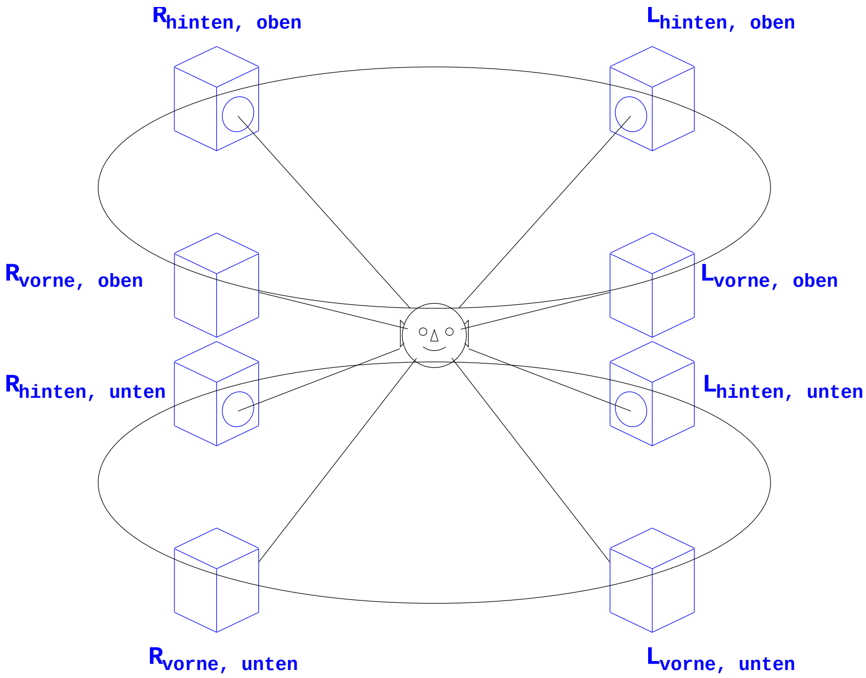


Abbildung 1.7: periphone 8-Spurwiedergabe

gen[6] Mitteln nicht.

Der ambisonic-Ansatz ist nicht nur weniger komplex, sondern auch besser skalierbar, wodurch er sich für eine Vielzahl von Anwendungen und Umgebungen eignet.

Aus diesem Grunde wurde entschieden, die Beschallungsanlage auf *ambisonic*-Basis zu implementieren.

[6]Ende 2001

Kapitel 2

Ambisonic

Als Vereinfachung eines Sonderfalles von Holophonie wird bei ambisonic die Annahme getroffen, dass sowohl im wiederzugebenden als auch im wiedergebenden System nur "Ebene Wellen" auftreten. Dies ist im Allgemeinen der Fall, wenn die Schallquellen (sowohl reproduzierte (virtuelle) als auch reproduzierende (Lautsprecher)) "weit genug" von der Hörposition entfernt sind.

Mit der Einschränkung auf Ebene Wellen lässt sich durch einfache mathematische Operationen (Lautstärken-Manipulation) das Originalschallfeld mit einer Lautsprecher-Matrix beliebig genau resynthetisieren.

2.1 Definitionen

2.1.1 Koordinatensystem

Entsprechend der menschlichen Lokalisationsfähigkeit, die anstelle von kartesischen Koordinaten dazu neigt, Positionsangaben in Richtung und Entfernung zu trennen, und aus Gründen der Vereinfachung bei der Herleitung wird ein sphärisches Koordinatensystem verwendet (s. Abb 2.1).

Es ist dabei üblich, das Koordinatensystem so zu drehen, dass die x-Achse nach "vorne" und die y-Achse nach "links" zeigt.[1]

[1] Im Gegensatz zum normalen kartesischen Koordinatensystem, bei dem die y-Achse nach vorne/oben und die x-Achse nach rechts zeigt.

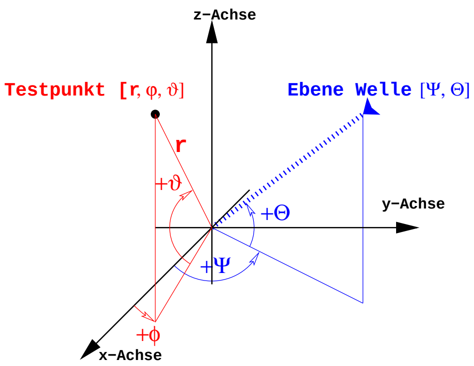


Abbildung 2.1: Koordinatensystem

2.1.2 Encodierung, Decodierung

Das Ziel von ambisonic ist es, eine Kopie einer Ebenen Welle zu synthetisieren. Dazu ist es notwendig, in einem Analyseschritt eine geschlossene Beschreibung einer solchen Ebenen Welle zu finden.

Eine Mono-Schallquelle strahlt aus einer Richtung eine solche Ebene Welle. Im folgenden Kapitel wird gezeigt, wie eine solche Ebene Welle an jedem Punkt im Raum vollständig beschrieben werden kann. Das Verfahren, mit dem eine Mono-Quelle mit Richtungsangabe als Ebene Welle beschrieben wird, nennt man *Encodierung*.

In einem weiteren Schritt kann die so beschriebene Welle mit einem Lautsprecher-Array wieder in ein reales Schallfeld zurück gewandelt werden. Diesen Vorgang nennt man *Decodierung*.

2.2 Encodierung

Die Ausbreitung eines freien Schallfeldes, in dem sich keine Schallquellen befinden, kann nach [11] beschrieben werden als eine Differentialgleichung 2. Ordnung:

$$(\Delta + k^2)p = 0 \tag{2.1}$$

Dabei ist p der Schalldruck und k die Wellenzahl $k = \frac{2\pi f}{c}$

In einem sphärischen Koordinatensystem kann man dies schreiben als:

$$\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right)+\frac{1}{r^2\sin\vartheta}\frac{\partial}{\partial\vartheta}\left(\sin\vartheta\frac{\partial}{\partial\vartheta}\right)+\frac{1}{r^2\sin^2\vartheta}\frac{\partial^2}{\partial\varphi^2}+k^2\right]p=0$$

Die Lösung dieser Differentialgleichung mit Hilfe der Variablenseparierung führt zu einer Entwicklung in eine Fourier-Bessel-Reihe mit den Koeffizienten $A^\sigma_{m,\eta}$ und $B^\sigma_{m,\eta}$

$$p(r,\varphi,\vartheta)=\sum_{m,\eta,\sigma}Y^\sigma_{m,\eta}(\vartheta,\varphi)j^m(A^\sigma_{m,\eta}J_m(kr)+jB^\sigma_{m,\eta}N_m(kr)) \tag{2.2}$$

J_m bezeichnet die *Bessel*-Funktion m-ter Ordnung, N_m die *Neumann*-Funktion m-ter Ordnung. $Y^\sigma_{m,\eta}$ steht für die *sphärische Harmonische*, die definiert wird als:

$$Y^\sigma_{m,\eta}(\varphi,\vartheta)=\bar{P}_{m,\eta}(\sin\vartheta)\cdot\begin{cases}\cos(\eta\varphi) & \text{für}\quad \sigma=1\\ \sin(\eta\varphi) & \text{für}\quad \sigma=-1\quad\text{und}\quad 1\le\eta\le m\end{cases} \tag{2.3}$$

Dabei bezeichnet $\bar{P}_{m,\eta}$ das halb-normalisierte *Legendre*-Polynom:

$$\bar{P}_{m,\eta}(\zeta)=\sqrt{\varepsilon_\eta\frac{(m-\eta)!}{(m+\eta)!}}P_{m,\eta}(\zeta)\quad\text{mit}\begin{cases}\varepsilon_0=1\\ \varepsilon_\eta=2\quad\text{für}\quad \eta\ge1\end{cases} \tag{2.4}$$

Betrachten wir nun eine Ebene Welle, die sich in Richtung $\vec{u}_p$ ausbreitet, an einem Punkt im Raum $\vec{r}$. Die Ausbreitung der Welle wird dabei vollständig durch die Ausbreitungsrichtung, die Schallgeschwindigkeit und die Frequenz beschrieben. Man definiert über Richtung und Wellenzahl den Vektor

$$\vec{k}=k\vec{u}_p=\frac{2\pi f}{c}\vec{u}_p \tag{2.5}$$

wobei die Schallgeschwindigkeit $c\approx 340m/s$ beträgt. Da $\vec{u}_p$ nur die Richtung beschreibt, gilt weiters $|\vec{u}_p|=1$.

Da die Welle als eben angenommen wird, ist die Ausbreitungsrichtung über alle Messpunkte $\vec{r}$ konstant.

Eine Ebene Welle kann vollständig beschrieben werden als

$$S_{\vec{k}}(\vec{r})=p_{\vec{k}}\mathrm{e}^{j\vec{k}\cdot\vec{r}}=p_{\vec{k}}\mathrm{e}^{jkr\cos\angle(\vec{k},\vec{r})} \tag{2.6}$$

Dadurch lässt sich der allgemeine Ansatz in Gl. 2.2 mit

$$\cos\delta = \cos\angle(\vec{k}, \vec{r}) = \frac{\vec{k} \cdot \vec{r}}{|\vec{k}||\vec{r}|}$$

vereinfachen zu:

$$S_{\vec{k}}(\vec{r}) = p_{\vec{k}} \sum_{m=0}^{\infty} (2m+1) j^m P_m(\cos\delta) J_m(kr) \tag{2.7}$$

In sphärischen Koordinaten bedeutet dies eine ebene Welle im Punkt (r, φ, ϑ) aus der Richtung (Φ, Θ). Das *Legendre*-Polynom P_m lässt sich dann als folgende Summe ausdrücken:[2]

$$P_m(\cos\delta) = \sum_{\eta=0}^{m} \varepsilon_\eta \frac{(m-\eta)!}{(m+\eta)!} P_{m,\eta}(\cos(\frac{\pi}{2} - \vartheta)) P_{m,\eta}(\cos(\frac{\pi}{2} - \Theta)) \cos[\eta(\varphi - \Phi)] \tag{2.8}$$

Damit und unter Verwendung der Definition der *sphärischen Harmonischen* (Gl. 2.3) erhält man:

$$\begin{aligned}
S_{\Phi,\Theta}(r, \varphi, \vartheta) &= p_{\Phi,\Theta} \sum_{m=0}^{\infty} (2m+1) j^m P_m(\cos\delta) J_m(kr) = \\
&= p_{\Phi,\Theta} \sum_{m=0}^{\infty} (2m+1) j^m J_m(kr) \cdot \\
&\quad \cdot \sum_{\eta=0}^{m} \varepsilon_\eta \frac{(m-\eta)!}{(m+\eta)!} P_{m,\eta}(\sin\vartheta) P_{m,\eta}(\sin\Theta) \cos[\eta(\varphi - \Phi)] = \\
&= p_{\Phi,\Theta} \sum_{m=0}^{\infty} (2m+1) j^m J_m(kr) \cdot \\
&\quad \cdot \sum_{\eta=0}^{m} \bar{P}_{m,\eta}(\sin\vartheta) \bar{P}_{m,\eta}(\sin\Theta) \cos[\eta(\varphi - \Phi)] = \\
&= p_{\Phi,\Theta} \sum_{m=0}^{\infty} \sum_{\eta=0}^{m} \sum_{\sigma=\pm 1} (2m+1) j^m J_m(kr) Y_{m,\eta}^{\sigma}(\Phi, \Theta) Y_{m,\eta}^{\sigma}(\varphi, \vartheta) = \\
&= p_{\Phi,\Theta} \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y_{m,\eta}^{\sigma}(\Phi, \Theta) Y_{m,\eta}^{\sigma}(\varphi, \vartheta)
\end{aligned} \tag{2.9}$$

[2] δ ist aus Gründen der einfacheren Herleitung an der Z-Achse orientiert ($\delta = 0$ zeigt nach oben). Da im verwendeten Koordinatensystem die Elevation θ jedoch von der XY-Ebene aus gesehen wird ($\theta = \frac{\pi}{2}$ zeigt nach oben), gilt $\delta = \frac{\pi}{2} - \theta$.

Wird nur der Koordinatenursprung betrachtet, vereinfacht sich diese Beschreibung zu:

$$B^{\sigma}_{m,\eta} = Y^{\sigma}_{m,\eta}(\Phi,\Theta) p_{\Phi,\Theta} \tag{2.10}$$

Eine Ebene Welle aus einer bestimmten Richtung $\vec{u}_p = [\Phi,\Theta]^{\mathrm{T}}$ kann also durch eine unendlich hohe Anzahl von eindimensionalen ambisonic-Signalen vollständig repräsentiert werden.

Die einzelnen ambisonic-Kanäle lassen sich einfach durch die Multiplikation eines Mono-Signals mit dem Funktionswert der sphärischen Harmonischen für die Richtung, aus der die Quelle die Ebene Welle abstrahlt darstellen.

Durch die vollständige Beschreibung wird es möglich, das Originalschallfeld mit einem Wiedergabesystem akkurat zu resynthetisieren.

Natürlich ist die Übertragung und Wiedergabe von unendlich vielen Kanälen technisch nicht möglich. Im Allgemeinen wird die Reihenentwicklung daher nach einem bestimmten Glied abgebrochen und es werden nur L Übertragungskanäle verwendet. Die Ordnung M der sphärischen Harmonischen, nach der abgebrochen wird, bezeichnet man daher auch die *Ordnung* des ambisonic-Systems.

Dadurch tritt ein Informationsverlust auf, da die ebene Welle nicht mehr vollständig repräsentiert wird. Es ist in Folge auch nicht mehr möglich, das originale Schallfeld mathematisch korrekt zu resynthetisieren

Das reduzierte ambisonic-Schallfeld lässt sich in Matrixschreibweise darstellen als:

$$\vec{B}_{\Phi,\Theta} = \vec{Y}_{\Phi,\Theta} \cdot p_{\Phi,\Theta} \tag{2.11}$$

Die einzelnen Kanäle des ambisonic-Schallfeldes $\vec{B}_{\Phi,\Theta}$ werden traditionell mit den Buchstaben des Alphabetes bezeichnet, und zwar in folgender Form:

$$\vec{B} = \begin{pmatrix} B_{0,0} \\ B^{1}_{1,1} \\ B^{-1}_{1,1} \\ B^{1}_{1,0} \\ \vdots \end{pmatrix} = \begin{pmatrix} W \\ X \\ Y \\ Z \\ \vdots \end{pmatrix} \tag{2.12}$$

Durch Einsetzen der ersten paar Werte für m und η in die *assoziierten Legendre*-Polynome erhält man die Sphärischen Harmonischen:

$$\begin{array}{c|cc|l|l}
M & m & \eta & Y^{+1}_{m,\eta} & Y^{-1}_{m,\eta} \\
\hline
0 & 0 & 0 & 1 & 0 \\
\hline
1 & 1 & 0 & \sin\Theta & 0 \\
 & 1 & 1 & \cos\Theta\cos\Phi & \cos\Theta\sin\Phi \\
\hline
2 & 2 & 0 & \frac{1}{2}(3\sin^2\Theta - 1) & 0 \\
 & 2 & 1 & \frac{3}{2}\sin 2\Theta\cos\Phi & \frac{3}{2}\sin 2\Theta\sin\Phi \\
 & 2 & 2 & \frac{3}{2}\cos^2\Theta\cos 2\Phi & \frac{3}{2}\cos^2\Theta\sin 2\Phi
\end{array} \tag{2.13}$$

Daraus lässt sich zum Beispiel ein ambisonic-Schallfeld zweiter Ordnung ableiten:

$$\begin{aligned}
B_{0,0} &= W = p_{\Phi,\Theta} \\
B^{1}_{1,1} &= X = (\cos\Phi\cos\Theta)p_{\Phi,\Theta} \\
B^{-1}_{1,1} &= Y = (\sin\Phi\cos\Theta)p_{\Phi,\Theta} \\
B_{1,0} &= Z = (\sin\Theta)p_{\Phi,\Theta} \\
B_{2,0} &= R = (\tfrac{1}{2}(3\sin^2\Theta - 1))p_{\Phi,\Theta} \\
B^{1}_{2,1} &= S = (\tfrac{3}{2}\cos\Phi\sin 2\Theta)p_{\Phi,\Theta} \\
B^{-1}_{2,1} &= T = (\tfrac{3}{2}\sin\Phi\sin 2\Theta)p_{\Phi,\Theta} \\
B^{1}_{2,2} &= U = (\tfrac{3}{2}\cos 2\Phi\cos^2\Theta)p_{\Phi,\Theta} \\
B^{-1}_{2,2} &= V = (\tfrac{3}{2}\sin 2\Phi\cos^2\Theta)p_{\Phi,\Theta}
\end{aligned} \tag{2.14}$$

Für $\eta = 0, \sigma = -1$ sind die sphärischen Harmonischen $Y^{-1}_{m,0} \equiv 0$. Dadurch errechnet sich die Anzahl L der für die ambisonic-Repräsentation M-ter Ordnung notwendigen Kanäle als

$$\begin{aligned}
L^{3D} &= \left[\sum_{k=0}^{M}\sum_{\sigma} k+1\right] - (M+1) = \\
&= 2\sum_{k=1}^{M+1} k - M - 1 = \\
&= (M+1)^2
\end{aligned} \tag{2.15}$$

Wird nur der zweidimensionale Fall betrachtet (mit $\Theta = 0$), so sind überhaupt nur noch

$$L^{2D} = 2M + 1 \tag{2.16}$$

Übertragungskanäle notwendig.

2.2.1 Ambisonic-Aufnahmen

Um ein Schallfeld mit einem ambisonic-Wiedergabesystem zu reproduzieren, muss zuerst ein ambisonic-codiertes Schallfeld aufgezeichnet werden. Um ein ambisonic-Schallfeld zu synthetisieren, muss, wie oben gezeigt, eine Mono-Schallquelle $p_{\Phi,\Theta}$ mit der Sphärischen Harmonischen des gesuchten ambisonic-Kanals für die gewünschte Richtung multipliziert werden. Dadurch erhält jeder ambisonic-Kanal eine bestimmte Richtcharakteristik.

Sollen reale Schallfelder direkt in ambisonic aufgenommen werden, müssen daher Mikrophone mit entsprechender Richtcharakteristik verwendet werden. Der ambisonic-Kanal nullter Ordnung hat eine kugelförmige Richtcharakteristik weshalb sich druckempfindliche Mikrophone anbieten. Die Kanäle erster Ordnung haben eine Achter-Charakteristik, die man mit Schallschnelle-Empfängern erhält.

Diese Richtcharakteristika gelten nach Gl. (2.10) nur im Koordinatenursprung. Es ist daher notwendig, alle Schallwandler im Koordinatenursprung zu positionieren, was in der Praxis durch die endliche Kleinheit der Mikrophonkapseln nur schwer möglich ist.

ambisonic-Kanäle höherer Ordnung können nicht mehr so einfach aufgezeichnet werden, da es noch keine Mikrophone mit entsprechenden Richtcharakteristiken gibt.

Eine Möglichkeit, dies zu umgehen, besteht darin, Aufnahmetechniken zu verwenden, die eine genaue Erfassung des Schallfeldes mit anderen Mitteln erlauben, wie zum Beispiel Holophonie, und das so gewonnene Schallfeld in ein ambisonic-Schallfeld zu konvertieren.

2.2.2 Standard-Codierungsvorschriften

In Gl. 2.10 ist eine allgemeine Beschreibung der ambisonic-Kanäle gegeben. In der Literatur finden sich diverse Abarten der sphärischen Harmonischen als “Codierungsvorschriften”.

Diese Abarten unterscheiden sich jedoch nur durch einen konstanten multiplikativen Faktor für jeden Kanal.

So wird zum Beispiel der Faktor $w = B_{0,0}^{(-1)}$ zur Codierung des W-Kanals (0. Ordnung) historisch nicht als 1 sondern aus Gründen der ausgewogeneren Leistungsverteilung über die einzelnen Kanäle als $\frac{1}{\sqrt{2}}$ verwendet.

Gerzon hat das sogenannte *B-Format* als Standard für ambisonic 1. Ordnung vorgeschlagen

w	$=$	$\frac{1}{\sqrt{2}}$
x	$=$	$\cos\varphi\cos\vartheta$
y	$=$	$\sin\varphi\cos\vartheta$
z	$=$	$\sin\vartheta$

Daraus wurde Ende der 1990er Jahre von Furse und Malham das *Furse-Malham*-Set entwickelt [28], welches von ihnen als Standard für voll-periphones ambisonic bis zur 3. Ordnung vorgeschlagen wurde.

w	$=$	$\frac{1}{\sqrt{2}}$	$=$	w
x	$=$	$\cos\varphi\cos\vartheta$	$=$	x
y	$=$	$\sin\varphi\cos\vartheta$	$=$	y
z	$=$	$\sin\vartheta$	$=$	z
r	$=$	$\frac{1}{2}(3\sin^2\vartheta - 1)$	$=$	$\frac{1}{2}(3z^2 - 1)$
s	$=$	$\cos\varphi\sin 2\vartheta$	$=$	$2xz$
t	$=$	$\sin\varphi\sin 2\vartheta$	$=$	$2yz$
u	$=$	$\cos 2\varphi\cos^2\vartheta$	$=$	$x^2 - y^2$
v	$=$	$\sin 2\varphi\cos^2\vartheta$	$=$	$2xy$
k	$=$	$\frac{1}{2}\sin\vartheta(5\sin^2\vartheta - 3)$	$=$	$\frac{1}{2}z(5z^2 - 3)$
l	$=$	$\frac{8}{11}\sin\varphi\cos\vartheta(5\sin^2\vartheta - 1)$	$=$	$\frac{8}{11}y(5z^2 - 1)$
m	$=$	$\frac{8}{11}\cos\varphi\cos\vartheta(5\sin^2\vartheta - 1)$	$=$	$\frac{8}{11}x(5z^2 - 1)$
n	$=$	$\sin 2\varphi\sin\vartheta\cos^2\vartheta$	$=$	$2xyz$
o	$=$	$\cos 2\varphi\sin\vartheta\cos^2\vartheta$	$=$	$z(x^2 - y^2)$
p	$=$	$\sin 3\varphi\cos^3\vartheta$	$=$	$y(3x^2 - y^2)$
q	$=$	$\cos 3\varphi\cos^3\vartheta$	$=$	$x(x^2 - 3y^2)$

Eine Visualisierung des *Furse-Malham*-Sets ist Abb.2.2 bis 2.5 zu entnehmen.

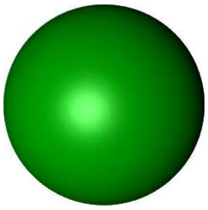


Abbildung 2.2: 0. Ordnung (W)

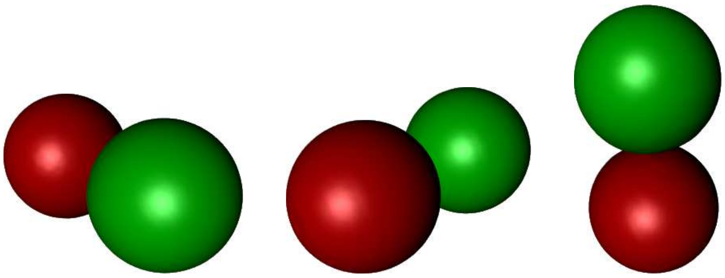


Abbildung 2.3: 1. Ordnung (X, Y, Z)

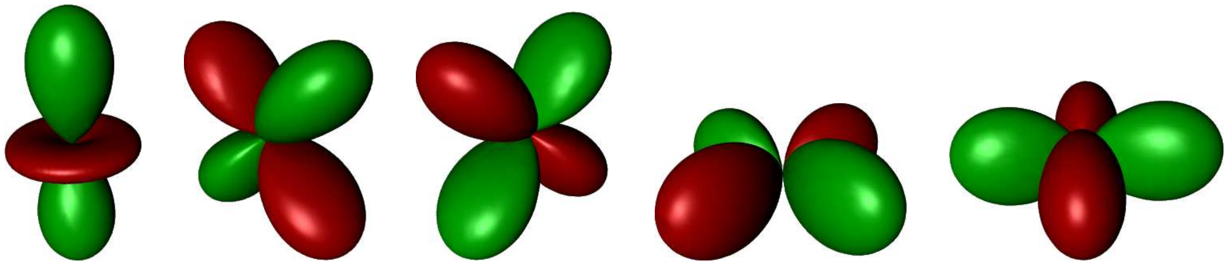


Abbildung 2.4: 2. Ordnung (R, S, T, U, V)

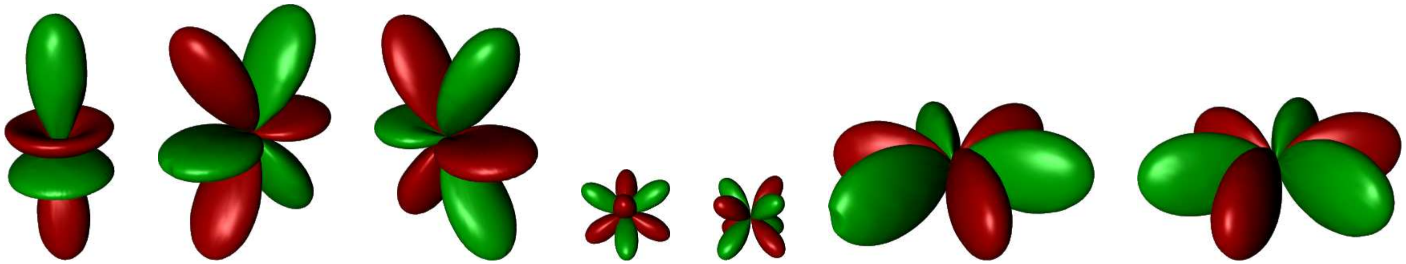


Abbildung 2.5: 3. Ordnung (K, L, M, N, O, P, Q)

2.2.3 Alternative Codierungsvorschriften

Um ein ambisonic-Schallfeld zu speichern oder zu übertragen sind entsprechend der Ordnung des Schallfeldes mehr oder weniger Kanäle erforderlich.

Mehrspur-Aufzeichnungsformate bieten traditionellerweise acht Kanäle an. Bei Verwendung von zwei parallelen 8-Spur-Bandgeräten kann also das gesamte Furse-Malham-Set bis zur dritten Ordnung übertragen werden.

Steht nur ein 8-Spur-Aufzeichnungsgerät zur Verfügung, so kann man rein horizontale Soundfields bis zur dritten Ordnung bei relativ guter Bandbreitenausnützung (sieben von acht Spuren werden bei einem ambisonic-Soundfield dritter Ordnung belegt) übertragen. Allerdings können keine periphonen Schallfelder, deren Ordnung über eins hinausgeht, in acht Spuren untergebracht werden. Jedoch werden bei 3D-ambisonic erster Ordnung nur vier Kanäle benötigt, wodurch 50% der zur Verfügung stehenden Bandbreite ungenutzt bleibt.

Durch geschickte Wahl der Codierungsvorschrift und gemischte Ordnungen für die drei Dimensionen, lässt sich aber eine bessere Bandbreitenausnutzung erreichen.

Codierungsvorschrift *3rd+1st*

Codiert man ein Schallfeld ohne Höheninformation mit Ambisonic dritter Ordnung, so werden sieben Übertragungskanäle benötigt. Über einen zusätzlichen achten Kanal, kann Höheninformation mit ambisonic erster Ordnung übertragen werden:

$$
\begin{array}{ccc}
w & = & \frac{1}{\sqrt{2}} \\
\hline
x & = & \cos\varphi\cos\vartheta \\
y & = & \sin\varphi\cos\vartheta \\
z & = & \sin\vartheta \\
\hline
s & = & \cos 2\varphi\cos\vartheta \\
t & = & \sin 2\varphi\cos\vartheta \\
u & = & \cos 3\varphi\cos\vartheta \\
v & = & \sin 3\varphi\cos\vartheta
\end{array}
$$

2.3 Manipulation

Wie in Kapitel 2.2 gezeigt, kann aus einem Mono-Signal mit Richtungsangabe $[\Phi, \Theta]^{\mathrm{T}}$ mit Hilfe der Sphärischen Harmonischen durch die einfache lineare Operation der Multiplikation eine ambisonic-Repräsentation einer ebenen Welle gefunden werden.

Da die Encodierungsvorschrift linear ist, kann das encodierte Schallfeld nachträglich noch durch einfache Operationen manipuliert werden.

2.3.1 Amplifikation

Durch Multiplikation des gesamten Schallfeldes mit einem Faktor α kann tatsächlich die Lautstärke manipuliert werden.

$$\vec{B}' = \alpha\vec{B} = \alpha\vec{Y} \cdot p = \vec{Y} \cdot \alpha p \tag{2.17}$$

2.3.2 Superposition

Werden mehrere ambisonic-Schallfelder (mit beliebigen Mono-quellen an beliebigen Positionen) addiert, so ist das resultierende ambisonic-Schallfeld eine Repräsentation von mehreren virtuellen Quellen.

Ein ambisonic-Schallfeld einer bestimmten Ordnung kann also beliebig viele virtuelle Schallquellen bei gleich bleibenden Eigenschaften beinhalten.

$$\vec{B}' = \sum_i \vec{B}_i \tag{2.18}$$

2.3.3 Rotation

Die Sphärischen Harmonischen bilden eine Orthonormalbasis. Dadurch kann man sie durch lineare Operationen ineinander überführen und somit das ambisonic-Schallfeld durch Matrizenmultiplikation um alle Achsen rotieren lassen.

$$\vec{B}' = \mathbf{R} \cdot \vec{B} \tag{2.19}$$

Sinnvollerweise werden Rotationsmatrizen um die drei Hauptachsen definiert, um die Drehungen *tilt-tumble-rotation* zu ermöglichen.

$$\vec{B}' = \mathbf{R}_x(\Psi) \cdot \mathbf{R}_y(\Theta) \cdot \mathbf{R}_z(\Phi) \cdot \vec{B} \tag{2.20}$$

Rotationsmatrizen für das Furse-Malham Set

Rotationsmatrizen für das *Furse-Malham*-Set sind den Tabellen 2.1 (Tilt Ψ um X-Achse), 2.2 (Tumble Ψ um Y-Achse) und 2.3 (Rotation Φ um Z-Achse) zu entnehmen.

$$
\begin{pmatrix} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \end{pmatrix}_{\Psi}
=
\left(\begin{array}{c|ccc|ccccc}
1 & & & & & & & & \\ \hline
 & 1 & 0 & 0 & & & & & \\
 & 0 & \cos\Psi & -\sin\Psi & & & & & \\
 & 0 & \sin\Psi & \cos\Psi & & & & & \\ \hline
 & & & & \frac{1}{4}(1+3\cos 2\Psi) & 0 & \frac{3}{4}\sin 2\Psi & \frac{3}{8}(\cos 2\Psi - 1) & 0 \\
 & & & & 0 & \cos\Psi & 0 & 0 & \sin\Psi \\
 & & & & -\sin 2\Psi & 0 & \cos 2\Psi & -\frac{1}{2}\sin 2\Psi & 0 \\
 & & & & \frac{1}{2}(\cos 2\Psi - 1) & 0 & \frac{1}{2}\sin 2\Psi & \frac{1}{4}(\cos 2\Psi + 3) & 0 \\
 & & & & 0 & -\sin\Psi & 0 & 0 & \cos\Psi
\end{array}\right)
\begin{pmatrix} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \end{pmatrix}
$$

$$
\begin{pmatrix} K \\ L \\ M \\ N \\ O \\ P \\ Q \end{pmatrix}_{\Psi}
=
\begin{pmatrix}
\frac{1}{8}(3\cos\Psi + 5\cos 3\Psi) & \frac{33}{256}(\sin\Psi + 5\sin 3\Psi) & 0 & 0 & \frac{15}{16}(-\cos\Psi + \cos 3\Psi) & \frac{5}{32}(-3\sin\Psi + \sin 3\Psi) & 0 \\
-\frac{2}{11}(\sin\Psi + 5\sin 3\Psi) & \frac{1}{16}(\cos\Psi + 15\cos 3\Psi) & 0 & 0 & \frac{5}{11}(\sin\Psi - 3\sin 3\Psi) & \frac{5}{22}(-\cos\Psi + \cos 3\Psi) & 0 \\
0 & 0 & \frac{1}{8}(3 + 5\cos 2\Psi) & \frac{20}{11}\sin 2\Psi & 0 & 0 & \frac{5}{11}(\cos 2\Psi - 1) \\
0 & 0 & -\frac{11}{32}\sin 2\Psi & \cos 2\Psi & 0 & 0 & -\frac{1}{4}\sin 2\Psi \\
\frac{1}{4}(-\cos\Psi + \cos 3\Psi) & \frac{11}{128}(-\sin\Psi + 3\sin 3\Psi) & 0 & 0 & \frac{1}{8}(5\cos\Psi + 3\cos 3\Psi) & \frac{1}{16}(5\sin\Psi + \sin 3\Psi) & 0 \\
\frac{1}{4}(3\sin\Psi - \sin 3\Psi) & -\frac{33}{128}(\cos\Psi - \cos 3\Psi) & 0 & 0 & -\frac{3}{8}(5\sin\Psi + \sin 3\Psi) & \frac{1}{16}(15\cos\Psi + \cos 3\Psi) & 0 \\
0 & 0 & \frac{33}{64}(\cos 2\Psi - 1) & \frac{3}{2}\sin 2\Psi & 0 & 0 & \frac{1}{8}(3\cos 2\Psi + 5)
\end{pmatrix}
\begin{pmatrix} K \\ L \\ M \\ N \\ O \\ P \\ Q \end{pmatrix}
$$

Tabelle 2.1: Tilt Ψ um die X-Achse

$$\left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \end{array}\right)_{\Theta} = \left(\begin{array}{c|ccc|ccccc} 1 & & & & & & & & \\ \hline & \cos\Theta & 0 & -\sin\Theta & & & & & \\ & 0 & 1 & 0 & & & & & \\ & \sin\Theta & 0 & \cos\Theta & & & & & \\ \hline & & & & \frac{1}{4}(1+3\cos 2\Theta) & \frac{3}{4}\sin 2\Theta & 0 & \frac{3}{8}(1-\cos 2\Theta) & 0 \\ & & & & -\sin 2\Theta & \cos 2\Theta & 0 & \frac{1}{2}\sin 2\Theta & 0 \\ & & & & 0 & 0 & \cos\Theta & 0 & sin\Theta \\ & & & & \frac{1}{2}(1-\cos 2\Theta) & -\frac{1}{2}\sin 2\Theta & 0 & \frac{1}{4}(3+\cos 2\Theta) & 0 \\ & & & & 0 & 0 & -\sin\Theta & 0 & \cos\Theta \end{array}\right) \left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \end{array}\right)$$

$$\begin{pmatrix} K \\ L \\ M \\ N \\ O \\ P \\ Q \end{pmatrix}_{\Theta} = \begin{pmatrix} \frac{1}{8}(3\cos\Theta+5\cos 3\Theta) & 0 & \frac{33}{256}(\sin\Theta+5\sin 3\Theta) & 0 & \frac{15}{16}(\cos\Theta-\cos 3\Theta) & 0 & \frac{5}{32}(3\sin\Theta-\sin 3\Theta) \\ 0 & \frac{1}{8}(3+5\cos 2\Theta) & 0 & \frac{20}{11}\sin 2\Theta & 0 & \frac{5}{11}(1-\cos 2\Theta) & 0 \\ -\frac{2}{11}(\sin\Theta+5\sin 3\Theta) & 0 & \frac{1}{16}(\cos\Theta+15\cos 3\Theta) & 0 & \frac{5}{11}(-\sin\Theta+3\sin 3\Theta) & 0 & \frac{5}{22}(cos\Theta-\cos 3\Theta) \\ 0 & -\frac{11}{32}\sin 2\Theta & 0 & \cos 2\Theta & 0 & \frac{1}{4}\sin 2\Theta & 0 \\ \frac{1}{4}(\cos\Theta-\cos 3\Theta) & 0 & \frac{11}{128}(\sin\Theta-3\sin 3\Theta) & 0 & \frac{1}{8}(5\cos\Theta+3\cos 3\Theta) & 0 & \frac{1}{16}(5\sin\Theta+\sin 3\Theta) \\ 0 & \frac{33}{64}(1-\cos 2\Theta) & 0 & -\frac{3}{2}\sin 2\Theta & 0 & \frac{1}{8}(5+3\cos 2\Theta) & 0 \\ \frac{1}{4}(-3\sin\Theta+\sin 3\Theta) & 0 & \frac{33}{128}(\cos\Theta-\cos 3\Theta) & 0 & -\frac{3}{8}(5\sin\Theta+\sin 3\Theta) & 0 & \frac{1}{16}(15\cos\Theta+\cos 3\Theta) \end{pmatrix} \begin{pmatrix} K \\ L \\ M \\ N \\ O \\ P \\ Q \end{pmatrix}$$

Tabelle 2.2: Tumble Θ um die Y-Achse

$$
\left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \\ \hline K \\ L \\ M \\ N \\ O \\ P \\ Q \end{array}\right)_{\Phi}
=
\left(\begin{array}{c|ccc|ccccc|ccccccc}
1 & & & & & & & & & & & & & & & \\ \hline
 & \cos\Phi & -\sin\Phi & 0 & & & & & & & & & & & & \\
 & \sin\Phi & \cos\Phi & 0 & & & & & & & & & & & & \\
 & 0 & 0 & 1 & & & & & & & & & & & & \\ \hline
 & & & & 1 & 0 & 0 & 0 & 0 & & & & & & & \\
 & & & & 0 & \cos\Phi & -\sin\Phi & 0 & 0 & & & & & & & \\
 & & & & 0 & \sin\Phi & \cos\Phi & 0 & 0 & & & & & & & \\
 & & & & 0 & 0 & 0 & \cos 2\Phi & -\sin 2\Phi & & & & & & & \\
 & & & & 0 & 0 & 0 & \sin 2\Phi & \cos 2\Phi & & & & & & & \\ \hline
 & & & & & & & & & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & \cos\Phi & \sin\Phi & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & -\sin\Phi & \cos\Phi & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & \cos 2\Phi & \sin 2\Phi & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & -\sin 2\Phi & \cos 2\Phi & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & \cos 3\Phi & \sin 3\Phi \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & -\sin 3\Phi & \cos 3\Phi
\end{array}\right)
\left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline R \\ S \\ T \\ U \\ V \\ \hline K \\ L \\ M \\ N \\ O \\ P \\ Q \end{array}\right)
$$

Tabelle 2.3: Rotation Φ um die Z-Achse

Rotation von 3rd+1st

Es ist nicht möglich, durch lineare Manipulationen ein Soundfield niedrigerer Ordnung in eines höherer Ordnung zu überführen.[3]

Da bei der Codierungsvorschrift *3rd+1st* Horizontalebene und Höhe mit verschiedenen Ordnungen kodiert werden, kann das Schallfeld durch Rotationsmatrizen nur innerhalb der Horizontalebene gedreht werden,

Rotation Φ um die Z-Achse

$$\left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline S \\ T \\ U \\ V \end{array}\right)_{\Phi} = \left(\begin{array}{c|ccc|cccc} 1 & & & & & & & \\ \hline & \cos\Phi & -\sin\Phi & 0 & & & & \\ & \sin\Phi & \cos\Phi & 0 & & & & \\ & 0 & 0 & 1 & & & & \\ \hline & & & & \cos 2\Phi & -\sin 2\Phi & 0 & 0 \\ & & & & \sin 2\Phi & \cos 2\Phi & 0 & 0 \\ & & & & 0 & 0 & \cos 3\Phi & -\sin 3\Phi \\ & & & & 0 & 0 & \sin 3\Phi & \cos 3\Phi \end{array}\right) \left(\begin{array}{c} W \\ \hline X \\ Y \\ Z \\ \hline S \\ T \\ U \\ V \end{array}\right)$$

2.4 Decodierung

In der Praxis ist es nicht möglich, unendlich viele Kanäle wiederzugeben. Stattdessen erfolgt in einem linearen Decodierungsprozess eine Verteilung der L ambisonic-Kanäle auf N Lautsprecher.

Seien

$$\begin{aligned} \vec{p} &= [p_1, p_2, \cdots, p_N]^{\mathrm{T}} \\ \vec{B} &= [Y^1_{0,0}(\Phi,\Theta), Y^1_{1,0}(\Phi,\Theta), \cdots, Y^{-1}_{M,M}(\Phi,\Theta)]^{\mathrm{T}} s \end{aligned} \tag{2.21}$$

$\vec{p}$ ist dabei die Menge aller Lautsprechersignale p_n. $\vec{B}$ ist die ambisonic-Beschreibung einer Ebenen Welle mit dem Signalverlauf s aus der Richtung $[\Phi,\Theta]^{\mathrm{T}}$.

[3]Umgekehrt ist es natürlich sehr leicht möglich, eine Soundfield höherer Ordnung durch 0-setzen der Komponenten höherer Ordnung auf eine niedrigere Ordnung zu überführen.

Nun kann man den allgemeinen Decodierungsprozess in Matrixschreibweise notieren als:

$$\vec{p} = \mathbf{D} \cdot \vec{B} \tag{2.22}$$

2.4.1 *basic* Decodierung

Betrachten wir eine beliebige Quelle, die eine ebene Welle aus der Richtung $\vec{u}_p = [\Phi, \Theta]^\mathrm{T}$ abstrahlt. Nach Gleichung 2.7 und 2.9 kann das gesamte Schallfeld (an allen Punkten $\vec{r}$) beschrieben werden als:

$$S_{\text{Analyse}} = S_{\Phi,\Theta}(r, \varphi, \vartheta) = s \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y^\sigma_{m,\eta}(\Phi, \Theta) Y^\sigma_{m,\eta}(\varphi, \vartheta) \tag{2.23}$$

Bei der Wiedergabe ist es aus technischen Gründen nicht möglich, unendlich viele Kanäle abzuspielen. Im Allgemeinen erfolgt die Wiedergabe vielmehr über eine Anzahl von N Lautsprechern.

Ist jeder Lautsprecher n weit genug vom Auditorium entfernt, so kann man die von einem einzelnen Lautsprecher emittierte Welle als Ebene Welle idealisieren, die aus der Richtung des Schallwandlers $[\varphi_n, \vartheta_n]^\mathrm{T}$ kommt. Jede Welle, die von einem solchen Lautsprecher abgestrahlt wird, lässt sich also ebenfalls beschreiben durch Gl. 2.9:

$$S_n(r, \varphi, \vartheta) = p_n \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y^\sigma_{m,\eta}(\varphi_n, \vartheta_n) Y^\sigma_{m,\eta}(\varphi, \vartheta) \tag{2.24}$$

Am Messpunkt überlagern sich die Schallfelder der einzelnen Lautsprecher zu einem Synthese-Schallfeld:

$$\begin{aligned} S_{\text{Synthese}} &= S^\Sigma(r, \varphi, \vartheta) = \sum_{n=1}^{N} S_n(r, \varphi, \vartheta) = \\ &= \sum_{n=1}^{N} p_n \sum_{m,\eta,\sigma} (2m+1) j^m J_m(kr) Y^\sigma_{m,\eta}(\varphi_n, \vartheta_n) Y^\sigma_{m,\eta}(\varphi, \vartheta) \end{aligned} \tag{2.25}$$

Das Ziel eines ambisonic-Wiedergabesystems ist es, das ursprüngliche Schallfeld so genau wie möglich zu resynthetisieren. Es soll also gelten:

$$S_{\text{Synthese}} \equiv S_{\text{Analyse}} \tag{2.26}$$

Durch Koeffizientenvergleich erhält man für die Äquivalenz folgende Randbedingung (matching condition) mit $m = 1, \ldots, \infty$ und $\eta = 1, \ldots, m$:

$$sY^{\sigma}_{m,\eta}(\Phi, \Theta) = \sum_{n=1}^{N} p_n Y^{\sigma}_{m,\eta}(\varphi_n, \vartheta_n) \tag{2.27}$$

Oder ausgeschrieben:

$$\begin{aligned} s &= \sum_{n=1}^{N} p_n \\ sP_{m,\eta}(\cos\Theta)\cos(\eta\Phi) &= \sum_{n=1}^{N} p_n P_{m,\eta}(\cos\vartheta_n)\cos(\eta\varphi_n) \\ sP_{m,\eta}(\cos\Theta)\sin(\eta\Phi) &= \sum_{n=1}^{N} p_n P_{m,\eta}(\cos\vartheta_n)\sin(\eta\varphi_n) \end{aligned} \tag{2.28}$$

Diese Gleichung kann mit Gl. (2.21) und

$$\mathbf{C} = \begin{bmatrix} Y^{1}_{0,0}(\varphi_1, \vartheta_1) & Y^{1}_{0,0}(\varphi_2, \vartheta_2) & \cdots & Y^{1}_{0,0}(\varphi_N, \vartheta_N) \\ Y^{1}_{1,0}(\varphi_1, \vartheta_1) & Y^{1}_{1,0}(\varphi_2, \vartheta_2) & \cdots & Y^{1}_{1,0}(\varphi_N, \vartheta_N) \\ \vdots & \vdots & \ddots & \vdots \\ Y^{-1}_{M,M}(\varphi_1, \vartheta_1) & Y^{-1}_{M,M}(\varphi_2, \vartheta_2) & \cdots & Y^{-1}_{M,M}(\varphi_N, \vartheta_N) \end{bmatrix} \tag{2.29}$$

wobei die trivialen Sphärischen Harmonischen $Y^{-1}_{m,0} \equiv 0$ selbstverständlich nicht Bestandteil von $\vec{B}$ und $\mathbf{C}$ sind, auch in Matrixschreibweise dargestellt werden:

$$\vec{B} = \mathbf{C} \cdot \vec{p} \tag{2.30}$$

Dieses Geleichungssystem ist lösbar:

$$\vec{p} = \mathbf{C}^{-1} \cdot \vec{b} \qquad \text{für } M = N \tag{2.31}$$

Im Allgemeinen ist jedoch $M \neq N$. In diesem Fall können die Lautsprechersignale p_n mit Hilfe der Pseudoinversen

$$\begin{aligned} \mathbf{D} &= \mathrm{pinv}(\mathbf{C}) \\ &= \mathbf{C}^{\mathrm{T}} \cdot (\mathbf{C} \cdot \mathbf{C}^{\mathrm{T}})^{-1} \end{aligned} \tag{2.32}$$

gewonnen werden.

Damit die gesamte (Richtungs-)Information der ambisonic-Repräsentation in die Decodierung einfließt, muss Gl. (2.30) unterbestimmt sein, weshalb gelten muss:

$$N \geq L \tag{2.33}$$

Man beachte, dass hier nicht von einer regelmäßigen Aufstellung der Lautsprecher ausgegangen wurde, wie dies bei ambisonic-Betrachtungen oft der Fall ist. Mit dieser Methode kann jede Lautsprecheraufstellung, für die man die Pseudoinverse der Reencodierungsmatrix $\mathbf{C}$ bilden kann, verwendet werden.

Beschränkt man jedoch die Lautsprecheraufstellung auf regelmäßige Polyeder, so vereinfacht sich die Pseudoinverse der Encodierungsmatrix zu

$$\mathbf{D}_{\text{Projection}} = \frac{1}{N}\mathbf{C}^{\mathrm{T}} \tag{2.34}$$

Diese Decodierung kann auch auf ein unregelmäßiges Lautsprecherlayout angewendet werden. Man spricht dann von einer *Projektion* der ambisonic-Kanäle auf die Richtungen der Lautsprecher.

2.4.2 *in-phase* Decodierung

Der Ansatz

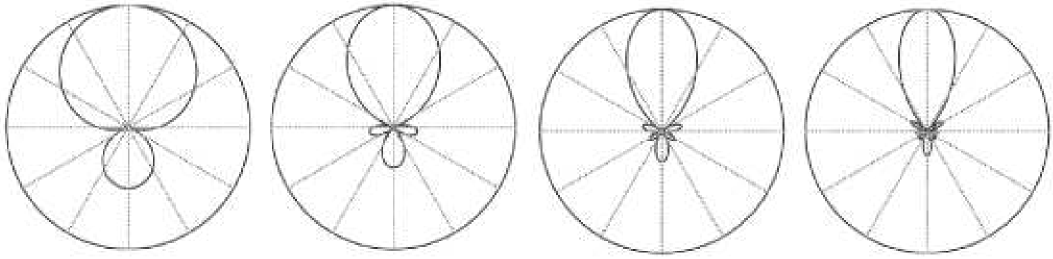
$$\mathbf{D}_{\{g_m\}} = \mathbf{D} \cdot \mathbf{\Gamma}_{\{g_m\}} \tag{2.35}$$

beschreibt eine allgemeine modifizierte Decodiermatrix. Die Modifikationsmatrix $\mathbf{\Gamma}_{\{g_m\}}$ ist eine Diagonalmatrix der Form:

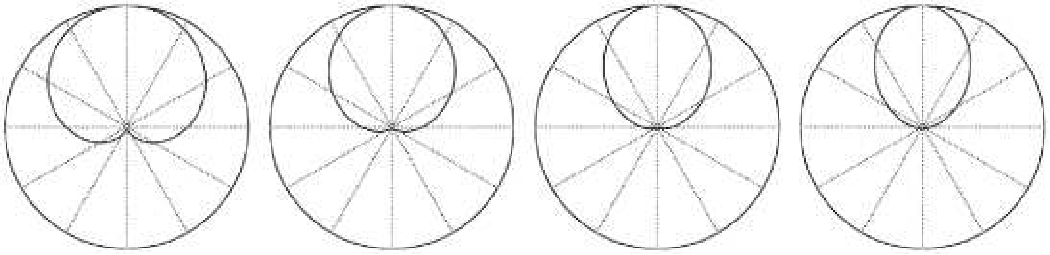
$$\begin{aligned} \mathbf{\Gamma}^{2D}_{\{g_m\}} &= \mathrm{Diag}(g_0, g_1, g_1, g_2, g_2, \cdots, g_M, g_M) \\ \mathbf{\Gamma}^{3D}_{\{g_m\}} &= \mathrm{Diag}(g_0, \cdots, \underbrace{g_m \cdots g_m}_{2m+1}, \cdots, \underbrace{g_M \cdots g_M}_{2M+1}) \end{aligned} \tag{2.36}$$

Eine Modifikation der *basic*-Decodiermatrix macht Sinn, um Unzulänglichkeiten der herkömmlichen Decodierung zu verringern.

In Gl.(2.10) wurde die Entwicklung der Ebenen Welle in eine Reihe vereinfacht, indem der Meßpunkt im Koordinatenursprung angesetzt wurde.



(a)



(b)

Abbildung 2.6:
Richtcharakteristik für (a) *basic*- und (b) *in-phase*-Decodierung

Durch diese Vereinfachung wird die im letzten Kapitel vorgestellte Methode das Schallfeld nur in eben diesem Bezugspunkt korrekt resynthetisiert. Dies ist aber unerwünscht, wenn eine Beschallungsanlage für ein größeres Auditorium zufriedenstellende Höreindrücke liefern soll.

Die Lautsprechersignale bei der *basic*-Decodierung können zueinander gegenphasig sein. Theoretisch würden sich solche gegenphasigen Signale bei der Superposition der Ebenen Wellen im Bezugspunkt auslöschen. Dies ist jedoch unter anderem durch Beugungseffekte um den Kopf der Hörerin in der Praxis nicht der Fall.

Weiters liefern auch Lautsprecher, die gegenüber der virtuellen Quelle liegen, einen Beitrag zum Synthese-Feld. (siehe Abb. 2.6) Ist nun die Abhörposition in der Nähe eines solchen Lautsprechers, so wird durch den Haas-Effekt die Lokalisation stark gestört.

Dem kann abgeholfen werden, indem man es zu einer Bedingung der De-codiervorschrift macht, dass alle Lautsprechersignale in Phase sind.

Eine solche Lösung mit Hilfe von Decodierung über eine Projektionsmatrix (s. Gl.(2.34) wurde von Malham [28] für zweidimensionale Wiedergabesysteme vorgeschlagen und von Daniel [11] für den dreidimensionalen Fall verallgemeinert:

$$g_m^{3D} = \sqrt{\frac{N(2M+1)}{M+1} \frac{M!(M+1)!}{(M+m+1)!(M-m)!}} \tag{2.37}$$

Die Decodierungsgleichung lautet dann:

$$\vec{p} = \frac{1}{N}\mathbf{C}^{\mathrm{T}} \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ 0 & g_1 & 0 & \cdots & 0 \\ 0 & 0 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & g_M \end{bmatrix} \vec{B} \tag{2.38}$$

2.5 Fehlerbetrachtungen

Eine Methode zur Evaluierung der Güte eines resynthetisierten Schallfeldes ist der *Integrated Wavefront Error*, der die Differenz zwischen Analyse- und Synthese-Schallfeld über eine Kreis- bzw. Kugeloberfläche integriert. Nach [1] wird dieser Fehler definiert:

$$D = \frac{1}{|2\pi P_\Psi|} \int_0^{2\pi} |S_\Psi - S^\Sigma| d\phi \tag{2.39}$$

Für den 3D-Fall kann dies erweitert werden zu

$$D = \frac{1}{|4\pi P^2_{\Psi,\Theta|}} \int_0^{2\pi} \int_{-\pi/2}^{+\pi/2} |S_{\Psi,\Theta} - S^\Sigma| d\theta d\phi \tag{2.40}$$

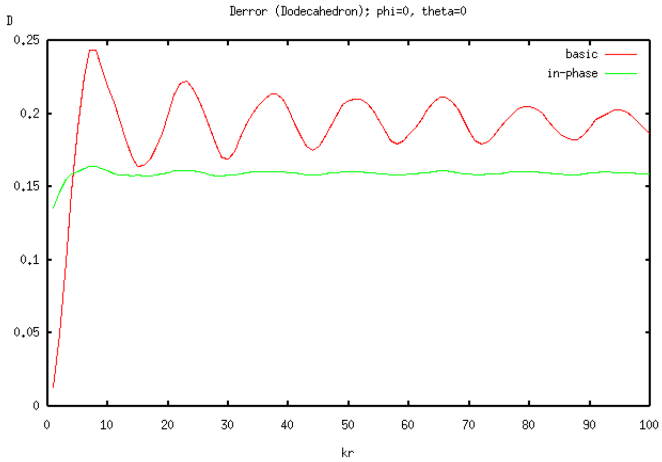
Durch die Normalisierung $|4\pi P^2_{\Psi,\Theta|}$ (bzw. $|2\pi P_\Psi|$) ist bei fehlendem Synthese-Schallfeld ($S = 0$) der Fehler $D = 1$.

Dabei ist für Beschallungsanlagen für ein größeres Publikum vor allem der Verlauf des Fehlers in Abhängigkeit von Frequenz und Entfernung vom Mittelpunkt interessant:

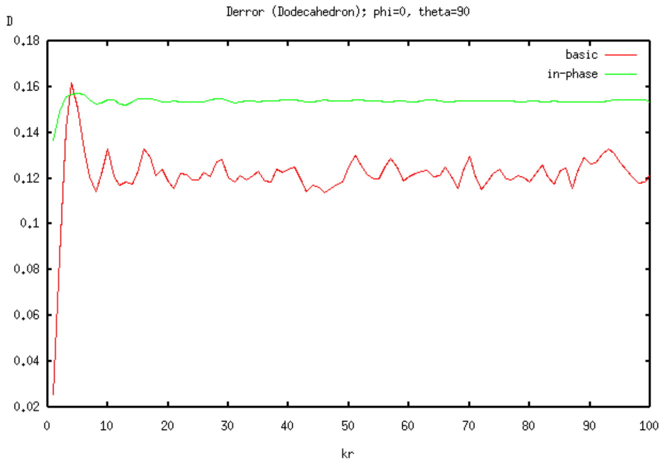
$$D = D(kr)$$

Bei der Gegenüberstellung von regelmäßigen (Dodecahedron, Abb. 2.7) und unregelmäßigen (IEM-cube, Abb. 2.8) Lautsprecherlayouts fällt dann vor allem auf, dass der Fehler D_{basic} bei der *basic*-Decodierung stark vom kr und von der Einfallsrichtung der analysierten/synthetisierten Schallquelle abhängt. Im Gegensatz dazu, ist der Fehler D_{inphase} nahezu konstant.

Durch die Fähigkeit ein Schallfeld überall in etwa gleich gut abbilden zu können, eignet sich die *in-phase*-Decodierung also besser für große *sweet areas*.

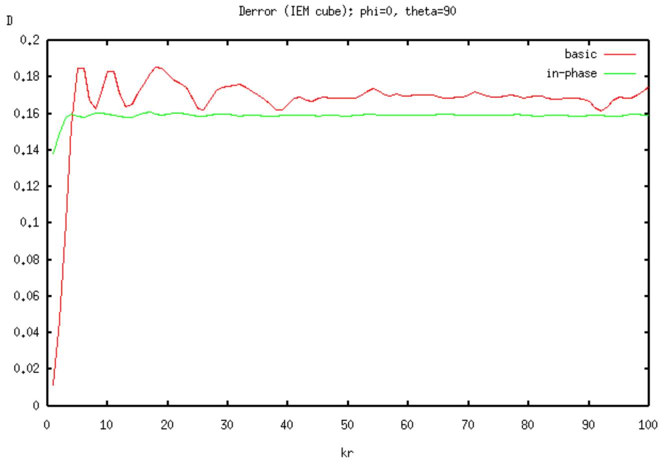


(a)

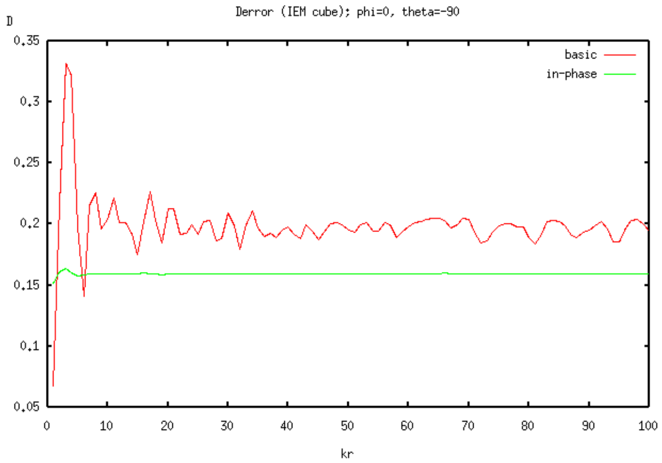


(b)

Abbildung 2.7: D-Fehler bei regelmäßigem LS-Layout (Dodecahedron) für verschiedene Einfallsrichtungen; ambisonic 2.Ordnung



(a)



(b)

Abbildung 2.8: D-Fehler bei unregelmäßigem LS-Layout (IEM-cube) für verschiedene Einfallsrichtungen; ambisonic 2.Ordnung

Kapitel 3

Hardware

3.1 Anforderungen

Die Anforderungen an die Hardware wird bestimmt durch die Forderung nach hoher, gleich bleibender Klangqualität für alle Wiedergabekanäle. Das Equipment sollte auf Grund folgender Kriterien ausgewählt werden:

- linearer Frequenzgang im hörbaren Bereich
- Dynamik
- Signal-Rauschabstand
- ausreichend große Anzahl an Kanälen
- Sicherheit

Die anfallende Signalverarbeitung wird in der digitalen Domäne durchgeführt werden.

Sinnvollerweise kann also die Beschallungseinheit in drei Teile geteilt werden (siehe Abb. 3.1):

- *Eingangsstufe* (analog/digital): Die vorliegenden (analogen oder digitalen) Signale werden der Verarbeitungsstufe in digitaler Form angeboten.

- *Verarbeitungsstufe* (digital): Diskrete virtuelle Quellen (auch virtuelle Lautsprecher bei der Wiedergabe zum Beispiel von herkömmlichen Stereoaufnahmen) müssen zuerst in ein ambisonic-Format codiert werden (Dies kann auch getrennt vom eigentlichen Wiedergabeprozess geschehen). Anschließend wird die ambisonic-Repräsentation auf die reale Lautsprecher-Konfiguration decodiert.

- *Wiedergabestufe* (analog): Die einzelnen diskreten Lautsprechersignale, die vom ambisonic-Decoder errechnet wurden, werden auf einer Lautsprechermatrix ausgegeben.

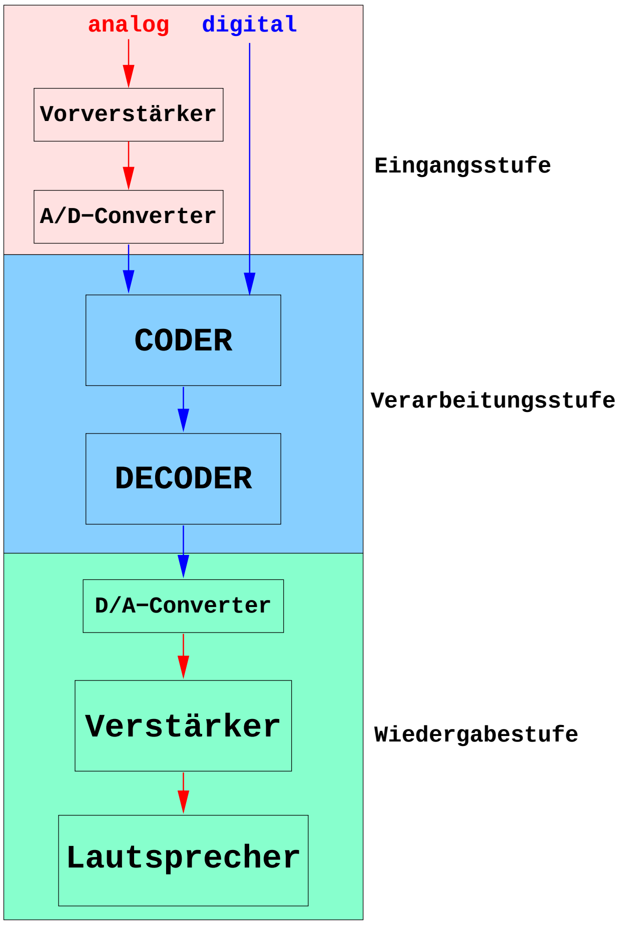


Abbildung 3.1: Hardware-Grundstruktur

Zur Übertragung der vielen Audiokanäle stellt sich in der digitalen Domäne die Frage, welches Mehrkanal-Übertragungsformat verwendet werden sollte:

- *TDIF*: Das von der Firma *Tascam* faßt acht digitale Kanäle nach AES/EBU-Norm in einem geschirmten Multicore-Kabel mit einem Sub-D-Stecker zusammen.
- *ADAT-optical*[1]: kann ebenfalls acht Kanäle übertragen und ist von der Firma *Alesis* entwickelt worden. Dabei werden alle acht Kanäle nach einem modifizierten S/P-DIF-Format über ein einziges Optokabel nach *Toslink*-Spezifikation übertragen.
- *MADI*: Dieser Mehrkanalstandard der AES mit mehr als 56 Audio-Kanälen konnte sich bis jetzt nicht durchsetzen. Es gibt daher kaum Geräte am Markt, die *MADI* (Multichannel Digital Audio Interface) unterstützen.
- Von der Firma *Yamaha* wird derzeit ein Mehrkanalformat, das über eine IEEE1394-Verbindung übertragen wird, propagiert. Allerdings gibt es noch kaum Geräte, die dieses Format auch verwenden.

Da es am weitesten verbreitet und daher leicht erhältich ist, wurde das *ADAT*-Format ausgewählt.

3.2 Sicherheit

Da das Wiedergabesystem für akustische Aufführungen mit sehr großer Dynamik für Publikum geeignet sein soll, muss das System dermaßen abgesichert werden, dass es unter normalen Umständen zu keinen systembedingten gesundheitsschädigenden Lautstärken kommen kann.

[1]Auch als *Alesis fiberoptic multichannel*-Format bekannt. Im Folgenden kurz *ADAT*-Format oder *ADAT* genannt.

Es muss aber auch die Möglichkeit des Ausfalls einzelner Komponenten einkalkuliert werden. Sollte ein Bestandteil der analogen Wiedergabekette ausfallen, wird dies im Allgemeinen keine Sicherheitsprobleme bereiten, da es meist in den wohldefinierten Zustand "stumm" übergeht.[2]

3.2.1 Synchronisation

Digitale Geräte müssen untereinander synchronisiert sein, damit sichergestellt werden kann, dass durch voneinander abweichende Taktfrequenzen keine Samples verloren gehen. Fällt ein synchronisierendes Gerät aus, so kann dies allerdings bei den (nicht mehr) synchronisierten Geräten zu undefiniertem Verhalten führen, da diese nun versuchen, sich auf ein "Nicht-Signal" zu synchronisieren. Undefiniertes Verhalten und Synchronisationsverlust sind im günstigsten Fall als "Knacksen" und "Prizzeln" hörbar. Im schlimmsten Fall wird digitales Rauschen mit Vollpegel generiert.

Im Hinblick auf das aufzuführende Material muss die Wiedergabeeinheit so dimensioniert werden, dass die gesamte Dynamik des menschlichen Hörvermögens, die für Musik 110dB_A beträgt[3][15], wiedergegeben werden kann. Da das erwähnte "Knacksen" oder Rauschen im schlimmsten Falle gleichzeitig mit Vollpegel von allen Lautsprechern wiedergegeben wird, können dabei Schalldrücke entstehen, die die Schmerzgrenze übersteigen und zu irreparablen Schädigungen des Gehörs führen. Dies ist unter allen Umständen zu verhindern.

Das problematischste Glied der Beschallungsanlage ist der D/A-Converter, da nur sein Synchronisationsverlust direkt über die Lautsprecher wiedergegeben wird. Glücklicherweise erkennen hochwertige Converter einen Synchronisationsverlust, und schalten dann den analogen Ausgang auf stumm.

[2]Es kann allerdings durch Feedback-Schleifen auch zum Aufschwingen eines analogen Systems kommen. Dies kommt jedoch praktisch so selten vor, dass diese Fälle vernachlässigbar sind.

[3]Die Schmerzgrenze liegt höher, bei 120dB_A

3.2.2 Maximalpegel

(Experimentelle) elektro-akustische Musik fordert oft einen großen Dynamikbereich ein. Dabei kann das produzierte Schallfeld durchaus Lautstärken aufweisen, die bis an die Grenze des Hörvermögens von Menschen mit gesunden Ohren heranreichen.

Hinzu kommt, dass bei einer mehrkanaligen Beschallungsanlage sich im schlimmsten Fall die Amplituden der einzelnen Wiedergabekanäle im Hörraum addieren.

$$\begin{aligned} L_{\text{Gesamt}} &= 20\log\frac{Kp_{\text{EinzelLS}}}{p_0} = \\ &= 20\log K + L_{\text{EinzelLS}} < 110\text{d}B_A \end{aligned} \tag{3.1}$$

Es ist daher darauf zu achten, dass das Signal eines einzelnen Lautsprechers bei digitaler Vollaussteuerung(!) einen Pegel

$$L_{\text{LS,max}} = 120 - 20\log K$$

nicht übersteigt.

Da Hörschäden schneller bei länger einwirkenden Pegeln entstehen, empfiehlt sich die Verwendung von nichtlinearen Modulen (Limiter). Die Limiterbank soll für alle Kanäle so eingestellt werden, dass kurzfristige Pegelspitzen weniger gedämpft werden als lang anhaltenden hohe Pegel.

Die Limiter dienen nur der Sicherheit und verändern den Wiedergabedynamikbereich so wenig wie möglich.

3.3 Hardware-Module

3.3.1 DSP-Modul

Das Herz der Beschallungsanlage, die Coder/Decoder-Einheit, wird in der digitalen Domäne realisiert. Dabei bieten sich auf der Hardwareseite zwei verschiedene Herangehensweisen an, nämlich die Realisierung auf einem proprietären Hardware-Modul (DSP-board) oder auf einer allgemeinen Hardware-Plattform (PC).

proprietäre Hardware vs. PC

Bis vor wenigen Jahren konnte digitale Signalverarbeitung nur auf spezieller Hardware durchgeführt werden. Diese sogenannten "DSP-boards" bestehen aus kleinen hochspezialisierten Rechner-Einheiten. Der Vorteil dieser DSP-boards besteht darin, dass durch die optimierte Zusammenarbeit von Hard- und Software Prozesse ausgeführt werden können, die auf herkömmlichen PC-basierten Systemen nur mit ungleich höherer Rechen-Performance möglich sind.

Daraus ergibt sich aber auch zwangsläufig ein Nachteil von DSP-Systemen: Durch die hohe Optimierung bleiben Hard- und Software eng miteinander verbunden und sind kaum zu entflechten. (Maschinen)Code, der für ein DSP-System geschrieben wurde, läuft höchstwahrscheinlich nur auf diesem DSP-System. Durch den hohen Grad an Optimierung und Hardwarenähe ist ein hoher Entwicklungsaufwand notwendig, der resultierende Code ist (meist) schlecht les- und wartbar. Solche Systeme sind wenig modular, und dadurch schwer skalier- und erweiterbar.

PC-Systeme haben im Gegensatz dazu bei gleichen Kosten weniger Rechenleistung, bieten aber den Vorteil, durch höhere Programmiersprachen (zum Beispiel graphische Programmiersprachen ´a la *pure data*), Code in einer Hardware-unabhängigen Form zu schreiben, wodurch dieser besser zu lesen und warten wird. Hard- und Software können unabhängig von einander entwickelt, erweitert und erneuert werden, wodurch ein solches System den aktuellen Entwicklungen besser angepasst werden kann.

Während auf PCs mit allgemeinen Entwicklungswerkzeugen gearbeitet werden kann, ist man bei proprietärer Hardware auf die (lizensierte) Entwicklungsumgebung des Herstellers angewiesen.

Bei proprietärer Hardware wird im Allgemeinen noch ein Steuer-PC als BenutzerInnen-Oberfläche benötigt. Bei PC-Systemen kann diese getrennte Aufgabe natürlich von der selben Hardware (und eventuell auch Software), die die DSP-Aufgabe rechnet hat, übernommen werden.

Waren allgemeine Hardwarelösungen bis vor wenigen Jahren noch zu kostspielig, sind mittlerweile herkömmliche PCs leistungsfähig genug, um damit ernsthaft mehrkanalige Audioverarbeitung betreiben zu können.

Mit Augenmerk auf die rasche Weiterentwicklung und die verfallenden Preise am PC-Sektor, wurde daher entschieden, die Spatialisierung auf einem PC-System zu realisieren.

Audiokarte

Nach der Entscheidung für ein PC-basiertes System, ergibt sich folgerichtig die Notwendigkeit einer mehrkanaligen Audiokarte, die folgenden Ansprüchen gerecht wird:

- mindestens 16-kanalige Ausgabe (für periphones *ambisonic* dritter Ordnung)
- mehrkanaliger Eingang
- geringe THD+N (zum Beispiel digitale I/O-Schnittstelle mit 24bit Wordbreite)
- Master/Slave-Betrieb (Synchronisation)
- Möglichkeit geringer Latenzzeit
- geringe CPU-Belastung durch die Hardware
- Treiber für das verwendete Betriebssystem vorhanden

Diese Anforderungen werden alle von der Audiokarte *RME-9652 Project Hammerfall*[4] erfüllt:

- 26 Ein-/Ausgänge, ausgeführt als
  - 3 ADAT optical format
  - 1 S/P-DIF

[4]Im Folgenden *Hammerfall-Audiokarte* oder *RME-9652* genannt.

- Wordclock-Ein-/Ausgang
- 24bit Auflösung
- Latenz: Die Durchlaufzeit wird maßgeblich von den Spezifikationen des PCI-Busses und der verwendeten Software (inklusive Treiber) begrenzt.
- Da die Karte PCI-Master-fähig ist und daher via DMA[5] direkt in den Hauptspeicher des PCs schreiben (bzw. daraus lesen) kann, ist die CPU-Belastung minimal.
- Treiber für alle gängigen PC-Plattformen (Linux, Mac, Windows) vorhanden

3.3.2 AD/DA-Converter

Da die Audiokarte *RME-9652* 24 Ein-/Ausgabekanäle im ADAT-Format (optische Übertragung) zur Verfügung stellt, werden entsprechende Umsetzereinheiten benötigt.

- digital(ADAT) <-> analog(line Pegel) Umsetzer
- 24bit Unterstützung
- Master/Slave-Betrieb
- definiertes Verhalten im Slave-Betrieb bei Synchronisationsverlust
- hohe Dynamik
- geringe THD(+N)
- geringes Kanalübersprechen in der analogen Domäne
- linearer Frequenzgang
- Speicherung der Einstellung auch bei Stromausfall

Eine Umsetzereinheit die alle diese Kriterien erfüllt ist die *RME ADI-8DS*.

[5] Digital Memory Access

3.3.3 Limiter

Zwischen D/A-Converter und Endstufe soll aus Sicherheitsgründen eine Limiterbank geschalten werden. Dieser soll folgende Eigenschaften aufweisen:

- einstellbarer Threshold für Dauerpegel
- getrennt einstellbarer Threshold für Spitzenpegel

3.3.4 Lautsprecher-Aufstellung

Um eine möglichst gleichmäßig gute ambisonic-Decodierung zu ermöglichen, sollte Clusterbildung vermieden und die Lautsprecher möglichst an den Knoten eines regelmäßigen Polyeders angeordnet werden.

Aus baulichen Gründen ist es meist schwer, Lautsprecher entlang einer Kugeloberfläche zu montieren, in deren Mittelpunkt sich die Zuhörerschaft befindet. Es macht aber Sinn, die Lautsprecher auf einer Halbkugel über den Köpfen des Publikums anzubringen.

Da das menschliche Gehör eine bessere Lokalisationsfähigkeit in der Horizontalebene hat, ist es psychoakustisch sinnvoll, in der Horizontalen relativ mehr Lautsprecher unterzubringen als in der Höhe.

Die Anzahl der verwendeten Lautsprecher richtet sich meist nach praktischen Kriterien, wie bauliche Gegebenheiten, zur Verfügung stehende (Übertragungs-)Kanäle und natürlich dem Budget.

Bei Verwendung einer *Hammerfall-Audiokarte* stehen 24 Kanäle im ADAT-Format zur Verfügung. Damit lässt sich ein ambisonic-Schallfeld 3. Ordnung resynthetisieren.

Eine einfache Anordnung wäre, die 24 Lautsprecher in drei (in sich symmetrischen) Ringen anzuordnen, wobei die Anzahl der Schallwandler pro Ring nach oben hin abnimmt. Dadurch vereinfacht sich das nicht-triviale Problem der Findung einer regelmäßigen Anordnung für eine quasi-beliebige Anzahl von Knoten. Die größte Regularität findet man dann mit Ringen zu 12, 8 und 4 Lautsprechern.

Ein solches Layout wurde zum Beispiel im *Cube* des IEM's realisiert (siehe Abb. 3.2 und 3.3). Da das menschliche Gehör "vorne" die kleinste Lokalisationsunschärfe aufweist, wurden in dieser Richtung auch die Lautsprecher enger positioniert (siehe auch App. A.3)

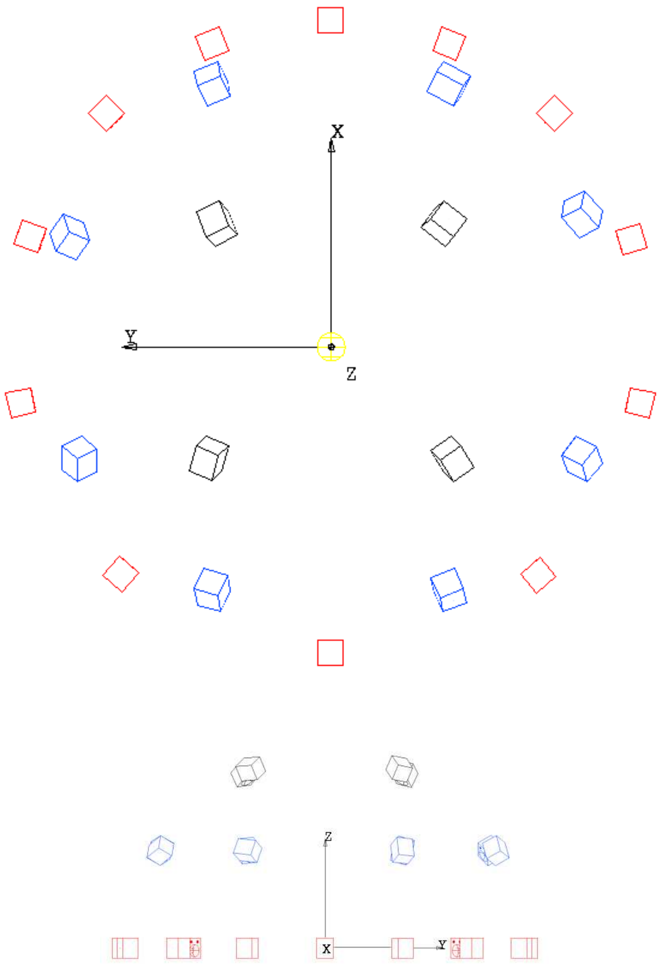


Abbildung 3.2: *IEM Cube*: Grund- und Seitenriss

Die Minimalvariante für ambisonic 3. Ordnung benötigt 16 Lautsprecher. Eine gute Verteilung der Lautsprecher erhält man, wenn man sie zu Ringen mit 8, 5 und 3 Ecken anordnet.

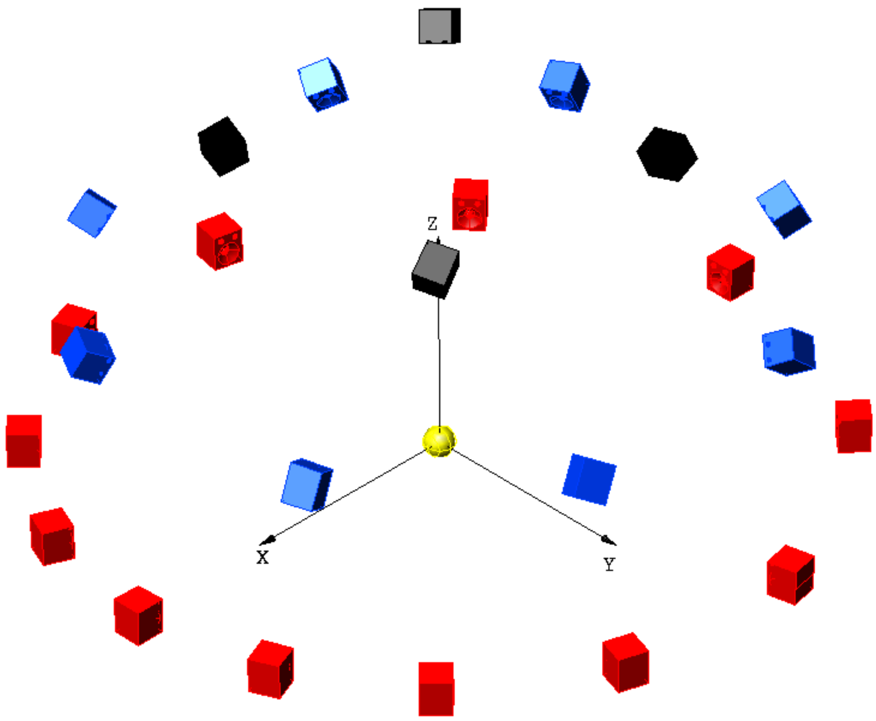


Abbildung 3.3: *IEM Cube*: Perspektive Darstellung

Aufgrund des verwendeten 8-kanaligen ADAT-Formates wird die Lautsprecheranzahl aus dem Wunsch nach einer möglichst guten Ausnützung aller Module zumeist ein Vielfaches von acht sein.

3.3.5 Verstärker

Die Lautsprecher werden von Endstufen getrieben, für die gilt, dass sie möglichst gut an die Lautsprecher angepasst sein sollten.

Kapitel 4

Systemsoftware

4.1 Betriebssystem

An das Betriebssystem werden folgende Anforderungen gestellt:

- Unterstützung von verwendeter Hardware
- Vorhandensein von echtzeitfähiger DSP-Software
- Treiber für gewählte Audiokarte verfügbar
- geringe Latenzzeiten möglich
- einfache Prozessinterkommunikation, wodurch Programme auf einem PC oder über mehrere Rechner verteilt verwendet werden können
- offene Architektur, um alle Vorzüge des Betriebssystems auch nutzen zu können

handelsübliche Hardware Da die entwickelte Software auf *handelsüblicher Hardware* funktionieren sollte, werden nur Betriebssysteme für ebensolche Hardware-Plattformen in Betracht gezogen. Es sind dies:

- *Windows*: PC-Plattformen
- *Linux*: PC- (und Alpha-) Plattformen

- *MacOS*: Macintosh-Plattformen

Audiokartentreiber Im Lieferumfang der *Hammerfall*-Audiokarte sind Treiber für die Betriebssysteme *Windows* und *MacOS* enthalten.

Für die Plattform *Linux* sind sowohl freie OSS-Treiber (*Open Sound-System*) als auch ALSA-Treiber (*Advanced Linux Sound-Architecture*) vorhanden.

Für andere System-Plattformen stehen (noch) keine Treiber zur Verfügung.

Echtzeitverhalten – Latenzzeit Nach einer Untersuchung von MacMillan et al. [14] ergaben sich für die Betriebssystem *Windows 2000*, *Linux 2.4.x* (Testplattform bei beiden: PIII-933MHz) und *MacOS 9.04* (Testplattform G4-500MHz) mit Verwendung von professioneller Mehrspur-Audiohardware (unter *Windows* und *Linux* die *RME Hammerfall*-Audiokarte, unter *MacOS* die *MOTU 2408*) folgende Latenzzeiten:

OS	Treiber	Latenz (no load)	Latenz (heavy load)
Linux	ALSA	4.30 ms	4.30 ms
MacOS	ASIO	6.8 ms	6.8 ms
Windows	ASIO	3.11 ms	6.03 ms
Windows	MME	11.45 ms	245.17 ms

Trotz der sehr guten Performance aller drei Betriebssysteme ist bemerkenswert, dass *Linux* und *MacOS* die Latenzzeiten auch bei starker CPU-Belastung konstant halten konnten, während die Durchlaufzeit unter *Windows* stark von der Rechnerauslastung abhängig ist.

Netzwerkfähigkeit *UNIX*-basierte Systeme (wie *Linux* oder *MacOS-X*) erleichtern das gemeinsame Nutzen von Dateien. Vor allem das verteilte Rechnen auf mehreren Computern bietet Vorzüge bei Anwendungen, die aus Prozessorleistungsgründen nicht auf einem einzelnen PC ausführbar sind.

Conclusio

Als Entwicklerumgebung ist *Linux* den Anwenderumgebungen *Windows* und *MacOS* überlegen. Die hohe Netzwerkfähigkeit, die Tatsache, dass *Linux* bereits in einer der Umgebungen (*IEM*), in der die hier diskutierte Beschallungsanlage realisiert wird, verwendet wird, sowie die weite Verbreitung speziell im akademischen Bereich bestimmten die Entscheidung für *Debian/GNU Linux*.

4.2 Audiokarten-Treiber

Seit Kernel 2.4.9 sind die OSS-Treiber für die Audiokarte *RME – Project Hammerfall* im Betriebssystem-Kernel enthalten. ALSA-Treiber, die niedrigere Latenzzeiten ermöglichen, sind für diese Karte seit ALSA-0.9 inkludiert.

Das überlegene ALSA-System wird allerdings durch politische Grabenkämpfe innerhalb des Entwicklungsteams so weit geschwächt, dass die Zukunft dieses Projektes nicht abzusehen ist. Da sich aus diesen Gründen auch die API (*Advanced Programm Interface* der Treiber monatlich ändert, ist eine ernsthafte Software-Entwicklung auf der Grundlage des ALSA-Projektes derzeit nicht machbar.

Daher wurde die Beschallungsanlage auf Basis der OSS-Treiber entwickelt, die zur Zeit noch ein schlechteres Latenzverhalten zeigen. So ist unter OSS bei voller Belastung mit 26 Ein-/Ausgabekanälen eine minimale Latenz von 512 Samples (entspricht bei einer Abtastrate von 44.1kHz einer Latenzzeit von 11.61ms) möglich. Unter besonderen Umständen (zum Beispiel bei Verwendung von weniger Ein-/Ausgabekanälen) kann eine weitere Senkung der Latenz auf 256 Samples (entspricht 5.80 ms) erfolgen.

Da nur die Basissoftware *pure-data* (siehe Cap. 4.3) direkt mit den Treibern kommuniziert, ist es wahrscheinlich, dass in Zukunft, sobald die ALSA-Treiber stabilisiert sind und das Interface von *pd* angepasst ist, die Implementierung auch problemlos auf ALSA-Basis laufen wird und von der niedrigeren Latenzzeit profitieren kann.

4.3 Echtzeitcomputerprogramm *pure-data*

pure-data[1] ist eine graphische Echtzeit-Computermusikprogrammiersprache von Miller S. Puckette et al. in der Tradition von IRCAM's *ISPW-max*.

pd ist für die Betriebssysteme *Windows* (PC), *Irix* (SGI), und *Linux* (PC und Alpha) als *Open Source*-Projekt erhältlich.

pd stellt ein einfach handhabbares und mächtiges Framework zur Audio-Bearbeitung in Echtzeit zur Verfügung.

Wie alle *max*-Derivate und die meisten Echtzeit-Computermusikprogramme stellt auch *pd* dafür zwei Scheduling-Ebenen zur Verfügung:

- **Signal-Ebene**: In Echtzeit, synchron zu den Ein- bzw. Ausgabedevices der Audiokarte, werden Signale manipuliert (addiert, multipliziert, transformiert, ...).

- **Control-Ebene**: Asynchrone "Messages" werden bei Bedarf ausgeführt und können das Verhalten der Signal-Manipulation bestimmen.

Durch die Trennung in zwei Ebenen kann der Rechenaufwand minimiert werden: Nur die nötigsten Operationen werden in der Signal-Ebene für jedes Sample gerechnet, aufwendigere Berechnungen werden nur "on demand" in der Control-Ebene durchgeführt.

Signaloperationen müssen, damit sie synchron zur Audiokarte bleiben, fortlaufend berechnet werden. Die Prozessorauslastung ist also konstant, wodurch es leicht einschätzbar wird, ob Signalbearbeitung mit gegebener Komplexität auf einem gegebenem Prozessor möglich ist.

Da Messages bei Bedarf sofort bearbeitet werden, wird hier der Prozessor nur kurzzeitig belastet. Bei sehr aufwendigen Message-Operationen (die vor den Signaloperationen abgearbeitet werden, siehe Abb. 4.1) kann es geschehen, dass der Prozessor kurzfristig überlastet ist und den Signal-Schedule nicht einhalten kann. Bei genügend kleinem Signalbuffer kommt es daher zu

[1] Im Folgenden mit *pd* abgekürzt

einem Buffer-Unterlauf, der Ausgabebuffer konnte noch nicht angefüllt werden und ist daher entweder 0 oder hat noch die Werte des letzten Blocks gespeichert.

Es muss daher vom Programmierer darauf geachtet werden, dass die Messages zeitlich so verteilt auftreten, dass es zu keiner Zeit zu einer Überbelastung des Prozessors kommt.

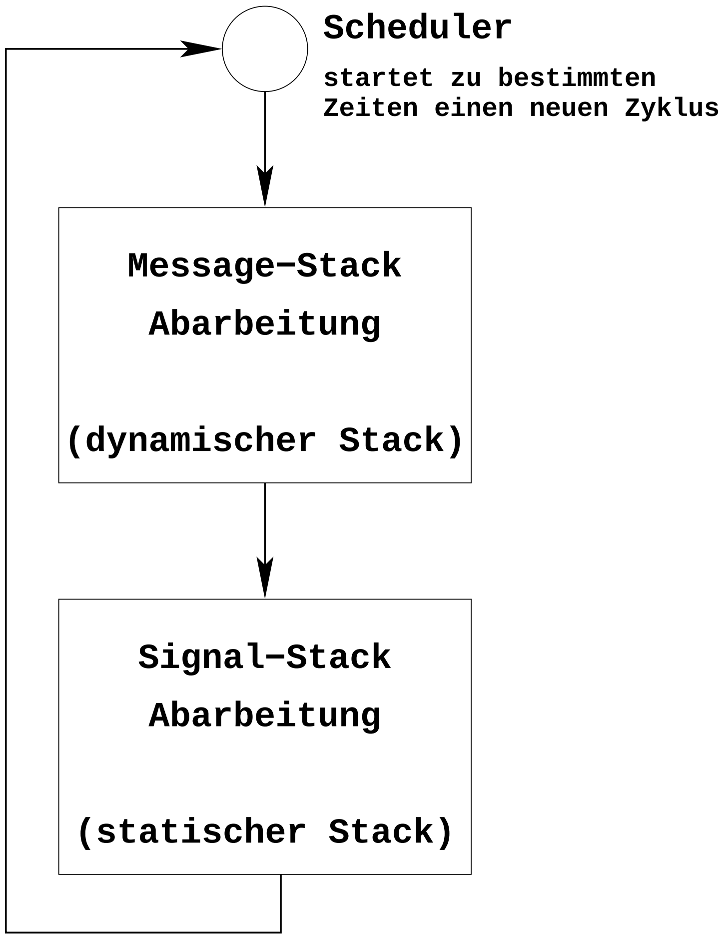


Abbildung 4.1: Scheduling Struktur von *pd*

Kapitel 5

Software-Entwurf

Die Audio-Berechnungen sind in Abhängigkeit von der Anzahl der Eingangskanäle, der Ordnung des ambisonic-Systems und der Lautsprecherkanäle relativ aufwendig.

Den BenutzerInnen sollten solche Details bei der Bedienung des 3D-Mixers verborgen bleiben.

Durch Auftrennung der Software in Audio-Engine und User-Interface ergibt sich eine mehrschichtige Architektur (s. Abb. 5.1

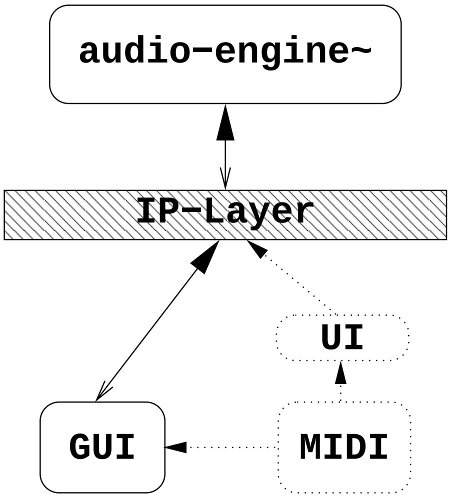


Abbildung 5.1: Software-Architektur

Das User-Interface (egal ob ein 3D-Mixer-Programm, ein MIDI-Interface oder dergleichen) ist also von der Audio-Engine völlig abgetrennt und kommuniziert mit dieser ausschließlich über ein beschränktes Kommunikationslayer.

5.1 Audio-Engine

In der *Audio-Engine* werden die eigentlichen Signal-Manipulationen durchgeführt.

Diese rechenaufwendige Aufgabe soll bei Bedarf auf mehrere PCs verteilt werden können.

5.1.1 Aufgaben

Zu ihren Aufgaben gehören:

- *Codierung* von virtuellen Schallquellen mit Positionsangaben in ein ambisonic-Schallfeld
- *Schallfeldmanipulationen* (Rotation, Lautstärke)
- virtuelle *Raum-Akustik* (Verhallung)
- Regelung der *Gesamtlautstärke* und der Einzellautstärken
- *Decodierung* eines ambisonic-Schallfeldes auf eine gegebene Lautsprecheranordnung
- automatische *Angleichung* der einzelnen Lautsprecherkanäle (Phasen- und Amplitudenausgleich)

5.1.2 Struktur

Sinnvollerweise lässt sich die Audio-Engine in zwei Teile auftrennen, eine CODER-Einheit und eine DECODER-Einheit.

Dadurch ist es auch möglich, nur die Coder-Einheit, zur Aufnahme eines Sets von ambisonic-Kanälen, oder nur die Decoder-Einheit, zur Wiedergabe einer vorher-aufgenommenen ambisonic-Repräsentation, zu verwenden.

Zwischen Coder und Decoder müssen keine Steuerdaten ausgetauscht werden, die Übertragung erfolgt ausschließlich unidirektional über Audio-Kanäle (deren Anzahl natürlich von der verwendeten Codierungsvorschrift (s. 2.2.2) abhängig ist).

Coder

Im Coder werden die einzelnen diskreten Quellen in ein ambisonic-Soundfield gerendert.

Zu seinen Aufgaben gehören demnach:

- *Lautstärkenmanipulation* der einzelnen Quellsignale durch Multiplikation mit Verstärkungsfaktoren α_n

$$\begin{aligned} \vec{s}' &= \mathbf{A} \cdot \vec{s} \\ &= \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \alpha_K \end{bmatrix} \cdot \vec{s} \end{aligned} \tag{5.1}$$

- *Rendering der (statisch) positionierten Quellen* in das ambisonic-Soundfield durch Multiplikation der Quellsignale mit den Sphärischen Harmonischen $Y_{m,\eta}^{\sigma}(\varphi_k, \vartheta_k)$ und anschließender Superposition der so gewonnenen ambisonic-Kanäle. Die Entfernungsbewertung erfolgt über die Pegelabnahme nach dem $\frac{1}{r}$-Gesetz und die tiefpassähnlich Luftdämpfung

- *Rendering der (dynamischen) Quellenbewegung*: Bewegte Quellen verändern nach dem Doppler-Effekt ihre Tonhöhe.

- künstliche *Raumsimulation*

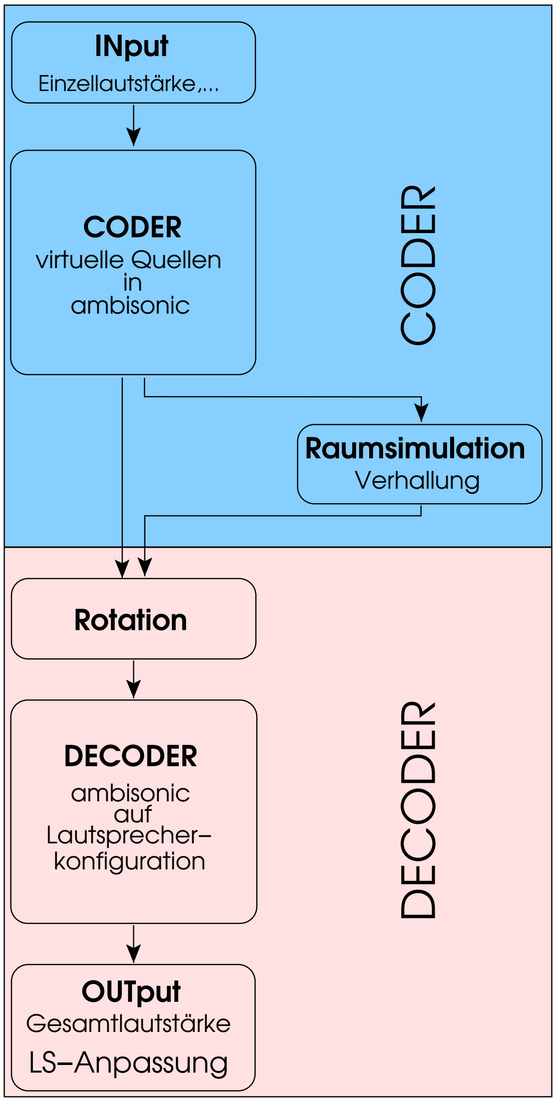


Abbildung 5.2: Aufteilung der Audio-Engine in zwei Module mit mehreren Submodulen

Der eigentliche Coder lässt sich nach Operationstypen in zwei Teile zergliedern:

- *lineare Manipulationen*: Lautstärkenmanipulation (für K Eingangskanäle) und ambisonic-Encodierung (von K Eingangskanälen auf L ambisonic-Kanäle) sind lineare Operationen, die durch Matrixoperationen ausgedrückt werden können

$$\begin{aligned} \vec{B} &= \mathbf{C} \cdot \mathbf{A} \cdot \vec{s} \\ \vec{B}^{\mathrm{T}} &= \vec{s}^{\mathrm{T}} \cdot \mathbf{A} \cdot \mathbf{C}^{\mathrm{T}} \end{aligned} \tag{5.2}$$

- *Manipulationen im Zeit-/Frequenzbereich*: Die Entfernungs-/ Bewegungsbewertung einer Mono-Quelle wird durch Filterung mit einem Tiefpass 1. Ordnung (Luftdämpfung) und Verzögerung (Doppler-Effekt) durchgeführt. Für jeden Eingangskanal muss eine eigene Entfernungs-/ Bewegungsbewertung durchgeführt werden.

Am Coder-Ausgang liegt also eine, von der Anzahl der Eingangskanäle unabhängige, Menge an Kanälen an, die das gesamte Schallfeld à la ambisonic repräsentiert. Es ist daher nach dem Coder nicht mehr möglich, die einzelnen Schallquellen getrennt zu manipulieren.

Der Coder-Ausgang kann auch als Mehrspuraufnahme aufgezeichnet werden. Es ist daher sinnvoll, dass an seinem Ausgang eine komplette Repräsentation eines musikalischen Werkes vorliegt.

Eine wesentliche Forderung an den Coder ist es, eine beliebige Anzahl von Eingangskanälen sowie verschiedene Codierungsvorschriften zu erlauben.

Verhallung

Für die Simulation von beliebigen Raumsituationen sind Verhallungsalgorithmen notwendig. Dabei wäre es eigentlich notwendig, für jede virtuelle Quelle einen eigenen Verhallungsalgorithmus zu implementieren. Dies ist aus Gründen der Rechenleistung (noch) schwer möglich.

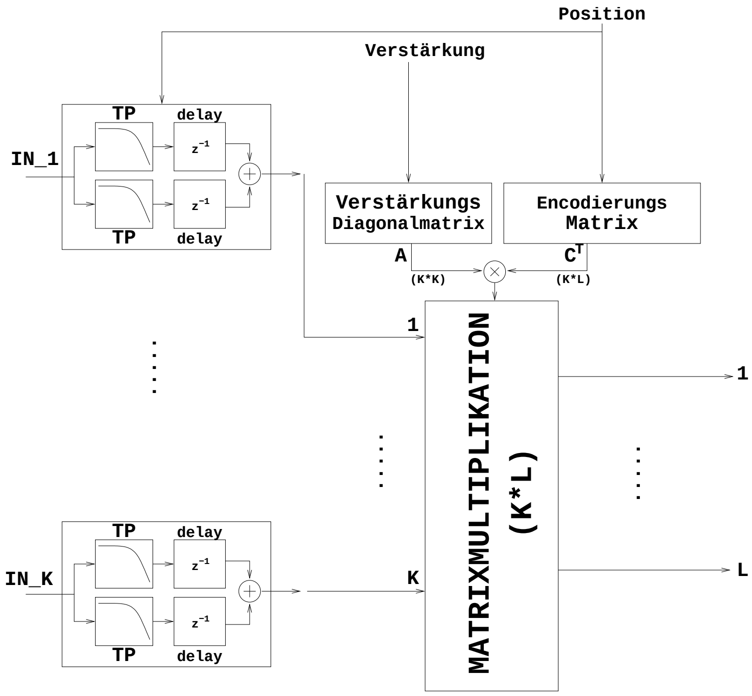


Abbildung 5.3: Struktur des Coders

Eine Alternative wäre, nur die ambisonic-Repräsentation des Schallfeldes zu verhallen, wodurch die aufzubringende Rechenleistung (in Abhängigkeit von der Codierungsvorschrift) unabhängig von der Anzahl der virtuellen Schallquellen konstant bleibt. Allerdings können in der ambisonic-Repräsentation die einzelnen Schallquellen nicht mehr getrennt manipuliert werden, wodurch die ersten Reflexionen nicht mehr korrekt simuliert werden können, da diese stark von der Position der jeweiligen virtuellen Quelle im Raum abhängig sind. Korrekt werden die ersten Reflexionen mit der Hilfskonstruktion der Spiegelquellen simuliert.

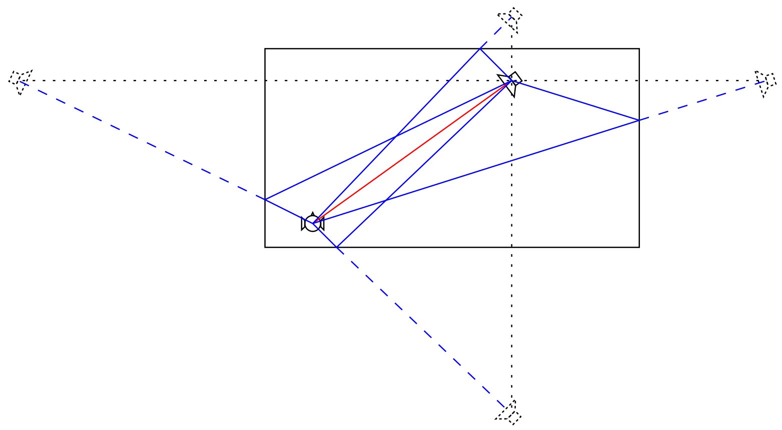


Abbildung 5.4: Simulation von Ersten Reflexionen mit Hilfe von Spiegelquellen in einem rechteckigen Raum

Wird der simulierte Raum von orthogonalen Ebenen begrenzt (Rechteck, Quader), so müssen für jede virtuelle Schallquelle zusätzlich vier (2D-Fall) oder sechs (3D-Fall) Spiegelquellen gerendert werden.

Unter der Annahme, dass sich der Zuhörer im Zentrum eines kugelförmigen Raumes befindet, reduziert sich die Anzahl der Spiegelquellen unabhängig von der Dimension auf eins. Die Richtung, aus der die Spiegelquelle strahlt, ist mit der Richtung, aus der die originale virtuelle Quelle kommt ident.

$$\begin{pmatrix}\varphi\\ \vartheta\end{pmatrix} \equiv \begin{pmatrix}\varphi'\\ \vartheta'\end{pmatrix}$$

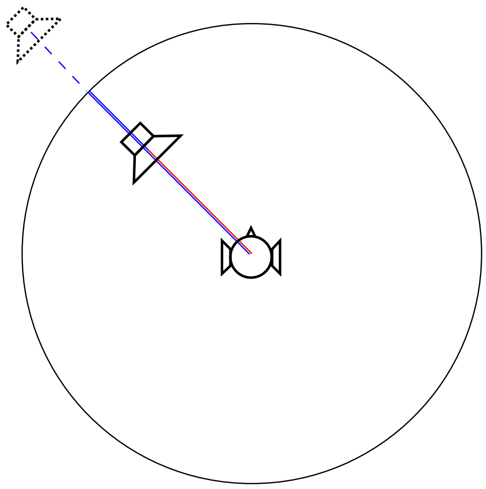


Abbildung 5.5: Simulation von Ersten Reflexionen mit Hilfe von Spiegelquellen in einem kugelförmigen Raum

Das diffuse Schallfeld ist unabhängig von der Position der einzelnen Quellen relativ zu den Begrenzungsflächen. Es kann daher gut als Manipulation der ambisonic-Gesamtrepräsentation implementiert werden.

- *einfache 3D-Verhallung*: ein geeigneter Mixdown der ambisonic-Kanäle (Decodierung) wird einem internen oder externen Hall-Algorithmus zugespielt. Das so gewonnene verhallte Signal wird recodiert und wieder dem codierten Soundfield zugeführt.

- *echter ambisonic-Hall*: (künstliche oder gemessene) dreidimensionale Raumimpulsantworten werden ambisonic-codiert und auf das gesamte Soundfield appliziert.[1]

Decoder

Im Decoder wird das ambisonic-Soundfield auf eine gegebene Lautsprechermatrix decodiert.

[1]Diese Methode wurde im Rahmen dieser Diplomarbeit nicht implementiert.

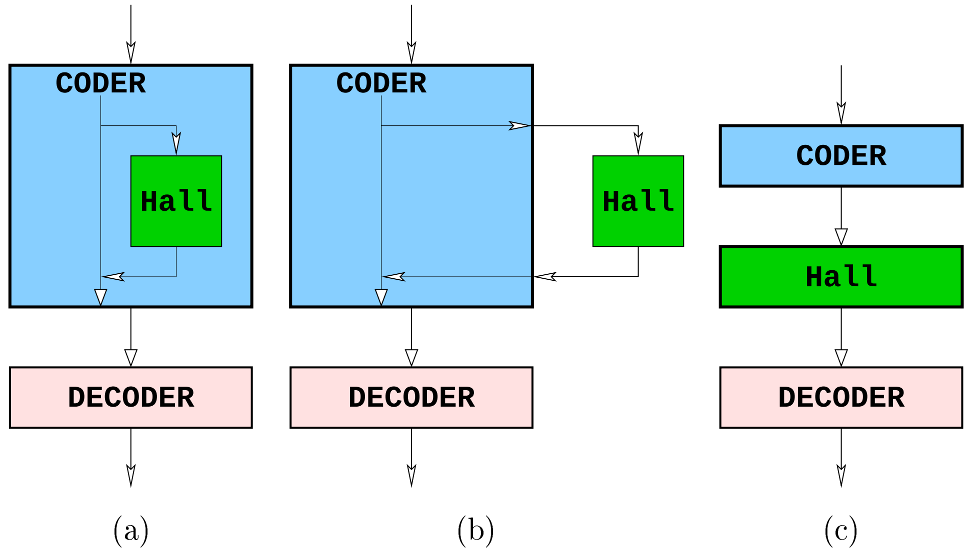


Abbildung 5.6: 3D-Verhallung: (a) interner Hall, (b) externer Hall, (c) echter ambisonic-Hall

Es ist daher wichtig, dass der Decoder für eine beliebige Codierungsvorschrift und eine beliebige Lautsprecheraufstellung die Decodierungsmatrix errechnen kann.

Weiteres können noch Manipulationen am gesamten Schallfeld, wie Rotation (s. 2.3.3) oder Gesamtlautstärkenanpassung, durchgeführt werden, um zum Beispiel mit einem fix montierten Lautsprecherarray auf aufführungstechnische Änderungen im Wiedergaberaum ("Vorne"-Orientierung) reagieren zu können.

Die gesamte Decoder-Funktionalität kann also durch Matrixoperationen ausgedrückt werden:

$$\vec{p} = \mathbf{D} \cdot \mathbf{R} \cdot \alpha \vec{B} \tag{5.3}$$

5.2 User-Interfaces

Bei der Benutzung der Beschallungsanlage sollten die Benutzenden mit einem (mehr oder weniger) intuitiv verständlichen Interface konfrontiert sein, ohne

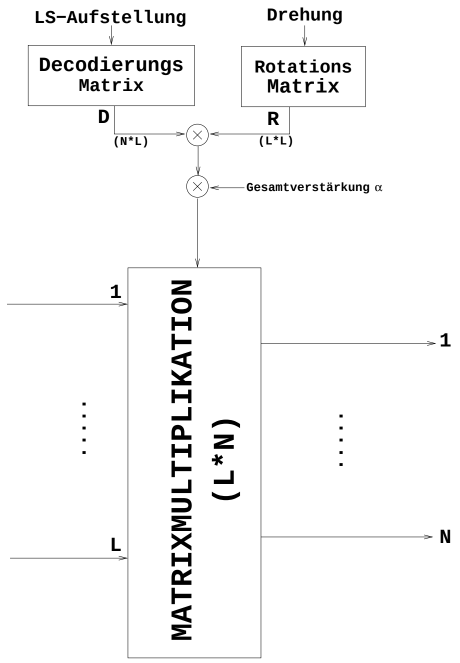


Abbildung 5.7: Struktur des Decoders

sich über die technischen Vorgänge des Renderings bewusst sein zu müssen.

Prinzipiell gibt es zwei Aufgaben an solche Interfaces:

- Abrufen von Presets
- frei (im Rahmen des Systems) konfigurierbarer periphoner Mixer

5.2.1 Presets

Eine sehr einfach zu bedienende Oberfläche soll die Möglichkeit bieten, Wiedergabekonfigurationen für Standardsituationen auf Knopfdruck zu aktivieren.

Zu den Standardsituationen zählen vor allem die Wiedergabe von traditionellen Aufnahmen:

- *Stereo*: Die (ersten) zwei Eingangskanäle werden auf ein virtuelles Lautsprecherpaar gerendert.

- *Dolby Surround*: Die sechs, von einem externen Dolby-Decoder aus einer zwei-kanaligen Surround-Aufnahme gewonnen Kanäle werden auf ebenso-vielen virtuellen Lautsprechern wiedergegeben.
- *Mehrspur-Aufnahmen*: Um traditionelle Mehrspur-Kompositionen abzuspielen, werden die Kanäle auf einem virtuellen Quadrat (bzw. Oktogon) wiedergegeben.
- *Mono*: Zum Beispiel kann eine Sprecherin über die Beschallungsanlage verstärkt werden, wobei die Quelle wahlweise gleichmäßig im ganzen Raum wiedergeben wird (wodurch sie nicht mehr lokalisierbar ist) oder als virtuelle Quelle auf eine bestimmte Position ("vorne") gerendert wird.

5.2.2 Anforderungen an einen 3D-Mixer

Ein komplexer 3D-Mixer, soll es erlauben, eine beliebige Positionierung und Bewegung von virtuellen Quellen vorzunehmen. Dadurch ist es möglich, periphone Kompositionen zu realisieren.

Zur einfachen Bedienung und Kontrolle ist es für einen 3D-Mixer angebracht, eine 3D-Visualisierung der virtuellen Quellen bereitzustellen.

5.3 Kommunikationslayer

Die Verbindung zwischen User-Interface und Audio-Engine erfolgt über ein Kommunikationslayer. Dieses beinhaltet prinzipiell Schnittstellen zwischen den drei Software-Modulen *Coder*, *Decoder* (Audio-Engine) und *(G)UI*.

5.3.1 Anforderungen

Folgende Punkte sind bei der Kommunikation zwischen Audio-Engine und BenutzerInnen-Schnittstelle zu beachten:

- *Stabilität* gegen Hardwareprobleme (Sender-/Empfängerausfall,...)

- *Verlässlichkeit* der Datenübertragung bezüglich verlorener und verfälschter Daten
- hohe *Geschwindigkeit* der Datenübertragung, da in der Echtzeitanwendung eines 3D-Mixers merkbare Latenzzeiten als störend wahrgenommen werden.
- *Kommunikation* zwischen Software-Modulen (nicht nur zwischen Hardware-Modulen), da Coder und Decoder (und eventuell auch eine graphische BenutzerInnen-Oberfläche) verteilt auf mehrere Computer oder auf dem selben Rechner laufen können

5.3.2 MIDI

Eine Möglichkeit zur Kommunikation zwischen den Modulen bietet MIDI[2]

Der Vorteil von MIDI liegt darin, dass es ein wohldefiniertes, erprobtes und stabiles Umfeld zur hardwareseitigen Verbindung und Steuerung (musikspezifischer) Geräte (wie Synthesizer, Mischpulte,...) bietet.

Allerdings ist MIDI für die Verbindung zwischen Hardware-Modulen ausgelegt und nicht zur Interkommunikation von Programmen. Über virtuelle Devices ist es zwar möglich, MIDI-Daten von einem Software-Modul in ein anderes zu routen, allerdings wird dadurch eine umständliche Fallbehandlung notwendig.

Komplexere Daten, also solche, die das Standarddatenformat von 7 Bit sprengen, können nur relativ aufwendig via MIDI übertragen werden.

5.3.3 Netzwerkverbindungen

Im Gegensatz zu MIDI, bieten Netzwerkverbindungen die Möglichkeit, Kontrolldaten zwischen Softwaremodulen auszutauschen.

Durch eingekapselte Protokolle können auch komplexe Datenstrukturen leicht übertragen werden.

[2] Musical Instrument Digital Interface

Diesen Metaprotokollen liegen Netzwerkprotokolle wie TCP/IP oder UDP zu Grunde, die wiederum auf das allgemeine Ethernet-Protokoll aufbauen.

TCP/IP

Das *Transfer Control Protocol / Internet Protocol* ist ein bidirektionaler Protokoll-Standard.

Dabei wird vom Protokoll sichergestellt, dass Datenpakete in der korrekten Reihenfolge empfangen werden. Dadurch ist eine hohe Sicherheit und Verlässlichkeit der Datenübertragung gewährleistet. Allerdings wird durch das dafür benötigten Handshaking dieses Protokoll relativ langsam.

Die bidirektionale Verbindung bedingt, dass Sender und Empfänger synchron gestartet und beendet werden, und nach Etablierung einer Verbindung online bleiben.

UDP

Das *User Datagram Protocol* ist ein standardisiertes unidirektionales Protokoll. Datenpakete können dabei umgeordnet, dupliziert oder gar nicht beim Empfänger ankommen.

Dadurch ist UDP sehr schnell. Die unsichere Datenübertragung ist bei Internet-Datentransfer über mehrere Hostrechner durchaus problematisch, in LANs[3] mit breitbandigen Verbindungen treten im Allgemeinen wenig Probleme auf.

5.3.4 Conclusio

Da MIDI sich schlecht für die Verbindung zwischen Software-Modulen eignet, wurde die Kommunikation auf Ethernet-Basis entworfen.

Da die vorgesehene Verwendung sich auf lokale Netzwerke, in denen im Allgemeinen die Bandbreite hoch und der Net-Traffic relativ gering ist, be-

[3] Local Area Networks

schränkt, kann auf die Verlässlichkeit von TCP/IP zugunsten der hohen Übertragungsgeschwindigkeit von UDP verzichtet werden.

Aus diesen Gründen wurde ein UDP-basiertes Ethernetprotokoll für die Steuerung der Audio-Engine gewählt.

Kapitel 6

Audio-Engine

6.1 Coder

Der Coder ist das zentrale Modul eines 3D-Mixers. Hier werden K Monoquellen s_n anhand von Verstärkungsfaktoren α_n und Positionsangaben $[r_n, \varphi_n, \vartheta_n]^{\mathrm{T}}$ in eine periphone ambisonic-Repräsentation unter Verwendung einer Codierungsvorschrift codiert. Anschließend kann das ambisonic-Schallfeld noch einer Verhallung zugeführt werden, um einen räumlichen Eindruck zu vermitteln.

```
coder~ 16 furse-malham
```

Abbildung 6.1: *pd*-Patch: Aufruf des coder˜-Moduls

Die zugehörige *pd*-Abstraktion benötigt daher zwei Argumente (siehe Abb. 6.1):

- die *Anzahl der Eingangskanäle*: z.B.: 16
- die zu verwendende *Codierungsvorschrift*: z.B. das Furse-Malham-Set

Berechnungen in der Kontrollebene

Durch das gewählte Übertragungsformat kann vom User-Interface die Position einer virtuellen Quelle nicht zu jedem (Sample-genauen) Zeitpunkt bestimmt werden. Stattdessen erfolgt die Positionsbestimmung zu diskreten Zeitpunkten in der Kontrollebene.

Da die Berechnung der ambisonic-Faktoren (sphärische Harmonische) durch die Anzahl an trigonometrischen Funktionen sehr rechenintensiv ist, liegt es nahe, diese Berechnungen in der Kontrollebene zu belassen und nicht in der Signal-Domäne durchzuführen.

Die Verstärkungsfaktoren für die einzelnen Eingangskanäle können dann gleich prozessorleistungssparend in der Kontrollebene in die Codierungsmatrix eingerechnet werden.

Die so gewonnen Faktoren werden mit den Eingangssignalen multipliziert. Damit keine Lautstärkesprünge durch die plötzliche Änderung in einem Kanal auftreten, muss zwischen den Faktoren übergeblendet werden.

Überblendung

Da diese Überblendung aus Recheneffiziengründen linear ist, wird die Richtung, die einem Faktoren-Set zugeordnet werden kann, durch die inversen sphärischen Harmonischen nichtlinear verzerrt. Da dies zu nicht gewollten Ergebnissen führt, ist die Richtungsänderungen zwischen den beiden Fixpunkten der Interpolation klein zu wählen, wodurch sich die Faktoren wenig ändern und die Verzerrung ebenfalls klein bleibt. Eine Bewegung entlang einer geschlossenen Linie (zum Beispiel Drehung um 90°) ist daher dem Coder nicht über Start- und Endpunkt (Φ und $\Phi + 90°$) mitzuteilen, sondern über viele Einzelpunkte der Linie ($\Phi, \Phi + 5°, \Phi + 10°, \cdots, \Phi + 90°$).

Das kurze Stück der Überblendung kann als Amplitudenmodulation des Eingangssignals mit einer Rampe gesehen werden. Dadurch kommt es zu einer störenden spektralen Veränderung des Eingangssignals.

Eine Möglichkeit, die spektrale Verschiebung möglichst klein zu halten, ist, die Überblendung so langsam wie möglich zu machen, sodass die Grund-frequenz des modulierenden Signals (der Rampe) unter die Hörschwelle des menschlichen Ohrs gedrückt wird. Dem sind jedoch durch die Forderung nach einem möglichst schnell reagierenden Systems Grenzen gesetzt.

Einen guten Kompromiss erhält man bei Überblendzeiten zwischen 100 und 50 Millisekunden ($\doteq 10 - 20$Hz)

Konstanter Kontrolldatenfluß

Werden dem Coder in einem kurzen Zeitraum sehr viele Kontrolldaten (wie Richtung(sänderung)en) geschickt, so fallen sehr schnell sehr viele komplexe ambisonic-Berechnungen an. Dies kann zu einer kurzzeitigen Überlastung der CPU führen, was, bei einem genügen kleinen Audiobuffer, zu einem Bufferunterlauf führen kann. Um dies zu verhindern, werden ankommende Steuerdaten in einen Puffer geschrieben, von wo sie mit einer konstanten Rate ausgelesen werden. Ändert sich ein Datensatz innerhalb eines Takt-Zyklus mehrmals, wird am nächsten Takt-Anfang nur der letzte gültige Wert ausgelesen und weiterverarbeitet. Durch das dauernde Auslesen tritt die Belastung des Systems durch die Kontrolldaten-Berechnungen nicht asynchron sondern gleichmäßig auf, wodurch sich das Systemverhalten besser evaluieren lässt.

Indem man verschiedene Berechnungen verzögert taktet, lässt sich die Rechenleistung besser über die Zeit verteilen.

6.1.1 Entfernung und Bewegung

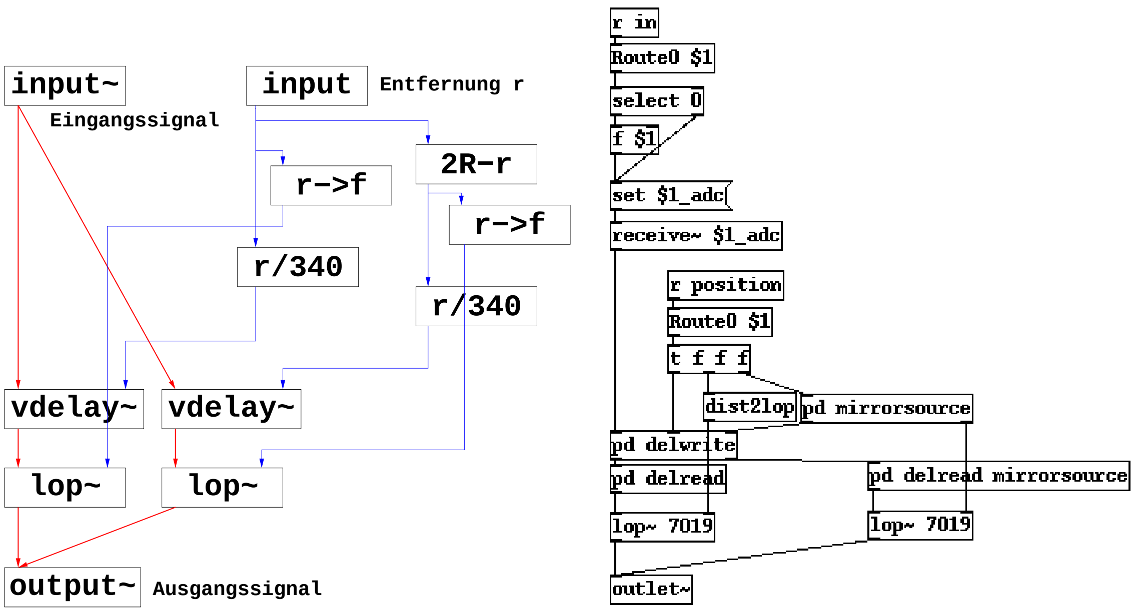
Um die Pegelabnahme und die Luftdämpfung proportional zur Entfernung zu berücksichtigen, kann als Näherung ein Tiefpaß-Filter erster Ordnung, dessen Grenzfrequenz mit steigender Entfernung sinkt, verwendet werden.

Da der Schall einer entfernten Quelle Zeit benötigt, um zum Ohr zu gelangen, wird jede virtuelle Quelle um $t = \frac{r}{340}$ verzögert.

Ändert sich diese Verzögerung innerhalb eines Zeitraumes, so ändert sich auch die Tonhöhe der Schallquelle. Der Doppler-Effekt bei bewegten Quellen lässt sich also einfach realisieren, indem man diese Schallquelle zu jedem

Zeitpunkt um den angemessenen, der Entfernung entsprechenden Betrag verzögert.

Die erste Reflexion einer Quelle in einem kugelförmigen Raum kann effektiv berechnet werden, indem man das Signal um einen der Entfernung der Spiegelquelle entsprechenden Betrag verzögert, und zum Original-Signal hinzuaddiert, bevor die virtuelle Quelle codiert wird. Die Berechnung von zwei getrennten Quellen (virtuelle Quelle und ihre Spiegelquelle) reduziert sich also auf zwei Glieder zur Entfernungsbewertung und das ambisonic-Codieren nur eines Kanals.



Schema *pd*-Patch

Abbildung 6.2: Entfernungs-/Bewegungsbewertung eines Mono-Eingangs

In Abb. 6.2 wird so aus einem Mono-Signal eine virtuelle Quelle mit ihrer Spiegelung (noch ohne Richtungsbewertung).

Um eine einfachere Steuerung von vielen gleichen Abstraktionen von einer zentralen Kontrollroutine zu erlauben, werden jeder Abstraktion ein Name (bzw. eine ID-Nummer) als Argument übergeben. Dieser Name wird bei allen über *send* an ein bestimmtes Label verteilten Messages als erstes Argument mitgeschickt.[1] Soll eine Message für alle Empfänger eines Labels gelten, so wird die ID-Nummer *0* verwendet.

Im oberen Drittel des *pd*-Patches wird über die Message "<ID> <adc#>" an das Label "in" die Zuordnung eines beliebigen adc-Signals der Soundkarte zur virtuellen Quelle ermöglicht.

Vom Positionsvektor *position*="$< ID >< r >< \varphi >< \vartheta >$" geht nur die Entfernung "r" in die Bewertung ein, und zwar einerseits über das Verzögerungsglied (mit $t = \frac{r}{c_0}$, bzw. $t' = \frac{2R-r}{c_0}$ für die Spiegelquelle) und andererseits über die Grenzfrequenz des Tiefpasses. Dabei erwies sich eine lineare Abhängigkeit dieser Grenzfrequenz von der Entfernung mit $f_g = f_0 \quad \text{für } r_{\min} = 1$ und $f_g = 0 \quad \text{für } r_{\max} = 100$ als gute Approximation.

Die beiden Signale für die virtuelle Quelle (mit $r = r$) und der Spiegelquelle (mit $r' = 2R - r$) werden wieder zusammengemischt und der Richtungsencodierung zugeführt.

6.1.2 Encodierung

(vgl. Abb. 6.3)

Die asynchron vom User-Interface kommenden "position"-Messages werden in einer Positionsmatrix $\mathbf{P}$ gespeichert.

$$\mathbf{P} = \begin{bmatrix} 1 & r_1 & \varphi_1 & \vartheta_1 \\ 2 & r_2 & \varphi_2 & \vartheta_2 \\ \vdots & \vdots & \vdots & \vdots \\ K & r_K & \varphi_K & \vartheta_K \end{bmatrix}$$

Diese Matrix wird mit einer konstanten Taktrate ausgelesen (siehe Cap. 6.1). Die in der Matrix gespeicherten Positionen werden über die Codierungsvorschrift in ambisonic-Faktoren überführt.

Der Coder soll mit verschiedenen Codierungsvorschriften betrieben werden können. Deshalb wird die Berechnung der Codierungsfaktoren (sphäri-

[1] zB: Wird über das Label *position* die Message "1 10 0 0" geschickt, so bedeutet dies, dass die virtuelle Quelle *1* die Position $r = 10, \varphi = 0, \vartheta = 0$ haben sollte.

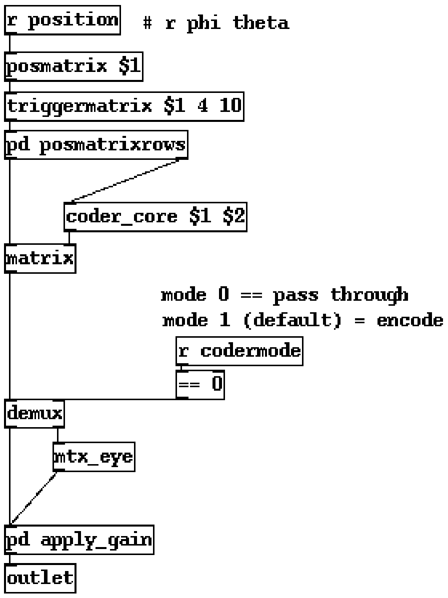


Abbildung 6.3: *pd*-Patch: Matrixberechnung im Coder

sche Harmonische) in eine eigene Abstraktion ausgelagert, die einen Satz von ambisonic-Faktoren für eine bestimmte Richtung errechnet.

Dem Coder wird als Argument der Name der Codierungsvorschrift mitgeteilt (z.B. “furse-malham” in Abb. 6.1). Es wird versucht, die Abstraktion *<such-pfad>/coder/<codername>.pd* zu öffnen. (Im Beispiel wird also *<such-pfad>/coder/furse-malham.pd* geöffnet.) Wird eine solche Abstraktion gefunden, so wird sie ohne weitere Überprüfung als Modul, das eine Codierungsvorschrift implementiert, betrachtet.

Dem Coder-Hauptprogramm ist also erst zur Laufzeit die Anzahl der *ambisonic*-Ausgabekanäle bekannt.

Implementierung einer Codierungsvorschrift

Ein Codierungsmodul gibt für jede Eingangsliste $[r, \varphi, \vartheta]^{\mathrm{T}}$ eine Liste von ambisonic-Faktoren aus. Die Entfernung r ist für die ambisonic-Codierung (Ebene Welle !) eigentlich nicht von Bedeutung. In der Praxis erscheinen aber Schallquellen, die sehr nahe sind, (durch die physikalische Ausdehnung) viel größer als entfernte und sind daher nicht so gut lokalisierbar. Dieses Verwischen kann von einer (modifizierten) Codierungsvorschrift berücksichtigt werden, indem zum Beispiel die Faktoren niedriger Ordnung (zum Beispiel $Y_{0,0} = 1$, ohne jede Richtungsinformation) gegenüber denen höherer Ordnung (= mehr Richtungsinformation) stärker bewertet werden [28].

Eine typische Codierungsabstraktion ist in Abb. 6.4 zu sehen.

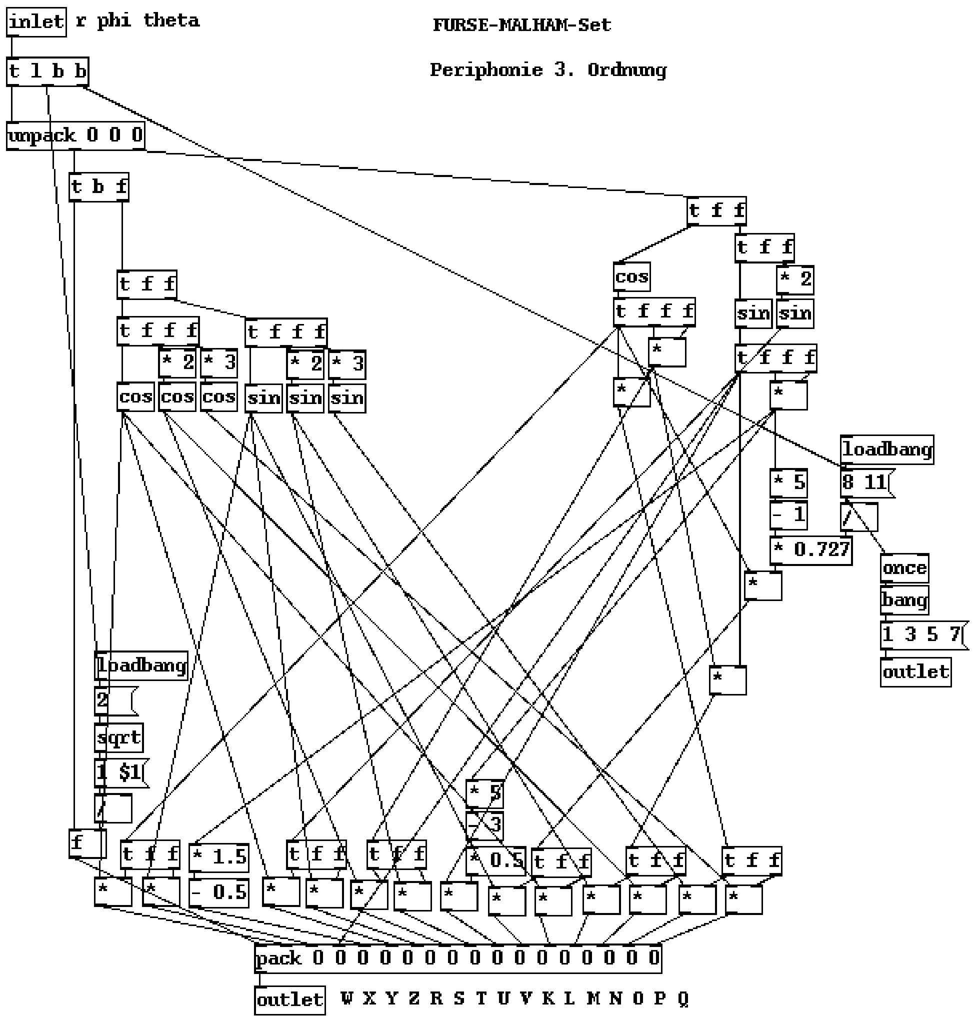


Abbildung 6.4: *pd*-Patch: Codierungsmodul für das Furse-Malham-Set (3. Ordnung)

Da dieselbe Codierungsabstraktion später im Decoder verwendet wird, um die Decodierungsmatrix zu berechnen muss diese Abstraktion noch Zusatzinformationen über die Codierungsvorschrift zur Verfügung stellen: An einen zweiten Outlet wird daher einmal (bei der ersten Verwendung des Codierungsmoduls) eine Liste ausgegeben, die die ambisonic-Ordnung der Faktoren beschreibt ausgegeben. Die Liste $[1, 3, 5, 7]$ in Abb. 6.4 bedeutet, dass das Set aus 1 Faktor 0ter Ordnung, 3 Faktoren 1ter Ordnung, 5 Faktoren 2ter Ordnung und 7 Faktoren 3ter Ordnung (in dieser Reihenfolge !) besteht.

Mit diesem Wissen sollte es einfach sein, eine neue Codierungsvorschrift in das vorhandene Programm einzufügen. (siehe auch Cap. 6.3.1)

Das eigentliche Codierungsmodul führt immer nur eine Positionsangabe $[r, \varphi, \vartheta]^{\mathrm{T}}$ in ein Set von ambisonic-Faktoren $[wxyz\cdots]^{\mathrm{T}}$ über.

Die zu Synchronisationszwecken in einer Matrix vorliegenden Positionsangaben werden daher zuerst wieder in Einzelvektoren zerlegt und codiert. Die einzelnen Codierungsvektoren werden wieder zur Codierungsmatrix $\mathbf{C}^{\mathrm{T}}$ zusammengefasst.

Überbrücken des Codierers

Sollte die ambisonic-Codierung rasch überbrückt werden, sodass nur noch Lautstärkenmanipulation (mit Tiefpass-Filterung und Verzögerungsgliedern), so genügt es, die Codierungsmatrix durch eine Einheitsmatrix zu ersetzen.

Die Message "coder_mode" wählt zwischen Codierungsmatrix und Einheitsmatrix.

Kanalverstärkung

Die Verstärkungsfaktoren für die einzelnen Eingangskanäle kommen ebenfalls asynchron vom User-Interface. Sie werden in einer Diagonalmatrix $\mathbf{A}$ gespeichert, die synchron zur Positionsmatrix ausgelesen wird.

Diese wird mit der Codierungsmatrix (oder alternativ mit einer Einheitsmatrix) multipliziert.

Die resultierende Matrix beschreibt alle linearen Operationen des Coders. Sie muss daher nur noch mit der Menge der (entfernungsbewerteten) Eingangssignale multipliziert werden um die ambisonic-Repräsentation des periphonen Schallfeldes zu erhalten.

6.1.3 Dynamische Kanalallokation

Abhängig von der verwendeten Codierungsvorschrift (und implizit natürlich von der Ordnung der Codierungsvorschrift) wird eine bestimmte Menge an Ausgabekanälen benötigt. Die Anzahl der Eingangskanäle soll ebenfalls vom Benutzer bestimmbar sein.

Während der Benutzung (also nach dem Zeitpunkt des Einschaltens / der Konfiguration) ist die Anzahl der Ein- und Ausgabekanäle konstant.

Auf Message-Ebene ergibt sich aus der Kanal-Verwaltung zur Laufzeit kein Problem, da alle mehrdimensionalen Konstrukte (Listen und Matrizen) eine dynamische Vergrößerung/Verkleinerung unterstützen.

Es gibt bei *pd* jedoch keine mehrdimensionalen Strukturen im Signal-Bereich. Jedes Signal muss explizit repräsentiert werden.

pd bietet jedoch Methoden für die dynamischen Patch-Generierung. Da die Anzahl der benötigten Ein-/Ausgabekanäle bei der Initialisierung explizit (Eingangskanäle) oder implizit (die Anzahl der ambisonic-Kanäle wird über die Codierungsvorschrift bestimmt) bekannt ist, werden die Signal-Objekte erst dann kreiert und untereinander verbunden.

Dadurch wird sichergestellt, dass nur die geforderte Anzahl an Ein- und Ausgabekanälen zur Verfügung steht (und Rechenleistung beansprucht).

6.2 Verhallung

In einem natürlichen Schallfeld besitzt auch der diffuse Nachhall eine Richtungsinformation, da Decke (zum Beispiel aus gut reflektierendem Beton), Boden (dämpfender Teppich) und Wände (heterogene Struktur) verschiedene Reflexionsverhalten aufweisen.

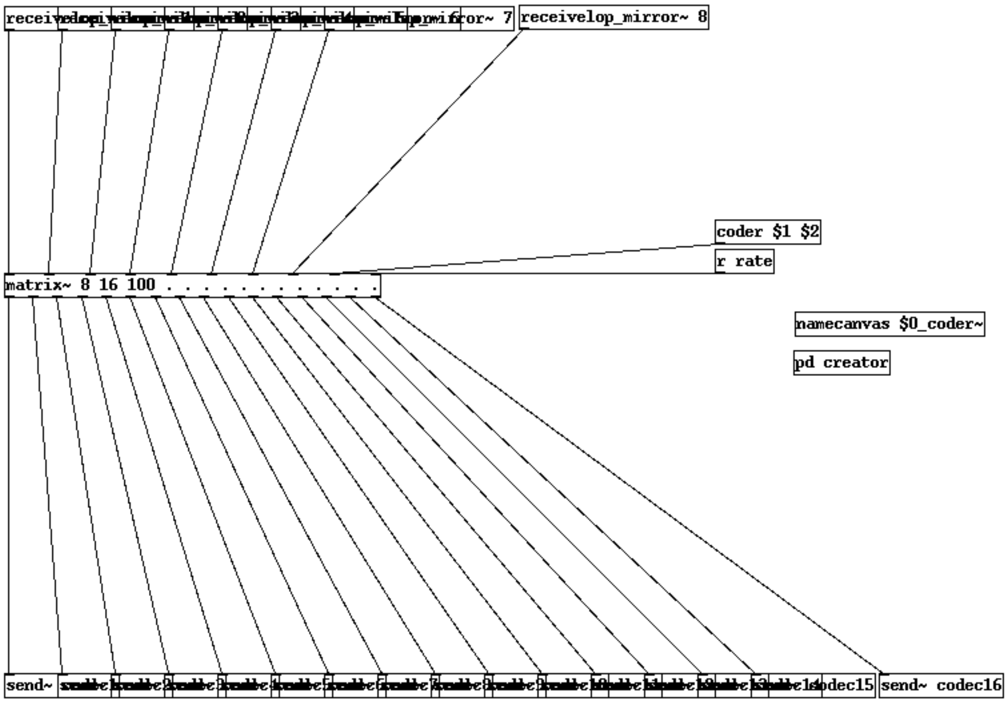


Abbildung 6.5: dynamisch generierter *pd*-Patch: Signalmatrizierung im Coder

Um eine möglichst plastische Raumsimulation zu realisieren, ist es daher notwendig einer diffusen Verhallung Richtungsinformation aufzuprägen. Allerdings muss diese Richtungsinformation "schwammig" sein, um den unnatürlichen Eindruck von konkreten "Hallquellen" zu vermeiden. Daher ist es sinnvoll, den Diffushall nur mit einer niedrigen ambisonic-Ordnung (1. Ordnung) zu codieren.

Ein ambisonic-Hallgerät hat daher eine wie in Abb. 6.6 dargestellte Struktur.

Eine ambisonic-Repräsentation 1.Ordnung (4 Kanäle) wird in eine Minimalkonfiguration decodiert. Die einfachste Lautsprecheraufstellung zur Wiedergabe von 3D-Schallfeldern ist ein regelmäßiger Tetraeder.

In einem vierkanaligen Hallgerät kann nun jede Richtung über ein eigenes Hallprogramm verräumlicht werden. Die diffusen Signale werden anschließend wieder ambisonic-codiert, zum Beispiel ebenfalls in ein Tetraeder.

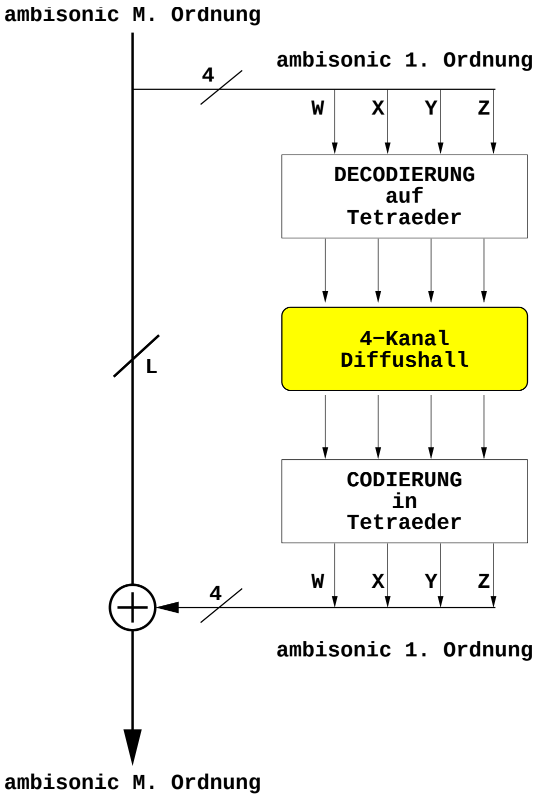


Abbildung 6.6: Struktur zur Verhallung eines ambisonic-Signales

6.2.1 Interner Hallalgorithmus

Die vier Signale werden einem Hallpatch zugeführt.

Der Name des Hall-Patches wird dem ambisonic-Hallmodul übergeben (siehe Abb. 6.7(a)) und muss sich im Verzeichnis *<such-pfad>/reverb/* befinden.

(In Abb. 6.7(a) wird also eine Datei *<such-pfad>/reverb/abstrus4˜* gesucht.) Der Verhallungsalgorithmus kann über das User-Interface gesteuert werden. (siehe Cap. 7.1). Steuerbefehle für den Hall werden an das Label *reverb* gesendet. Zahl und Bedeutung der Parameter bleiben sind in keinem Protokoll festgeschrieben und können so der Hall-Implementierung angepasst werden.

Gute Hallalgorithmen sind sehr rechenintensiv. Da der eigentliche Codie-

`reverb~ abstrus4~`　　`reverb~ 25 26 35 36`

(a)　　(b)

Abbildung 6.7: Aufruf des ambisonic-Hallmoduls mit (a) internem Algorithmus ("*abstrus4~*") und (b) externem Hallgerät (Ein-/Ausgabe über Kanäle 25, 26, 35, 36)

rungsprozess (siehe Cap. 6.1) ebenfalls viel CPU-Leistung beansprucht, sind interne Reverberationen meist nicht sehr zufriedenstellend.

6.2.2 Externes Hallgerät

Externe Hallgeräte haben mehr Rechenleistung und können daher besser klingende Hallalgorithmen implementieren.

Um ein externes Hallgerät an die ambisonic-Verhallungsstruktur anzubinden, werden die decodierten Kanäle über vier freie Ausgänge der Soundkarte zum Hallgerät gespielt und die Rücksignale über vier freie Eingänge in den Hall-Coder eingespielt.

Welche Kanäle "frei" sind, hängt von der verwendeten Hardware ab. Daher können die Kanalnummern als Argumente übergeben werden.

Da die Ein- und Ausgabe über die Audiokarte mit einer gewissen Latenz behaftet ist, muss diese bei der Programmierung des Hallgerätes beachtet werden.

6.3 Decoder

Die Aufgabe des Decoders ist, eine Anzahl von ambisonic-Kanälen korrekt über ein gegebenes Lautsprecherarray wiederzugeben. Alle Manipulationen der Repräsentation des Schallfeldes haben daher ausschließlich den Charakter der Anpassung der vorgefundenen Situation an eine Idealisierte.

Zur korrekten Decodierung ist es notwendig, dieselbe Codierungsvorschrift zu benützen wie bei der Encodierung. Der Name der Codierungsabstraktion

```
decoder~ furse-malham
```

(a)

```
decoder~ furse-malham speakers.mtx
```

(b)

Abbildung 6.8: *pd*-Patch: Aufruf des decoder~-Moduls

wird daher (wie beim Coder) als Argument übergeben. (z.B. "furse-malham" in Abb. 6.8)

Es werden also wie beim Coder werden alle Operationen in der Message-Ebene durchgeführt und dann über eine Matrixmultiplikation die ambisonic-Signale in die Lautsprecher-Signale überführt.

6.3.1 Schallfeld-Rotation

Durch Drehung des Schallfeldes ist es möglich, die "Vorne"-Lokalisierung an den Bedarf auszurichten.

Im Allgemeinen werden hiefür nur Rotationen um die senkrechte Z-Achse benötigt werden. Nichtsdestotrotz werden Rotationen um alle kartesischen Achsen implementiert.

Die Rotationswinkel um die drei Hauptachsen, die asynchron vom User-Interface geschickt werden, werden zwischengespeichert und mit konstantem Takt ausgelesen. Danach werden sie den einzelnen Rotationsmatrizen zugeführt.

Da die Rotationsmatrizen von der verwendeten Codierungsvorschrift abhängig sind, müssen für jede implementierte Codierungsregel auch Rotationsmatrizen als Abstraktionen vorliegen.

Das Decoderprogramm versucht den Rotationspatch für die Codervorschrift *<codername>* in *<such-pfad>/rotation/Z-<codername>.pd* zu öffnen.[2] Die Abstraktion gibt für einen bestimmten Drehwinkel Φ die zugehörige Drehmatrix aus.

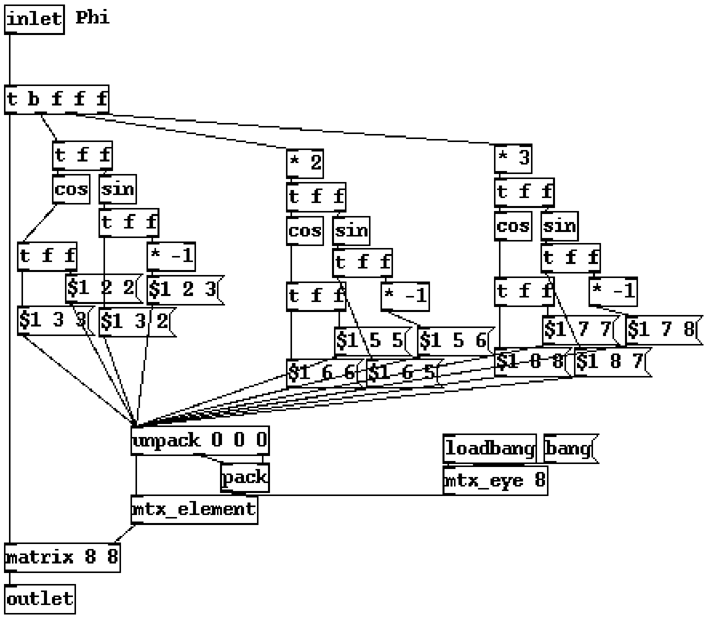


Abbildung 6.9: *pd*-Patch: Drehmatrix um Z-Achse für *3rd+1st*

Wird eine solche Abstraktion nicht gefunden, wird sie durch eine Einheitsmatrix ersetzt. Dadurch müssen nur Rotationsabstraktionen für diejenigen Achsen geschrieben werden, für die auch tatsächlich eine Drehung erwünscht ist.

Die Gesamtrotationsmatrix ergibt sich nach Gl. (2.20) als

$$\mathbf{R} = \mathbf{R}_x(\Psi) \cdot \mathbf{R}_y(\Theta) \cdot \mathbf{R}_z(\Phi)$$

6.3.2 Parametrisierung des Lautsprecherarrays

Das Herzstück des Decoders ist die Decodierungsmatrix für eine gegebene Lautsprecheraufstellung.

[2]Sinngemäß erweitert, werden auch die Abstraktionen *<such-pfad>/rotation/X-<codername>.pd* und *<such-pfad>/rotation/Y-<codername>.pd* geöffnet und verwendet.

Um diese zu berechnen, müssen dem Decoder die Richtungen der Lautsprecher bekannt sein. Defaultmäßig wird daher im Suchpfad nach einer Datei *angles.mtx* gesucht, in der Anzahl sowie Azimuth und Elevation der Lautsprecher beschrieben sind (siehe Anhang C).

Ist die Datei, in der das Lautsprecherlayout gespeichert ist, eine andere, so wird deren Name als zweites Argument der *decoder*-Abstraktion übergeben (z.B. *speakers.mtx* in Abb.6.8(b)).

Diese Layout-Datei[3] muss für jede verwendete Lautsprecherkonfiguration vorhanden sein.

Die Lautsprecherpositionen werden dann über die Codierungsvorschrift in die Reencodierungsmatrix $\mathbf{C}$ codiert.

Sollte dabei die Anzahl der Lautsprecher kleiner sein als die Anzahl der ambisonic-Kanäle (also Gl.(2.30) überbestimmt sein), so wird die Ordnung von $\mathbf{C}$ automatisch soweit reduziert, dass Gl.(2.30) genau oder unterbestimmt ist.

Anschließend werden zwei Decodierungsmatrizen berechnet, und zwar $\mathbf{D}_{\text{basic}}$ und $\mathbf{D}_{\text{in-phase}}$ (siehe 2.4.1 und 2.4.2). Zur Bildung der Pseudoinversen bei der Berechnung von $\mathbf{D}_{\text{basic}}$ wird die Reencodierungsmatrix aus Stabilitätsgründen zuerst pivotisiert und nach der Inversion wieder depivotisiert.

Durch lineare Interpolation zwischen den beiden Matrizen, die vom User-Interface aus geregelt werden kann, wird die tatsächlich verwendete Decodierungsmatrix $\mathbf{D}$ gewonnen.

$$\mathbf{D} = (1-\beta)\mathbf{D}_{\text{basic}} + \beta\mathbf{D}_{\text{in-phase}}$$

Durch Multiplikation der Rotationsmatrix mit der Decodierungsmatrix kann das ambisonic-Schallfeld korrekt decodiert werden.

$$\mathbf{D}' = \mathbf{D} \cdot \mathbf{R}$$

Sollte der Decoder überbrückt werden, so genügt es, $\mathbf{D}'$ durch die Einheitsmatrix zu ersetzen.

[3] Das Dateiformat sowie eine Beispieldatei sind im Anhang C zu finden.

Die Message "decoder_mode" wählt zwischen Decodierungsmatrix und Einheitsmatrix.

Gesamtlautstärke

Die Gesamtlautstärke kann über die Multiplikation der Decodierungsmatrix mit einem Faktor α geregelt werden.

Die endgültige Decodierungsmatrix wird mit der Menge der ambisonic-Kanäle multipliziert und über die Lautsprecher wiedergegeben.

6.4 Ausgabemodul

Die Decodierungsmatrix **D** wurde mit Rücksicht auf die tatsächlichen Lautsprechersignale an einem Bezugspunkt entworfen. Die Ausgangssignale des Decoders repräsentieren daher die Signale am Meßpunkt und nicht die Signale an den Endstufen-Eingängen. Durch abweichende Einstellungen an Verstärkern, Bauteilschwankungen,..., aber auch durch die verschieden großen Entfernungen der Lautsprecher zum Bezugspunkt, entsprechen jedoch die Decoderausgänge nicht mehr unbedingt den am Meßpunkt aufgenommenen.

Es ist daher notwendig, dass Maßnahmen ergriffen werden, die dazu führen, dass alle wiedergebenden Lautsprecher (wenigstens bezüglich eines kleinen Bereiches im Raum) praktisch idente Eigenschaften aufweisen.

Da diese Forderung nicht nur für ambisonic-Systeme sondern für die allermeisten Beschallungsanlagen gilt, wurde diese Lautsprechercalibrierung in einem eigenen Ausgabemodul und nicht über die Decodierungsmatrix implementiert.

6.4.1 Calibrator

Bevor die Eigenschaften von Lautsprechern ausgeglichen werden können, müssen sie erfasst und die Unterschiede gemessen werden.

Für Musik-Wiedergabe relevante Eigenschaften sind:

- *Pegel*
- *Laufzeit*
- *Klangfarbe*

Pegel und Laufzeitunterschiede resultieren in erster Linie aus der räumlichen Entfernung von Bezugspunkt und Lautsprecher. Lautsprecher, die näher am Bezugspunkt sind, erzeugen bei gleichem Ausgangssignal höhere Schalldrücke an diesem Punkt als Lautsprecher, die entfernter sind.

Durch die endliche Ausbreitungsgeschwindigkeit der Luft

$$c \approx 331.3 + 0.6\tau \tag{6.1}$$

erreichen Wellen von weiter entfernten Lautsprechern den Bezugspunkt später als solche von nahen Lautsprechern.

Die Entfernungen zwischen Lautsprechern und Hörposition(en) sind bei Wiedergabesystemen im Allgemeinen relativ gering ($< 20m$), wodurch sich der Einfluss der Luftdämpfung auf die Klangfarbe der einzelnen Lautsprecher-Wiedergabekanäle kaum bemerkbar macht. Klangfarbenunterschiede ergeben sich jedoch durch unterschiedliche Eigenschaften der Übertragungswege, namentlich seien hier vor allem Fertigungsunterschiede von Verstärkern und Lautsprechern erwähnt.

Während also Pegel- und Laufzeitunterschiede in einem Bezugspunkt alleine schon systembedingt durch die Aufstellung der Lautsprecher entstehen, ergeben sich Klangfarbenunterschiede vor allem aus der Verwendung von nicht-homogenen Wiedergabeelementen.

Amplitudenausgleich

Um den Pegelunterschied der einzelnen Lautsprecher ausgleichen zu können, wird jeder Lautsprecher mit einem Pegel L eingerauscht.

Wird über einen Lautsprecherkanal ein Signal mit dem Pegel L wiedergegeben, so kann durch die unterschiedliche Verstärkungen in den einzelnen Verstärker-Endstufen und die entfernungsabhängige Luftdämpfung für jeden

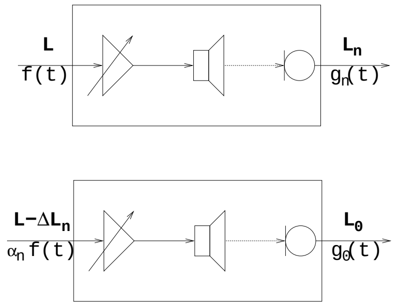


Abbildung 6.10: Lautstärkenanpassung

Wiedergabekanal am Bezugspunkt mit einem Messmikrophon ein Wiedergabepegel L_n gemessen werden.

Man kann sich nun zu einem beliebigen Bezugspegel L_0 eine Pegeldifferenz errechnen:

$$\Delta L_n = L_n - L_0 \tag{6.2}$$

Werden die einzelnen Kanäle vor der Wiedergabe über die Lautsprecher um $-\Delta L_n$ verstärkt, so wird (vorausgesetzt die Wiedergabekanäle verhalten sich annähernd linear) am Bezugspunkt von jedem Lautsprecher der Pegel L_0 erzeugt.

Bei Verstärkungen > 1 $(-\Delta L_n > 0)$ können in der digitalen Signalverarbeitung zum Überschreiten des zulässigen Wertebereichs führen (Clipping!). Um dies zu vermeiden, ist es sinnvoll, ΔL_n auf den Wiedergabekanal mit dem niedrigsten Pegel zu normieren

$$L_0 = L_{\min} = \min\{L_1, L_2, \ldots, L_N\} \tag{6.3}$$

Verlässt man die Pegeldomäne, so ergeben sich Verstärkungsfaktoren aus der Betrachtung der Effektivwerte $F = \lim\limits_{\tau\to\infty} \sqrt{\frac{1}{\tau}\int\limits_0^\tau f^2(t)dt}$ der gemessenen Signale $g(t)$:

$$\alpha_n = \frac{G_0}{G_n} \tag{6.4}$$

Als Messsignal wird Rauschen verwendet, da sich so keine stehenden Wellen im Wiedergaberaum ausbreiten können. Es würde das Messergebnis verfälschen, wenn das Messmikrophon sich einmal im Druckbauch (Pegelmaximum) und ein andermal an einem Druckknoten (Pegelminimum) einer stehenden Welle befände.

Phasenausgleich

Um die Laufzeiten von Lautsprechern zum Bezugspunkt zu erhalten, misst man die Übertragungseigenschaften des Wiedergabesystems (Abb.6.10) und untersucht diese.

Das Übertragungssystem kann durch seine Impulsantwort $h(t)$ beschrieben werden. Wird einem solchen System ein beliebiges Signal $f(t)$ zugespielt, so erhält man die Antwort des Systems, indem man das Eingangssignal mit der Impulsantwort faltet:

$$\begin{aligned} g_n(t) &= \sum_{k=-\infty}^{+\infty} h_n(k) f(t-k) \\ &= h_n(t) * f(t) \end{aligned} \tag{6.5}$$

In der Frequenzdomäne vereinfacht sich die Faltung zu einer Multiplikation:

$$G_n(\omega) = H_n(\omega) \cdot F(\omega) \tag{6.6}$$

Sind Eingangs- und Ausgangssignal bekannt, lässt sich die Impulsantwort berechnen:

$$H_n(\omega) = \frac{G_n(\omega)}{F(\omega)} \tag{6.7}$$

$$\begin{aligned} h_n(t) &= \mathfrak{F}^{-1}\left\{\frac{G_n(\omega)}{F(\omega)}\right\} \\ &= \mathfrak{F}^{-1}\left\{\frac{\mathfrak{F}\{g_n(t)\}}{\mathfrak{F}\{f(t)\}}\right\} \end{aligned} \tag{6.8}$$

Zur Messung der Raumimpulsantwort wird weißes Rauschen in einen Raum eingespielt und mit einem Mikrophon aufgenommen. Korreliert man das in sich unkorrelierte Originalsignal mit dem verräumlichten Signal, so erhält man die Impulsantwort des Übertragungssystems. Sie setzt sich sich im Wesentlichen zusammen aus:

$$h_n(t) = h_{\mathrm{Raum},n} * h_{\mathrm{LS},n} * h_{\mathrm{Mikrophon}} * h_{\mathrm{DSP}} \tag{6.9}$$

Im Idealfall (wenn die Übertragungseigenschaften des elektrischen Systems inklusive Lautsprecher und Mikrophon vernachlässigt werden können) ist sie gleich der Raumimpulsantwort. Im Allgemeinen kann man jedoch die Übertragungseigenschaften vor allem der Schallwandler $h_{\mathrm{LS},n} * h_{\mathrm{Mikrophon}}$ nicht vernachlässigen. Die Übertragungseigenschaft des DSP-Systems setzt sich zusammen aus der Latenz dieses Systems sowie den Anti-Aliasing-Filtern der Digital-Analog- und der Analog-Digital-Converter.

Das erste Maximum der Raumimpulsantwort entspricht dem Direktanteil des Signals. Man erhält also die Laufzeit Lautsprecher-Mikrophon τ durch Messung des ersten Maximums der Raumimpulsantwort. Bei Verwendung von hochwertigen Schallwandlern und A/D- bzw. D/A-Umsetzern sind deren Einschwingzeiten so gering, dass die daraus resultierende Laufzeit-Änderung gegenüber der tatsächlichen Laufzeit Lautsprecher-Mikrophon vom Ohr nicht mehr wahrgenommen werden kann. Die Latenzzeit des DSP-Systems verfälscht das Ergebnis jedoch um einen konstanten Faktor T.

$$\tau_{\mathrm{Messung}} = \tau + T \tag{6.10}$$

Um die Phasenverschiebung der einzelnen Lautsprecherwiedergabesignale auszugleichen, werden alle Signale mit demjenigen mit der längsten Laufzeit durch Verzögerungsglieder in Phase gebracht:

$$\begin{aligned} \tau_0 &= \max\{\tau_1, \tau_2, \cdots, \tau_N\} && (6.11) \\ \Delta\tau_n &= \tau_0 - \tau_n && (6.12) \end{aligned}$$

Durch die Subtraktion fällt dabei der konstante Offset der DSP-Latenzzeit heraus.

Die Berechnung der Impulsantwort über die Fouriertransformation ist in der Praxis unmöglich, da man kein Signal unendlich lange beobachten kann. Um die Laufzeit des Signals zu erhalten, interessiert jedoch glücklicherweise nur der erste Teil der Raumimpulsantwort. Es genügt also, nur einen zeitlich begrenzten Teil des Aus- und Eingangssignales zu betrachten.

Dabei ist es wichtig, dass das gewählte Zeitfenster groß genug ist, um die Laufzeitverschiebung enthalten zu können. Um die Artefakte der Diskreten

Fourier-Transformation (DFT) so gering wie möglich zu halten, werden Ein- und Ausgangssignal mit einem *von Hann*-Fenster multipliziert.

Um die so gemessene Impulsantwort stabiler gegen mathematische Ungenauigkeiten bei der Berechnung der DFT zu machen, werden mehrere Messungen gemacht und die gewonnen Impulsantworten gemittelt. Das erste Maximum, das ja den Direktschallanteil kennzeichnet, muss nicht unbedingt dem globalen Maximum der Raumimpulsantwort entsprechen, da sich durch den längeren Weg stärker gedämpfte Reflexionen gleichphasig überlagern können, sodass die so addierten Signale einen höheren Pegel aufweisen, als der Direktschall. Durch die Artefakte der DFT, sowie durch die tatsächliche Periodizität der digitalen Pseudo-Rauschgeneratoren, wird die gewonnenen Impulsantwort allerdings von einem rauschhaften Störanteil überlagert.

Dadurch ist es nötig, statt des ersten *tatsächlichen* lokalen Maximums das erste *signifikante* Maximum zu detektieren. Da eine Raumimpulsantwort immer mehrere *signifikante* Maxima aufweist, kann man das erste dieser Maxima detektieren, indem man das erste von N Maxima findet.

Klangfarbenegalisierung

Unabhängig von der Aufstellung der Lautsprecher, können sich die einzelnen Wiedergabekanäle aus produktionstechnischen Gründen klangfarblich unterscheiden. Ein digitales Equalizing ist durch Filterung möglich. Allerdings ist diese Methode sehr rechenaufwendig.

Im Gegensatz dazu ist es sehr leicht, die Notwendigkeit zu einer solchen Klangfarben-Anpassung kleinzuhalten, indem gleichartige D/A-Converter, vor allem aber gleichartige Verstärker und Lautsprecher verwendet werden. Idealerweise sollen Produkte aus einer Charge zur Anwendung kommen.

6.4.2 Ausgabemodul

Die vom *Calibrator* errechneten Korrekturwerte für Amplituden und Phasen der einzelnen Lautsprecher werden in einer Matrix *calibrate.mtx* gespeichert.

Damit die die calibrierten Ausgänge benutzt werden können, wird ein Ausgabemodul *main-out˜.pd* verwendet. Dieses generiert soviele *dac˜* -Objekte wie laut *calibrate.mtx* Wiedergabekanäle existieren. In jeden Ausgabekanal wird ein Verzögerungsglied und eine Multiplikationsstufe eingefügt, die mit den Korrekturwerten für den jeweiligen Kanal initialisiert werden. Zusätzlich wird noch eine *panic*-Methode hinzugefügt, die alle Kanäle auf “stumm” schaltet. Mit *unpanic* kann die Stummschaltung wieder aufgehoben werden. Defaultmäßig sind alle Ausgabekanäle gemutet!

Will man einen so kalibrierten Ausgabekanal (z.B. Kanal “1”) verwenden, schickt man das betreffende Signal nicht an ein *dac˜ 1*-Objekt sondern über `throw~ 1_spk` an den Soundkartenausgang.

Kapitel 7

Kommunikationslayer

7.1 Einleitung

Die Kommunikation zwischen der BenutzerInnen-Schnittstelle und dem Coder bzw. Decoder wird über eine Ethernet-Verbindung hergestellt.

Dabei kommt das in *pd* implementierte *FUDI*-Protokoll zur Anwendung.

7.1.1 FUDI-Protokoll

Das FUDI-Protokoll wurde als einfaches Kommunikationsprotokoll auf ASCII-Basis zum Austausch von Kontrolldaten zwischen Musik-Programmen von Miller S. Puckette entworfen.

Prinzipiell besteht jedes FUDI-Paket aus einer Liste von Atomen, die durch Leerzeichen getrennt sind. Ein Atom kann eine Zahl oder ein Symbol (symbolischer Name, Nicht-Zahl) sein und wird in ASCII übertragen. Die Zahl "41" wird daher nicht als ASCII-Wert *41* (")"), sondern durch zwei Zeichen "4" (ASCII-Wert *52*) und "1" (ASCII *49*). Die Floatingpoint-Zahl "3.14159265" wird als 10 Byte langer String übertragen !

Das Ende eines Paketes wird durch das Zeichen ";" angezeigt.

Durch die Analogie zwischen FUDI und dem internen Message-System von *pd* (das ebenfalls auf durch Leerzeichen getrennte Listen von Atomen basiert), erscheint die Verwendung von FUDI als Message-nah.

7.1.2 UDP

Die TCP/IP-Implementierung in *pd* bei Beendigung eines verbundenen *pd*-Prozesses einen Prozess, der trotzdem versucht, über die (nunmehr ungültige) Verbindung Daten zu schicke, zum Absturz bringt. Aus Stabilitätsgründen muss daher UDP gegenüber TCP/IP als der Übertragung zugrundeliegendes Protokoll vorgezogen werden.

Um Probleme durch fehlerhaften UDP-Übertragungen möglichst klein zu halten, muss der Ethernet-Traffic möglichst gering gehalten werden.

Die einfachste Methode den Traffic so gering wie möglich zu halten, ist, nur sehr kurze FUDI-Pakete zu schicken. Dies impliziert, dass nur ganzzahlige Werte anstelle von Gleitkommazahlen geschickt werden können. Daher werden zur Richtungsangabe φ und ϑ in Grad anstatt in Radiant übertragen.[1]

Verstärkungsfaktoren werden als dB-Werte übertragen, wobei die *pd*-Konvention gilt, dass Unity Gain $1 \doteq 100$dB.

Um mehrere verschiedenartige Messages über eine Ethernet-Verbindung zu übertragen, beginnt jede Message-Liste mit einem symbolischen Identifier. Dieser Identifier sollte ebenfalls möglichst kurz sein.

Da Übertragungsfehler trotzdem nicht auszuschließen sind, ist darauf zu achten, dass Übertragungsfehler keine Schäden verursachen können. Besonders Verstärkungsfaktoren sollten begrenzt werden.[2]

Aus all diesen Gründen ist zusätzlich noch eine *Panik*-Funktion zu implementieren, die es erlaubt, die Wiedergabe auf "stumm" zu schalten.

Fällt das User-Interface aus, so soll die Audio-Engine aus Sicherheitsgründen in den *Panik*-Zustand gehen. Damit die Audio-Engine erfährt, ob das

---

[1]Dies macht besonders Sinn, da die Gradauflösung des menschlichen Gehörs bestenfalls etwa $1°$ ausmacht.

[2]Zum Beispiel kann durch Zeichnverdoppelung bei der Übertragung aus einer Verstärkung um "100"dB (Unity Gain) plötzlich eine Verstärkung um "1000"dB (Faktor 1.7710^{19}). Auch wenn durch geeignete Sicherheitsmaßnahmen der tatsächliche Ausgangspegel noch keine irreparablen Gehörschäden verursacht, so ist dieser Fauxpas doch sehr unangenehm und kann durch die Schrecksituation durchaus auch lebensbedrohlich werden.

User-Interface noch funktioniert, wird eine einfache Watchdog-Funktionalität implementiert. Die Audio-Engine muss in regelmäßigen Abständen durch die Message "1" (ohne Identifier) reaktiviert werden. Fällt diese Reaktivierung länger als 5 Sekunden aus, so schaltet sich die Audio-Engine ab.

7.1.3 Kommunikationsprotokolle

Die Interprozesskommunikation basiert auf einen eingeschränkten Satz von "Befehlen", der, je nach gesteuertem Modul, in einem von drei Sub-Protokollen (Tabellen 7.1, 7.2,7.3) definiert ist.

Über ein sehr einfaches Protokoll (Tab. 7.3) können Coder und Decoder der GUI Statusinformationen übermitteln. Mächtiger sind die Netzschnittstelle zum Coder sowie das Decoder-Protokoll, mit denen die volle Funktionalität der Audio-Module gesteuert werden kann.

Die Implementierung der Protokolle erfolgte in drei Abstraktionen. Deren Defaultwerte sind so gewählt, dass sie bei Verwendung von Coder, Decoder und (G)UI auf einem einzigen Rechner nicht umkonfiguriert werden müssen.

7.2 coder_netinterface

Empfängerseitige Implementation des Coder-Protokolls (Tab. 7.1), sowie senderseitige teilweise Implementation des GUI-Protokolls (Tab. 7.3).

Als (optionale) Argumente können *<control_host>* *<sendport>* *<receiveport>* angegeben werden. Defaultmäßig wird der *control_host* (auf dem das User-Interface läuft) als der lokale Rechner *localhost* angenommen. Der standardmäßige *receiveport*, an dem die Messages abgefangen werden, ist 8086. Der *sendport*, über den Statusinformationen zurück zum User-Interface geschickt werden, ist 8090.

7.3 decoder_netinterface

Empfängerseitige Implementation der Decoder-Protokolls, sowie senderseitige teilweise Implementation des GUI-Protokolls.

Als (optionale) Argumente können <*control_host*> <*sendport*> <*receiveport*> angegeben werden. Defaultmäßig wird der *control_host* (auf dem das User-Interface läuft) als der lokale Rechner *localhost* angenommen. Der standardmäßige *receiveport*, an dem die Messages abgefangen werden, ist 8088. Der *sendport*, über den Statusinformationen zurück zum User-Interface geschickt werden, ist wie beim *coder_netinterface* 8090.

7.4 gui_netinterface

Senderseitige Implementation von Coder- und Decoderprotokollen, sowie empfängerseitige Implementation des GUI-Protokolls.

Hier können als Argumente <*coder_host*> <*decoder_host*> <*send2coderport*> <*send2decoderport*> <*receiveport*> übergeben werden. <*coder_host*> und <*decoder_host*> sind beide standardmäßig auf *localhost*> (den lokalen Rechner) gesetzt. <*send2coderport*> ist per Default 8086, <*send2decoderport*> ist 8088 und *receiveport* ist 8090.

ID	*data*	*Beschreibung*
`DSP`	<dsp_flag>	dsp_flag=1/0 schaltet den *Coder* ein/aus
`gain`	<#slot> <slot_level>	Eingangsverstärkung für die virtuelle Quelle #slot
`pos`	<#slot> <r> <φ> <ϑ>	Position der virtuellen Quelle #slot

`in`	<#slot> <channel>	weist einer virtuellen Quelle das Mono-Signal zu, das am Audiokarten-Kanal channel liegt
		channel=0: *default*-Kanal (zB: adc3 bei #slot 3)
		channel=-1: NULL-Kanal (stumm)

`rev`	<reverb_gain>	Reverb-Return Verstärkung
`revmod`	<reverb_mode>	
`reverb`	{<reverb_factor>}	Parameterliste für den (internen) Reverberator
`mode`	<coder_mode>	coder_mode=1: *ambisonic-Coder* eingeschaltet coder_mode=0: *ambisonic-Coder* wird überbrückt (through)
`talk`	<talkback_flag>	flag=1 veranlasst den *Coder*, Informationen zu senden (s. 7.3)
`QUIT`	<quit>	quit=1 beendet den *Coder*

Tabelle 7.1: Netzschnittstelle: Coder

ID	*data*	*Beschreibung*
`DSP`	<dsp_flag>	dsp_flag=1 schaltet den *Decoder* ein dsp_flag=0 schaltet den *Decoder* aus
`panic`		schaltet den *Decoder* stumm
`unpanic`		schaltet den stummgeschaltetenen *Decoder* wieder laut
`master`	<master_level>	Gesamtverstärkung
`head`	<Ψ> <Θ> <Φ>	dreht das Schallfeld um die X-, Y- und Z-Achse
`phase`	<in-phase>	Interpolation zwischen *basic*-Decoding (in-phase=0) und *in-phase*-Decoding (in-phase=100)
`out`	<out_mode>	out_mode=1: Wiedergabe mit Lautsprechercalibrierung out_mode=0: Wiedergabe ohne Lautsprechercalibrierung
`mode`	<decoder_mode>	decoder_mode=1: *ambisonic-Decoder* eingeschaltet decoder_mode=0: *ambisonic-Decoder* wird überbrückt (through)
`talk`	<talkback_flag>	flag=1 veranlasst den *Decoder*, Informationen zu senden (s. 7.3)
`QUIT`	<quit>	quit=1 beendet den *Decoder*

Tabelle 7.2: Netzschnittstelle: Decoder

ID	*data*	*Beschreibung*
`DSP1`	<dsp_state>	ist *Coder* eingeschaltet ?
`cpu1`	<cpu_load>	CPU-Auslastung des *Coder*s in Prozent
`DSP2`	<dsp_state>	ist *Decoder* eingeschaltet ?
`cpu2`	<cpu_load>	CPU-Auslastung des *Decoder*s in Prozent
`level`	<level1>...<levelN>	Eingangspegel der virtuellen Quellen 1...N

Tabelle 7.3: Netzschnittstelle: GUI

Kapitel 8

User-Interface

8.1 PRESETter

Eine sehr leicht les- und bedienbare Oberfläche stellt den benutzenden Personen die wichtigsten Standardwiedergabesituationen zur Verfügung.

Da diese Oberfläche für einen Touchscreen mit 800*600 Pixel Auflösung geeignet sein soll ist sie relativ grob strukturiert.

Über große Buttons ist einer der folgenden Wiedergabe-Modi auswählbar:

- *MONO*: Alle Eingangssignale werden ohne Richtungsbewertung gleichmäßig von allen Lautsprechern wiedergegeben.
- *FRONT*: Alle virtuellen Quellen befinden sich "vorne".
- *STEREO*
- *QUADRO*
- *5+1*
- *OCTOGON*

Rechts von der Auswahlleiste wird in einer einfachen 2D-Grafik die Aufstellung der virtuellen Lautsprecher visualisiert.

Über einen Masterregler kann die Gesamtlautstärke der Wiedergabe geregelt werden. Eine Pegelanzeige zeigt an, ob an den Eingängen ein Signal anliegt.[1]

Zum Starten der Wiedergabe muss zuerst der grüne “start”-Button aktiviert werden. Zum kurzfristigen Beenden dient der große rote “PANIK”-Knopf. Zwischen diesen START/STOP-Knöpfen gibt eine Anzeige Auskunft über die CPU-Belastung von Coder- und Decoderprozessor.

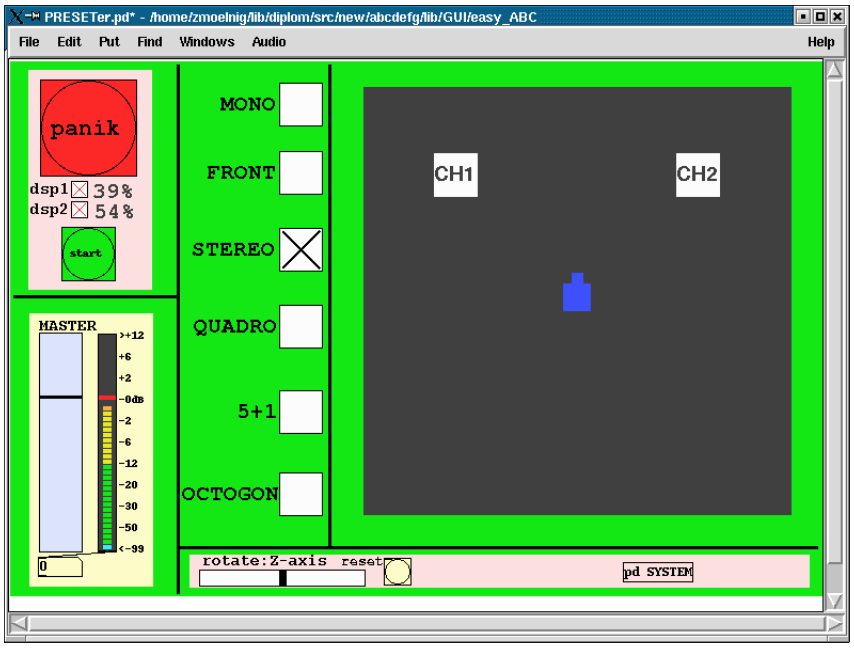


Abbildung 8.1: PRESETter

8.2 3D-Mixer

Der 3D-Mixer lässt eine völlig freie Positionierung beliebiger virtueller Quellen sowie deren visuelle Kontrolle zu.

[1]Dabei wird der Maximalpegel aus allen anliegenden Signalen angezeigt.

8.2.1 Steuerung

Das System-Modul dient wieder zum Starten/Beenden der Audio-Engine via *start* und *PANIK*, sowie zur Regelung der Gesamtlautstärke.

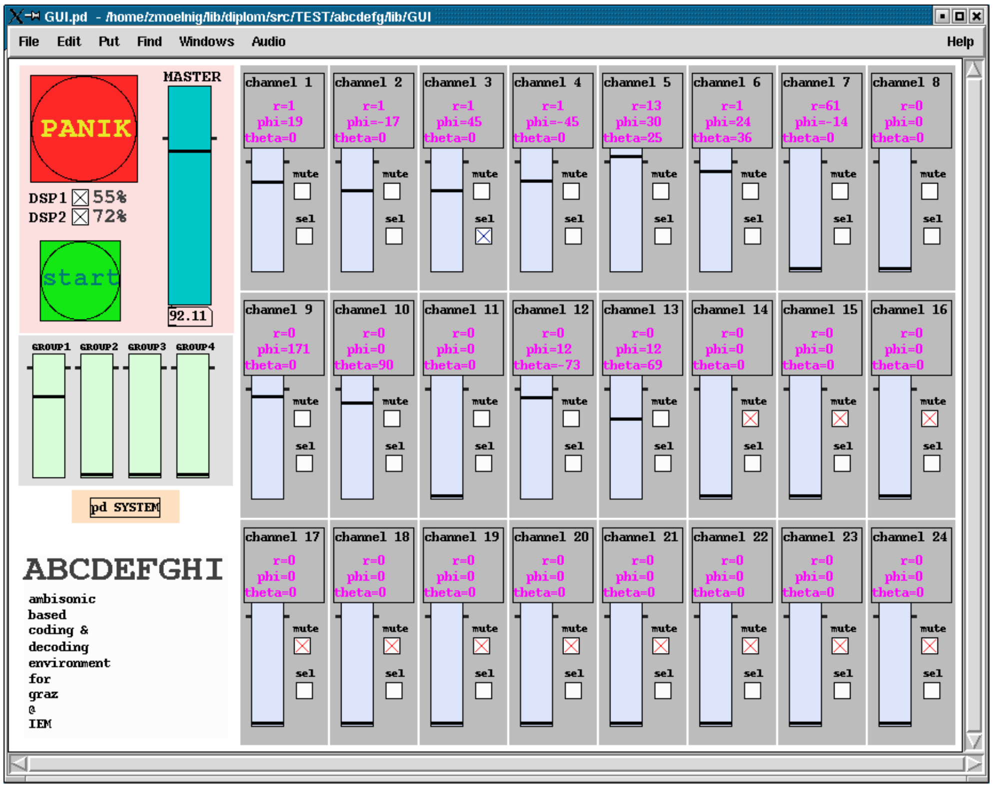


Abbildung 8.2: 3D-Mixer

Die einzelnen virtuellen Quellen können vom Hauptfenster aus gesteuert werden. Dafür gibt es einen Fader zur Lautstärkenänderung sowie drei numerische Werte r, φ, ϑ. Zusätzlich kann noch jeden Kanal muten.

Eine etwas größere Kontrolle über eine virtuelle Quelle erlaubt der Patch, der sich öffnet wenn man auf einen Kanalzug clickt (Abb 8.3).

Neben den Reglern für Verstärkung und Position (hier als Fader ausgeführt) kann hier der Kanal einer von vier Subgruppen zugeordnet werden. Die Zuordnung zu einer Subgruppe erlaubt, dass mehrere Kanäle über einen einzigen Lautstärkeregler (den der Subgruppe) gesteuert werden können. Die

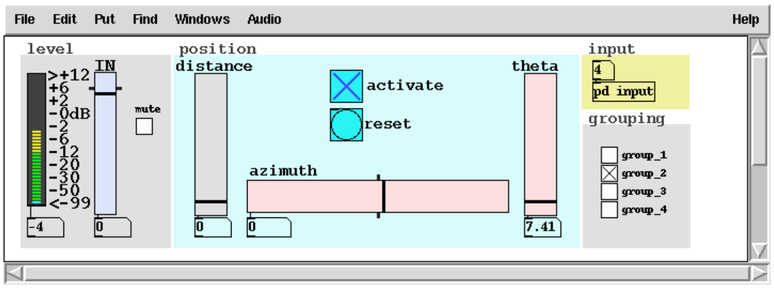


Abbildung 8.3: 3D-Kanalzug

Zuordnung erlaubt jedoch keine gemeinschaftliche Manipulation der Positionen der Gruppenmitglieder.

Außerdem kann man hier bestimmen, welcher Kanal der Audiokarte für die jeweilige virtuelle Quelle verwendet wird. (0 ist der Defaultwert: zB hat die virtuelle Quelle (=Kanal) 4 standardmäßig das Signal des vierten Audiokartenkanals) Zusätzlich gibt es eine Anzeige, die den Pegel des Eingangssignals (Pre-fader !) darstellt.

Im Hauptfenster (Abb. 8.2) gibt es noch einen Subpatch *system*, der grundlegende Systemeinstellungen erlaubt (siehe Abb. 8.4) Dazu gehören

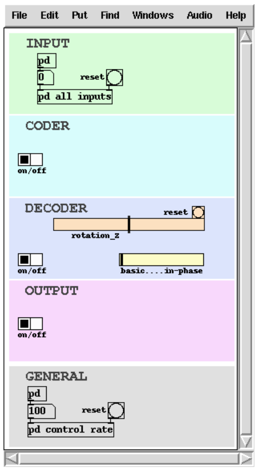


Abbildung 8.4: 3D-Mixer: Systemeinstellungen

neben (De-)Aktivierung von Coder, Decoder und der Calibrierung des Ausgangsmoduls auch die Rotation des Wiedergabeschallfeldes und der Faktor zur Interpolation zwischen *basic* und *in-phase* Decodierung.

Letztendlich können noch komfortabel allen virtuellen Quellen ein bestimmter, für alle gleicher Eingangskanal oder der jeweilige Default-Eingang auf einmal zugewiesen werden.

8.2.2 Visualisierung

Beim periphonen Abmischen von einer größeren Anzahl von Quellen, ist es schwer möglich, immer den Überblick zu behalten.

Eine 3D-Visualisierung sollte dabei helfen (siehe Abb. 8.5).

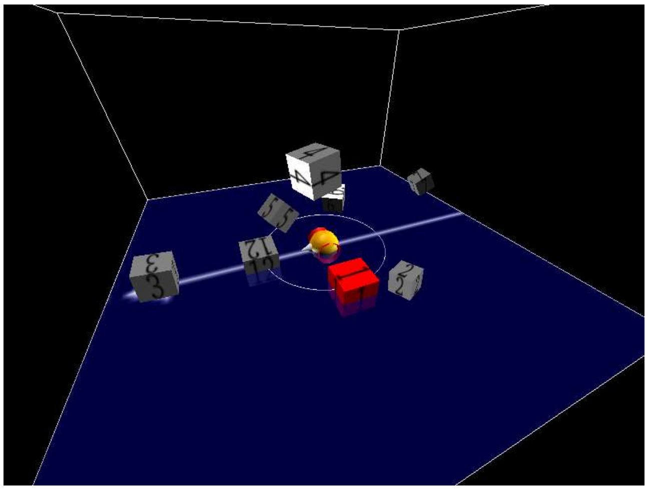


Abbildung 8.5: 3D-Mixer: Visualisierung

Im Zentrum der Visualisierung befindet sich ein stilisierter Kopf als HörerIn. Wird das Schallfeld gedreht, so dreht sich auch der Kopf mit, sodass die relative "vorne"-Richtung sichtbar wird. Die absolute "vorne"-Richtung ($\varphi = \vartheta = 0$) wird durch einen großen Pfeil auf der Horizontalebene kenntlich gemacht.

Die virtuellen Quellen werden als Quader dargestellt, wobei die Ordnungszahl jeder Quelle sichtbar auf ihre Visualisierung texturiert wird. Während die aktuelle Quelle rot hervorgehoben wird, verbleiben die nicht aktiven grau.

Die aktuelle Quelle kann auch sofort im Visualisierungsfenster mit der gedrückten linken Maustaste verschoben werden.

3D-Visualisierungen in Echtzeit sind sehr rechenintensiv. Da die Visualisierung als Open-GL-Applikation implementiert ist, kann jedoch ein Großteil der benötigten Rechenleistung an die Grafikkarte abgegeben werden, sofern diese eine Hardware-Beschleunigung für Open-GL unterstützt. Sollte dies nicht der Fall sein, so muss aus praktischen Gründen das GUI auf einem separaten Prozessor gerechnet werden und kann nicht am gleichen Rechner wie die Audio-Engine betrieben werden.

Kapitel 9

Konkret : alles in Beton gegossen

9.1 Einzelrechnerlösung

Für kleinere Settings (wenige Eingangskanäle und/oder Lautsprecher) ist es möglich, die gesamte Software auf einem einzigen Rechner laufen zu lassen.

Sinnvollerweise werden dann Coder- und Decodermodul in einem einzigen *pd*-Patch zusammengefasst. Dadurch kann umständliches Signal-Routing im Computer über ein *virtual audio cable* vermieden werden.

Da garantiert werden muss, dass der Audio-Engine immer genügend Rechenleistung zur Verfügung steht, wird diese mit einer sehr niedrigen Priorität ausgestattet. Das als getrenntes Programm laufende User-Interface kann dann mit einer hohen Priorität benutzt werden. Die beiden getrennten Programme kommunizieren (lokal am Rechner *localhost*) über eine Netzwerkverbindung.

9.1.1 Audio-Engine

Da im Codec sowohl Coder als auch Decoder vereint sind, sind im Hauptpatch (siehe Abb. 9.1) die Implementationen beider Schnittstellen-Protokolle (1) zu finden. Der *main-in~*-Patch (2) liest $K(=16)$ Kanäle von der Soundkarte, und sendet sie an *send~ <k>_adc*. Der *coder~*-Patch (3) weist diese Signale den einzelnen virtuellen Schallquellen zu und codiert diese über die angegebene Codiervorschrift (hier: Furse-Malham) in L ambisonic-Kanäle. Diese werden mit *send~ codec<l>* an Decoder und Hall (5) verteilt. Das verhallte Signal muss wieder zum eigentlichen Ambisonic-Schallfeld dazugemischt werden. Dazu wird die Tatsache ausgenutzt, dass mit *catch~* die Signale mehrerer *throw~*-Sender überlagert werden können. In (4) werden daher die ambisonic-Signale mit *receive~ codec<l>* von einer verteilenden Struktur über *throw~ codec<l>* in eine sammelende Struktur (gleichen Namens) convertiert. Das Hallmodul empfängt ebenfalls mit *receive~ codec<l>* und sendet das verhallte Signal über *throw~ codec<l>*. Da *pd* die Labels für *send/receive* und *throw/catch* getrennt behandelt, kommt es durch diese Struktur zu keiner Rückkopplung. Es könnten so noch andere Module parallel in den Verarbeitungsweg geschalten werden.

Im *decoder~*-Patch (6) werden die ambisonic-Signale auf ein gegebenes Lautsprecherarray (die Richtungen (zum Bezugspunkt) an denen sich die N Lautsprecherquellen befinden, sind in der Datei *<such-pfad>/angles.mtx* gespeichert) decodiert. Die einzelnen diskreten Lautsprecherkanäle werden mit *throw~ <n>_spk* über den Wiedergabepatch *main-out~* (7) so wiedergegeben, dass Amplituden- und Phasenunterschiede in einem Bezugspunkt ausgeglichen sind.

Der Takt für die Matrixberechnungen von Coder und Decoder in der Messagedomain wird in einem eigenen Patch *calc_trigger* (8) generiert sobald die Audio-Engine eingeschalten wird.

9.2 Zweirechnerlösung

Wird das User-Interface auf einem anderen Rechner ausgeführt, so müssen im Unterschied zur Einrechnermethode, den Protokoll-Schnittstellen die anzusprechenden Rechner mitteilen.

Der (noch immer Coder und Decoder beinhaltende) Patch der Audio-Engine erweitert sich um eben diese Konfiguration. (siehe (1) in Abb. 9.2)

Auf der GUI-Seite wird der Patch *GUI/gui_netinterface* mit den Namen von Coder-Rechner und Decoder-Rechner, die in diesem Fall ident sind aufgerufen (siehe Abb. 9.3)

9.3 Dreirechnerlösung

Muss eine große Anzahl von Quellen auf eine große Lautsprechermatrix codiert und decodiert werden, so reicht unter Umständen die Prozessorleistung nur eines Rechners für das Audio-Processing nicht aus.

Daher werden in solchen Fällen der Coder (evtl. mit Hall) und der Decoder auf getrennten Rechnern ausgeführt.

Der Coder-Ausgang (also die Summe von codierten ambisonic-Signal und Hallsignal) wird über einen Patch *coder-out˜* (Abb.9.4(2)) auf der Audio-Karte ausgegeben.

Ergänzend dazu (Abb.9.5) wird das ambisonic-Schallfeld über einen Patch *decoder-in˜* (2) vom Soundkarteneingang des Decoder-Rechners in den Decoder eingespeist.

Dem User-Interface muss mitgeteilt werden, auf welchen über das Ethernet erreichbaren Rechnern der Coder bzw. der Decoder laufen (siehe Abb. 9.6).

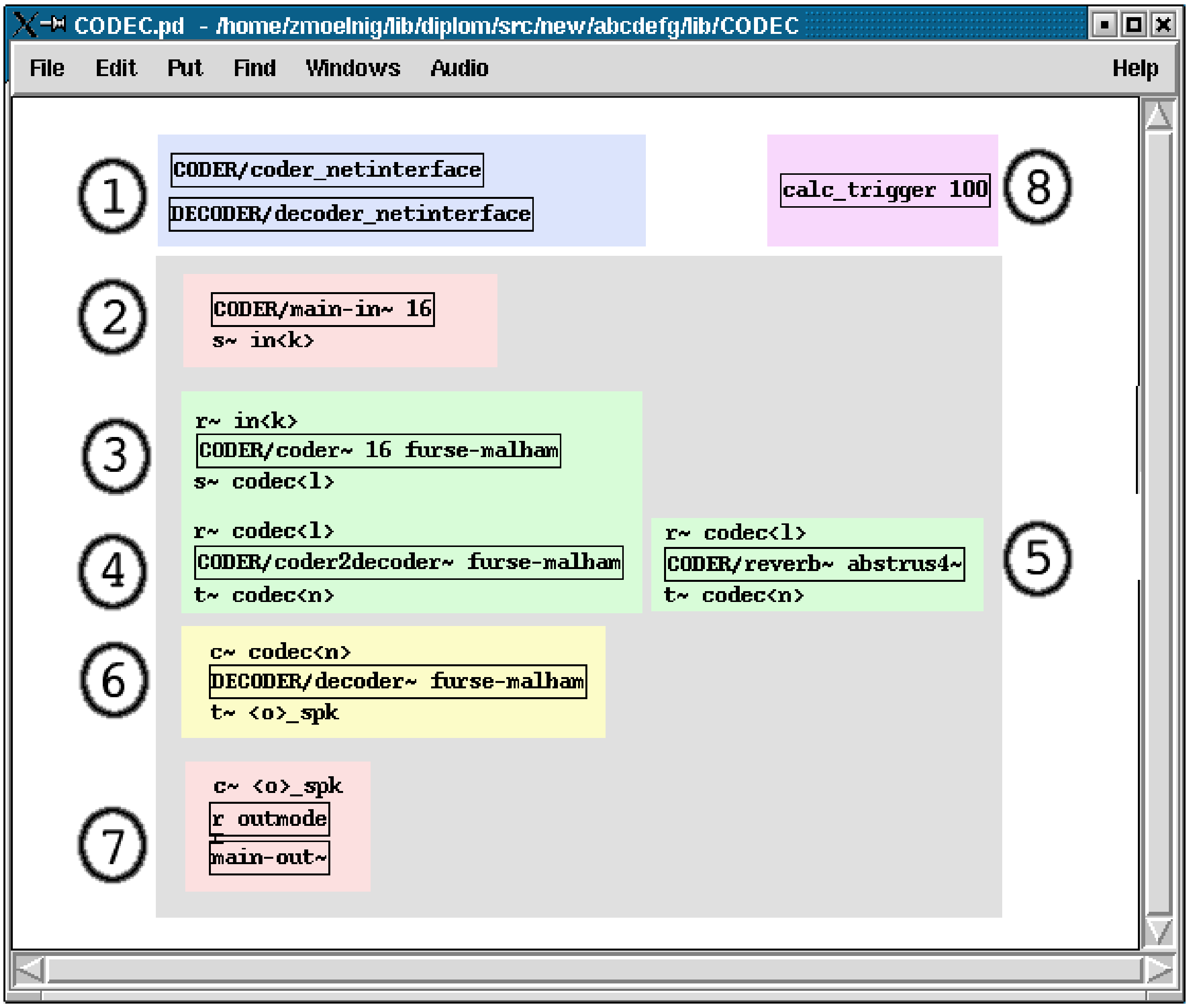


Abbildung 9.1: Einzelrechnerlösung: audio-engine

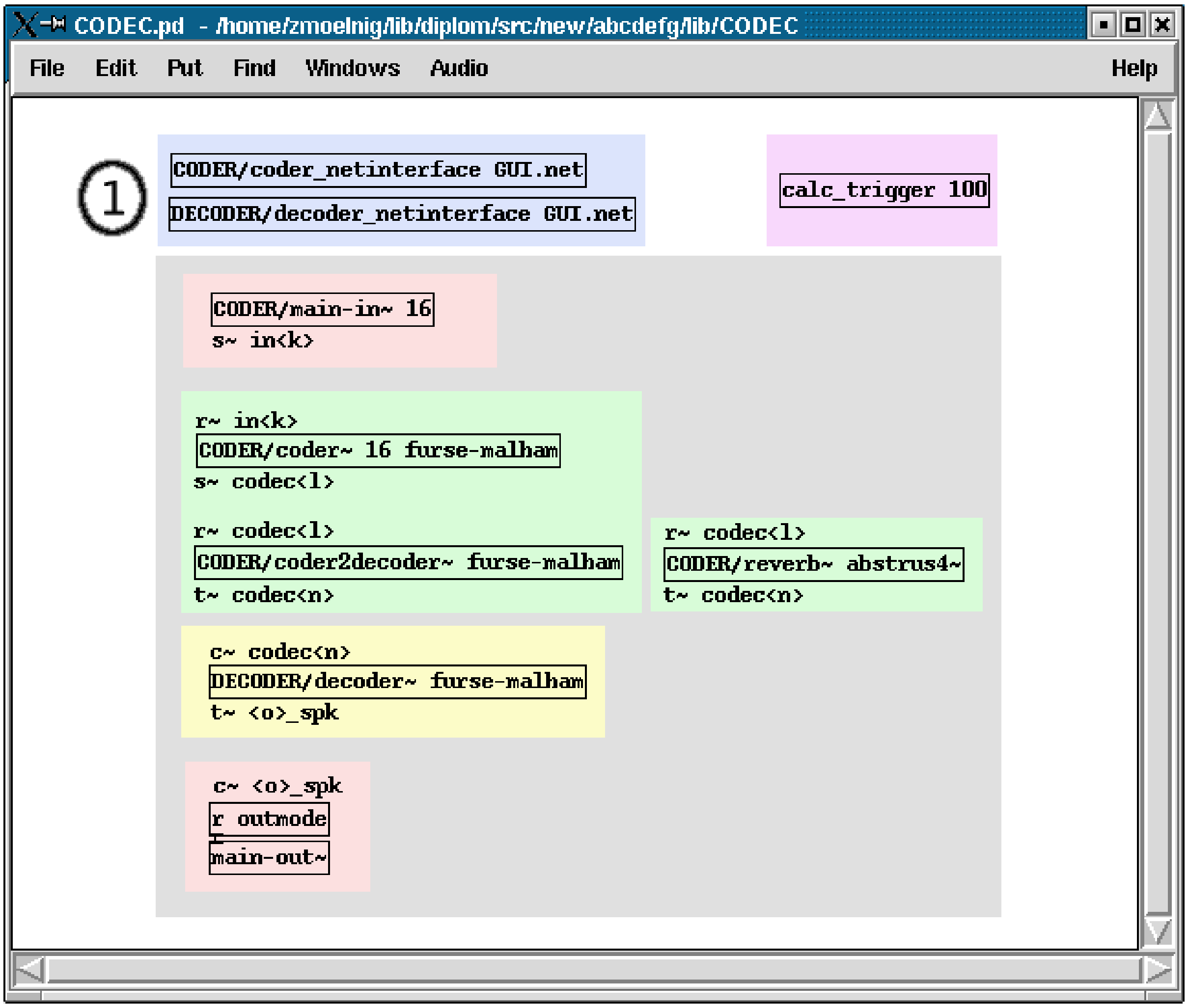


Abbildung 9.2: Zweirechnerlösung: Audio-Engine

```
GUI/gui_netinterface CODEC.net CODEC.net
```

Abbildung 9.3: Zweirechnerlösung: *gui_ interface*-Aufruf

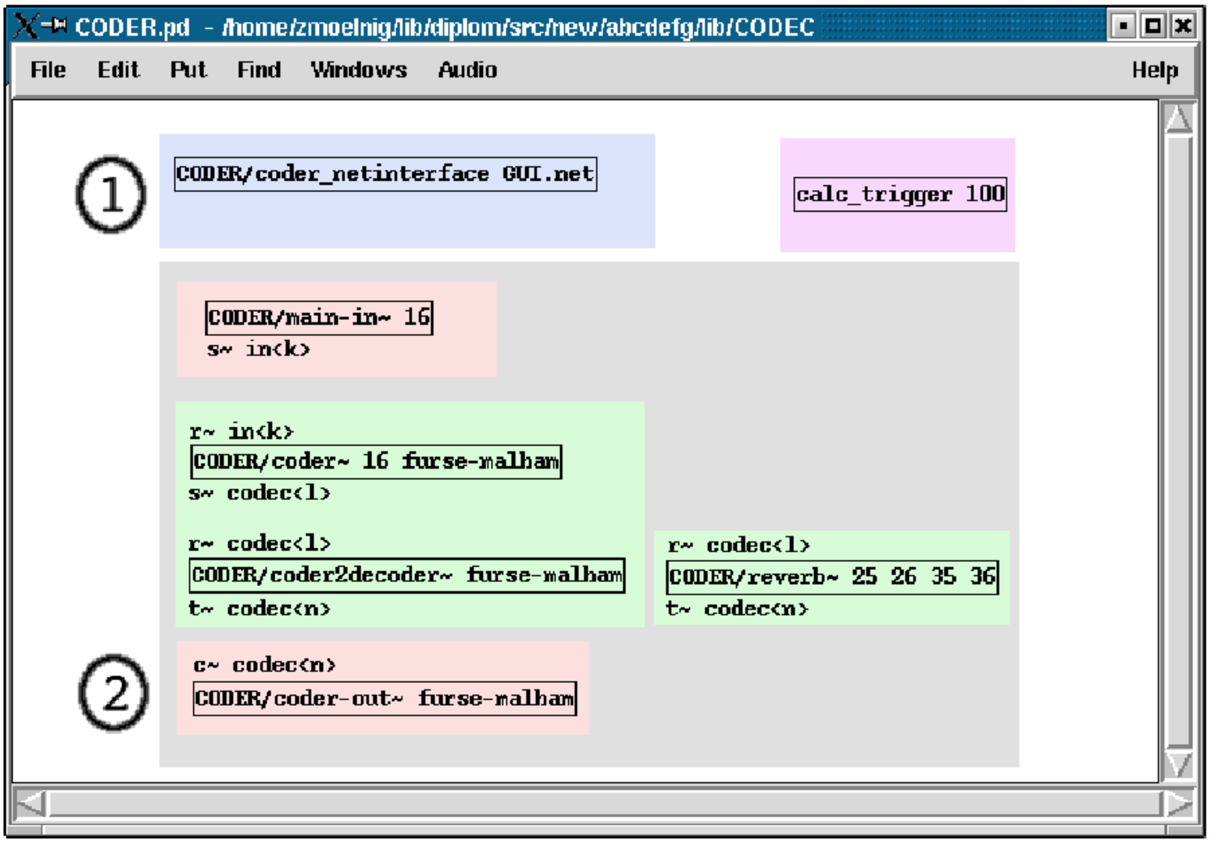


Abbildung 9.4: Dreirechnerlösung: CODER

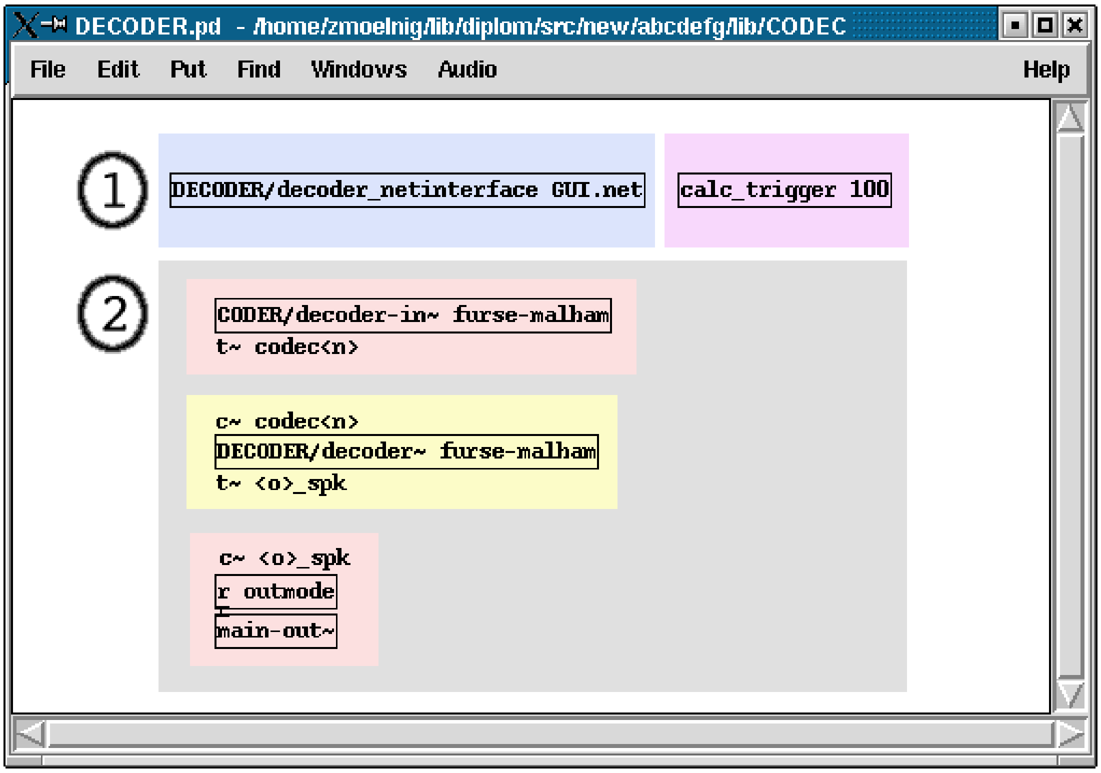


Abbildung 9.5: Dreirechnerlösung: DECODER

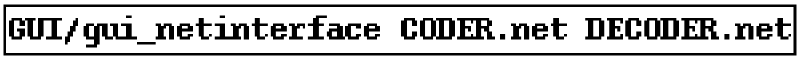


Abbildung 9.6: Dreirechnerlösung: *gui_ interface*-Aufruf

Kapitel 10

Heiteres und Weiteres

Die Muhlipps[1]

Such nicht die Muhlipps! Ihre Welt
ist tintenschwarz und klamm.
Die Glocke schlägt, der Moorhund bellt,
und du versinkst im Schlamm.

Im Schlamm versinkst du nun dafür,
dass du mit frechem Mut
gewagt, zu klopfen an die Tür,
wo es kein Kluger tat.

Die Trauerweiden weinen pech-
schwarz dort am Moderstrand.
Die Schauerweihen brüten, kräch-
zend noch im Schlaf, an Land.

Jenseits der Buckelberge, versteckt in einem Tal,
wo Mond nicht scheint noch Sonne, dort stehn die Bäume fahl,
wo tote Äste faulen im nassen, finstern Kar,
dort hausen sie, die Muhlipps, am grundlos tiefen Maar.

[1] John Ronald Reuel Tolkien, [41]

Die Muhlipps hocken dort beim Licht
einer trüben Kerz zusamm'
und zählen ihr Gold. Sie stört es nicht,
dass es kalt ist im Keller und klamm.

Die Wände schimmeln, die Decke trieft,
man hört wie's ringsum tropft.
Die Muhlipps schlurfen mit sanftem "klieft-
klaft-klieft" zur Tür, wenn's klopft.

Sie öffnen, spähn durch den engen Spalt,
sie strecken die Finger herfür,
sie packen dich, eh du's begreifst, und bald
sind Knochen der Rest von dir!

Über die Buckelberge einen einsamen Weg
durch die Finsterschluchten, über den Spinnensteg
jenseits der Schwarzen Sümpfe ins tiefe Modertal
gehst du die Muhlipps suchen – und wirst für sie zum Mahl.

Anhang A

Realisierung im IEM-Cube

Im *Cube* des Instituts für Elektronische Musik und Akustik, Graz, wurde die hier diskutierte Beschallungsanlage mit folgender Hardware realisiert.

A.1 Audio-Hardware

24 Lautsprecher	**Tannoy**	**System 1200**
	Leistung	180 W (RMS)
	Impedanz	8Ω
	Frequenzgang	40Hz–20kHz
	Empfindlichkeit	95 dB / 1 W @ 1m
	Distortion	$<0.5\%$
	Dispersion	90° (konisch) @ -6dB
	Crossover-Frequenz	1300Hz
12 Endstufen	**Thomann**	**TA-500**
	Leistung	2*170 W @ 8Ω 2*250 W @ 4Ω
	S/N-Abstand	>110 dB
	Klirrfaktor	$<0.05\%$
	Frequenzgang	18Hz–25kHz: -1dB

24 Limiter	**dbx**	**1046**
	Arbeitspegel	+4dBu
	max. Eingangspegel	>+22 dBu
	max. Ausgangspegel	>+22 dBu
	Dynamik	>118 dB
	Rauschen	<-96 dBu
	THD+N	0.008% @ 4 dBu 0.08% @ 20 dBu
	Frequenzgang	20Hz–20kHz: -0.5 dB 0.35Hz–90kHz: -3dB
	Gleichtaktunterdrückung	>50dB @ 1kHz

3 AD/DA-Converter	**RME**	**ADI-8DS**
	Eingangspegel	LoGain (+19dBu @ 0 dBFS) +4 dBu (+13 dBu @ 0 dBFS) -10 dBV (+2 dBV @ 0 dBFS)
	Dynamik (AD)	117 dB
	THD (AD)	<-110 dB
	THD+N (AD)	<-104 dB
	Übersprechdämpfung (AD)	> 130 dB
	Ausgangspegel	HiGain (+19 dBu @ 0 dBFS) +4dBu (+13 dBu @ 0 dBFS) -10dBV (+2 dBV @ 0 dBFS)
	Dynamik (DA)	112 dB (unmuted)
	THD (DA)	<-104 dB
	THD+N (DA)	<-102 dB
	Übersprechdämpfung (DA)	>110 dB
	Frequenzgang	5Hz–21.5kHz: -0.1 dB

A.2 PCs

matrix1		ambisonic-Decoder
	CPU	PIII-800MHz
	Speicher	256MB
	Audio-I/O	3*ADAT 1*AES/EBU
	MIDI	1
	OS	Debian/GNU Linux 2.4.12
matrix2		(ambisonic-Coder)
	CPU	PIII-800MHz
	Speicher	256MB
	Audio-I/O	5*ADAT 2*AES/EBU
	MIDI	1
	OS	Debian/GNU Linux 2.4.12
rockmore		ambisonic-Coder (et al.)
	CPU	Athlon K7-1.4GHz
	Speicher	512MB
	Audio-I/O	6*ADAT 2*AES/EBU
	MIDI	0
	OS	Debian/GNU Linux 2.4.14
cube		GUI (PRESETter)
	CPU	Celeron 500MHz
	Speicher	256MB
	Audio-I/O	0
	MIDI	1
	OS	Debian/GNU Linux 2.2.18

A.3 Lautsprecherlayout

Das Layout der Lautsprecher in der Hemisphäre im *Cube* (siehe Abb. 3.2 und 3.3) ist folgendermaßen:

LS	φ	ϑ
1	0°	0°
2	22°	0°
3	45°	0°
4	71.5°	0°
5	102°	0°
6	138.5°	0°
7	180°	0°
8	221°	0°
9	258°	0°
10	288°	0°
11	315°	0°
12	338°	0°
13	25°	28°
14	69°	28°
15	116°	28°
16	155°	28°
17	205°	28°
18	244°	28°
19	296°	28°
20	335°	28°
21	45°	59.5°
22	135°	57°
23	225°	57°
24	315°	59.5°

Anhang B

Polynome und sphärische Funktionen

Im Kapitel 2 werden Polynome und Funktionen verwendet, die dem geneigten Leser vielleicht nicht geläufig sind. Sie sollten daher hier in aller Kürze wiederholt werden:

B.1 *Legendre*-Polynome

Die *Legendre*-Polynome P_m und die *assoziierten Legendre*-Polynome $P_{m,n}$ folgen der Rekursion:

$$P_{m+1}(\eta) = P_{m+1,0}(\eta) = \frac{1}{m+1}\Big((2m+1)\,\eta P_m(\eta) - mP_{m-1}(\eta)\Big) \quad \text{(B.1)}$$

$$P_{m+1,n+1}(\eta) = P_{m-1,n+1}(\eta) + (2m+1)\sqrt{1-\eta^2}P_{m,n}(\eta) \quad \text{(B.2)}$$

mit den Startwerten

$$P_0 = P_{0,0} = 1 \quad \text{(B.3)}$$

$$P_1 = P_{1,0} = \eta \quad \text{(B.4)}$$

$$P_{1,1} = \sqrt{1-\eta^2} \quad \text{(B.5)}$$

$$P_2 = P_{2,0} = \frac{1}{2}(3\eta^2 - 1) \quad \text{(B.6)}$$

$$P_{2,1} = 3\eta\sqrt{1-\eta^2} \tag{B.7}$$

$$P_{2,2} = 3(1-\eta^2) \tag{B.8}$$

Dies lässt sich auch geschlossen ausdrücken durch

$$P_m(\eta) = P_{m,0}(\eta) = \frac{1}{2^m m!}\frac{d^m}{d\eta^m}(\eta^2-1)^m \tag{B.9}$$

$$P_{m,n}(\eta) = (-1)^n(1-\eta^2)^{\frac{n}{2}}\frac{d^n}{d\eta^n}P_m(\eta) \quad \text{für} \quad 0 \leq n \leq m \tag{B.10}$$

Die *Halbnormalisierung* im Sinne Schmidts ist wie folgt:

$$\bar{P}_{m,n}(\eta) = \sqrt{\varepsilon_n\frac{(m-n)!}{(m+n)!}}P_{m,n}(\eta) \tag{B.11}$$

B.2 Sphärische Funktionen

B.2.1 *Bessel*-Funktionen

Es folgen die Definition der sphärischen *Bessel*-Funktionen n-ter Ordnung $J_n(x)$ und explizit die ersten ausgerechneten Funktionen:

$$J_n(x) = \sqrt{\frac{\pi}{2x}}J_{n+1/2}(x) = (-1)^n x^n\left(\frac{d}{xdx}\right)^n\frac{\sin x}{x} \tag{B.12}$$

$$J_0(x) = \frac{\sin x}{x} \tag{B.13}$$

$$J_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x} \tag{B.14}$$

$$J_2(x) = \left(\frac{3}{x^3}-\frac{1}{x}\right)\sin x - \frac{3}{x^2}\cos x \tag{B.15}$$

B.2.2 *Neumann*-Funktionen

Analog zu den *Bessel*-Funktionen lassen siche die sphärischen *Neumann*-Funktionen n-ter Ordnung $N_n(x)$ definieren als:

$$N_n(x) = (-1)^{n+1}\sqrt{\frac{\pi}{2x}}J_{-n-1/2}(x) = (-1)^{n+1}x^n\left(\frac{d}{xdx}\right)^n\frac{\cos x}{x} \tag{B.16}$$

$$N_0(x) = -\frac{\cos x}{x} \tag{B.17}$$

$$N_1(x) = -\frac{\cos x}{x^2} - \frac{\sin x}{x} \tag{B.18}$$

$$N_2(x) = -\left(\frac{3}{x^3} - \frac{1}{x}\right)\cos x - \frac{3}{x^2}\sin x \tag{B.19}$$

Anhang C

Lautsprecherlayout-Datei

Das verwendete Lautsprecherlayout muss dem Decoder mitgeteilt werden (siehe 6.3.2). Dies geschieht über eine Datei, die defaultmäßig *angles.mtx* heißt.

Diese Datei hat folgendes Format:

$$\begin{array}{ll} \texttt{\#matrix} & < N > \;\; \texttt{2} \\ < \varphi_1 > & < \vartheta_1 > \\ < \varphi_2 > & < \vartheta_2 > \\ \vdots & \vdots \\ < \varphi_N > & < \vartheta_N > \end{array}$$

Der Header gibt an, dass es sich bei dieser Datei um eine Matrix der Dimension $N \times 2$ handelt. Die folgenden N Zeilen bilden die Zeilen einer Matrix, die mit den Wertepaaren $< \varphi_n >< \vartheta_n >$ (in Grad) die Positionen der N Lautsprecher beschreiben.

Sind die Lautsprecher in Form eines regelmäßigen Tetraeders angeordnet, so sieht diese Datei folgendermaßen aus:

```
#matrix 2 4
0 0
120 0
-120 0
0 90
```

Anhang D

Software-Downloads (Auswahl)

Softwarepacket	Beschreibung	URL
Debian/GNU-Linux	Linux-Distribution	http://www.debian.org
rme96xx-0.7	Audiotreiber für RME-Hammerfall-Audiokarte	ftp://xdv.org/pub/gige/rme
pure-data	Audioprocessing-Environment	http://iem.kug.ac.at/pd http://crca.ucsd.edu/~msp/software
zexy	pd-External für Matrizen,...	ftp://iem.kug.ac.at/pd/Externals/ZEXY

Literaturverzeichnis

[1] Jeffery S. Bamford: "An Analysis of Ambisonic Sound Systems of First and Second Order", Thesis presented to the University of Waterloo; Waterloo, Ontario, Canada, 1995

[2] Jeffrey S. Bamford, John Vanderkooy: "Ambisonic Sound for Us", presented at the 99th Convention AES, Preprint 4138; Okt. 1995

[3] Hans-Jochen Bartsch: "Taschenbusch mathematischer Formeln", 17., neubearbeitete Auflage (Carl Hanser Verlag, München, Wien, 1997)

[4] Durand R. Begault: "Challenges to the Successful Implementation of 3-D Sound", *J. Acoust. Soc. Am.*, vol.39, no.11, pp.864-870 (1991)

[5] Durand R. Begault: "3-D Sound for Virtual Reality and Multimedia", (AP Professional, Boston, MA, 1994)

[6] J. Blauert: "Räumliches Hören", S. Hirzel Verlag, Stuttgard, 1974

[7] Walter Jung (Hrsg.), Gerhard Lanz: "Weltbild Kolleg: Abiturwissen Physik", Weltbild Verlag, Augsburg, 1994

[8] A. S. Bregman: "Auditory Scene Analysis – The Perceptual Organization of Sound" (MIT Press, Cambridge, MA, 1990)

[9] A. W. Bronkhorst: "Localization of Real and Virtual Sources", *J. Acoust. Soc. Am.*, vol.98, pp.2542-2553 (1995)

[10] Brengt-Inge Dalenbäck, David McGrath, Andrew Reilly: "Using Auralisation for Creating Animated 3-D Sound Fields Across Multiple Speakers", presented at the 99th Convention AES, Preprint 4127; Okt. 1995

[11] Jérôme Daniel: "Representation de champs acoustiques, application á la transmission et á la reproduction de scénes sonores complexes dans un contexte multiméda", Thèse de doctorat de l'Université Paris 6, Paris, France, 2000

[12] Jérôme Daniel, Jean-Bernard Rault, Jean-Dominique Polack: "Ambisonics Encoding of Other Formats for Multiple Listening Conditions", presented at the 105th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol.46. pp.1034-1035 (1998 Nov.), preprint 4795

[13] Jérôme Daniel, Jean-Bernard Rault, Jean-Dominique Polack: "Acoustic Properties and Perceptive Implications of Stereophonic Phenomena", in *Proc. AES 16th Int. Conf.* (1999), pp.91-102

[14] Michael Droettboom, Ichiro Fujinaga, Karl MacMillan: "Audio Latency Measurements of Desktop Operating Systems" (Peabody Institute of the John Hopkins University; ICMC 2001)

[15] dtv-Atlas: "Musik", Band 1 und 2, 17.Auflage 1997 (Deutscher Taschenbuchverlag GmbH&Co.KG, München, 1977)

[16] R. Nicol, M. Emerit: "3d Sound Reproduction over an Extensive Listening Area: A Hybrid Method Derived from Holphony and Ambisonics", in *Proc. AES 16th Int. Conf.* (1999), pp.436-453

[17] Michael J. Evans, A. I. Tew, James A. S. Angus: "Perceived Performance of Loudspeaker-Spatialized Speech for Teleconferencing" *J. Audio Eng. Soc.*, vol.48, no.9, pp.771-785 (2000)

[18] Angelo Farina, Emanuele Ugolotti: "Software Implementation of B-Format Encoding and Decoding", presented a the 104th Convention of

the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol.46, p.574 (1998 June), preprint 4691

[19] P. Fellgett: "Ambisonics Part One: General System Description", *Studio Sound*, vol.17, pt.8, pp.20-40 (1975)

[20] Richard Furse: "3d audio links and information", 1999. URL: http://www.muse.demon.co.uk/3daudio.html

[21] Hans H. Hiebel, Heinz Hiebler, Karl Kogler, Herwig Walitsch: "Die Medien", Wilhelm Fink Verlag, München, 1998

[22] Robert R. Höldrich, Alois Sontacchi: "Konzepte zur Schallfeldsynthese und Schallfeldreproduktion", Procedings des FA für Akustik, Jahrestagung der ÖPG, Graz

[23] Robert R. Höldrich, Alois Sontacchi: "Schallfeldreproduktion durch ein Verbessertes Holophonie–Ambisonic System",

[24] J. M. Jot, V. Larcher, J.-M. Pernaux: "A Comparative Study of 3-d Audio Encoding and Rendering Techniques", in *Proc. AES 16th Int. Conf.* (1999), pp.281-300

[25] Dave G. Malham: "3-D sound for virtual reality systems using Ambisonic techniques" presented at the VR93 Conference in London in April 1993; University of York, 1993. URL: http://www.york.ac.uk/inst/mustech/3d_audio/vr93papr.htm

[26] Dave G. Malham: "Spatial hearing mechanisms and sound reproduction", University of York, England 1998 URL: http://www.york.ac.uk/inst/mustech/3d_audio/ambis2.htm

[27] Dave G. Malham: "Homogenous and Nonhomogenous Surround Sound Systems", Updated (Mai 2000) version of paper presented at AES UK SSecond Century of AudioConference, London, 7.-8. Juni 1999 URL: http://www.york.ac.uk/inst/mustech/3d_audio/homogeneous.htm

[28] Dave G. Malham: "Second Order Ambisonics – the Furse-Malham Set", Oktober 1999.
URL: http://www.york.ac.uk/inst/mustech/3d_audio/secondor.html

[29] Dave G. Malham: "Experience with Large Area 3D Ambisonic Sound Systems", *Proc. Inst. Acoust.*, vol.14, no.5, pp.209-215 (1992)

[30] D. G. Malham, A. Myatt: "3-D Sound Spatialization Using Ambisonic Techniques", *Computer Music J.*, vol.19, no.4, pp.58-70 (1995)

[31] Markus Nussbaumer: "Aufnahme- und Wiedergabeverfahren für Surround Sound", Diplomarbeit am Institut für Elekronische Musik und Akustik, Graz, November 2000

[32] Alan V. Oppenheim, Ronald S. Schafer: "Digital Signal Processing", (Prentice-Hall Inc., USA, 1975)

[33] pure data mailinglist: (technisches Diskussionsforum). Subskription: mailto:pd-list-request@iem.kug.ac.at?subject=subscribe Archiv: http://iem.kug.ac.at/mailinglists/pd-list/

[34] Mark A. Poletti: "The Design of Encoding Functions for Stereophonic and Polyphonic Sound Systems", *J, Audio Eng. Soc.*, vol.44, no.11?, pp.948-963 (1996 Nov.)

[35] Mark A. Poletti: "A Unified Theory of Horizontal Holographic Sound Systems", *J, Audio Eng. Soc.*, vol.48, no.12, pp.1155-1182 (2000)

[36] Miller S. Puckette: "pure-data", Source-Code (San Diego, 2001; http://www.crca.ucsd.edu/~msp/software.html , 24.12.2001)

[37] Miller S. Puckette: "Pd Documentation", Teil der pure-data Distribution (San Diego, 2001; http://www.crca.ucsd.edu/~msp/software.html , 24.12.2001)

[38] Curtis Roads [et al.]: “The computer music tutorial”, (The MIT Press, Cambridge, Massachusetts, London, England; 1996; fourth reprinting, 1999)

[39] Anthony Romano: “Three-Dimensional Image Reconstruction in Audio”, *J. Audio Eng. Soc.*, vol.35, no.10, pp.749-759

[40] Surround-Sound Mailing List (technisches Diskussionsforum). Subskription: mailto:majordomo@lists.uoregon.edu

[41] J. R. R. Tolkien: “Die Abenteuer des Tom Bombadil und andere Gedichte aus dem Roten Buch”, (Klett-Cotta Verlag, Stuttgard, 1984)

[42] Edwin N. Verheijen: “Sound Reproduction by Wave Field Synthesis”, Proefschrift ter verkrijging van de graad von doctor aan de Technische Universiteit Delft, 1998

[43] Diemer de Vries, Edo Hulsebos, Emmanuelle Bourdillat: “Auralization by Wave Field Synthesis”, Technische Universität Delft, Niederlande

[44] Johannes M. Zmölnig: “How to write externals for pure-data”, (Graz, 2001; http://iem.kug.ac.at/pd/externals-HOWTO/ , 16.09.2001)