



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Robert Johns

**Klangquellen-Positionierung in verschiedenen Surround  
Formaten auf Basis von Positionsdaten einer  
Wellenfeldsynthese-Anlage**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Robert Johns

**Klangquellen-Positionierung in verschiedenen Surround  
Formaten auf Basis von Positionsdaten einer  
Wellenfeldsynthese-Anlage**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Wolfgang Fohl  
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 15. Dezember 2014

**Robert Johns**

**Thema der Arbeit**

Klangquellen-Positionierung in verschiedenen Surround Formaten auf Basis von Positionsdaten einer Wellenfeldsynthese-Anlage

**Stichworte**

SPAOP, Wellenfeldsynthese, VST, Cubase, DAW, Surround 5.1, WFS, Audio-Plugin, Panning

**Kurzzusammenfassung**

Diese Arbeit beschreibt die Umsetzung eines Formatkonverters, der Klangquellen auf der Basis von Positionsdaten einer Wellenfeldsynthese-Anlage in verschiedene Surroundformate konvertiert. Im Verlauf dieser Arbeit wird der Konverter als Erweiterung in das Open Source Audioplugin SPAOP integriert. Für die Verifikation des Konverters wird ein Hörtest durchgeführt und ausgewertet. Dieser veranschaulicht das Ergebnis einer Konvertierung für eine 5.1 Surround-Anlage.

**Robert Johns**

**Title of the paper**

Spatializing in various surround sound formats based on position data of a wave field synthesis system

**Keywords**

SPAOP, wave-field-synthesis, VST, Cubase, DAW, Surround 5.1, WFS, audio-plugin, panning

**Abstract**

The intention of this thesis is to describe the implementation of a format converter. Based on position data of a wave field synthesis system it converts sound sources into different surround sound formats. Over the period of the thesis the converter was integrated into the open source plugin SPAOP. To verify the functionality of the converter a hearing test has been performed with a row of test persons. The results of the tests are visualised in the last chapter.

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>1. Einführung</b>   | <b>1</b>  |
| 1.1. Ausgangssituation und Motivation . . . . .                      | 1         |
| 1.2. Thema dieser Arbeit . . . . .                                   | 2         |
| 1.3. Inhalt . . . . .  | 2         |
| <b>2. Grundlagen</b>   | <b>3</b>  |
| 2.1. Räumliches Hören . . . . .                                      | 3         |
| 2.1.1. Richtungshören . . . . .                                      | 4         |
| 2.1.2. Entfernungswahrnehmung . . . . .                              | 5         |
| 2.1.3. Unschärfe der Lokalisation . . . . .                          | 6         |
| 2.2. Verfahren zur räumlichen Darstellung von Klangquellen . . . . . | 6         |
| 2.2.1. Die Wellenfeldsynthese . . . . .                              | 7         |
| 2.2.2. Mehrkanaltonsysteme mit diskreter Übertragung . . . . .       | 9         |
| 2.3. Panning Laws für Mehrkanaltonsysteme . . . . .                  | 15        |
| 2.3.1. Die „Constant Gain“ Methode . . . . .                         | 16        |
| 2.3.2. Die „Constant Power“ Methode . . . . .                        | 17        |
| 2.3.3. Weitere Methoden . . . . .                                    | 18        |
| 2.4. Tools, Bibliotheken und Werkzeuge . . . . .                     | 19        |
| 2.4.1. Entwicklungsumgebung . . . . .                                | 19        |
| 2.4.2. JUCE-Bibliothek . . . . .                                     | 20        |
| 2.5. Digitale Audioproduktion . . . . .                              | 20        |
| 2.5.1. Digital Audio Workstations . . . . .                          | 20        |
| 2.5.2. VST-Plugins . . . . .   | 21        |
| <b>3. Softwarekonzept und Integration</b>                            | <b>23</b> |
| 3.1. Anforderung . . . . .   | 24        |
| 3.1.1. Allgemein . . . . .   | 25        |
| 3.1.2. Einstellungen im Konverter . . . . .                          | 25        |
| 3.2. Analyse der Ausgangssituation . . . . .                         | 27        |
| 3.2.1. Softwarearchitektur von SPAOP . . . . .                       | 28        |
| 3.2.2. Audiodaten in SPAOP . . . . .                                 | 30        |
| 3.2.3. Definition der Arbeitsschritte . . . . .                      | 31        |
| 3.3. Implementierung von SPAOP-MFC . . . . .                         | 32        |
| 3.3.1. Die Algorithmen . . . . .                                     | 33        |
| 3.3.2. Lineare Interpolation der Samples . . . . .                   | 41        |

|           |   |           |
|-----------|---|-----------|
| 3.3.3.    | Implementierung der MFC-Klassen . . . . .       | 41        |
| 3.3.4.    | Erweiterung der grafischen Oberfläche . . . . . | 44        |
| 3.3.5.    | Manipulation bestehender Klassen . . . . .      | 44        |
| <b>4.</b> | <b>Testen der Funktionalität</b>                | <b>48</b> |
| 4.1.      | Allgemein . . . . .                             | 48        |
| 4.2.      | Hörtest . . . . .                               | 49        |
| 4.2.1.    | Aufbau . . . . .                                | 49        |
| 4.2.2.    | Ergebnisse des Hörtests . . . . .               | 53        |
| 4.2.3.    | Auswertung der Ergebnisse . . . . .             | 59        |
| 4.2.4.    | Zusammenfassung . . . . .                       | 61        |
| <b>5.</b> | <b>Fazit / Ausblick</b>                         | <b>62</b> |
| 5.1.      | Zusammenfassung . . . . .                       | 62        |
| 5.2.      | Fazit . . . . .                                 | 62        |
| 5.3.      | Ausblick . . . . .                              | 62        |
| <b>A.</b> | <b>Anhang</b>                                   | <b>64</b> |
| <b>B.</b> | <b>Glossar</b>                                  | <b>71</b> |
|           | <b>Abbildungsverzeichnis</b>                    | <b>72</b> |
|           | <b>Tabellenverzeichnis</b>                      | <b>74</b> |
|           | <b>Literaturverzeichnis</b>                     | <b>75</b> |

# 1. Einführung

Das Departement Informatik der Hochschule für Angewandte Wissenschaften in Hamburg hat auf dem Campus Berliner Tor das Wellenfeldsynthese-Labor. In diesem Labor ist eine Wellenfeldsynthese-Anlage installiert [Foh13], auf die in einem späteren Kapitel noch näher eingegangen wird. Diese Anlage steht Studenten aus verschiedenen technischen Studiengängen zur Verfügung, um für ihren Fachbereich zu unterschiedlichen Themen und Problemstellungen rund um das Thema „Audio“ zu forschen oder eigene Projekte zu verfolgen und umzusetzen. Um die Funktionalität dieser Anlage zu erweitern und die Bedienbarkeit kontinuierlich zu verbessern, müssen Tools und Audio-Plugins entwickelt werden, die es ermöglichen, das Arbeiten mit dieser Anlagen so intuitiv und einfach wie möglich zu machen.

## 1.1. Ausgangssituation und Motivation

Martin Hansen hat in seiner Bachelorarbeit zum Thema „Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines Audio-Plugins“ das Audio-Plugin „Spatial Audio Object Positioner“, kurz SPAOP, entwickelt [Han14]. Dieses Plugin ermöglicht es, Klangquellen für die Wellenfeldsynthese-Anlage auf einer grafischen Oberfläche zu steuern, Automatisierungsdaten aufzuzeichnen und als Projekt wieder abzuspielen. Die Funktionalität, wie sie SPAOP bietet, ist für die Wellenfeldsynthese-Anlage neu und ein wichtiger Schritt dahin, sich der gewohnten Arbeitsumgebung von professionellen Tontechnikern zu nähern, die sie bereits aus dem Bereich der Mehrkanal-Stereofonie gewohnt sind.

Ein Nachteil bleibt weiterhin, dass Audio-Projekte doppelt erstellt werden müssen. Produktionen für handelsübliche Mehrkanal-Tonformate können zwar auf der Wellenfeldsynthese-Anlage abgespielt werden, indem die Lautsprecher als virtuelle Klangquellen abgebildet werden, sie können jedoch nicht die Eigenschaften und Vorteile nutzen, die diese Anlage bietet. Umgekehrt ist es noch problematischer, Audio-Produktionen für die Wellenfeldsynthese-Anlage können nicht auf standardmäßigen Surround-Anlagen wiedergegeben werden. Will man die Vorteile einer Wellenfeldsynthese-Anlage nutzen und die Wiedergabe gleichzeitig auf Mehrkanaltonsystemen unterstützen, muss man derzeit den Nachteil in Kauf nehmen, zwei oder mehr Projekte zu erstellen.

### 1.2. Thema dieser Arbeit

Im Rahmen dieser Bachelorarbeit wird das Audio-Plugin SPAOP von Martin Hansen aufgegriffen und mit einem Formatkonverter erweitert. Dieser soll es ermöglichen, Audio-Projekte die für die Wellenfeldsynthese-Anlage erstellt worden sind, auch auf anderen Surround-Anlagen wiederzugeben. Diese Erweiterung soll das mehrfache Produzieren von Audio-Projekten überflüssig machen.

Hauptziel ist es, dieses Plugin im Bereich des Wellenfeldsynthese-Labors der HAW Hamburg einzusetzen. Dennoch wird in dieser Erweiterung das Konzept von Martin Hansen fortgeführt, Kompatibilität zu andere Anlagen zu gewährleisten.

### 1.3. Inhalt

In Kapitel 2 werden die Grundlagen aufgearbeitet, die für die Umsetzung dieser Arbeit notwendig sind. Dazu gehören die Grundlagen des räumlichen Hörens, die Eigenschaften der verwendeten Formate und die Möglichkeiten, Klangquellen in Surround Formaten zu positionieren. Am Ende des zweiten Kapitels werden die in dieser Arbeit verwendeten Tools, Werkzeuge und Bibliotheken vorgestellt.

Die Umsetzung des Multi-Format-Konverters (ab hier nur noch kurz SPAOP-MFC genannt) wird in Kapitel 3 beschrieben. Dabei wird einleitend das zu erweiternde Plugin SPAOP von Martin Hansen vorgestellt.

Der am Ende der Arbeit durchgeführte Hörtest wird in Kapitel 4 dokumentiert. Dort werden die verfolgten Testziele und das Vorgehen bei der Durchführung des Testes beschrieben und im Anschluß daran die Testergebnisse präsentiert und bewertet.

Kapitel 5 fasst abschließend die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf die Zukunft.

## 2. Grundlagen

In diesem Kapitel werden die Grundlagen des räumlichen Hörens erläutert und ein Überblick über gängige Fachbegriffe geschaffen. Dieser Abschnitt erhebt keinen Anspruch auf Vollständigkeit, er dient lediglich dazu, die für diese Arbeit notwendigen Begriffe einleitend zu erklären und ein Grundverständnis aufzubauen. Im Anschluss daran werden die Wellenfeldsynthese erläutert sowie die Verfahren zur Darstellung von Raumklang vorgestellt, die vom Format-Konverter eingesetzt werden sollen. Daraufhin folgt ein Überblick über die „Panning Laws“, mit denen Klangquellen in stereofonen Mehrkanalton-Formaten räumlich positioniert werden, die Werkzeuge, die bei der digitalen Erstellung von Audioinhalten Einsatz finden sowie die Bibliothek und die Schnittstellen, die bei diesem Plugin verwendet wurden.

### 2.1. Räumliches Hören

Neben den Fähigkeiten, Geräusche anhand ihrer Tonhöhe, der Lautstärke oder ihrer Klangfarbe zu unterscheiden, hat das menschliche Ohr zudem die Fähigkeit, durch Richtungs- und Entfernungsempfinden deren Positionen zu bestimmen. Geräusche, denen man folgen will, lassen sich lokal bestimmen und hervorheben [Web07, S.118]. Eine Fähigkeit, die tagtäglich ihre Anwendung findet, wie etwa in Gesprächen. Beim räumlichen Hören muss man das Hörereignis vom Schallereignis unterscheiden. Das Schallereignis ist der physikalische Vorgang selbst, wodurch Schall an einem exakten Ort entsteht. Im Gegensatz dazu entsteht das Hörereignis durch Informationsauswertung im Gehirn und muss nicht auf ein Schallereignis zurückzuführen sein [Sau05]. Vor allem aber kann ein ausgewertetes Hörereignis in seiner Wahrnehmung von dem real stattfindenden Schallereignis abweichen, hierauf wird später noch einmal eingegangen.

Für die einheitliche Betrachtung von Hörereignissen existiert ein kopfbezogenes Koordinatensystem, das in Abbildung 2.1 dargestellt ist. In diesem Koordinatensystem sind drei Achsen und zwei Winkel definiert. Wie zu erkennen, ist der Kopf des Hörers der Ursprung des Koordinatensystems und die Blickrichtung ist fest  $0^\circ$  auf allen Ebenen. Ereignisse, die auf der Horizontalebene lokalisiert werden, werden mit dem Horizontalwinkel  $\varphi$  angegeben. Der Winkel  $\vartheta$  ist der Erhebungswinkel und stellt Lokalisationen auf der Medianebene dar.



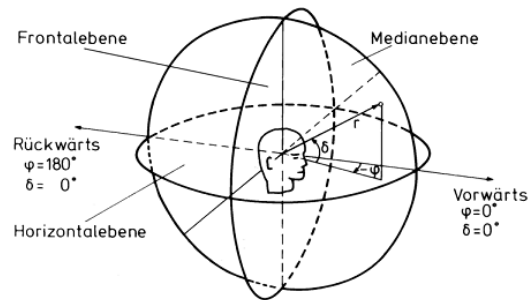


Abbildung 2.1.: Kopfbezogenes Koordinatensystem aus [BH89]

Da sich diese Arbeit mit der Übersetzung von Rauminformationen aus dem zweidimensionalen Raum bezieht, bleibt die Medianebene im weiteren Verlauf unbetrachtet. Somit wird sich auf die Lokalisation auf der Horizontalebene begrenzt.

### 2.1.1. Richtungshören

Um Schall im Raum lokalisieren zu können, benötigen wir beide Ohren. So ist es uns möglich, Signal-Merkmale auszuwerten und zu Rauminformationen zu verarbeiten. Diese Merkmale, die wir dabei aufnehmen und verarbeiten, sind die sogenannten interauralen Merkmale. Der Begriff „interaural“ leitet sich vom lateinischen ab und bedeutet „zwischen den Ohren“. Das Gehirn wertet Informationen, die sich zwischen den Ohren unterscheiden, aus. Die Unterschiede, die hierbei auftreten können, sind der Zeitpunkt des Auftretens am Ohr und die Lautstärke, die am Ohr anliegt [Web07, S.119]. Diese Merkmale, die zur Wahrnehmung auf der Horizontalebene führen, heißen interaurale Laufzeitdifferenz und interaurale Pegeldifferenz [Bla74].

#### Interaurale Laufzeitdifferenz

Die interaurale Laufzeitdifferenz ist der Zeitunterschied, der beim Auftreffen eines Schallereignisses zwischen beiden Ohren entsteht. Das Ohr, das der Schallquelle zugewandt ist, nimmt das Signal hierbei eher wahr als das abgewandte Ohr. Abbildung 2.2 zeigt, wie der Schall erst den Umweg um den Kopf herum nehmen muss, um am abgewandten Ohr wahrgenommen zu werden.

Bei Schall, der direkt von vorne oder von hinten auf den Kopf auftritt, beträgt die Laufzeitdifferenz 0s, da der Schall die gleiche Strecke zu beiden Ohren zurücklegen muss. Seitlich bei  $90^\circ$  (beziehungsweise  $270^\circ$ ) liegt die größte Laufzeitdifferenz von  $630\mu\text{s}$  vor, weil der Schall zum abgewandten Ohr um den ganzen Kopf herum gebeugt werden muss. Die für das Gehör kleinste

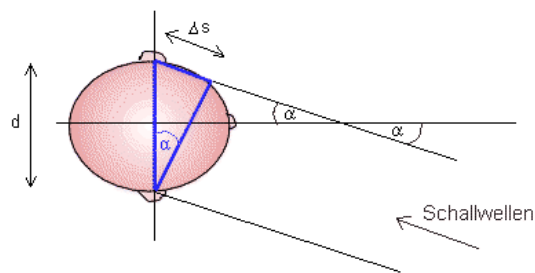


Abbildung 2.2.: Laufzeitunterschied bei seitlichem Schallereignis [Kre14]

wahrnehmbare Laufzeitdifferenz beträgt  $10\mu\text{s}$  [Gör08]. Teilt man die größte Laufzeit, die auftreten kann, durch die Schrittlänge von  $10\mu\text{s}$ , erhält man eine Schrittzahl von 63. Nimmt man die maximale Ausdehnung von  $90^\circ$ , die von der kleinsten bis zur größten Laufzeit entstehen kann und teilt diese durch 63, leitet sich daraus eine theoretische Auflösungsgenauigkeit von  $1,5^\circ$  ab. Praktisch konnten in Versuchen dazu bisher nur Genauigkeiten von  $3^\circ$  nachgewiesen werden. Aber auch  $3^\circ$  zeigen, dass schon geringe Positionsveränderungen mit der interauralen Laufzeitdifferenz wahrgenommen werden können [Wei08].

### Interaurale Pegeldifferenz

Bei der interauralen Pegeldifferenz kommen ein entfernungsabhängiger und ein frequenzabhängiger Effekt zur Geltung. Bei dem entfernungsabhängigen ist die Lautheit differenzierbar, was sich durch den Abstand zur Schallquelle und der damit zusammenhängenden Abnahme der Schallenergie ergibt. Das zugewandten Ohr ist in der Distanz der Schallquelle dichter, als es das abgewandten Ohr ist und wird dadurch mit einem wahrnehmbar höheren Pegel beschallt. Dieser Effekt verringert sich allerdings mit zunehmender Entfernung zur Schallquelle [Web07, S.121]. Der Effekt, der dann zum Tragen kommt, ist die Tatsache, dass der Kopf selbst für den Schall ein Hindernis darstellt und der Pegel frequenzabhängig vermindert wird. Die Ohren stellen dann einen frequenzabhängigen Intensitätsunterschied fest [Web07]. Diese beiden Effekte zusammen bilden die interaurale Pegeldifferenz.

### 2.1.2. Entfernungswahrnehmung

Die Entfernungswahrnehmung wird durch die Einflüsse von Hallerscheinung, Klangfarbenunterschiede und Lautstärkeänderung bestimmt. Nimmt das Gehör ein Geräusch wahr, dem ein Nachhall unterliegt, wird automatisch eine gewisse Entfernung assoziiert, je nachdem wie

intensiv der Hall ist. Zudem spielt bei der Entfernungswahrnehmung die Klangfarbe eine Rolle, je dumpfer ein Geräusch ist, desto weiter entfernt wirkt es für den Hörer. Der Grund dafür ist, dass die hohen Frequenzanteile mit der Entfernung abnehmen. Der Entfernungswahrnehmung durch Lautstärkeänderung ist vorausgesetzt, dass dem Hörer die Klangquelle und deren charakteristische Lautstärke bekannt ist. Trifft das nicht zu, kann die Entfernung nur schwer korrekt abgeschätzt werden [Web07].

### 2.1.3. Unschärfe der Lokalisation

Dieser Abschnitt soll grundsätzlich ein Verständnis dafür schaffen, dass Fehler bei der Lokalisation auftreten und anhand von Beispielen mögliche Gründe aufzeigen. Das Ziel ist nicht die Unschärfe in der Lokalisation exakt wiederzugeben oder im Detail zu erklären.

Bei der Lokalisation von Schallquellen spielen viele Faktoren eine Rolle, die dazu führen können, dass die Ortung unscharf oder sogar komplett falsch angenommen werden kann. Ein Faktor bei der Richtungswahrnehmung ist die Frequenz eines Geräusches. Unterhalb von 300 Hz zum Beispiel ist die Wellenlänge so groß, dass der Schall um den Kopf gebeugt wird. Die Schallquelle kann nur noch am Pegel oder mit Hilfe der Laufzeit geortet werden [Web07, S.124]. Darum spielt es bei Surroundanlagen auch keine große Rolle, wo genau der LFE-Lautsprecher aufgestellt wird. Bei einer Frequenz über 1600 Hz wiederum kann das Gehirn Geräusche nicht mehr anhand der interauralen Laufzeitdifferenz orten, da die Phasen an beiden Ohren nicht mehr in Laufzeit unterscheidbar sind. Weitere Probleme bei der Lokalisation anhand der interauralen Laufzeit entstehen durch gleiche Laufzeiten bei verschiedenen Richtungen. Mit dem Begriff „*cone of confusion*“ wird ein kegelförmiger Bereich beschrieben, dargestellt in Abbildung 2.3, auf dessen Oberfläche dargestellte Klangquellen gleiche Laufzeitunterschiede erzeugen. Der Hörer kann bei ruhiger Kopfhaltung die Quellen nicht korrekt lokalisieren, wodurch eine andere Position mit dem selben Auftrittswinkel angenommen wird. Ein weiterer Fall gleicher Laufzeiten ist die *Vorne-Hinten-Ambiguität*, wodurch der Hörer nicht zwischen Quellen von vorne und hinten unterscheiden kann, wie zum Beispiel bei  $0^\circ$  und bei  $180^\circ$ . Durch leichte Bewegungen mit dem Kopf können diese Täuschungen aber beseitigt werden und die Quelle kann korrekt wahrgenommen werden [Zim04].

## 2.2. Verfahren zur räumlichen Darstellung von Klangquellen

Die in dieser Arbeit betrachteten Verfahren begrenzen sich auf die klassischen Formate, die seit Jahrzehnten Anwendung finden, sowie die modernen aktuellen Formate. Nicht betrachtet

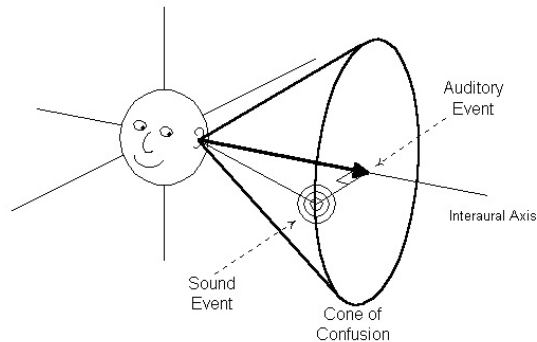


Abbildung 2.3.: Cone of Confusion [Wes98]

werden alle Formate, die heutzutage keine oder kaum noch praktische Anwendung finden, ein klassischer Vertreter ist zum Beispiel die Quadrofonie.

### 2.2.1. Die Wellenfeldsynthese

Die Wellenfeldsynthese ist ein an der TU Delft entwickeltes Verfahren zur räumlichen Darstellung von Klangquellen. Anders als bei der Mehrkanal-Stereofonie werden keine **Phantom-schallquellen** nachgebildet, sondern virtuelle Klangquellen erzeugt, bei denen versucht wird, das Schallereignis nachzubilden. Das Prinzip, nach dem die Wellenfeldsynthese funktioniert, ist das Huygens-Prinzip.

#### Das Huygens-Prinzip

Das Huygens-Prinzip wurde im 17. Jahrhundert durch den gleichnamigen niederländischen Astronom, Mathematiker und Physiker Christiaan Huygens entdeckt. Es sagt aus, dass jeder Punkt auf einer Wellenfront einer sich ausbreitenden Welle wieder als Ausgangspunkt einer neuen Welle gesehen werden kann. Die an diesen Punkten entstehenden Wellen nennt man Elementarwellen [Gör08]. Betrachtet man alle einzelnen Elementarwellen und schaut sich die Überlagerungspunkte an, erhält man wieder die ursprüngliche Wellenfront.

#### Verfahren der Wellenfeldsynthese

Bei der Wellenfeldsynthese nutzt man dieses Prinzip, indem man eine Vielzahl von Lautsprechern in einem sehr dichten Abstand zueinander anordnet und einzeln ansteuert [Ber88]. So lassen sich Elementarwellen erzeugen und Wellenfronten nachbilden, die einer realen Klangquelle gleichkommen. Abbildung 2.4 zeigt das Funktionsprinzip, wie in der Wellenfeldsynthese

eine virtuelle Schallwelle am Lautsprecher-Array nachgebildet wird. Die von den Lautsprechern ausgehenden Elementarwellen rekonstruieren diese virtuelle Wellenfront.

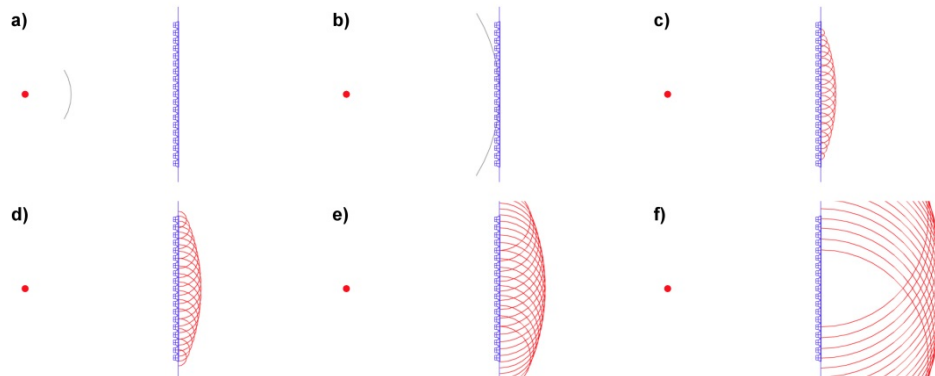


Abbildung 2.4.: Ablauf der Synthese einer Schallwelle [Oel13]

Mit diesem Verfahren ist die Lokalisation von Klangquellen nicht mehr nur auf den sogenannten **Sweet Spot** begrenzt, der Bereich innerhalb beziehungsweise hinter den wiedergebenden Lautsprechern bildet eine Hörzone.

### Aufbau der Anlage im Labor der HAW

Das Labor für Wellenfeldsynthese an der HAW wurde 2011 eingerichtet [Foh13]. Die im Labor installierte Anlage stammt von der Firma Four Audio<sup>1</sup> und wird mit der Software WONDER<sup>2</sup> betrieben, welche die Wellenfeldsynthese berechnet. Die Anlage selbst verfügt über 26 Lautsprecher, die zu einem Rechteck angeordnet auf einer Höhe von zwei Meter aufgehängt sind. Jeder Lautsprecher hat 8 separate Audiokanäle, mit denen er angesteuert werden kann, woraus sich insgesamt 208 Wiedergabekanäle ergeben [Han14]. Die Abmessungen der Anlage betragen ca. 5 x 5,7 m und werden als voreingestellte Startwerte im SPAOP-MFC verwendet, wobei die Tiefe von 5,7 m auf 6 m gerundet wird. Abbildung 2.5 zeigt den Aufbau der Anlage im Labor mit Powerwall und Infrarotkameras eines Tracking-Systems.

Betrieben wird diese Anlage mit drei Linux-Rechnern im Cluster, auf denen WONDER das Rendering übernimmt. Für die Übertragung der Audiodaten und der Bereitstellung der grafischen Oberfläche von WONDER ist ein Apple Mac zuständig. Damit die Datenübertragung fehlerfrei zwischen den verschiedenen Komponenten abläuft, existieren zwei separate kabelge-

---

<sup>1</sup><http://fouraudio.com/> - Abruf: 2014-11-04

<sup>2</sup>Quellcode: <http://sourceforge.net/projects/wonder/> - Abruf: 2014-11-04

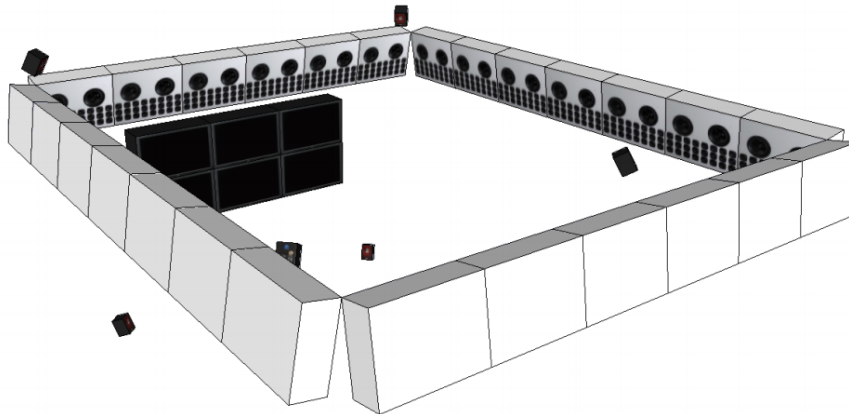


Abbildung 2.5.: Skizze der Wellenfeldsyntheseanlage im Labor der HAW [Nog12, S.19]

bundene Netzwerke. Ein Netzwerk dient der Kommunikation der WONDER-Komponenten und das andere der Kommunikation mit den Lautsprecher-Panels. Das verwendete Netzwerkprotokoll, das bei der Kommunikation aller Komponenten verwendet wird, ist Dante, ein Protokoll zur Audioübertragung von der Firma Audinate<sup>3</sup> [Han14].

Neben der Wellenfeldsynthese-Anlage verfügt das Labor noch über eine 5.1 Surround-Anlage, mit der Vergleiche und Hörtests durchgeführt werden können. An zwei neu beschafften Audio-Arbeitsplätzen können Audioinhalte für diesen Zweck erzeugt werden. Eine Powerwall, bestehend aus sechs HD-Monitoren, dient als großer Bildschirm und wird von einem gesonderten Windows-Rechner angesteuert. Ein Tracking-System der Firma Advanced Realtime Tracking<sup>4</sup> kann mittels Infrarotkameras zum Positionstracking verwendet werden. Zukünftig kommen noch zwei Video-Arbeitsplätze in das Labor, mit denen später im Zusammenspiel mit den Audio-Arbeitsplätzen auch Audio-Video-Schnitt-Bearbeitung für die Wellenfeldsynthese-Anlage durchgeführt werden soll.

### 2.2.2. Mehrkanaltonsysteme mit diskreter Übertragung

Die Schreibweisen, wie Formate abgekürzt werden, sind sehr unterschiedlich. In der Implementierung von SPAOP-MFC wird die Codierung in folgender Form verwendet: *Frontlautsprecher / Surroundlautsprecher / Effektgerät*. Daraus lässt sich die verwendete Konstellation besser erkennen, da zwischen Front- und Surroundlautsprechern unterschieden wird. Eine oft gebräuchliche Bezeichnung wie „5.1“ wird demnach mit 3/2/1 beschrieben.

---

<sup>3</sup><http://www.audinate.com> - Abruf: 2014-11-04

<sup>4</sup><http://www.ar-tracking.com> - Abruf: 2014-11-04

Seit der Einführung der Stereophonie in der Mitte des 20. Jahrhunderts und der darauf aufbauenden Entwicklung der stereofonen Mehrkanaltonformate spricht man allgemein vom Mehrkanalton. Jeder Lautsprecher in einer Mehrkanalton-Anordnung verfügt über einen eigenen Tonkanal, in dem die für den jeweiligen Lautsprecher zu einem beliebigen Zeitpunkt vorgesehenen Toninformationen enthalten sind. Anders als bei der klassischen Monophonie, in der jeder Lautsprecher zu jeder Zeit exakt dieselben Töne wiedergibt, können diese im Mehrkanalton auf den Kanälen verteilt werden, wodurch sich eine räumliche Darstellung erzeugen lässt.

### Phantomschallquellen

Wie einleitend bereits erwähnt, stimmt bei einer realen Schallquelle das wahrgenommene Hörereignis mit dem tatsächlichen Schallereignis überein. Bei der räumlichen Darstellung von Schallquellen in der Mehrkanal-Stereophonie bildet man diese nach, indem man die Schallsignale einer nachzubildenden Schallquelle über ein Lautsprecherpaar wiedergibt. Die Schallquelle wird aus der Richtung lokalisiert, wo sich die Schallwellen überlagern. Zwischen den Lautsprechern entsteht dadurch eine sogenannte Phantomschallquelle. Sind die Signale aus beiden Lautsprechern kohärent, lokalisiert man die Phantomschallquelle auf der Mitte zwischen beiden Lautsprechern [Gör08, S.123]. Abbildung 2.6 zeigt das Beispiel, wie der von beiden Lautsprechern wiedergegebene Schall ein in der Mitte liegendes Hörereignis erzeugt.

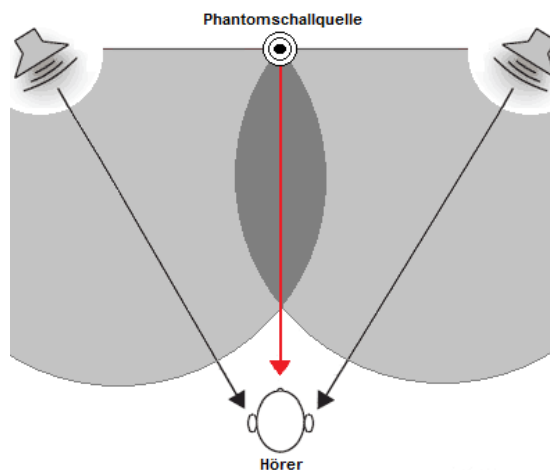


Abbildung 2.6.: Phantomschallquelle zwischen zwei Lautsprechern

Um eine Phantomschallquelle auf der Horizontalebene zwischen den Lautsprechern zu positionieren, gibt es verschiedene Möglichkeiten bei der Aufnahme. Durch Verzögerung der

Signale in der Laufzeit bewegt sich die Quelle in Richtung des zuerst auftretenden Signals. Man spricht in diesem Fall von Laufzeitstereofonie. Eine andere Möglichkeit ist die Intensitätsstereofonie, bei der eine Änderung des Pegelverhältnisses beider Signale eine Verschiebung auf der Achse bewirkt. Die Anwendung beider Varianten nennt sich Äquivalenzstereofonie [Gör08].

Die Verschiebung einer Phantomschallquelle während der Wiedergabe wird in der Praxis häufig mit einem Panoramaregler am Mischpult vorgenommen. Diese pegelabhängige Verteilung von Signalen wird als „Panning“ bezeichnet und eignet sich vor allem für die Verteilung von Monosignalen. Dieses Verfahren wird auch bei der Umsetzung des SPAOP-MFC eingesetzt, um die virtuellen Schallquellen der Wellenfeldsynthese-Anlage in korrekte Phantomschallquellen zu überführen. Welche Möglichkeiten es dafür gibt und welche Regeln dabei zu beachten sind, wird im Abschnitt „Panning Laws“ innerhalb dieses Kapitels diskutiert.

### **Sweet Spot**

Die Gültigkeit einer Phantomschallquelle ist bei Lautsprecheranordnungen von Mehrkanaltonsystemen auf einen sehr geringen Bereich begrenzt. In einer Anordnung mit zwei Lautsprechern ist der gültige Bereich noch die gesamte Mittelachse zwischen diesen. Bewegt sich der Zuhörer auf ihr vor und zurück, ändert sich das Empfinden nicht und die Lokalisation kann korrekt wahrgenommen werden (Abbildung 2.7). Kommen mehr Lautsprecher hinzu, hat jedes vorhandene Lautsprecherpaar eine eigene Mittelachse, auf der die von ihr dargestellten Phantomschallquellen gültig sind. Bewegt der Hörer sich allerdings auf einer der Mittelachsen vor oder zurück, dann verlässt er die anderen Mittelachsen und damit auch deren gültigen Bereich. Die gültige Hörposition in einer Anordnung mit mehr als zwei Lautsprechern begrenzt sich demnach auf einen sehr geringen Bereich, dem sogenannten „Sweet Spot“, der an dem Schnittpunkt aller Mittelachsen entsteht. Das ist ein großer Nachteil gegenüber der Wellenfeldsynthese-Anlage, wo im Idealfall der gesamte Bereich innerhalb der Lautsprecher eine „Sweet Area“ bildet.

### **Zweikanal Stereofonie**

Der Begriff „Stereo“ ist die heutzutage gebräuchliche Bezeichnung für die zweikanalige Übertragung und seit Mitte der 1950er Jahre das Standardverfahren bei der Produktion von Musik [Gör08]. Durch die Einführung eines zweiten Kanals, den es bei der Monofonie zuvor nicht gab, wurde es möglich, Richtungsinformationen in die Kanäle zu codieren. Die verschiedenen Möglichkeiten hierzu werden in einem späteren Kapitel beschrieben. Obwohl sich der Begriff „Stereo“ nicht auf die zweikanalige Übertragung allein begrenzt, wird der Begriff oft und



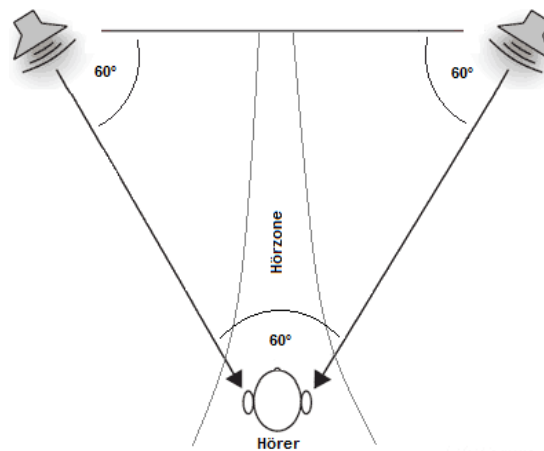


Abbildung 2.7.: Stereohörfläche bei Aufstellung im Stereodreieck

auch innerhalb dieser Arbeit alleinstehend nur in diesem Kontext verwendet. Des Weiteren beschäftigt sich diese Arbeit auch nur mit der raumbezogenen Stereophonie.

Die Standardaufstellung der Lautsprecher für eine Stereowiedergabe ist das sogenannte Stereodreieck. Abbildung 2.7 zeigt, dass Lautsprecher und Hörer ein gleichschenkliges Dreieck bilden, dessen Winkel alle  $60^\circ$  betragen. Es wird darüber hinaus empfohlen, die Lautsprecher mit einer Basisbreite größer als 2,50 m [Gri96] und einer Höhe von 1,20 m (Ohrhöhe) aufzustellen. Diese Empfehlung dient dazu, ein optimales Klangerlebnis beim Hörer hervorzurufen.

Obwohl Stereo heute für Kino- und Videoproduktionen eine eher untergeordnete Rolle spielt, unterstützen alle Surroundverfahren weiterhin dieses Format, da gerade auf Fernsehgeräten und Kompaktanlagen die Wiedergabe immer noch auf zwei Kanälen stattfindet. In der Praxis wird das mittels Downmixen umgesetzt. Die nicht verwendeten Kanäle werden mit einer Absenkung um 3dB dem linken beziehungsweise rechten Kanal zugemischt [Gör08].

### 3-,5-,7-Kanal Stereophonie

Ein großer Nachteil von Stereo beim Einsatz im Filmbereich ist seine sehr schmale Hörzone. Bei der Wiedergabe in großen Räumen, wie etwa einem Kinosaal, wird das sehr deutlich. Zuschauer, die außerhalb der Hörzone sitzen, lokalisieren mittig stattfindende Hörereignisse eher von der Seite des näher stehenden Lautsprechers [Gri96]. Erklärt werden kann dies durch den Präzedenzeffekt [Gör08]. Eine Schallquelle wird aus der Richtung geortet, aus der der Schall zuerst auf den Kopf trifft.



## 2. Grundlagen

„Links Surround“ und „Rechts Surround“ ist es möglich, Phantomschallquellen im seitlichen Bereich des Hörers darzustellen. Dazu werden Lautsprecher, wie in Abbildung 2.9 (Abbildung 1) dargestellt, auf  $110^\circ$  und  $250^\circ$  aufgestellt. 5.1 Surround ist gerade im Heimbereich beliebt, weil die Aufstellung von fünf Lautsprechern in fast jedem Wohnzimmer gut realisierbar ist.

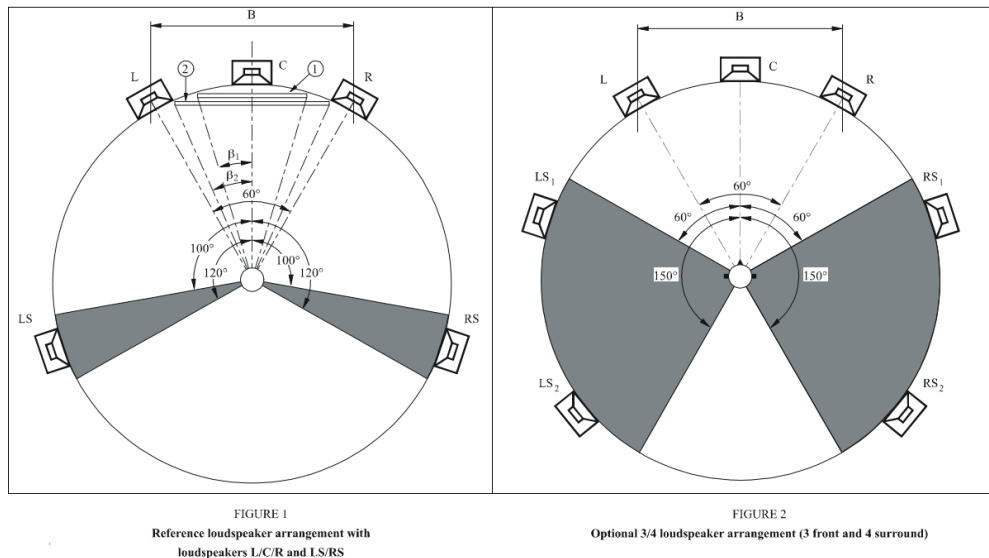


Abbildung 2.9.: Anordnung für 5-Kanal und 7-Kanal Surround nach ITU Recommendation BS.775-3 [ITU12]

Die Nachteile, über die bei Stereo in der Darstellung im Frontbereich diskutiert wurden, hat 5.1 Surround im Seitenbereich. Wie in der Abbildung 2.9 (Figure 1) gut zu erkennen ist, stehen die Seitenlautsprecher  $L$ ,  $R$  vergleichsweise weit von den Surround-Lautsprechern  $LS$ ,  $RS$  entfernt. Der schon zuvor erwähnte Präzedenzeffekt sorgt jetzt bei der Verwendung in großen Räumen dafür, dass die Lokalisation im Seitenbereich eher dem Lautsprecher zugeordnet wird, der dichter zur eigenen Sitzposition aufgestellt ist.

Um diesen Effekt zu minimieren, werden zwei weitere Surround-Kanäle im Seitenbereich verwendet, wodurch die Wahrnehmung stabilisiert und der Sweet Spot vergrößert wird. Dieses Format ist vorrangig für den Einsatz in Kinosälen vorgesehen. Es gibt zwar auch 7.1 Surround Anlagen für den Heimbereich, jedoch ist die korrekte Aufstellung nicht in jedem Wohnzimmer möglich. Abbildung 2 in Abbildung 2.9 zeigt die in der ITU Recommendation BS.775-3 vorgeschlagene Aufstellung für eine Surround Anordnung mit 3 Frontlautsprechern und 4 Surround-Lautsprechern.

Eine Weiterentwicklung von Formaten auf der Basis steigender diskreter Kanäle findet weiterhin statt. Es gibt einen konkreten Vorschlag von der NHK, der japanischen Rundfunk

Gesellschaft, die ein Mehrkanalsystem mit 22 Kanälen und zwei Effektkanälen vorsieht [ITU09]. Auch Hersteller wie *Dolby Laboratories* bringen weiterhin neue Produkte aus diesem Bereich auf den Markt, wie das erst kürzlich vorgestellte *Dolby Atmos*<sup>5</sup>. Jedoch ist die Datenmenge bei so vielen diskreten Kanälen enorm hoch und es gibt mit Ambisonics und der bereits vorgestellten Wellenfeldsynthese flexible Alternativen, die nicht kanalorientiert arbeiten. Bei diesen Formaten ist die Datenmenge unabhängig von der Anzahl der verwendeten Kanäle.

### Ambisonics

Ambisonics ist ein Syntheseverfahren zur Aufnahme und Wiedergabe von Schallereignissen mit zwei Ausprägungen. Anders als die kanalorientierten Formate hat Ambisonics keine feste Anzahl an Lautsprechern. Im B-Format, oder auch *First Order Ambisonics* genannt, werden die Audiosignale in vier Kanälen kodiert. Der omnidirektionale W-Kanal enthält hierbei die Schalldruckkomponente und die Kanäle X, Y und Z enthalten die Schallschnellekomponenten und stellen die orthogonalen Richtungen dar [Pau10]. Das entspricht der Aufnahme mit einem Soundfield-Mikrofon, das die Schalldruckkomponenten mit kugelförmiger Richtercharakteristik aufnimmt und die Schallschnellekomponenten mit achtförmiger Richtercharakteristik [Han14]. *Higher Order Ambisonics*, als die zweite Ausprägung, ist ein Oberbegriff für die Ausprägungen mit steigender Ordnung, *Second Order*, *Third Order*, ... . Der Anteil der Schallschnellekomponenten wird in eine beliebige Mehrzahl an Einzelkomponenten aufgeteilt, so dass das B-Format um die entsprechende Anzahl an Kanälen erweitert wird. Dadurch wird die räumliche Auflösung mit steigender Anzahl an Einzelkomponenten genauer, wodurch die Darstellung einer Klangquelle präziser und der Darstellungsbereich stabiler wird.

Mit der Möglichkeit, aus dem B-Format Signale für eine beliebige Anzahl von Lautsprechern zu dekodieren und mittels Z-Achse die Darstellung dreidimensional zu gestalten, ist dieses Format gegenüber den diskreten Mehrkanaltonsystemen sehr attraktiv. Der direkte Vergleich mit 22.2-Surround zeigt schon beim Kanalaufwand die Vorteile dieses Formates. Eine große kommerzielle Bedeutung hat Ambisonics derzeit jedoch noch nicht.

### 2.3. Panning Laws für Mehrkanaltonsysteme

In der Diskussion um die verschiedenen Panning Laws, beschränken sich diese auf die Positionierung von Mono-Schallquellen, weil alle Rohdaten aus dem Quellformat stets im Monoformat vorliegen. Ausgehend von der Tatsache, dass die Positionierung von Schallquellen in

---

<sup>5</sup><http://www.dolby.com/us/en/technologies/dolby-atmos.html> - Abruf: 2014-12-07

der Mehrkanalstereofonie lediglich eine Erweiterung der Gesetzmäßigkeiten der Zweikanal-Stereofonie darstellt [Sau05], werden die Grundlagen auf Basis von Stereo erläutert.

Unabhängig vom angewendeten Gesetz existieren für das Panning zwischen zwei Lautsprechern drei Fälle, in denen sich die Gesetze gleich verhalten [Sau05] :

1. Die Schallquelle ist ganz links dargestellt. In diesem Fall übernimmt der linke Lautsprecher allein die Wiedergabe.
2. Die Schallquelle ist ganz rechts dargestellt. In diesem Fall übernimmt der rechte Lautsprecher allein die Wiedergabe.
3. Die Schallquelle wird in der Mitte beider Lautsprecher dargestellt. Beide Lautsprecher geben das Signal mit gleichem Pegel wieder.

Die Höhe des Pegels und wie die Verteilung dessen auf der gesamten Strecke zwischen beiden Lautsprechern verläuft sind abhängig vom verwendeten Gesetz und haben unterschiedliche Wahrnehmungen beim Hörer zur Folge.

### 2.3.1. Die „Constant Gain“ Methode

Die wohl einfachste Form des Pannings ist die „Constant Gain“ Methode. Hierbei wird die Summe der beiden Pegel von beiden Kanälen über die gesamte Strecke konstant gehalten. Damit der Pegel in seiner Summe konstant bleibt, ist das Verhalten der einzelnen Kanäle linear. Die Berechnung der Pegel erfolgt im Verhältnis von darzustellender Phantomschallquelle  $\theta_{\text{pan}}$  zu den Winkeln  $\theta_1$  und  $\theta_2$  der Lautsprecher. Wie in Abbildung 2.10 dargestellt, ist die Sichtlinie des Hörers  $0^\circ$ .

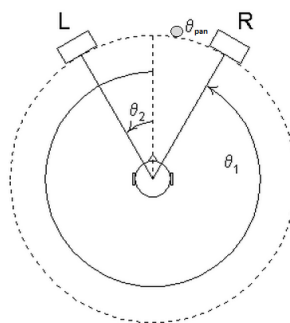


Abbildung 2.10.: Zweikanal-Anordnung [Wes98]

Für den Pegel  $p_i$  der einzelnen Gains der Kanäle folgt daraus das folgende Gesetz:

$$p_1 = \frac{\theta_2 - \theta_{pan}}{\theta_2 - \theta_1}, \text{ mit } \theta_2 \geq \theta_{pan} \geq \theta_1 \quad (2.1)$$

$$p_2 = \frac{\theta_{pan} - \theta_1}{\theta_2 - \theta_1} \quad (2.2)$$

$$gain = \sum_{i=1}^2 g_i \quad (2.3)$$

$$power = \sum_{i=1}^2 g_i^2 \quad (2.4)$$

In der Abbildung 2.11 ist der Verlauf für dieses Gesetz dargestellt. Der rechte Kanal ist mit  $\theta_1 = 45^\circ$  und der linke Kanal mit  $\theta_2 = -45^\circ$  angegeben. Es ist gut zu erkennen, dass die Pegelsumme auf der gesamten Strecke konstant bleibt, allerdings ist die Leistung und damit die subjektiv wahrgenommene Lautstärke<sup>6</sup> an keiner Stelle konstant. Eine auf der Horizontalen verlaufenden Phantomschallquelle fällt in ihrer wahrgenommenen Lautstärke ab, obwohl die Distanz zum Hörer gleich bleibt. Eine automatische Konvertierung ist mit diesem Gesetz demnach nicht möglich, weil die Darstellung ungewollt verändert werden würde. Die Empfindung der Lautstärke beim Hörer und damit die Leistung muss auf der Strecke zwischen den Lautsprechern konstant bleiben.

### 2.3.2. Die „Constant Power“ Methode

Die Lösung für dieses Problem ist die „Constant Power“ Methode [Wes98], bei der die Lautstärke auf der gesamten Strecke konstant ist. Grundlage für diese Methode ist das Tangens-Gesetz [Sau05, S.17].

$$\frac{\tan \theta_{pan}}{\tan \theta_0} = \frac{p_1 - p_2}{p_1 + p_2}, \text{ mit } |\theta_{pan}| \leq \theta_0 \quad (2.5)$$

Begrenzt man den Winkel  $\theta$  zwischen  $0^\circ$  und  $90^\circ$ , wie Gleichung 2.6 zeigt,

$$\theta_m = \frac{\theta_2 - \theta_{pan}}{\theta_2 - \theta_1} \cdot 90, \text{ mit } \theta_2 \geq \theta_{pan} \geq \theta_1 \quad (2.6)$$

ist das Tangens-Gesetz mit den Gleichungen 2.7 und 2.8 gleichgestellt.

$$p_1 = \sin \theta_m \quad (2.7)$$

---

<sup>6</sup><http://www.sengpielaudio.com/Rechner-pegelaenderung.htm> - Abruf: 2014-11-10

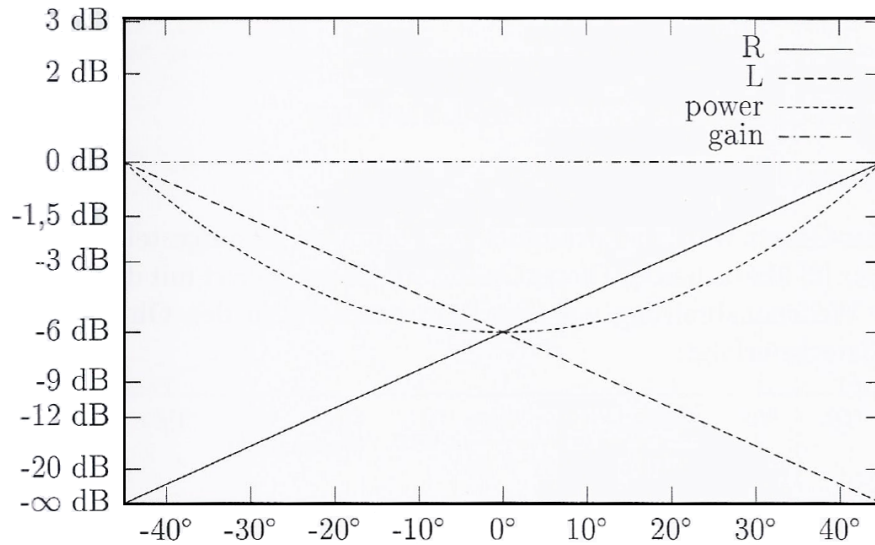


Abbildung 2.11.: Lineares Gesetz[Sau05]

$$p_2 = \cos \theta_m \quad (2.8)$$

Aus der zuvor diskutierten Formel 2.4 geht hervor, dass die Summe aller quadrierten Pegel die Gesamtleistung ergibt. Mit der trigonometrischen Funktion in 2.9 wird ersichtlich, dass diese Methode die Forderung erfüllt, die Lautstärke konstant zu halten.

$$\sin^2 + \cos^2 = 1 \quad (2.9)$$

Abbildung 2.12 zeigt den konstanten Verlauf der Leistung zwischen  $\theta_1 = 45^\circ$  und  $\theta_2 = -45^\circ$ .

Bei der Umsetzung von SPAOP-MFC wird dieses Verfahren verwendet. Entscheidend ist die Eigenschaft des Gesetzes, die Leistung auf der gesamten Strecke zwischen zwei Lautsprechern konstant zu halten.

### 2.3.3. Weitere Methoden

Zu diesen beiden genannten Gesetzen gibt es noch weitere *Panning Laws*. Ziel dieser ist es, das Gesamtergebnis noch „harmonischer“ zu gestalten, was durch Elimination der „Null-Kanäle“ umgesetzt wird. Damit sind die übrigen Kanäle in einem Mehrkanalformat gemeint, die nicht an der sonst paarweisen Wiedergabe bei einer Klangquellenberechnung beteiligt sind. Dadurch wird der Hörer in ein besseres Klangerlebnis gehüllt, da das Geräusch „um ihn

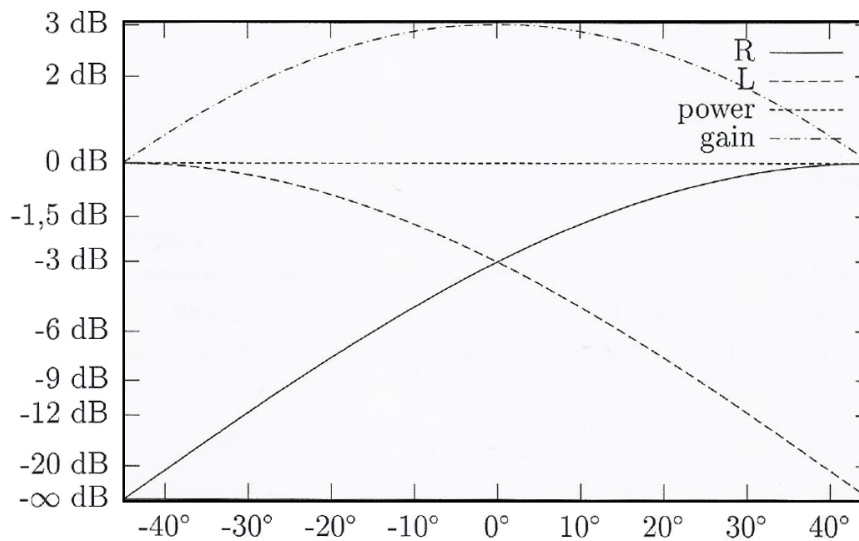


Abbildung 2.12.: Tangens-Gesetz[Sau05]

herum“ stattfindet. Das Ziel dieser Arbeit ist aber, Klangquellen in verschiedenen Formaten so zu positionieren, dass dabei ein vergleichbares Ergebnis zur WFS-Anlage erzielt wird. Das Hinzufügen von Kanalinformation würde an dieser Stelle dem widersprechen. Aus diesem Grund werden diese *Panning Laws* nicht näher diskutiert.

## 2.4. Tools, Bibliotheken und Werkzeuge

In diesem Abschnitt werden die bei der Umsetzung zum Einsatz kommenden Werkzeuge und Tools kurz erläutert. Die verwendete Programm-Bibliothek ist dabei die schon zuvor bei SPAOP eingesetzte JUCE-Bibliothek [Han14].

### 2.4.1. Entwicklungsumgebung

Die Entwicklung des SPAOP-MFC wurde für die Betriebssysteme Microsoft Windows und Apple OSX vorgenommen, wodurch auch zwei Entwicklungsumgebungen zum Einsatz kamen. Für die Entwicklung auf Windows-Ebene wurde dabei Microsofts hausinterne IDE Visual Studio Professional 2013<sup>7</sup> verwendet. Die Entwicklung für Apple OSX wurde mit Apples Xcode<sup>8</sup> in der Version 6.0.1 vorgenommen.

---

<sup>7</sup><http://www.visualstudio.com/>

<sup>8</sup><https://developer.apple.com/xcode/>



### 2.4.2. JUCE-Bibliothek

Die bereits in SPAOP eingesetzte Programm-Bibliothek JUCE<sup>9</sup> enthält eine Sammlung von C++ Klassen, die als Unterstützung bei der Entwicklung von Programmen dienen soll. Was diese Bibliothek so interessant für den Einsatz bei der Entwicklung eines Audio-Plugins macht, ist die Tatsache, dass es die verschiedenen Audio-Unit-Schnittstellen abstrahiert. Für die Entwicklung eines Plugins muss lediglich gegen die JUCE-Schnittstelle implementiert werden und bei der Kompilierung kann die gewünschte Version (RTAS-, AAX-, AU-, VST2 und VST3) erzeugt werden [Han14].

Ein weiteres Feature dieser Bibliothek ist das Programm *IntroJucer*, mit dem die Verwaltung und Konfiguration eines Projektes vereinfacht wird. Entkoppelt von Entwicklungsumgebung und Zielplattform können damit alle gewünschten Einstellungen vorgenommen werden. Außerdem enthält es einen WYSIWYG-Editor<sup>10</sup> für die Erzeugung von GUI-Klassen. Dieser generiert automatisch den Quellcode, der einfach um die gewünschte Funktionalität erweitert werden muss.

## 2.5. Digitale Audioproduktion

Dieser Abschnitt beschäftigt sich mit der digitalen Arbeitsumgebung für Audioproduktionen. Einleitend wird die Frage geklärt, was eine „Digital Audio Workstation“ ist und welches Programm verwendet wurde. Danach wird erläutert, was im Allgemeinen ein Audio-Plugin ist und auf welche Weise dieses mit „Digital Audio Workstations“ interagiert.

### 2.5.1. Digital Audio Workstations

Ein enormer Vorteil, den ein Computer in der Arbeitsumgebung von Audioproduktionen bietet, ist, dass er nahezu sämtliche Studiogeräte ersetzen kann. Die Möglichkeiten sind nicht mehr in der Funktionalität des Gerätes begrenzt sondern auf die Leistungsfähigkeit des Computers [Gör08]. Programme, die diesen Funktionsumfang bieten, werden „Digital Audio Workstation“ genannt. Im weiteren Verlauf wird für den Begriff nur noch die Abkürzung DAW verwendet.

Das Sortiment an nachzubildenden Studiogeräten beginnt bei analogen Geräten wie Mehrspur-Bandmaschinen oder Mischpulten bis hin zu digitalen Effektgeräten wie Synthesizer und Sequencer oder sogar virtuellen Instrumenten. Der Funktionsumfang ist eigentlich unbegrenzt, weil das, was eine DAW nicht bietet, durch eigens entworfene Plugins nachträglich geladen werden kann. Die verschiedenen Audiosignale, ob selbst generiert oder aufgenommen, können

---

<sup>9</sup><http://www.juce.com/about-juce>

<sup>10</sup>WYSIWYG - What you see is what you get

## 2. Grundlagen

---

in einer DAW zusammengemischt und in Echtzeit bearbeiten werden. Am Ende wird das fertige Produkt dann in das gewünschte Audioformat exportiert. Ein Beispiel für eine DAW-Benutzeroberfläche ist die Oberfläche von Cubase 7.5, dargestellt in Abbildung 2.13.

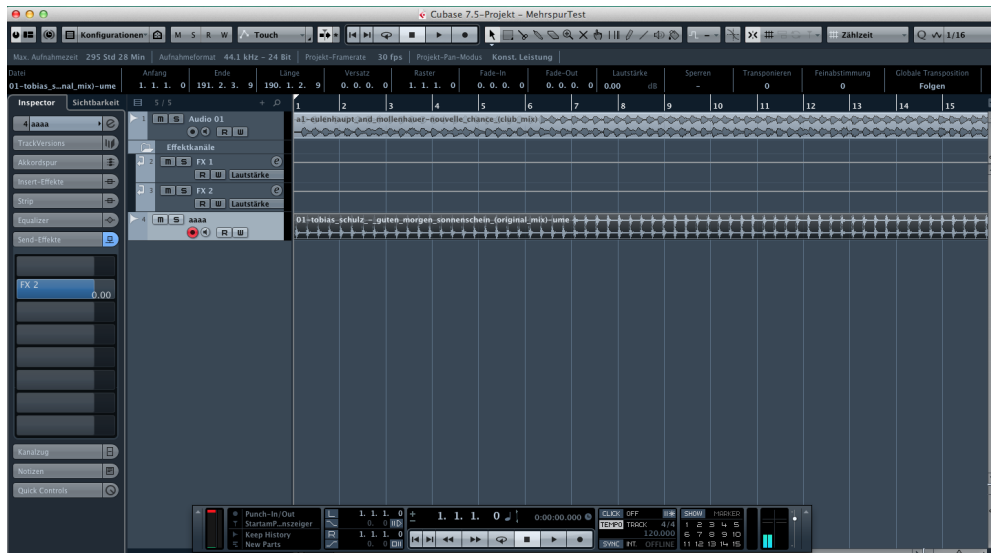


Abbildung 2.13.: Screenshot der DAW Software Cubase 7.5

Auf dem kommerziellen Markt der DAW's gibt es eine Vielzahl an Produkten, diese alle aufzulisten, erscheint wenig sinnvoll.. Einen kurzen Überblick verschafft Martin Hansen in seiner Arbeit [Han14]. Hervorzuheben ist an dieser Stelle Cubase<sup>11</sup> von der Firma Steinberg. Cubase 7 wurde bei der Umsetzung von SPAOP-MFC als DAW verwendet und soll deswegen gesondert genannt werden.

### 2.5.2. VST-Plugins

Wie bereits erwähnt, kann der Funktionsumfang einer DAW durch die Entwicklung von Audio-Plugins erweitert werden. Diese können dann zur Laufzeit in die DAW geladen und auch wieder entfernt werden. Die DAW-Software fungiert für die Plugins dabei wie ein „Host“, aus dessen Sicht das Plugin eine Blackbox mit einer beliebigen Anzahl an Eingängen, Ausgängen und zugehörigen Parametern darstellt. Der Host muss den internen Audioverarbeitungsprozess des Plugins nicht kennen. Für gewöhnlich wird ein Plugin konkret für jeweils eine Audiospur geladen und verarbeitet nur die Audiodaten dieser Spur [Han14][Sau05].

---

<sup>11</sup><http://www.steinberg.net/de/products/cubase/>

## 2. Grundlagen

---

Um die Entwicklung von Plugins für die eigene Software zu ermöglichen, haben diese oft von den Herstellern definierte Software-Schnittstellen und es werden passende Software Development Kits, kurz SDK's, dafür zur Verfügung gestellt. Für die in dieser Arbeit konkret verwendete Schnittstelle VST<sup>12</sup> stellt die Firma Steinberg Entwicklern das passende *VST Audio Plug-Ins SDK (aktuell in Version 3.6.0)*<sup>13</sup> kostenlos zur Verfügung.

Die zentrale Methode dieser Schnittstelle ist die *process*-Methode. Diese wird vom Host zyklisch wiederholt aufgerufen und dabei dem Plugin der aktuelle Block von Audio-Samples für die Bearbeitung übergeben. Eine Anforderung an Plugins ist dabei, dass dies in Echtzeit zu erfolgen hat. Die Anzahl an Samples, die vom Host an das Plugin übergeben werden, kann dabei von Aufruf zu Aufruf variieren, was ganz vom Scheduling der DAW abhängen kann. Die fertig bearbeiteten Daten werden abschließend vom Plugin in den dafür vom Host übergebenen Speicherbereich geschrieben.

---

<sup>12</sup>Virtual Studio Technology

<sup>13</sup><http://www.steinberg.net/en/company/developers.html> - Abruf: 2014-11-10

### 3. Softwarekonzept und Integration

Die derzeitige Situation, Projekte mehrmals zu erstellen, birgt einige Nachteile. Ein großer ist der enorme Arbeitsaufwand. Die Erstellung muss sehr präzise durchgeführt werden, wenn vergleichbare Ergebnisse erzielt werden sollen. Das erfordert hohe Konzentration und ist bei großen Projekten sehr anstrengend. Dadurch werden potentielle Projekte schon in der Entstehung wieder verworfen, weil der Arbeitsaufwand oft nicht im Verhältnis steht. Ein weiterer Nachteil ist, dass die Projekte dann vielleicht gleich sind, aber sie sind eben nicht identisch. Will man Studien oder Versuche durchführen, bei denen eine Anlage mit der WFS-Anlage verglichen wird, wäre eine identische Ausgangssituation aber äußerst wünschenswert. Um das zu ermöglichen, sollte im Rahmen dieser Arbeit eine Möglichkeit entwickelt werden, wodurch Klangquellen auf Basis von Positionsdaten der Wellenfeldsynthese-Anlage in verschiedene Surround Formate überführt werden.

Die Positionierung in Surround-Formaten wird mit sogenannten *Pannern* vorgenommen. Abbildung 3.1 zeigt ein Beispiel für einen Surround-Panner für ein 5.1 System. Damit wird die Verteilung der Audiodaten einer Klangquelle auf die gewünschten Kanäle vorgenommen. Die mit einem Panner erstellten Rauminformationen bilden aber keine Entfernung mit ab. Um einen solchen Effekt hervorzurufen, muss der Ton-techniker die Lautstärke *faden*. Faden bedeutet, die Lautstärke über einen Bereich kontinuierlich anzuheben oder abzusenken. Auf diese Art und Weise werden die vorliegenden Audiodaten mit den Rauminformationen in die entsprechenden Kanäle kodiert. Würde man versuchen, die Konvertierung aus diesem Format vorzunehmen, zeigt sich folgendes Problem. Durch die Nichtbetrachtung von Raumtiefe in diesem Panner kann bei der Konvertierung zur Laufzeit nicht bestimmt werden, wo genau sich die Klangquelle im Raum befindet. Theoretisch könnte man zwar versuchen, am höchsten Pegel das Verhältnis abzuleiten, um damit die Entfernung zu errechnen. Das käme aber eher einem Schätzwert gleich

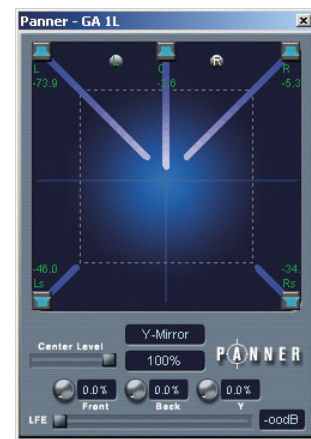


Abbildung 3.1.: 5.1 Surround Panner

als einem genauen Ergebnis. Dadurch ist eine weitere Untersuchung, ob die Konvertierung auf Basis von einem Mehrkanaltonformat stattfinden kann, unnötig. Eine korrekte Positionierung auf der WFS-Anlage wäre nicht möglich. Betrachtet man hingegen SPAOP, sieht man sehr deutlich die Unterschiede. Während sich der klassische Surround-Panner auf die Lautsprecher begrenzt, steht bei SPAOP die Betrachtung des Raumes im Vordergrund, wie Abbildung 3.2 zeigt.

Hier lassen sich die Klangquellen nicht nur rund um das Hörfeld positionieren sondern auch in der Raumtiefe. Außerdem können alle Klangquellen gemeinsam angezeigt werden, auch wenn die Steuerung nur für eine erfolgt. Die exakte Position der gesteuerten Klangquelle wird als kartesische Koordinate im OSC Format festgehalten. Das erleichtert die Berechnung für andere Formate, weil exakte Werte vorliegen. Dieses Beispiel zeigt, dass das eingesetzte Format bei der WFS-Anlage, wesentlich geeigneter als Ausgangsbasis für die Umsetzung eines „Multi-Format-Converters“ ist. Deshalb wird die Umsetzung auf der Grundlage von SPAOP erfolgen.

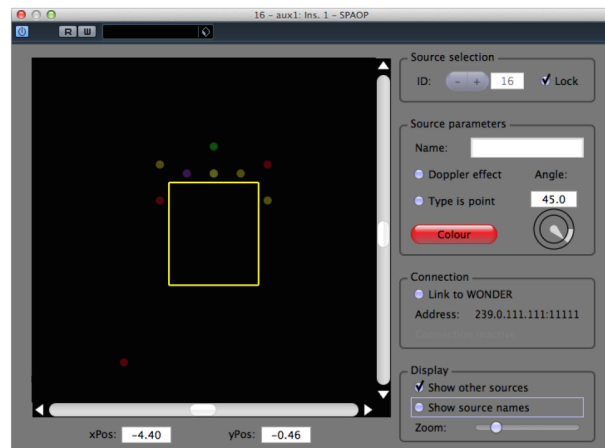


Abbildung 3.2.: Benutzeroberfläche von SPAOP

Mit dem Thema Format beschäftigen sich aktuell viele Arbeitsgruppen, die ein einheitliches Format einführen wollen, um die Inkompatibilität zwischen den bestehenden Surround-Formaten zu beseitigen. Auf der kürzlich veranstalteten *137. Audio Engineering Society Convention* wurde über neue Standards diskutiert und einige Vorschläge dazu unterbreitet [FHB<sup>+</sup>14] [HHKP14].

### 3.1. Anforderung

Das Ziel ist also eine Erweiterung in SPAOP zu integrieren, welche bei Aktivierung die Audiodaten vom Host aufnimmt, mit den Positionsdaten aus der OSC-Kommunikation in ein gewähltes Format konvertiert und in den vom Host bereitgestellten Speicher zurückschreibt.

#### 3.1.1. Allgemein

Die Formate, die in SPAOP-MFC aufgenommen werden, sind Stereo, 3/0/0, 3/0/1, 3/2/0, 3/2/1, 3/4/0, 3/4/1 und Ambisonics, wobei die Erweiterbarkeit um weitere Formate möglich sein soll. In allen Zielformaten müssen für die Konvertierung die kartesischen Koordinaten in Polarkoordinaten umgerechnet werden, weil die Darstellung in allen anderen Formaten auf der Grundlage eines Azimut-Winkels beruht. Neben der Berechnung der Richtung muss zudem ein Mechanismus entwickelt werden, der die Darstellung der Entfernung automatisiert. In der aktuellen Fassung des VST Plug Ins SDK, dem VST3, werden zwar sogenannte Ramping-Funktionen unterstützt, aber noch nicht durch die JUCE-Bibliothek. Daher muss ein Weg gefunden werden, wie die Wertänderungen von Block zu Block interpoliert werden können.

Neben den logischen Klassen muss auch die grafische Oberfläche des SPAOP-Plugins angepasst werden. Hierfür wird der vom *Introjuicer* zur Verfügung gestellte GUI-Builder verwendet. Die Anforderung besteht darin, auf einem Tab kompakt alle notwendigen Einstellungen verfügbar zu machen. Darüber hinaus sollen bei aktiviertem Konverter, passend zum gewählten Format, im Darstellungsbereich der Sweet Spot und die Lautsprecher in ihrer Aufstellung erscheinen. Die Ausdehnung der Anlage und die Position des Sweet Spots sollen zudem durch den Bediener anpassbar sein.

#### 3.1.2. Einstellungen im Konverter

Die in das Plugin einfließenden Einstellungen, die durch einen Benutzer vorgenommen werden können, werden nochmals in zwei Bereiche unterteilt: solche, die sich direkt auf die Klangquelle beziehen und dementsprechend von Instanz zu Instanz unterschiedlich sind und solche, die übergreifend Gültigkeit haben.

##### Globale Einstellungen

Um Einstellungen, die für alle geladenen SPAOP-MFC Instanzen gültig sind, untereinander auszutauschen, muss eine zentrale Komponente entworfen werden, mit der globale Einstellungen zentral gespeichert und wieder abgerufen werden können. Zu den globalen Einstellungen gehören das Zielformat, die Breite und Tiefe der WFS-Anlage, X-Offset und Y-Offset des Sweet Spots, der Radius sowie die Koordinaten der Lautsprecher des gewählten Formates. Die Felder Radius, Breite und Tiefe sowie der X- und Y-Offset des Sweet Spots haben voreingestellte Werte. Breite und Tiefe sind hierbei auf die Maße der Anlage im Labor der HAW Hamburg eingestellt und die Offset Felder werden mit null initialisiert.

Der Radius und die Koordinaten der Lautsprecher lassen sich nicht einstellen, sondern werden automatisch berechnet. Der Radius wird dabei aus den Werten Höhe und Tiefe gebildet (3.1). Wird einer der beiden Werte in der Benutzeroberfläche geändert, wird automatisch der Radius neu berechnet. Die Koordinaten der Lautsprecher, die zur Darstellung in der Benutzeroberfläche dienen, werden durch die Wahl des Zielformates initialisiert und sollen in einer dynamischen Liste gespeichert werden. Ändern sich Format oder Radius, ändern sich auch die Koordinaten und die Liste wird neu erstellt. Die Berechnung der Koordinaten wird wie in Formel 3.2 und 3.3 umgesetzt, wobei die Winkel der Lautsprecher die Angaben der Idealanordnung aus der ITU Recommendation sind [ITU12].

$$Radius = \frac{Breite + Tiefe}{4} \quad (3.1)$$

$$X_i = \cos(\varphi_i + 90) \cdot Radius + Offset_X \quad (3.2)$$

$$Y_i = \sin(\varphi_i + 90) \cdot Radius + Offset_Y \quad (3.3)$$

Wider Erwarten wird der Sweet Spot nicht extra als Datum gespeichert. Die Y-Koordinate des Sweet Spots wird als die Hälfte der eingestellten Tiefe der WFS-Anlagen bestimmt und die X-Koordinate mit null. Der Bediener kann bei Bedarf diese Position mit Hilfe der Offsets anpassen. In der Tabelle 3.1 sind alle globale Einstellungen mit ihren Eigenschaften zusammengefasst.

#### **Klangquellenbezogene Einstellungen**

Im Wesentlichen gibt es nur zwei Einstellungen, die sich direkt auf einzelne Klangquellen beziehen und durch den Benutzer vorgenommen werden können. Zum einen besteht die Möglichkeit, die Klangquelle für die Konvertierung komplett zu ignorieren. Wird dies durch den Benutzer angewählt, findet keine Konvertierung der Audiodaten dieser Klangquelle statt und der vom Host übergebene Speicher wird auf null gesetzt. Dies ist wichtig, um keinen

| Element                           | Initialisierungswert | Wertänderung   |
|-----------------------------------|----------------------|--|
| Radius                            | 2.75                 | automatisch durch Änderung von Breite oder Tiefe (3.1) |
| Breite                            | 5                    | Benutzereingabe in GUI                                 |
| Tiefe                             | 6                    | Benutzereingabe in GUI                                 |
| X - Offset                        | 0                    | Benutzereingabe in GUI                                 |
| Y - Offset                        | 0                    | Benutzereingabe in GUI                                 |
| Liste der Lautsprecherkoordinaten | NULL                 | durch Wahl des Zielformates                            |

Tabelle 3.1.: Globale Einstellungen des SPAOP-MFC

unbekannten Zustand im Speicher zu hinterlassen, da nicht garantiert werden kann, dass dieser leer ist. Das kann ungewollte Nebengeräusche zur Folge haben.

Zum anderen kann optional die Klangquelle nur auf Frontallautsprechern anstatt räumlich abgebildet werden. Der Impuls war die Tatsache, dass auf der WFS-Anlage dargestellte Klangquellen zwar überall präzise lokalisierbar sind, bei einer 3/2/x -Aufstellung aber zum Beispiel nicht. Im Bereich hinter dem Hörer können Klangquellen nur bedingt geortet werden. Betrachtet man nochmals die Abbildung 2.9, sieht man, dass die Surroundlautsprecher fast auf Höhe der Frontalebene des Hörers stehen. Präzise Darstellungen zwischen den Lautsprechern *LS* und *RS* sind so nicht möglich. Das hatte zur Folge, die Darstellung wie in Abbildung 3.3 auf die Frontlautsprecher umzulegen. Für den seitlichen Bereich stellen die Lautsprecher *L* und *R* die äußeren Grenzen dar. Die grau dargestellten Bereiche werden demnach komplett auf  $30^\circ$  oder  $330^\circ$  abgebildet und der untere Bereich auf den oberen gespiegelt. Zwar wird mit diesem Schritt die Darstellung der jeweiligen Klangquelle und dadurch auch die Gesamtdarstellung verändert, es gibt aber auch Anwendungsfälle, in denen nicht der direkte Vergleich sondern ein gutes Hörergebnis gefordert ist.

### 3.2. Analyse der Ausgangssituation

Vor der Erweiterung von SPAOP muss analysiert werden, an welcher Schnittstelle die Positionsdaten abgegriffen werden können und wie eine Erweiterung der GUI erfolgen kann. Darüber hinaus muss innerhalb der Architektur geprüft werden, wie der Konverter dort integriert werden kann.



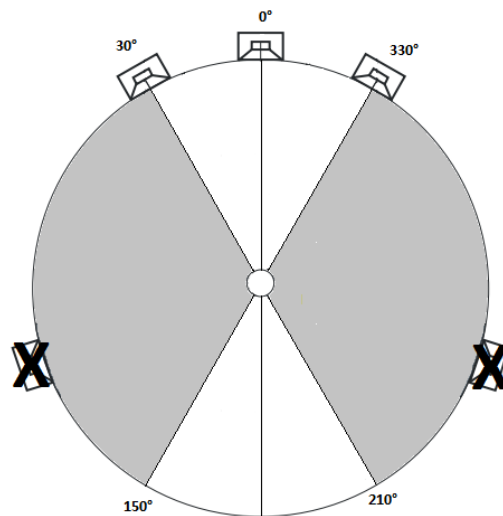


Abbildung 3.3.: Mapping der Klangquellen auf die Frontlautsprecher

#### 3.2.1. Softwarearchitektur von SPAOP

Im Wesentlichen ist die Architektur von SPAOP in die folgenden Komponenten unterteilt [Han14]:

1. *PluginEditor*: Der PluginEditor ist die grafische Oberfläche von SPAOP. Hier werden die Klangquellen dargestellt und durch den Benutzer gesteuert. Zusätzlich werden Möglichkeiten zur Benutzereingabe zur Verfügung gestellt, wie etwa die Wahl von ID und Kommunikationsmodi. Diese werden innerhalb der GUI durch ein sogenanntes *Tabbed-Component* in die Benutzeroberfläche eingebunden.
2. *PluginProcessor*: Der PluginProcessor ist die Schnittstelle zwischen SPAOP und Host. Hier sind die Methoden implementiert, die von den jeweiligen Schnittstellen eingefordert werden.
3. *SourceController*: Diese Komponente ist die Kernkomponente des Plugins. In ihm laufen alle SPAOP-spezifischen Eingaben auf, wie Automationsdaten und andere Parameter. Außerdem beinhaltet er die Logik, wie Signale und OSC-Nachrichten zu verarbeiten sind und verwaltet die Daten der virtuellen Quellen.

4. *OSC-Messaging*: Diese Komponente ist für das Versenden und Empfangen von OSC-Nachrichten verantwortlich. Da dies ausschließlich für die WFS-Anlage für diese Arbeit irrelevant, wird an dieser Stelle nicht weiter auf diese Komponente eingegangen.

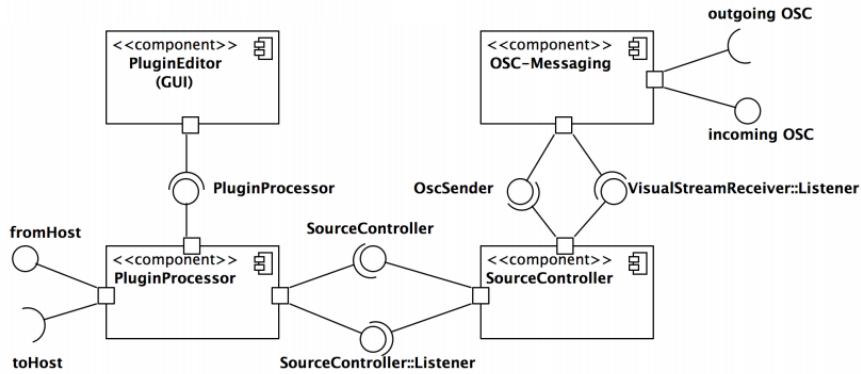


Abbildung 3.4.: Architektur von SPAOP [Han14]

Der Zusammenhang dieser Komponenten ist in Abbildung 3.4 dargestellt. Für die eigene Architektur lässt sich daraus folgendes ableiten:

1. Die grafische Erweiterung muss im PluginEditor durch Hinzufügen eines weiteren Tabs erfolgen. Auf diesem werden alle notwendigen MFC-Einstellungen eingebunden. Das Einbinden erfolgt automatisch durch den JUCE-eigenen GUI-Builder und muss nur abschließend angepasst und überprüft werden.
2. Die Positionsdaten werden durch die vom SourceController zur Verfügung gestellten Methoden *getXPos* und *getYPos* an der jeweiligen kontrollierten *Source*-Instanz zur Verfügung gestellt. Des Weiteren können auch alle weiteren Parameter hier abgefragt werden.
3. Im PluginProcessor ist die bereits implementierte *processBlock*-Methode, die zur Kommunikation mit dem Host notwendig ist und für die Umsetzung des Konverters angepasst werden muss. Des Weiteren hält der PluginProcessor einen *Smart-Pointer* auf die SourceController-Instanz und damit eine Referenz auf die kontrollierte Source. Über diese können die Positionsdaten abgegriffen werden. Der PluginProcessor muss also so angepasst werden, dass er über eine Instanz des Konverters verfügt. So kann in der *processBlock*-Methode die Abfrage der Positionsdaten über den SourceController erfolgen und die Konvertierung mit dem MFC angestoßen werden.

Die neue Plugin-Architektur, wie sie entworfen werden soll, ist in Abbildung 3.5 abgebildet.

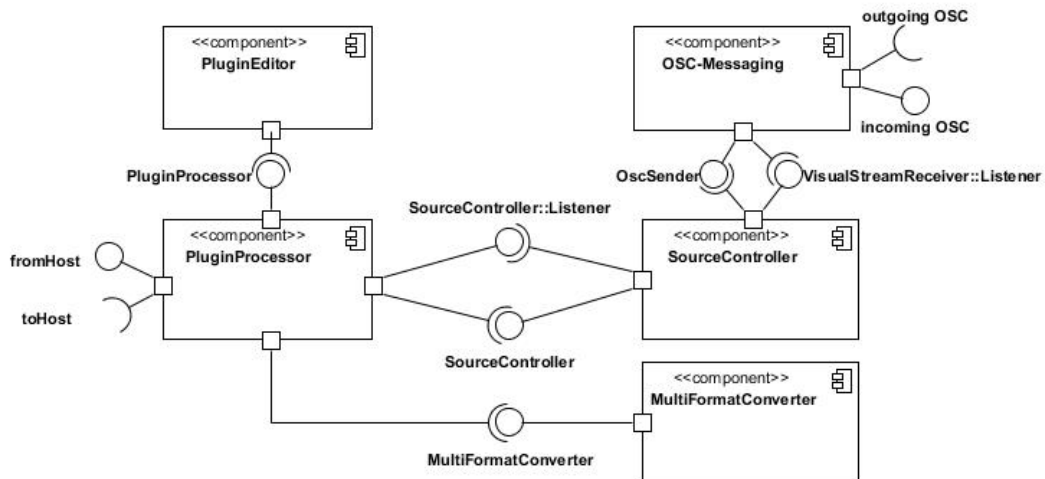


Abbildung 3.5.: Architektur von SPAOP-MFC

### 3.2.2. Audiodaten in SPAOP

Grundsätzlich muss unterschieden werden, wie die WFS-Anlage und die stereofonen Mehrkanalton-Formate die Kanäle betrachten. Bei der WFS-Anlage ist dies quellenbasiert, das heißt, in einem Kanal stehen die Toninformationen für genau eine Quelle. Bei den stereofonen Mehrkanalton-Formaten sind die Kanäle lautsprecherbasiert. In einem Kanal stehen sämtlichen Toninformationen für genau einen Lautsprecher. Ambisonics bleibt hier unbetrachtet, die Semantik der Kanäle ist im Grundlagenkapitel beschrieben.

Die quellenbasierten Audiodaten, die der Host an das Plugin übergibt, sind in einem Audiopuffer-Objekt gespeichert und werden der `processBlock`-Methode übergeben. SPAOP lässt diese Audiodaten unbehandelt, weil sie für die Verarbeitung auf der WFS-Anlage keine Rolle spielen. WONDER greift seine Audiodaten direkt über das Dante Audiointerface ab und bekommt von SPAOP die dazugehörigen Parameter, um für sich daraus seine räumliche Darstellung zu berechnen. In der derzeitigen Situation wird der übergebene Speicher also immer nur geleert.

An dieser Stelle müssen die Änderungen erfolgen, um die Daten für die eigene Konvertierung aufzunehmen. Dazu müssen die quellenbasierten Audiodaten im Audiopuffer dem korrekten Kanal entnommen werden und in die übrigen Kanäle kopiert werden. So liegen erst einmal in

allen Kanälen dieselben Rohdaten unbehindert vor. Danach müssen die formatspezifischen Berechnungen erfolgen, anhand dessen die Gewichtung des jeweiligen Kanals erfolgen kann. Zuletzt müssen diese Ergebnisse dann auf die jeweiligen Kanäle angewandt werden. Dadurch findet ein Wechsel von der quellenbasierten zur lautsprecherbasierten Semantik statt. Dieser Ablauf ist auch für Ambisonics korrekt, die Semantik ist aber nicht lautsprecherbasiert.

Bei dem bereits angesprochenen Audiopuffer handelt es sich um ein Objekt vom Typ *AudioSampleBuffer*<sup>1</sup> aus der JUCE-Bibliothek, das sowohl als Eingangs- als auch Ausgangspuffer dient. Hierfür stellt die *AudioSampleBuffer*-Klasse alle nützlichen Methoden bereit, um Samples innerhalb der Kanäle hin und her zu kopieren, einen bestimmten Gain auf einem Kanal anzuwenden oder auch die Samples zwischen einem Ausgangswert und einem Zielwert zu interpolieren. Der Puffer enthält dabei so viele Kanäle, wie das Wiedergabeformat hat, dass im Host eingestellt ist.

#### 3.2.3. Definition der Arbeitsschritte

Mit den aus der Analyse hervorgehenden Erkenntnissen können die nun folgenden Arbeitsschritte definiert werden, die für die Umsetzung des Konverters erforderlich sind:

1. Entwicklung eines Tabs für die grafische Erweiterung mit dem *Introjuicer*. Dieser muss die folgenden Einstellungen ermöglichen:
  - a) Aktivierung / Deaktivierung des Konverters durch einen Toggle-Button. Diese Einstellung ist global und hat Auswirkung auf alle Instanzen.
  - b) Wahl des Zielformates in einem Drop-Down Menü. (global)
  - c) Toggle-Button zur Aktivierung / Deaktivierung der Funktion, die Klangquelle nur auf den vorderen Lautsprechern abzubilden. Im Folgenden nur noch „no surround effect“ genannt. (lokal)
  - d) Aktivierung / Deaktivierung der Konvertierung für diese Klangquelle. (lokal)
  - e) Eingabe der Breite der WFS-Anlage mit einem Slider. (global)
  - f) Eingabe der Tiefe der WFS-Anlage mit einem Slider. (global)
  - g) Eingabe vom Sweet Spot Offset der X-Achse mit einem Slider. (global)
  - h) Eingabe vom Sweet Spot Offset der Y-Achse mit einem Slider. (global)

Die Abbildung 3.14 zeigt ein Mock-Up für den Konverter-Tab.

---

<sup>1</sup><http://www.juce.com/api/classAudioSampleBuffer.html> - Abruf: 2014-11-19

2. Berechnung der Rauminformation für die Konvertierung in die Zielformate. Dafür müssen die kartesischen Koordinaten unter der Berücksichtigung eines Sweet Spot aufgearbeitet werden, um anschließend in Polarkoordinaten umgerechnet zu werden.
3. Entwicklung der Algorithmen, die auf Basis des *Constant Power Laws* die Konvertierung in die jeweiligen Formate ermöglichen.
4. Implementierung des *MultiFormatConverters* mit den Konvertierungs-Methoden, die die Algorithmen umsetzen. Hierbei muss berücksichtigt werden, dass zu einem Zeitpunkt  $t$  die Daten von  $t-1$  noch im MFC vorliegen müssen, damit eine Interpolation der Werte erfolgen kann. Im Objekt müssen also die aktuellen *Gains* sowie die alten *Gains* gespeichert werden.
5. Implementierung einer globalen Klasse für die instanzübergreifende Kommunikation der Plugins und die Bereitstellung der Lautsprecherkoordinaten für die Darstellung auf der grafischen Benutzeroberfläche.
6. Integration der implementierten Klassen und Anpassen der bestehenden Klassen.

### 3.3. Implementierung von SPAOP-MFC

Der umgesetzte Entwurf für die interne Logik des Konverters basiert auf einem Funktionszeiger und statischen Methoden für die Konvertierung. Dieser Ansatz wurde gewählt, damit keine unnötige Logik bei jeder Iteration mit durchlaufen werden muss. Wechselt ein Benutzer das Zielformat, wird der Zeiger auf die gewünschte Übersetzungsmethode gesetzt, wodurch diese in jedem Durchlauf aufgerufen wird. Das Kommunikationsdiagramm in Abbildung 3.6 stellt den Ablauf vereinfacht dar. Der *PluginProcessor* ruft die Methode *processingToSelectedFormat* auf, die sozusagen das Kernstück des Konverters ist. Sie bekommt als Übergabeparameter die Positionsdaten und den Audiopuffer und kann über den Funktionszeiger die Berechnungen für das gewählte Format vornehmen.

Damit die formatbezogene Berechnung losgelöst von allen anderen Berechnungen stattfinden kann, wurden diese wie folgt unterteilt:

1. Zuerst findet die Berechnung der Rauminformationen statt, da dieses unabhängig vom Zielformat immer gleich abläuft.
2. Danach findet die Berechnung der *neuen Gains* auf Basis des gewählten Formates ab.
3. Zuletzt werden die *Gains* auf die Samples angewendet.

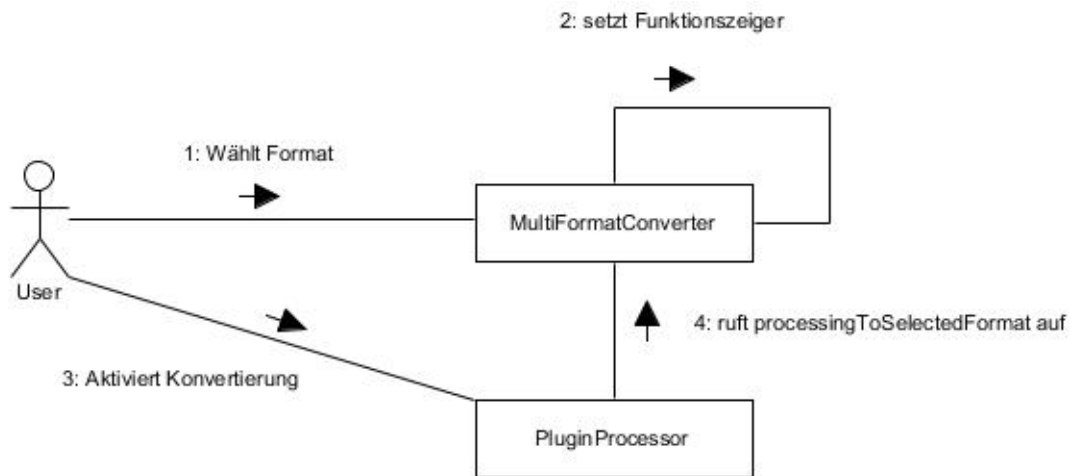


Abbildung 3.6.: Dargestellt ist die Beziehung zwischen Funktionszeiger und dessen Aufruf über die Methode `processingToSelectedFormat`.

Damit wie in den Anforderungen festgehalten die Werte interpoliert werden können, wurden zwei gleich große Arrays für die *Gain*-Werte eingesetzt. Nachdem die Konvertierungsmethoden die neuen Zielwerte berechnet haben, findet in der Hauptmethode `processingToSelectedFormat` die Interpolation statt. Im Anschluß daran werden die alten *Gains* mit den neuen überschrieben und stehen für den nächsten Durchlauf bereit. Abbildung 3.7 stellt vereinfacht den Ablauf der zyklischen Konvertierung dar.

#### 3.3.1. Die Algorithmen

Nachfolgend werden die Algorithmen beschrieben, die bei der Implementierung eingesetzt wurden.

##### Berechnen der Rauminformation

Wie bereits erwähnt, findet die Berechnung bei den verwendeten Tonformaten auf der Basis eines Azimuts statt, der auf der Horizontalebene des Hörer verläuft. Die Positionsdaten aus dem *SourceController* müssen dementsprechend in Polarkoordinaten umgerechnet werden. Diese Positionsdaten beziehen sich aber auf den Nullpunkt des Koordinatensystems der WFS-Anlage, der sich in der Mitte der Lautsprecher im Frontbereich befindet. Hierbei muss angemerkt werden, dass die Y-Achse bei der WFS-Anlage spiegelverkehrt zum normalen Koordinatensystem verläuft. Um also die Berechnung mit Werten vorzunehmen, die sich auf den Sweet Spot

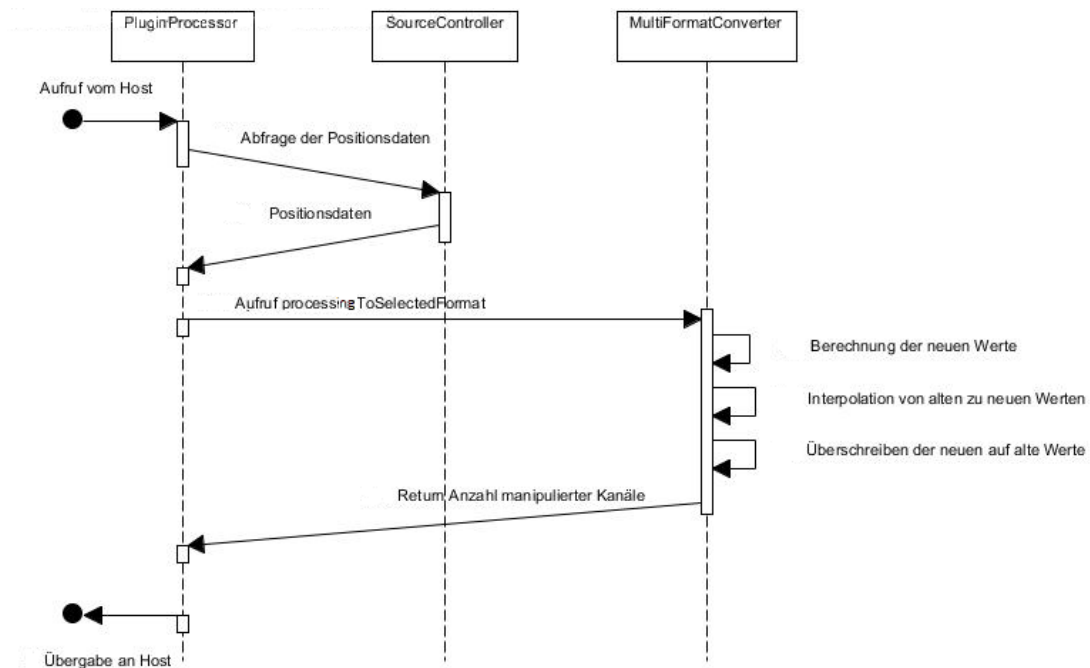


Abbildung 3.7.: Ablauf der Übersetzung eines Audioblocks.

beziehen, müssen die Koordinaten vorerst verrechnet werden, weil sonst der Winkel fehlerhaft ist. Veranschaulicht wird das in Abbildung 3.8.

Zu sehen ist in grün die originale Klangquelle aus der Darstellung und in blau die wahrgenommene Klangquelle nach der Übersetzung. Links im Bild sieht man, was geschieht, würden die Koordinaten nicht verrechnet werden, dort verschiebt sich die Wahrnehmung auf der Y-Achse. Auf der rechten Seite im Bild ist die Korrektur nach der Verrechnung bildlich dargestellt. Die dafür entworfene Methode *mappingCoordinatesToSweetSpot* nimmt dabei die Positionsdaten und verrechnet sie mit dem Sweet Spot und seinen Offsets. Ihr Rückgabewert ist ein Tupel(X,Y) von Floatwerten, das die auf den Sweet Spot abgebildeten Koordinaten enthält.

Mit diesen Werten kann die Berechnung der Polarkoordinaten umgesetzt werden. Um aus kartesischen Koordinaten Polarkoordinaten zu berechnen, wird der Arkustangens verwendet. Abbildung 3.9 zeigt eine Skizze. Die *math-Library* von C++ bietet hierfür die Funktion *atan2*<sup>2</sup> an, welche aus zwei Parametern die Polarkoordinate berechnet und intern die in Formel 3.4 dargestellten Fallunterscheidungen berücksichtigt. Da diese nur verwendet wird, wird an dieser Stelle für tiefere Informationen auf mathematische Standardwerke und die C++ Referenz verwiesen.

<sup>2</sup>Quelle: <http://www.cplusplus.com/reference/cmath/atan2/> - Abruf: 2014-11-16

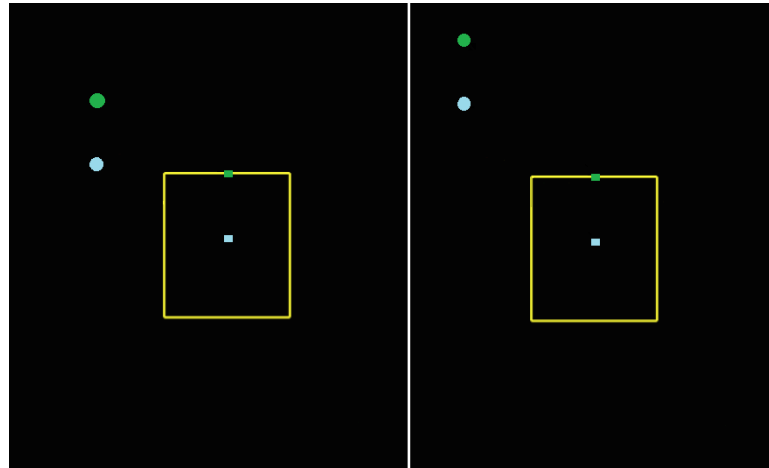


Abbildung 3.8.: Verschiebung der Klangquelle auf der Y-Achse um den Betrag des Sweet Spots. Links ist die Darstellung ohne Korrektur, rechts mit.

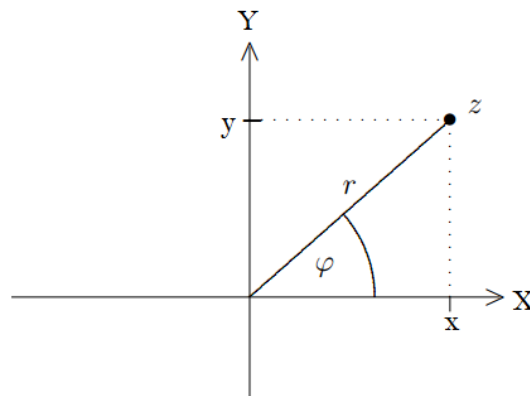


Abbildung 3.9.: Berechnung der Polarkoordinate  $\varphi$  anhand der kartesischen Koordinaten  $x$  und  $y$ .



$$atan2(x, y) := \begin{cases} \arctan(\frac{y}{x}), & x > 0 \\ \arctan(\frac{y}{x}) + \pi, & x < 0 \text{ und } y \geq 0 \\ \arctan(\frac{y}{x}) - \pi, & x < 0 \text{ und } y < 0 \\ \frac{\pi}{2}, & x = 0 \text{ und } y > 0 \\ -\frac{\pi}{2}, & x = 0 \text{ und } y < 0 \\ 0, & x = 0 \text{ und } y = 0 \end{cases} \quad (3.4)$$

Mit dem so berechneten Azimutwinkel ist die Richtung der Klangquelle die aus der Sicht des Hörers. Damit aber auch die Entfernung automatisiert berechnet werden kann, wird die Distanz zwischen Sweet Spot und Klangquellen benötigt. Diese Berechnung findet mit den auf den Sweet Spot normalisierten Positionsdaten der Klangquelle und dem Satz des Pythagoras statt. Formel 3.5 zeigt den Zusammenhang. Die Methode *processingCartesianToPolar* implementiert dabei beide Schritte, die Berechnung der Polarkoordinaten sowie die der Entfernung. Dazu wird sie mit den neu abgebildeten Koordinaten aufgerufen und liefert als Ergebnis den Azimutwinkel und die Distanz als Tupel( $\varphi$ ,  $r$ ) von Floatwerten.

$$Distance_i = \sqrt{(posX_i)^2 + (posY_i)^2} \quad (3.5)$$

### Ambisonics B-Format

Für die Übersetzung von Klangquellen aus dem Monoformat in das Ambisonics B-Format gibt es definierte Formeln. Diese berechnen anhand des Winkels die Anteile für die Schallschnellekomponenten. Der Anteil der Schalldruckkomponenten ist, unbetrachtet von Entfernungsdarstellung, immer gleich. Die Gains der Kanäle ergeben sich aus den Formeln 3.6, 3.7, 3.8 und 3.9, wobei  $\theta$  der Azimut ist und  $\alpha$  der Erhebungswinkel.

$$Gain_w = \frac{1}{\sqrt{2}} \quad (3.6)$$

$$Gain_x = \cos(\theta) \cdot \cos(\alpha), \text{ wobei } \cos(\alpha) \text{ immer 1 ist.} \quad (3.7)$$

$$Gain_y = \sin(\theta) \cdot \cos(\alpha), \text{ wobei } \cos(\alpha) \text{ immer 1 ist.} \quad (3.8)$$

$$Gain_z = \sin(\alpha), \text{ wobei dieser immer 0 ist.} \quad (3.9)$$

Da der Erhebungswinkel in einer zweidimensionalen Darstellung null ist und der  $\cos(0) = 1$ , wurden die Additionen mit  $\cos(\alpha)$  nicht implementiert. In den Formeln wird das aber korrekterweise mit abgebildet. Zu erkennen ist auch, dass der Gain vom Z-Kanal dauerhaft null ist, was bei einer zweidimensionalen Darstellung auch logisch erscheint.

#### 3-Kanal Surround

Bei der Umsetzung der Algorithmen für die stereofonen Formate wurde das im Grundlagenkapitel angesprochene „Constant Power Law“ angewandt. Wie schon dort beschrieben, findet das *Panning* immer zwischen zwei Lautsprechern statt. Bei der Verwendung in einem mehrkanaligen Format muss also vorher festgestellt werden, welches Lautsprecherpaar im Bereich zwischen  $0^\circ$  bis  $360^\circ$  für eine Klangquelle mit dem Winkel  $\theta_{pan}$  zuständig ist (Abbildung 3.10). Dafür wurde eine Abfragelogik eingesetzt, die überprüft, welches in Tabelle 3.2 gezeigte Panning angewendet werden muss [Wes98]. Die Winkel  $\theta_i$  der Lautsprecher, die bei dieser Logik eingesetzt wurden, sind die empfohlenen Vorschläge aus [ITU12]. Die Werte sind fest im Header der Klasse `wonderjuice::MultiFormatConverter` definiert.

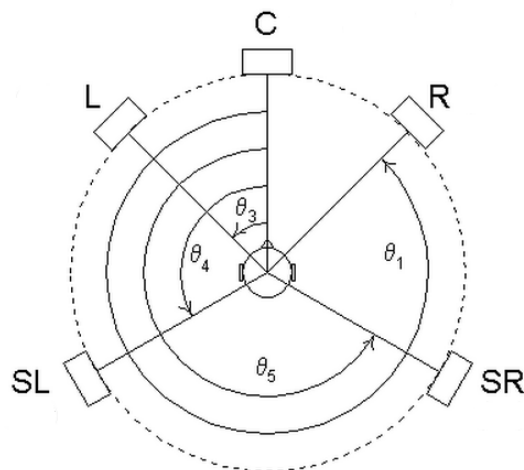


Abbildung 3.10.: Fünfkanal-Anordnung [Wes98]

Die Berechnung der *Gains* der Surroundlautsprecher (3.2) ist bei den Formaten 3/0/0 und 3/0/1 identisch. Der einzige Unterschied ist der Lfe-Kanal. Die Winkel in der Abfragelogik sind  $\theta_1 = 330^\circ$ ,  $\theta_2 = 0^\circ$  und  $\theta_3 = 30^\circ$ .

### 3. Softwarekonzept und Integration

| Position von $\theta_{pan}$             | Gains der Kanäle  |
|---|---|
| $\theta_{pan} \geq \theta_1$            | $\theta_m = \frac{\theta_{pan}-\theta_1}{360-\theta_1} \cdot 90$ , $g_1 = \cos \theta_m$ , $g_2 = \sin \theta_m$ , $g_3 = 0$      |
| $\theta_3 > \theta_{pan} \geq \theta_2$ | $\theta_m = \frac{\theta_{pan}-\theta_2}{\theta_3-\theta_2} \cdot 90$ , $g_2 = \cos \theta_m$ , $g_3 = \sin \theta_m$ , $g_1 = 0$ |

Tabelle 3.2.: Formeln für die Berechnung der Kanalanteile bei den Formaten 3/0/0 und 3/0/1

### 5-Kanal Surround

Dieses Verfahren konnte problemlos auf die Formate 3/2/0 und 3/2/1 erweitert werden (Tabelle 3.3). Die Abfrage wurde lediglich um die hinzugekommenen Lautsprecher ergänzt und das Panning Law für diese umgesetzt. Die Winkel in diesen Formaten sind  $\theta_1 = 330^\circ$  ,  $\theta_2 = 0^\circ$  ,  $\theta_3 = 30^\circ$  ,  $\theta_4 = 110^\circ$  und  $\theta_5 = 250^\circ$ .

| Position von $\theta_{pan}$             | Gains der Kanäle  |
|---|---|
| $\theta_{pan} \geq \theta_1$            | $\theta_m = \frac{\theta_{pan}-\theta_1}{360-\theta_1} \cdot 90$ , $g_1 = \cos \theta_m$ , $g_2 = \sin \theta_m$ , $g_3 = g_4 = g_5 = 0$      |
| $\theta_3 > \theta_{pan} \geq \theta_2$ | $\theta_m = \frac{\theta_{pan}-\theta_2}{\theta_3-\theta_2} \cdot 90$ , $g_2 = \cos \theta_m$ , $g_3 = \sin \theta_m$ , $g_1 = g_4 = g_5 = 0$ |
| $\theta_4 > \theta_{pan} \geq \theta_3$ | $\theta_m = \frac{\theta_{pan}-\theta_3}{\theta_4-\theta_3} \cdot 90$ , $g_3 = \cos \theta_m$ , $g_4 = \sin \theta_m$ , $g_1 = g_2 = g_5 = 0$ |
| $\theta_5 > \theta_{pan} \geq \theta_4$ | $\theta_m = \frac{\theta_{pan}-\theta_4}{\theta_5-\theta_4} \cdot 90$ , $g_4 = \cos \theta_m$ , $g_5 = \sin \theta_m$ , $g_1 = g_2 = g_3 = 0$ |
| $\theta_1 > \theta_{pan} \geq \theta_5$ | $\theta_m = \frac{\theta_{pan}-\theta_5}{\theta_1-\theta_5} \cdot 90$ , $g_5 = \cos \theta_m$ , $g_1 = \sin \theta_m$ , $g_2 = g_3 = g_4 = 0$ |

Tabelle 3.3.: Formeln für die Berechnung der Kanalanteile bei den Formaten 3/2/0 und 3/2/1

### 7-Kanal Surround

Genauso ist es auch bei den Formaten 3/4/0 und 3/4/1, wo lediglich zwei Surroundlautsprecher hinzugekommen sind. Die eingesetzten Winkel sind  $\theta_1 = 330^\circ$  ,  $\theta_2 = 0^\circ$  ,  $\theta_3 = 30^\circ$  ,  $\theta_4 = 70^\circ$  ,  $\theta_5 = 140^\circ$  ,  $\theta_6 = 220^\circ$  und  $\theta_7 = 290^\circ$ . Auf eine weitere Tabelle mit der Darstellung des Pannings wurde verzichtet, da dies aufgrund der beiden vorigen Tabellen trivial erscheint.

### Abstandsberechnung

Die Grundlage, auf der in allen Formaten die Berechnung von Abständen implementiert wurde, ist das *reziproke Abstandsgesetz*<sup>3</sup> oder auch  $\frac{1}{r}$ -Gesetz genannt. Es beschreibt die Schalldruck-Abnahme für punktförmige Schallquellen im Verhältnis zur Entfernung (Abbildung 3.11). Bei der Schall-Ausbreitung von einer Punktquelle bilden sich kugelförmige Wellen. Die Energie auf der Wellenfront ändert sich bei Ausbreitung nicht, sondern verteilt sich auf die Kugeloberfläche der Wellenfront, die proportional zu  $\frac{1}{r^2}$  ist. Da die Energie proportional zum Quadrat der Schallamplitude ist, muss die Amplitude mit  $\frac{1}{r}$  abnehmen. In den jeweiligen Konvertierungsmethoden wird dazu nach der Berechnung des Kanalanteils der jeweilige Gain in das Verhältnis zur Entfernung gesetzt. Die Kreislinie der Lautsprecher stellt dabei die Entfernung eins dar, was bewusst so gewählt wurde. Darstellungen auf der WFS-Anlage entfernen sich ebenfalls erst im rückwärtigen Raum der Lautsprecher. Bei der Übersetzung würde also fälschlicherweise eine Entfernung hinzugefügt werden, wenn diese direkt ab dem Sweet Spot berechnet werden würde.

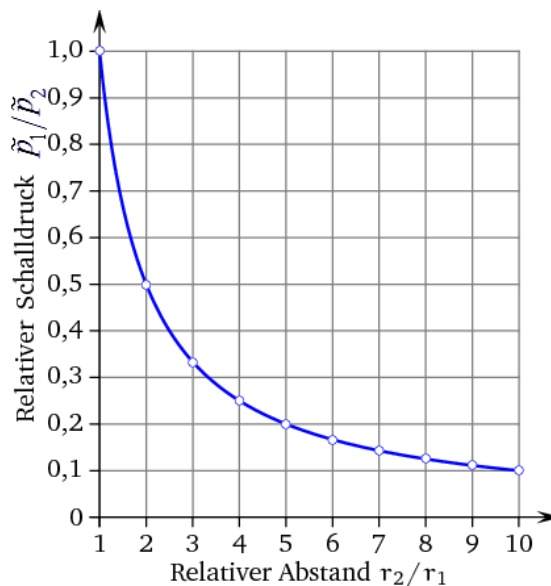


Abbildung 3.11.: Abnahme des Schalldrucks zur Entfernung - Sengpielaudio: Rechner Abstandsgesetz

Wie im Grundlagenkapitel zum räumlichen Hören bereits angemerkt, wird die Entfernung nicht nur an der Lautstärke bestimmt. Effekte wie Hall und Klangfarbenunterschiede können die Genauigkeit der Wahrnehmung steigern. Diese zu erzeugen beziehungsweise zu simulieren,

<sup>3</sup><http://www.sengpielaudio.com/Rechner-abstandsgesetz.htm> - Abruf: 2014-11-16

stellt allerdings eine Veränderung der ursprünglichen Wiedergabe dar und wurde nicht weiter betrachtet.

#### Besonderheiten

Wenn auch vereinzelt schon erwähnt, sollen hier noch einmal die Besonderheiten, die in dieser Arbeit aufgetreten sind, angesprochen werden. Da wäre zum einen die Y-Achse des Koordinatensystem von WONDER zu nennen, welche spiegelverkehrt verläuft. Im Code finden dadurch an einigen Stellen Koordinatentransformation statt, die diese Besonderheit berücksichtigen. Zum anderen ist die Betrachtung von  $0^\circ$  in der Winkelfunktion ungleich der Betrachtung von  $0^\circ$  bei der WFS-Anlage. Deswegen müssen in *processingCartesianToPolar* auf alle Winkel  $270^\circ$  aufgerechnet und *modulo*  $360^\circ$  gerechnet werden, damit die Betrachtung von  $0^\circ$  auf die Blickrichtung des Hörers gebracht wird.

Ein weiterer Punkt, der jedoch nicht ganz gelöst werden konnte, sind die sogenannten „fokussierten Quellen“ in der WFS-Anlage [Han14]. Diese verlaufen im inneren der Sweet Area und werden bei der Übersetzung aus der Sicht des Hörers im jeweiligen Winkel zu diesem auf die Kreislinie der Lautsprecher abgebildet. Bewegt sich diese Klangquelle allerdings geradlinig von einer Seite zur anderen und schneidet dabei exakt den Sweet Spot, kommt es zu plötzlichen Sprüngen. Der Grund dafür ist, dass die Klangquelle auf der Strecke über den Sweet Spot hinweg, zu keiner Zeit den Winkel zu diesem ändert. Erst bei Überschreiten dieser Position wechselt die Klangquelle schlagartig den Winkel und wird dadurch auf der gespiegelten Seite dargestellt. Die Wahrscheinlichkeit, dass fokussierte Quellen aber bei einem Audioprojekt verwendet werden, das für mehrere Formate geeignet sein soll, wird in dieser Arbeit als sehr gering eingeschätzt. Diese werden eher in sehr speziellen Projekten verwendet, die nur für die WFS-Anlage gedacht sind. Darüber hinaus müsste die Klangquelle genau über den Sweet Spot hinweg verlaufen, damit der genannte Effekt auftritt. Aus diesen Gründen wird diese Anomalie nicht als schwerwiegend eingestuft und die Vermeidung in die Verantwortung des Anwenders übergeben.

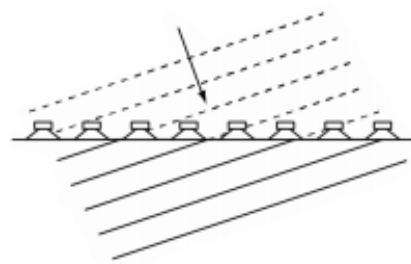


Abbildung 3.12.: Linearquelle [Han14]

Abschließend ist noch anzumerken, dass sich aufgrund der technischen Eigenschaften der WFS-Anlage sogenannte *Linearquellen* darstellen lassen. Die Wellenfront dieser Klangquellen breitet sich, wie die Abbildung 3.12 zeigt, nicht kreisförmig aus, sondern verläuft geradlinig. Die

Darstellung dieser Klangquellen wurde bei der Umsetzung nicht berücksichtigt. Deshalb werden die Audiodaten dieser Klangquellen bei der Übersetzung automatisch mit null überschrieben.

#### 3.3.2. Lineare Interpolation der Samples

Die lineare Interpolation wurde mit der Methode *applyGainRamp*<sup>4</sup> von Audiopuffer-Objekt umgesetzt. Diese wendet auf einen Bereich von Samples die Wertänderung zwischen den Parametern *startGain* und *endGain* auf den jeweiligen Kanal an. Hierfür wurden dem Konverter-Objekt die Arrays *oldGains* und *newGains* hinzugefügt, die jeweils die Gain-Werte zum Zeitpunkt *t* und *t-1* beinhalten. So kann über die Anzahl der Kanäle iteriert werden und für den jeweiligen Kanal die Interpolation von Startwert zu Zielwert erfolgen. Lediglich zu Beginn erfolgt das nicht, weil sonst ungewollt Bewegungen entstünden. Deswegen werden die Arrays bei der Erzeugung des Konverter-Objektes mit -1 vorinitialisiert. So kann der erste Durchlauf erkannt werden und anstelle von *applyGainRamp* die Methode *applyGain* aufgerufen werden. Diese wendet, wie der Name es vermuten lässt, die Gain-Werte direkt auf alle Samples an und interpoliert sie nicht.

#### 3.3.3. Implementierung der MFC-Klassen

Der Konverter wird repräsentiert von den beiden Klassen *wonderjuce::MultiFormatConverter* und *wonderjuce::GlobalMfcParameters*. Die Strukturierung des Programmcodes wird wie in SPAOP von Martin Hansen fortgeführt [Han14, S.45]. Daher wird der Programmcode wegen der Verwendung von JUCE-basierten Implementationen in den namespace *wonderjuce* eingliedert. Die Klassen und Ihre strukturelle Eingliederung ist in dem Klassendiagramm in Abbildung 3.13 vereinfacht dargestellt.

##### MultiFormatConverter

Der *MultiFormatConverter* ist die zentrale Komponente im SPAOP-MFC und wurde für die Berechnungen konzipiert. Er implementiert die zentrale Methode *processingToSelectedFormat*, beinhaltet alle statischen Methoden für die Konvertierung in die unterstützten Formate, implementiert die Methoden *mappingCoordinatesToSweetSpot* und *processingCartesianToPolar* für die Berechnung der Rauminformationen und setzt die bereits diskutierte Interpolation um. Hierfür verfügt er über den angesprochenen Funktionszeiger, der auf die aktuell über das Format gewählte statische Methode zeigt.

---

<sup>4</sup>AudioSampleBuffer: <https://www.juce.com/api/classAudioSampleBuffer.html> - Abruf: 2014-11-23

Zusätzlich sind noch zwei weitere Zeigervariablen vorhanden, mit denen auf die angesprochenen Arrays für die alten und neuen Gains referenziert wird. Dieser Ansatz wurde gewählt, damit die Arrays dynamisch zur Laufzeit angepasst werden können. Findet ein Wechsel des Formates statt, wird der gehaltene Speicher gelöscht und die Referenz auf ein neu angelegtes Array gesetzt.

Damit Berechnungen nicht unnötig wiederholt werden, wenn sich eine Klangquelle gar nicht bewegt, wurden die Datenfelder *oldX* und *oldY* integriert und mit einer Abfrage verknüpft. Wie die Bezeichnung andeutet, werden darin am Ende die Koordinaten gespeichert, auf dessen Grundlage die letzte Berechnung stattgefunden hat. Im nächsten Durchlauf werden diese mit den aktuellen Koordinaten verglichen. Sind sie gleich, findet keine Berechnung statt und der Gain kann mit *applyGain* wieder direkt auf die Samples angewandt werden.

Die Anzahl der verfügbaren Kanäle im Audiopuffer ist unabhängig vom gewählten Format, sie wird, wie schon erwähnt, durch das in der DAW als Ausgang gewählte Format bestimmt. Damit also keine Kanäle mit unbekanntem Inhalt an den Host zurückgegeben werden, wurde in den statischen Methoden und der *processingToSelectedFormat*-Methode ein Rückgabewert implementiert, der bei erfolgreichem Durchlaufen aller Methoden die Anzahl der manipulierten Kanäle angibt. In der Methode *processBlock* im *SourceController* können anhand dessen alle Kanäle terminiert werden, deren Index über dem Wert liegen.

#### **GlobalMfcParameters**

*GlobalMfcParameters* ist die Komponente für alle instanzübergreifenden Parameter des Konverters. Dafür wurde es als *Singleton* implementiert, da alle im Host instanziierten Plugins über einen gemeinsamen Speicher verfügen. Jedes Plugin kann mittels *wonderjuce::GlobalMfcParameters::getInstance* auf die Referenz des Objektes zugreifen und darüber auf die Schnittstellen. Die Eingaben der globalen Einstellungen aus dem Plugin werden mit den Attributen *offsetX*, *offsetY*, *wfsWidth*, *wfsDepth*, *Radius* und *formatType* verwaltet. Darüber hinaus beinhaltet die Komponente noch die Liste der Koordinaten der Lautsprecher. Diese wurde mit einer *std::list<float>* implementiert, welche bei jedem Wertwechsel der klasseneigenen Attribute gelöscht und neu erstellt wird. Dadurch konnte eine Logik zur Unterscheidung der unterschiedlichen Darstellungen der Lautsprecheranordnungen in der GUI eingespart werden. Eine komplette Neuberechnung der Liste ist zwar aufwändig, allerdings findet ein Wertwechsel deutlich seltener statt als die Aktualisierung der GUI und meist nur bei der initialen Einstellung des Plugins. Dadurch kann an einer anderen Stelle wesentlich mehr Rechenleistung eingespart werden.

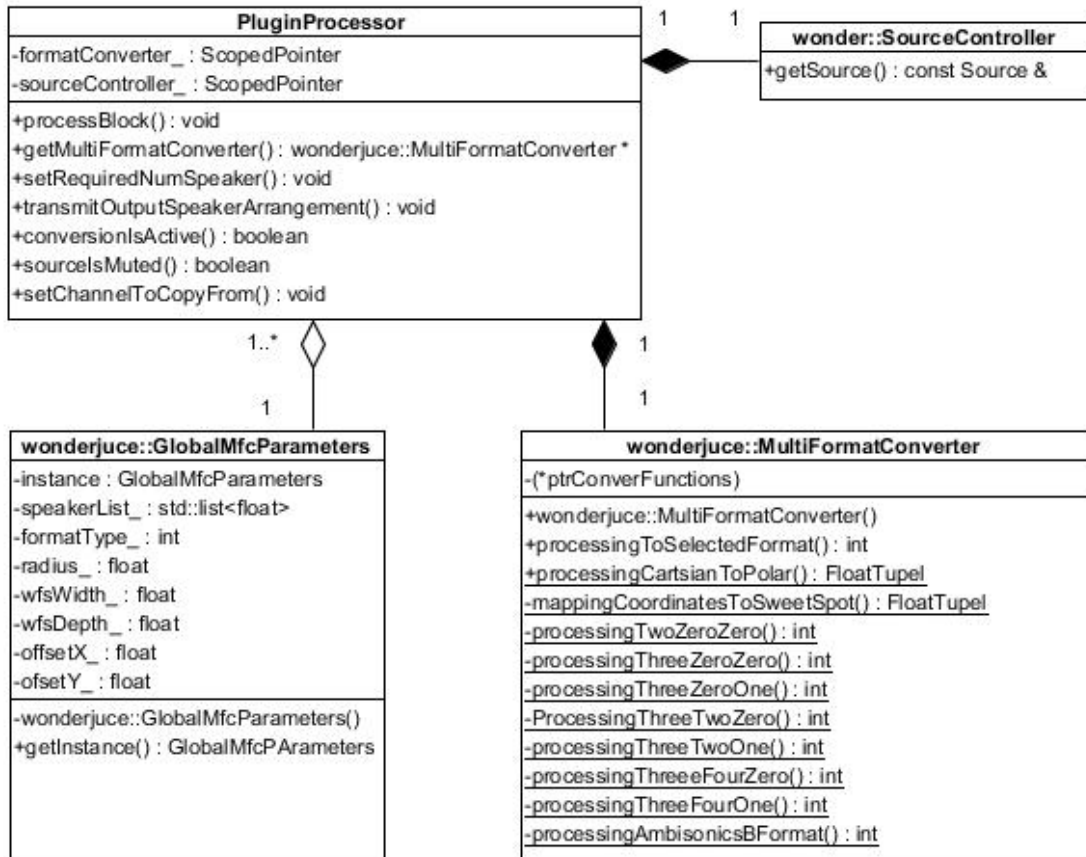


Abbildung 3.13.: Vereinfachtes Klassendiagramm von SPAOP-MFC.

Das Erstellen der Liste wurde mit der Methode *setParameters* umgesetzt. Umgekehrt zu dem Verfahren der Winkelberechnung bei der Konvertierung werden hier die Koordinaten ausgehend vom Winkel des Lautsprechers berechnet. Dabei wurden wieder die definierten Werte aus dem Header der Klasse *wonderjuce::MultiFormatConverter* verwendet, um keine redundanten Werte zu schaffen. Mit den implementierten Formeln 3.10 und 3.11 lassen sich die Koordinaten der Lautsprecher berechnen. Um diese in der GUI verwenden zu können, müssen die Werte auf einen Bereich zwischen null und eins umgerechnet werden. Dazu wurde die Methode *wonder::Source::normalizeParameters* eingesetzt, die diesen Schritt implementiert.

$$X_i = \cos\left(\theta_i \cdot \frac{\pi}{180}\right) \cdot radius + offsetX \quad (3.10)$$

$$Y_i = \sin\left(\theta_i \cdot \frac{\pi}{180}\right) \cdot radius - \frac{wfsDepth}{2} - offsetY \quad (3.11)$$



### 3.3.4. Erweiterung der grafischen Oberfläche

Für die grafische Darstellung der aus den Anforderung hervorgehenden Einstellungsmöglichkeiten wurde mit dem *Introjuicer* ein neues Tab erstellt und dem bereits enthaltenen *Tabbed-Component* hinzugefügt. Die Integration im *PluginEditor* erfolgte dabei automatisch durch den GUI-Builder. Die erforderlichen Button-Listener wurden ebenfalls mit vordefiniert und mussten nur logisch verknüpft werden. Die Eingaben der beiden lokalen Einstellungen „No surround-effect“ und „Ignore this source“ gehen über die Referenz des *SourceControllers* an die Referenz auf das *MultiFormatConverter*-Objekt, das hierfür öffentliche Schnittstellen hat. Die global gültigen Einstellungen können hingegen direkt an die Schnittstellen des Singleton-Objekt der *GlobalMfcParameters*-Klasse übergeben werden, von wo aus sich diese Einstellungen auf sämtliche aktiven SPAOP-MFC Instanzen auswirken. Das Mockup zu diesem Tab ist in der Abbildung 3.14 dargestellt.

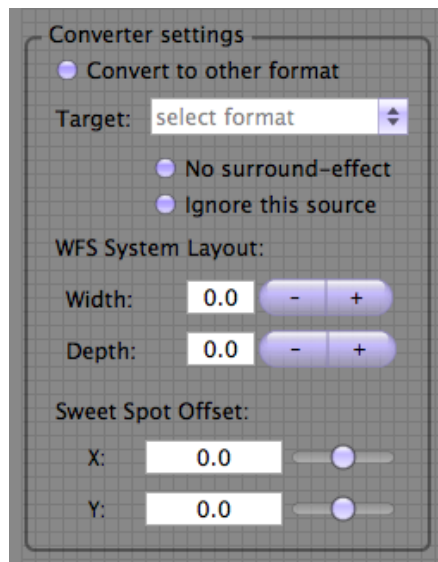


Abbildung 3.14.: Mockup der grafischen Erweiterung im Introjuicer

### 3.3.5. Manipulation bestehender Klassen

Um den MFC final in SPAOP zu integrieren, wurden die folgenden Klassen angepasst und um notwendige Funktionen und Felder erweitert.

#### PluginProcessor

Die Integration der *MultiFormatConverter*-Komponente erfolgte im *PluginProcessor*, der schon in SPAOP größtenteils auf die Vermittlung zwischen Host, GUI und SourceController ausgelegt war. Das hat die Entscheidung, die Integration an dieser Stelle vorzunehmen, einfach gemacht. Der *PluginProcessor* hält alle notwendigen Ressourcen mit Daten für die Konvertierung, bietet eine Schnittstelle, die Eingaben aus dem *PluginEditor* entgegen nimmt und beinhaltet darüber hinaus die *processBlock*-Methode mit dem zu manipulierenden Audiopuffer. Hierfür wurde dem *PluginProcessor* ein *ScopedPointer*<sup>5</sup> hinzugefügt, der auf das im Konstruktor erzeugte MFC-Objekt verweist. Dieser arbeitet nach dem *RAII*-Prinzip und ist eine JUCE-eigene Erweiterung des C++ *std::auto\_ptr*. Er verwaltet den Speicher des Objektes dynamisch und gibt ihn automatisch wieder frei, wenn das Objekt seine Gültigkeit verliert. Somit kann die Referenz des Objektes problemlos weitergereicht und bei Bedarf Kopien erzeugt werden.

Für die Implementierung der *processBlock*-Methode wurden der Komponente private Variablen und Methoden hinzugefügt, um die Prozesslogik zu steuern. Im Folgenden werden die wichtigsten erläutert:

1. *requiredNumSpeakers\_*: Diese Variable wird durch die Methode *setRequiredNumberOfSpeaker* gesetzt, wenn das Format gewählt wird. Sie beinhaltet, wie der Name schon aussagt, die Anzahl der Lautsprecher für das gewählte Format. In der Prozesslogik von *processBlock* findet ein Vergleich auf die Anzahl der im Audiopuffer enthaltenen Kanäle statt. Stellt der Host weniger Kanäle im Speicher zur Verfügung als das Format benötigt, wird *wonderjuce::MultiFormatConverter::processingToSelectedFormat* nicht aufgerufen. Es kann also passieren, dass eine Fehlbedienung des Plugins dazu führt, dass die Ausgabe stumm bleibt. Das ist zum Beispiel der Fall, wenn 3/2/0 im Host gewählt ist und 3/2/1 im Plugin.
2. *channelToCopyFrom\_*: Der Audiopuffer, der vom Host übergeben wird, befindet sich in nur einem Kanal. Wie bereits beschrieben, muss dieser vor der Bearbeitung auf alle Kanäle kopiert werden, es gibt dafür allerdings keinen „Standardkanal“. In *Cubase* war dies bei Formaten mit mehr als zwei Kanälen immer der „mittlere Kanal“ beziehungsweise „Kanal 2“ und bei Stereo der „linke Kanal“ beziehungsweise „Kanal 0“. Um das zu unterscheiden, wurde diese Variable integriert. Diese wird bei Eingabe des Formates durch die Methode *setChannelToCopyFrom* gesetzt.

---

<sup>5</sup><https://www.juce.com/api/classScopedPointer.html> - Abruf: 2014-11-25

3. *transmitOutputSpeakerArrangement*: Diese Methode wurde entworfen, um dem Konverter die Lautsprecher-Anordnung der Kanäle im Puffer mitzuteilen. Sie übergibt dazu dem MFC einen String, der durch Leerzeichen getrennt die Reihenfolge der Lautsprecher beinhaltet, zum Beispiel „L R C Lfe Ls Rs“. Den String stellt der PluginProcessor selbst zur Verfügung, da er die Methode *getOutputSpeakerArrangement* vom *AudioProcessor*<sup>6</sup> erbt. Leider konnte das im Zusammenspiel mit Cubase nicht getestet werden. Cubase übergibt keine Information über das Arrangement an die VST-Schnittstelle, wodurch der String leer bleibt und SPAOP-MFC dadurch auf voreingestellte Werte zurückgreift.

Damit die Konvertierung innerhalb von *processBlock* nur dann gestartet wird, wenn alle notwendigen Bedingungen zutreffen, wurde der Methode *processingToSelectedFormat* eine Abfragelogik vorgeschaltet. Die Übersetzung soll nur dann initiiert werden, wenn der Konverter aktiviert ist, die Anzahl der verfügbaren Kanäle im Puffer für das Format ausreichen, die Klangquelle nicht mit „ignore this source“ gekennzeichnet ist, es sich um eine Punktschallquelle handelt und ein Format gewählt wurde. Tabelle 3.4 führt die Bedingungen, die verwendet wurden, mit der Herkunft des Wertes und dem Sollwert auf, der vorliegen muss.

| Bezeichnung                | Herkunft             | Sollwert                            |
|----------------------------|----------------------|-------------------------------------|
| <i>conversionIsActive</i>  | MultiFormatConverter | TRUE                                |
| <i>requiredNumSpeaker_</i> | PluginProcessor      | $\geq$ <i>buffer.getNumChannels</i> |
| <i>sourceIsMuted</i>       | MultiFormatConverter | FALSE                               |
| <i>legalSource</i>         | wonder::Source       | TRUE if point                       |
| <i>functionIsSelected</i>  | MultiFormatConverter | TRUE                                |

Tabelle 3.4.: Bedingungen für die Konvertierung

#### SourcePanel und SourceDisplay

Die GUI-Klassen *wonderjuce::SourceDisplay* und *wonderjuce::SourcePanel* sind für die zweidimensionale Anzeige der Klangquellen im Plugin entwickelt worden. Dabei ist *wonderjuce::SourceDisplay* die Basisklasse von *wonderjuce::SourcePanel* [Han14]. Hier wurde die nicht öffentliche Methode *paintSpeakerArray* hinzugefügt, die das Layout des gewählten Formates in die GUI zeichnet. Diese iteriert über die vom *GlobalMfcParameters*-Singleton bereitgestellte *speakerList\_* und zeichnet das Layout anhand der Koordinaten aus der Liste. Die abgeleitete Klasse *wonderjuce::SourcePanel* implementiert die öffentliche *paint*-Methode und führt die *paintSpeakerArray*-Methode bei Aufruf aus.

---

<sup>6</sup><http://www.juce.com/api/classAudioProcessor.html> - Abruf: 2014-11-25

#### **XmlFactory**

Die XmlFactory bietet die Funktionen an, mit denen die Zustände des Plugins im XML-Format gespeichert werden können. Die Daten werden dann dem vom Host zur Verfügung gestellten Speicher übergeben und aus diesem wieder rekonstruiert. Dadurch können *Presets* für das Plugin erstellt werden, die *copy - paste* Funktion für Plugins wird unterstützt und ein Plugin behält seinen aktuellen Zustand, wenn es minimiert wird. Damit auch die Werte des Konverters gespeichert und rekonstruiert werden können, wurde die Funktionen *updateConverterFromXml* und *createConverterXml* implementiert.

## 4. Testen der Funktionalität

Dieser Abschnitt beschreibt, wie die Funktionalität des Konverters getestet und das Ergebnis der Übersetzung überprüft worden ist. Hierzu wurde zuerst die allgemeine Funktion getestet, womit in erster Linie die Stabilität der Software und das Verhalten bei verschiedensten Benutzereingaben gemeint ist. Im Anschluß daran ist ein Hörtest mit Probanden durchgeführt worden, um das Ergebnis des Konverters zu verifizieren.

### 4.1. Allgemein

Zuerst wurde getestet und festgestellt, inwieweit der Konverter funktional korrekt arbeitet. Diese Tests wurden mit einem Apple Macbook Air (OSX 10.9.5), Xcode 6.0.1 (6A317) und Cubase 7.5 vorgenommen. Mittels Bordwerkzeugen des Xcode Debuggers wurde ein intensives Testing der Software durchgeführt. Dieser bieten an, aus Xcode heraus das Plugin in Cubase auszuführen und während der Bedienung zu debuggen. Dort können dann beliebig viele Instanzen des Plugins geladen und getestet werden. Zum einen kann so nach Fehlern gesucht werden, treten welche auf, zeigt Xcode den gesamten Stack Trace an und ermöglicht eine gute Fehlersuche und Analyse. Zum anderen können gezielt Abläufe untersucht werden, indem man Breakpoints in den Code setzt und sich durch das Plugin arbeitet.

Bei den Tests wurde so vorgegangen, dass zuerst mit einem Komponententest die Funktionalität des MultiFormatConverters überprüft wurde. Danach wurde mit einem Integrationstest die Zusammenarbeit aller Schnittstellen überprüft und im darauffolgenden Systemtest die korrekte Zusammenarbeit aller Komponenten sichergestellt. Bei dem Systemtest wurde das Plugin auf verschiedenste Weise bedient. Dabei wurden Seiteneffekte aufgedeckt, die durch nicht vorhersehbare Reihenfolgen von Benutzereingaben aufgetreten sind. Hierzu war es wichtig, das Plugin abweichend von der Implementierungslogik zu bedienen. Im Debugger wurde währenddessen vor allem auf Wertänderungen von Attributen geachtet. In den eigens implementierten Set-Methoden wurden Breakpoints gesetzt, die bei jeder Wertänderung eines Attributes angeschlagen haben. Ebenso wurden Breakpoints dort in die Code-Strukturen gesetzt, von denen ausgegangen wurde, diese nicht zu durchlaufen.

Im Anschluß daran erfolgte in einem erweiterten Systemtest die Überprüfung der instanzübergreifenden Wertänderungen sowie die Funktionalität, einen Pluginzustand zu speichern und wiederherzustellen. Dazu wurden verschiedene Plugin-Presets erstellt und mehrere Instanzen des SPAOP-MFC in den Host geladen. Nacheinander wurden in abwechselnde Instanzen unterschiedliche Presets geladen und überprüft, ob das geladene Preset korrekt rekonstruiert wurde und ob die anderen Instanzen diese Werte mit übernommen haben.

Der letzte Testabschnitt hat sich auf die GUI bezogen. Hier musste vor allem überprüft werden, ob die dargestellten Lautsprecher mit den realen Aufstellungen übereinstimmen. Dazu wurden vor allem die Grenzwerte betrachtet. Eine Klangquelle wurde dazu mittels SPAOP-MFC auf den dargestellten Lautsprechern umher bewegt und die akustische Wahrnehmung mit der dargestellten Position verglichen. Geprüft wurde vor allem, dass die Klangquelle, wenn sie direkt auf den Lautsprecher bewegt wurde, auch nur von diesem wiedergegeben wurde.

Bis auf ein paar kleine leicht zu behebbende Fehler, bei denen es hauptsächlich um Logikfehler in der Implementierung ging, wurden keine größeren Probleme entdeckt. Nachdem diese beseitigt waren, konnten die Tests allesamt erfolgreich abgeschlossen und die geforderte Funktionalität des Plugins nachgewiesen werden.

## 4.2. Hörtest

Um neben der Funktionalität auch das Ergebnis des Konverters zu überprüfen, wurde ein Hörtest mit Probanden durchgeführt. Das Ziel dieses Hörtest war nicht zu beweisen, wie „gut“ der Konverter dem Hörer eine Phantomschallquelle darbieten kann. Das Ziel war es zu zeigen, dass sich die Wahrnehmungen auf der WFS-Anlage mit den Wahrnehmungen auf der konvertierten Variante decken. Damit sollte bewiesen werden, dass ein übersetztes WFS-Projekt zu einem gleichen Ergebnis kommt. Getestet wurde dabei die Lokalisation aus verschiedenen Richtungen, die Lokalisation bei unterschiedlichen Abständen, der Vergleich bei bewegten Quellen und der Vergleich von räumlichen Darbietungen mit mehreren Quellen. Der Testfragebogen, welcher durch die Probanden während des Tests ausgefüllt werden musste, befindet sich im Anhang.

### 4.2.1. Aufbau

Der Hörtest wurde dazu in drei Abschnitte unterteilt und der Anspruch stufenweise gesteigert. Bei der Reihenfolge der Einzeltests innerhalb der Abschnitte wurden darauf geachtet, diese zu streuen. Das sollte verhindern, dass dem Probanden automatisch bewusst ist, welche Anlage gerade die Wiedergebende ist. Vor allem bei den Lokalisationstests sollte dadurch vermie-



Abbildung 4.1.: Audio-Arbeitsplatz im WFS-Labor der HAW.

den werden, dass die Ergebnisse aufeinanderfolgender Varianten genauer werden, weil dem Probanden die Richtung bekannt ist.

Die Referenzanlagen, die bei dem Hörtest verwendet wurden, sind die im Grundlagenkapitel beschriebene WFS-Anlage und eine 3/2/1 Surround-Anlage aus demselben Labor. An einem erst vor kurzem für diese Zwecke beschafften Audio-Arbeitsplatz, zu sehen auf [Abbildung 4.1](#), wurden die Hörtests entwickelt und während des Testzyklus abgespielt. Die verschiedenen Projekte für die Tests wurden alle mit Cubase 7.5 entwickelt.

Die 3/2/1 Surround-Anlage wurde, wie in [Abbildung 4.2](#) gezeigt, aufgestellt. Der gewählte Radius von 2.75 m entspricht dabei dem voreingestellten Wert im Plugin. Der Proband hat während des Tests auf einem Stuhl gesessen, der auf dem Sweet Spot positioniert wurde. Die Höhe der Lautsprecher wurde dabei bewusst oberhalb der Ohrhöhe des Probanden gewählt. Dadurch waren die Lautsprecher aus der Sitzposition des Probanden heraus auf einer ähnlichen Höhe wie die WFS-Anlage. Dies sollte verhindern, dass durch Lokalisationen auf der Z-Achse die Anlagen auseinandergehalten werden können.

#### **Test 1: Richtungslokalisierung**

In diesem Test wurde eine Klangquellen schrittweise wiederholt abgespielt. Die Entfernung betrug dabei durchgehend einen Meter und die Klangquelle war unbewegt. Der Proband



Abbildung 4.2.: Aufbau des Hörtests im WFS-Labor der HAW.



#### 4. Testen der Funktionalität

---

sollte die Position der Quelle lokalisieren und in einer dafür vorgesehenen Grafik einzeichnen. Zur Vereinfachung wurde das Geräusch dreimal hintereinander in sehr kurzen Abständen abgespielt. Dadurch sollte die Lokalisation unterstützt werden, weil der Proband etwas Zeit hatte, sich auf das Geräusch einzustellen. Eine Übersicht der Einzeltests wird in Tabelle 4.1 dargestellt.

| Testbezeichnung | Wiedergebende Anlage | Richtung |
|-----------------|----------------------|----------|
| ET-A1           | WFS                  | 15°      |
| ET-A2           | WFS                  | 90°      |
| ET-A3           | 3/2/0                | 300°     |
| ET-A4           | 3/2/0                | 45°      |
| ET-A5           | 3/2/0                | 160°     |
| ET-A6           | 3/2/0                | 355°     |
| ET-A7           | WFS                  | 280°     |
| ET-A8           | WFS                  | 235°     |
| ET-A9           | 3/2/0                | 15°      |
| ET-A10          | 3/2/0                | 90°      |
| ET-A11          | WFS                  | 300°     |
| ET-A12          | WFS                  | 45°      |
| ET-A13          | WFS                  | 160°     |
| ET-A14          | WFS                  | 355°     |
| ET-A15          | 3/2/0                | 280°     |
| ET-A16          | 3/2/0                | 235°     |

Tabelle 4.1.: Einzeltests von Test 1

#### **Test 2: Entfernungslokalisation**

Im zweiten Test wurde eine Klangquelle aus einer Richtung nacheinander aus unterschiedlichen Entfernungen abgespielt. Die ersten beiden Entfernungen waren dabei immer 1 und 2 Meter. Diese Information sowie die Richtung wurden dem Probanden vorher mitgeteilt. Dadurch sollte es ihm ermöglicht werden, eine Relation zu der Entfernung aufzubauen. Außerdem wurde eine Stimme als Geräusch gewählt, da bekannte Geräusche eher räumlich zugeordnet werden können als unbekannte Geräusche. Eine Übersicht der Einzeltests wird in Tabelle 4.2 dargestellt.

#### 4. Testen der Funktionalität

| Testbezeichnung | Richtung | Wiedergebende Anlage | Dargestellte Entfernung in Meter |
|-----------------|----------|----------------------|----------------------------------|
| ET-B1           | 0°       | 3/2/0                | 1, 2, 3, 6, 12                   |
| ET-B2           | 90°      | 3/2/0                | 1, 2, 4, 6, 8                    |
| ET-B3           | 270°     | 3/2/0                | 1, 2, 3.5, 9.5, 11.5             |
| ET-B4           | 180°     | 3/2/0                | 1, 2, 3, 4, 5                    |
| ET-B5           | 0°       | WFS                  | 1, 2, 3, 6, 12                   |
| ET-B6           | 90°      | WFS                  | 1, 2, 4, 6, 8                    |
| ET-B7           | 270°     | WFS                  | 1, 2, 3.5, 9.5, 11.5             |
| ET-B8           | 180°     | WFS                  | 1, 2, 3, 4, 5                    |

Tabelle 4.2.: Einzeltests von Test 2

#### Test 3: Szenen

Der letzte Test bezog sich auf die Gesamtwahrnehmung einer Darstellung. Dafür wurden mehrere Klangquellen gleichzeitig abgespielt, die sich teils innerhalb der Szenen bewegt haben. Jeder Test wurde hintereinander erst auf der 3/2/1 Surround- und dann auf der Wellenfeldsynthese-Anlage abgespielt. Dem Proband wurde nicht mitgeteilt, welche Anlage die wiedergebende ist. Seine Aufgabe bestand darin, sich einen räumlichen Gesamteindruck von der Wiedergabe zu machen. Anhand dessen sollte er entscheiden, welche Variante ihm besser gefallen hat und in Stichworten begründen warum. Die Möglichkeit, beide Anlagen gleich zu bewerten, war gegeben. Eine Übersicht der Einzeltests wird in Tabelle 4.3 dargestellt.

| Testbezeichnung | Dargestellte Szene  |
|-----------------|---|
| MT-A1           | Zwei rhythmische Töne, die sich gegeneinander im Kreis um den Hörer bewegen |
| MT-A2           | Musikalische Darbietung mit vielen Klangquellen, aber ohne Bewegung         |
| MT-A3           | Musikalische Darbietung mit vielen Klangquellen, vereinzelt mit Bewegungen  |

Tabelle 4.3.: Einzeltests von Test 2

#### 4.2.2. Ergebnisse des Hörtests

Für die Darstellung der Ergebnisse von Test 1 und Test 2 wurde das arithmetische Mittel, der Median, sowie die Standardabweichung genommen. Diese Werte haben eine statistische Aussagekraft und können die Ergebnisse gut verdeutlichen. Im dritten Test gab es nur drei verschiedenen Aussagemöglichkeiten, daher werden hier die Summen der Einzeltests und der

Anteil am Gesamtergebnis dargestellt. Die schriftlichen Aussagen der Probanden wurden bei der Auswertung verarbeitet.

### Ergebnisse Test 1

Neben den Angaben der Probanden werden zusätzlich noch die Differenzen in absoluten Werten dargestellt, die sich zwischen den Angaben und den Sollwerten ergeben haben. Der Grund dafür ist, dass daraus in Einzelfällen genauere Rückschlüsse gezogen werden können. Die Angaben in diesem Test beziehen sich auf den Umkreis des Hörers im Bereich zwischen  $0^\circ$  und  $360^\circ$ , der am Ende wieder von vorne anfängt. Dadurch entstehen teils starke Verzerrungen im Bereich des Übergangs. Angenommen, eine Klangquelle ist auf  $350^\circ$  dargestellt und alle Probanden tippen bei der WFS-Anlagen um  $15^\circ$  zu niedrig und bei der 3/2/1 Surround-Anlage  $15^\circ$  zu hoch. Dann bedeutet das für die WFS-Anlage ein Gesamtergebnis von  $335^\circ$  und eine Abweichung zum Sollwert und  $15^\circ$ . Für die 3/2/1 Surround-Anlage aber ist das Gesamtergebnis  $5^\circ$  und die Abweichung zum Sollwert beträgt  $345^\circ$ . In der Darstellung in einem Balkendiagramm würde dies den Anschein erwecken, die Ergebnisse würden weit auseinander liegen, aus diesem Grund werden die absoluten Abweichungen mit aufgeführt.

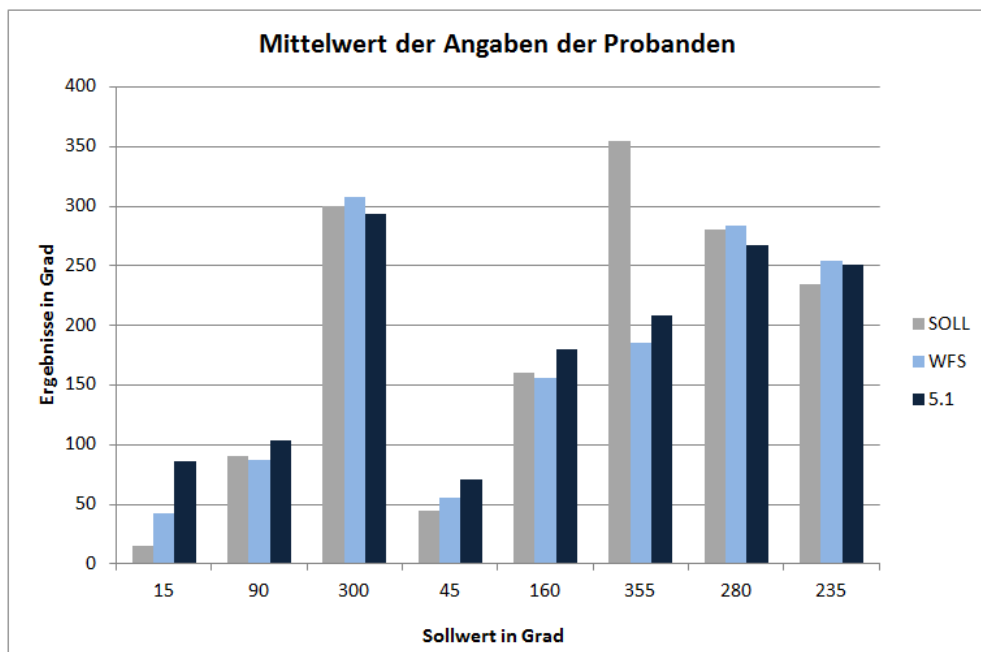


Abbildung 4.3.: Mittelwert der Angaben aus Test 1.

#### 4. Testen der Funktionalität

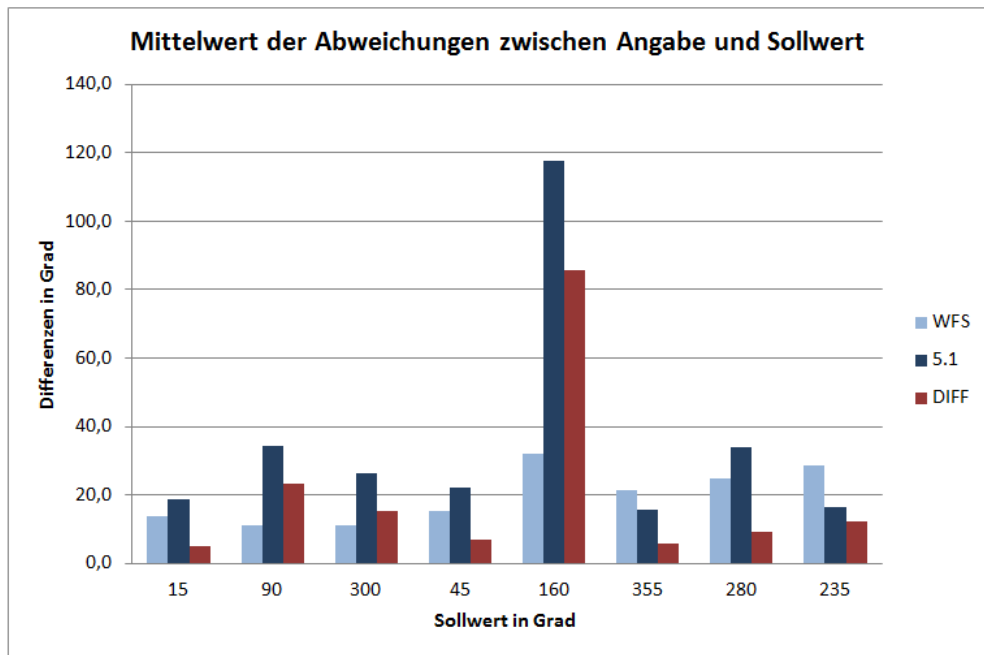


Abbildung 4.4.: Mittelwert der Abweichungen aus Test 1.

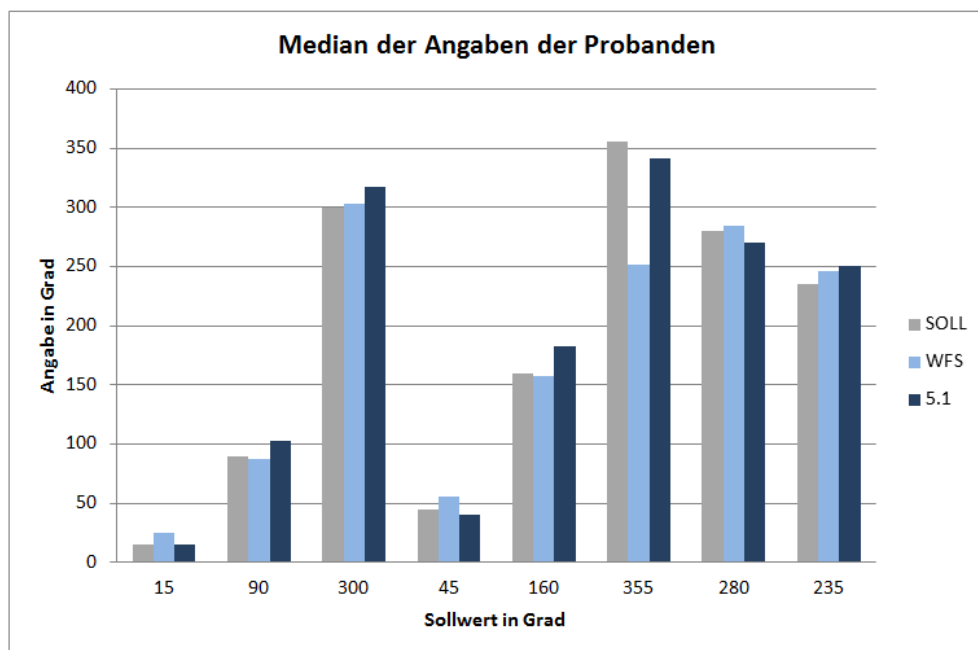


Abbildung 4.5.: Median der Angaben aus Test 1.

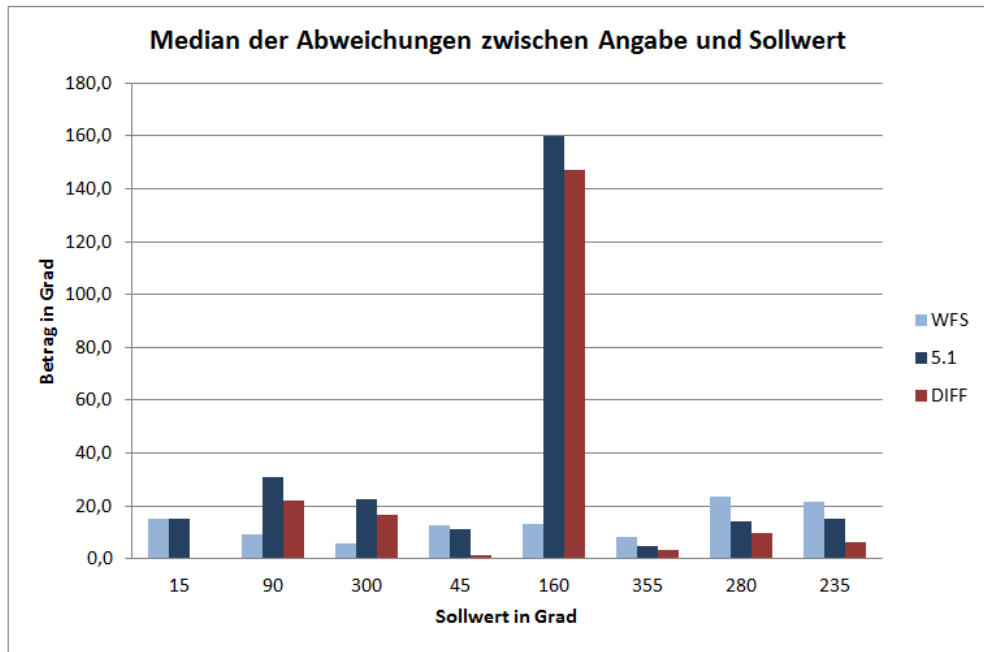


Abbildung 4.6.: Median der Abweichungen aus Test 1.

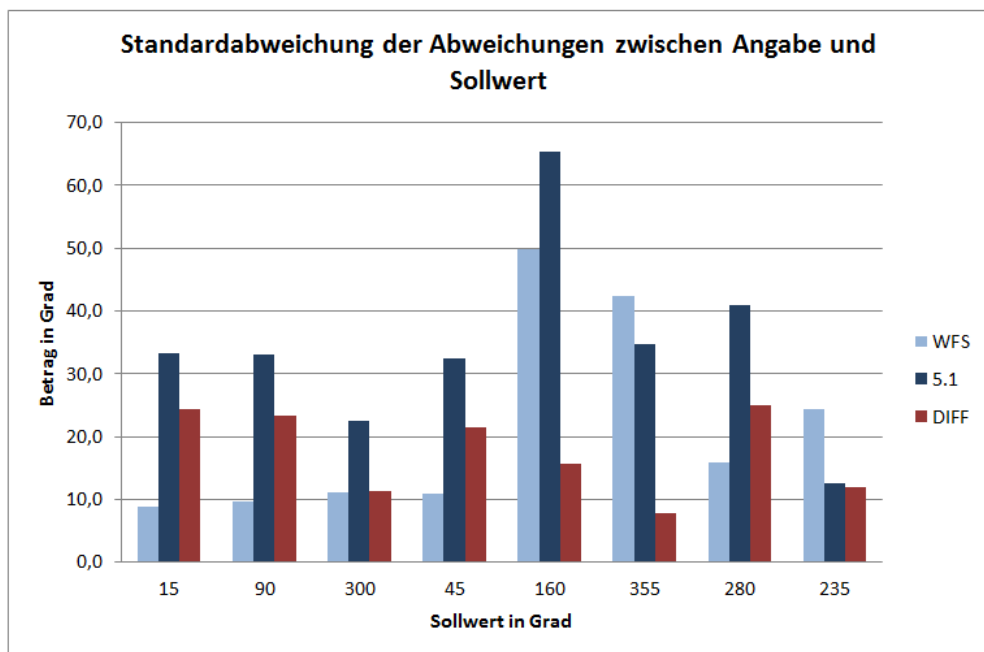


Abbildung 4.7.: Standardabweichungen der Abweichungen aus Test 1.

## Ergebnisse Test 2

In den folgenden Grafiken werden die Ergebnisse aus dem Entfernungstest dargestellt. Die Abbildung 4.8 stellt die Ergebnisse der 3/2/1 Surround-Anlage dar. In Abbildung 4.9 werden die Ergebnisse der WFS-Anlage gezeigt und Abbildung 4.10 fasst die Ergebnisse beider Anlagen zusammen. Die Einzeltests wurden bei den Darstellungen separiert, um die Ergebnisse jedes Entfernungsschrittes aufzuzeigen. Die Reihenfolge entspricht dabei der aufsteigenden Entfernungen innerhalb des Einzeltests. In jedem Abschnitt sind zu jeder Entfernung der Sollwert, das arithmetische Mittel, der Median sowie die Standardabweichung abgebildet. In der zusammenfassenden Darstellung wurden die drei statistischen Merkmale gegenübergestellt. Dabei wurden die Einzeltests, die identisch sind, zusammengefasst. Dieser Zusammenhang besteht jeweils zwischen ET-B1 und ET-B5, ET-B2 und ET-B6, ET-B3 und ET-B7 sowie ET-B4 und ET-B8.

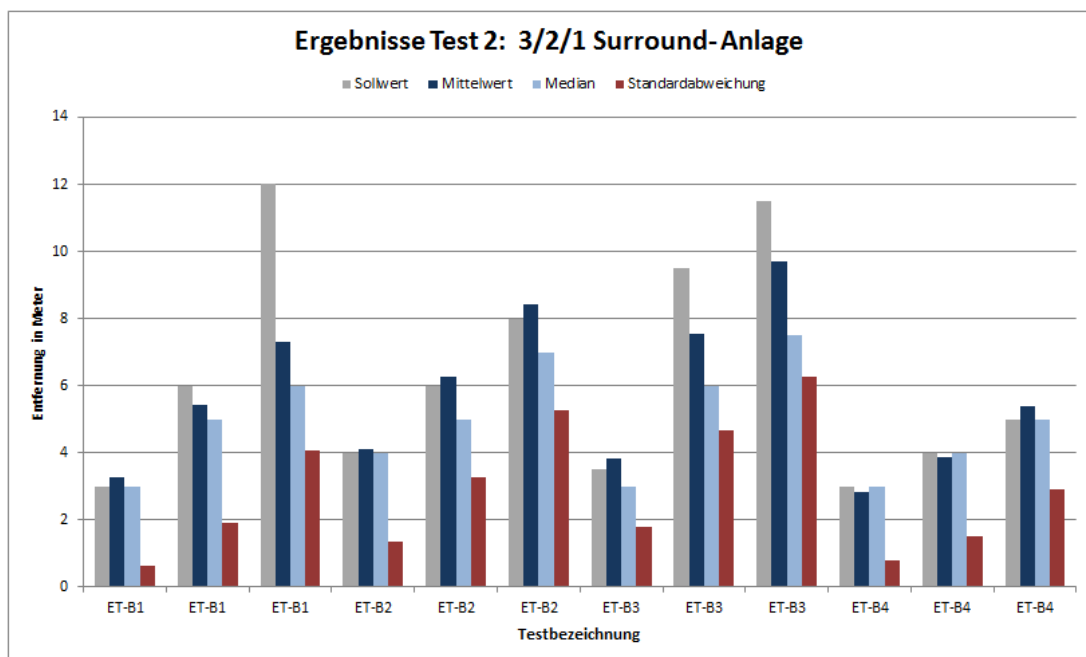


Abbildung 4.8.: Ergebnisse der 3/2/1 Surround-Anlage aus Test 2.

#### 4. Testen der Funktionalität

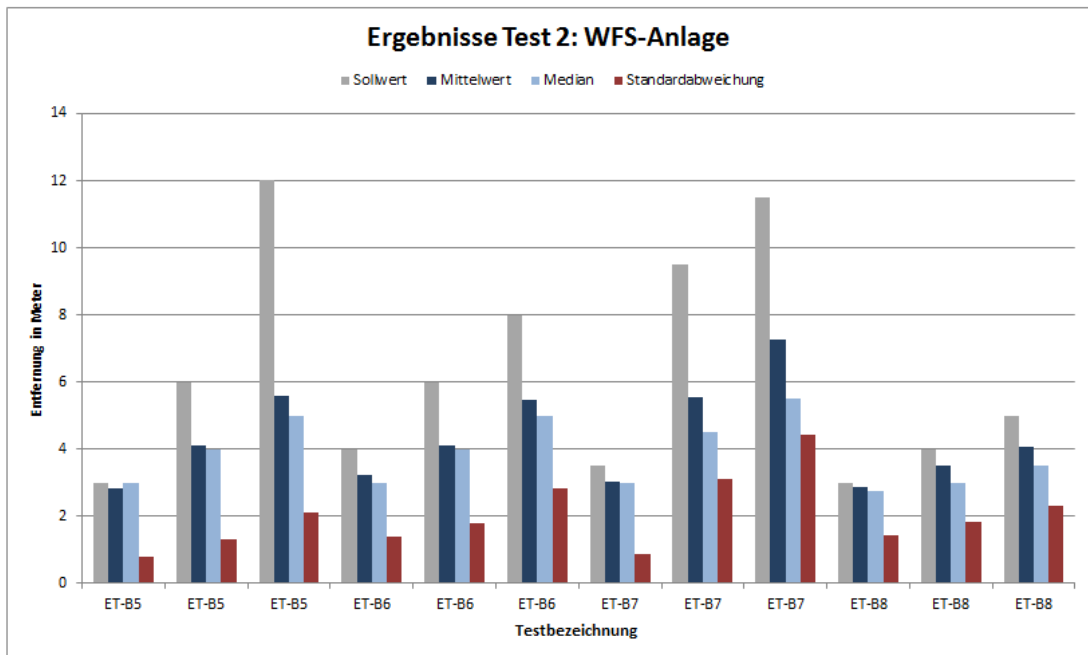


Abbildung 4.9.: Ergebnisse der WFS Surround-Anlage aus Test 2.

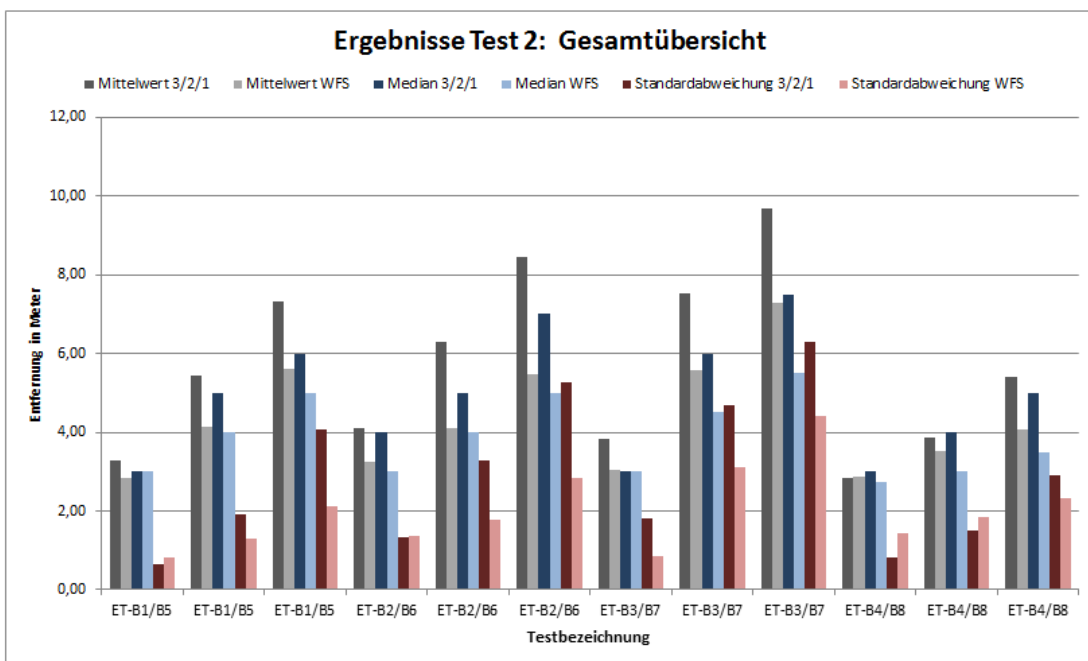


Abbildung 4.10.: Ergebnisse beider Anlagen im Vergleich aus Test 2.

### Ergebnisse Test 3

In der dargestellten Abbildung 4.11 wird das Ergebnis für den Szenenvergleich gezeigt.

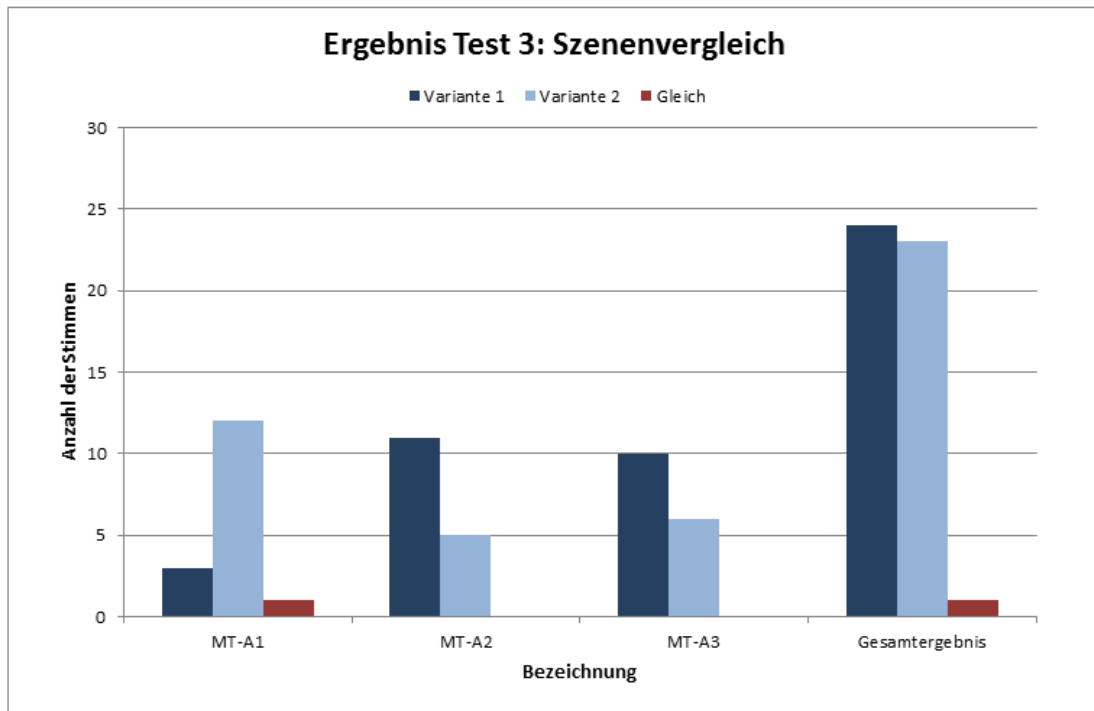


Abbildung 4.11.: Stimmenverteilung beim Szenenvergleich in Test 3.

#### 4.2.3. Auswertung der Ergebnisse

Das verfolgte Ziel des Hörtests war es zu belegen, dass die Wahrnehmung auf beiden Anlagen zu einem gleichen Ergebnis kommt. Betrachtet man die Ergebnisse der Einzeltests, zeichnet sich genau das ab. Lediglich in einzelnen Fällen wird dem widersprochen, worauf in der Einzelauswertung näher eingegangen wird.

#### Auswertung Test 1: Richtungslokalisierung

Die hier erhobenen Testdaten zeigen in Abbildung 4.3, dass die Wahrnehmung zwischen beiden Referenzanlagen gleich ist. Die Mittelwerte sind bei fast allen Werten stabil. Auch die Werte, die optisch den Anschein größerer Unterschiede erwecken, sind ebenfalls stabil. Betrachtet man die Werte  $15^\circ$  und  $355^\circ$  und deren absolute Abweichungen in Abbildung 4.4, zeigt sich was einleitend schon beschrieben wurde. Durch die Nähe zum Scheitelpunkt gibt es teils starke



Wertsprünge, die das Ergebnis in dieser Darstellung verfälschen. Der Median ist da schon etwas genauer, wie in Abbildung 4.5 zu sehen ist. Allerdings ist auch hier der Wert für  $355^\circ$  noch „verfälscht“, was aber an dem sehr geringen Abstand zum Scheitelpunkt liegt. Dadurch ist hier die Anzahl an Werte hinter dem Scheitelpunkt höher als es bei  $15^\circ$  der Fall ist.

Viel interessanter sind allerdings die Werte bei  $160^\circ$ . Zunächst ließe sich nicht erwarten, dass es dort im Ergebnis zu starken Unterschieden gekommen ist. Erst die Abweichungen zeigen einen großen Unterschied zwischen beiden Anlagen. Begründen lässt sich die abweichende Ortung auf der 3/2/1 Surround-Anlage, mit den bereits beschriebenen Schwächen und war somit zu erwarten. Die Surroundlautsprecher stehen beinahe auf Ohrhöhe und  $160^\circ$  liegen knapp mittig hinter dem Hörer, dadurch spielen beide Lautsprecher fast gleiche Signalanteile ab. Aufgrund der *Vorne-Hinten-Ambiguität* haben viele Probanden geglaubt, dieser Ton käme von vorne. Unter der Betrachtung dieser Schwäche, belegen die Ergebnisse aus Test 1, dass die Wahrnehmung gleich ist.

#### **Auswertung Test 2: Entfernungslokalisation**

Die Ergebnisse des Tests der Entfernungslokalisation waren sehr aufschlussreich. Wie zu erwarten, waren die Ergebnisse für große Entfernungen zunehmend unpräzise, aber auf beiden Anlagen wiederum fast gleich. Allerdings waren die Ergebnisse der 3/2/1 Surround-Anlage (Abbildung 4.8) immer etwas dichter am Sollwert als die Ergebnisse der WFS-Anlage (Abbildung 4.9).

Betrachtet man die Gegenüberstellung der Ergebnisse aus Abbildung 4.10 lässt sich erkennen, dass die Lokalisation bis zu einer Entfernung von fünf Metern genau und auf beiden Anlagen fast gleich ist. Erst danach beginnen die Ergebnisse leicht zu divergieren, liegen aber alle in einem noch vertretbaren Rahmen. Das stets bessere Ergebnis der 3/2/1 Surround-Anlage deutet darauf hin, dass die Umsetzung des Abstandsgesetz in WONDER ein wenig von der Umsetzung dieser Arbeit abweicht. Grundsätzlich kommen die Ergebnisse beider Anlagen aus Test 2 aber zu dem Schluss, dass die Wahrnehmung auch hier vergleichbar ist.

#### **Auswertung Test 3: Szenen**

Diese Ergebnisse (Abbildung 4.11) verdeutlichen, dass die aufgestellte Annahme zutrifft. Beide Anlagen liefern ein sehr vergleichbares Gesamtergebnis. Es zeigt vor allem auch, wo die Unterschiede der Anlagen liegen. Der erste Test *MT-A1* stellt ausschließlich bewegte Klangquellen dar. Hier fanden 75% der Probanden die Variante der WFS-Anlage besser, während bei den beiden anderen Tests, bei denen weniger mit Bewegungen gearbeitet wurde, immer die 3/2/1 Surround-Anlage Anlage favorisiert wurde. Das spiegelt sich auch in den schriftlichen

Aussagen der Probanden wieder. Die Entscheidung für die Variante der WFS-Anlage wurde immer mit „Lokalisationsschärfe“ begründet, während die 3/2/1 Surround-Anlage wegen des Klanges favorisiert wurde. Im Nachgespräch haben fast alle Probanden angegeben, dass die Unterschiede nur geringfügig waren und wenn sie gefragt worden wären, sie kaum in der Lage gewesen wären, zu sagen welche Anlage welche Variante abgespielt hat.

#### 4.2.4. Zusammenfassung

Die Abweichungen bei den Lokalisationstest waren allesamt gering und entstammen den bekannten Schwächen der Lautsprecheraufstellungen für Mehrkanalton-Formate. Der Hintergrund ist also nicht die Konvertierung mit SPAOP-MFC. Der Entfernungstest hat gezeigt, dass die Entfernungslokalisierung von künstlichen Klangquellen in geschlossenen Räumen insgesamt schwierig ist. Auch hier waren die direkten Unterschiede zwischen Mehrkanalton und WFS gering. Der Szenentest hat abschließend deutlich gemacht, dass das Konvertierungsergebnis von vielen verschiedenen Klangquellen ebenso vergleichbar ist wie das Ergebnis mit einzelnen Klangquellen. Zusammenfassend haben alle Testergebnisse gezeigt, dass auf Basis der Positionsdaten der WFS-Anlage im Mehrkanalton positionierte Klangquellen die gleiche Wahrnehmung beim Hörer hervorrufen. Das bestätigt die anfangs getroffene Annahme.

## 5. Fazit / Ausblick

### 5.1. Zusammenfassung

Ziel dieser Arbeit war es, Klangquellen anhand von Positionsdaten einer Wellenfeldsynthese-Anlage in verschiedenen Formaten zu positionieren. Dieses konnte erfolgreich umgesetzt werden. Dazu wurde das bestehende Plugin SPAOP von Martin Hansen [Han14] um einen MultiFormatConverter erweitert, woraus das Plugin SPAOP-MFC entstanden ist. Für die Umsetzung dieses MultiFormatConverters wurden verschiedene Mehrkanalton-Formate diskutiert, mögliche Panning Laws vorgestellt, die Unterschiede zwischen Quell- und Zielformaten untersucht und die Architektur von SPAOP analysiert. Die daraus hervorgehenden Erkenntnisse wurden zu Anforderungen geformt, auf dessen Basis die Implementierung stattgefunden hat. Nach der Fertigstellung wurde das Plugin ausgiebig getestet und in einem Hörtest mit Probanden nachgewiesen, dass das Verfahren funktioniert.

### 5.2. Fazit

Die Erweiterung von SPAOP zu SPAOP-MFC ist ein weiterer wichtiger Schritt zur Verbesserung der Arbeitsumgebung im WFS-Labor. Mit den neuen Möglichkeiten im Plugin können nun Vergleichstests zwischen verschiedenen Anlagentypen durchgeführt werden, ohne Projekte mehrfach erstellen zu müssen. Außerdem ist es jetzt möglich, die für die WFS-Anlage entwickelten Features, wie etwa die *Gestengesteuerte Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines kamerabasierten 3D-Tracking-Systems* von Malte Nogalski [Nog12], auch auf anderen Anlagentypen zu nutzen.

### 5.3. Ausblick

SPAOP-MFC ist eine gute Basis für künftige Erweiterungen. Es bietet sich immer an, den MFC mit anderen sinnvollen Formaten zu ergänzen. Davon losgelöst wurde auch schon diskutiert, das Zielformat anhand einer Datei einzulesen. Dadurch würde die strikte Unterteilung in einzelne Mehrkanalformate wegfallen und mehr Flexibilität erreicht werden. In der Datei

könnten alle für eine Wiedergabe notwendigen Informationen in einem noch festzulegenden Datenformat stehen. Diese Daten könnten sein:

- Wie viele Lautsprecher hat die gelesene Aufstellung?
- Wie sind die einzelnen Positionen der Lautsprecher?
- Welche Winkel haben diese untereinander?
- Informationen über die Aufstellung der Anlage (Radius, Position des Sweet Spot, etc.).

Mit einer generischen Konvertierungsmethode könnte daraus dann die Berechnung in einer Schleife erfolgen. Diese Erweiterung würde einen flexiblen Umgang mit verschiedenen Mehrkanalton-Formaten und Offenheit gegenüber Versuchen abseits standardisierter Formate ermöglichen.

Weitere sinnvolle Erweiterungen, die rein technisch SPAOP betreffen, führt Martin Hansen in seiner Bachelorarbeit auf [\[Han14\]](#).

## **A. Anhang**

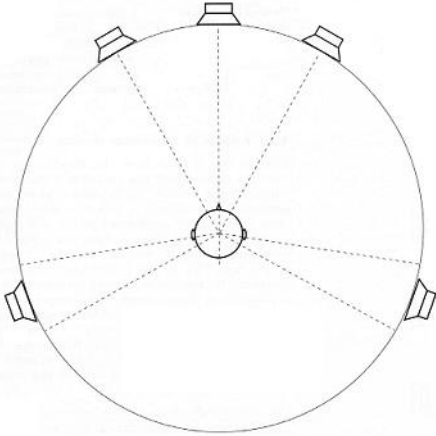
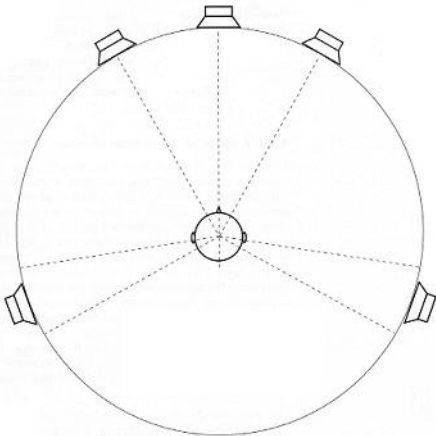
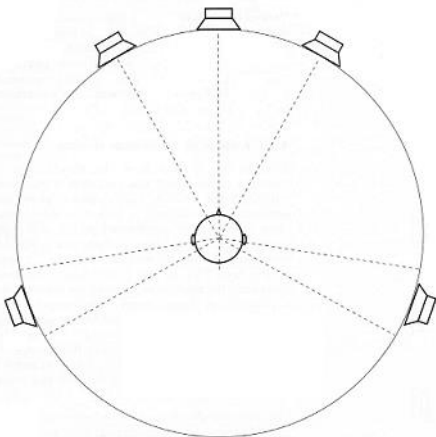
# Vergleichs-Hörprobe zur Raumakustik

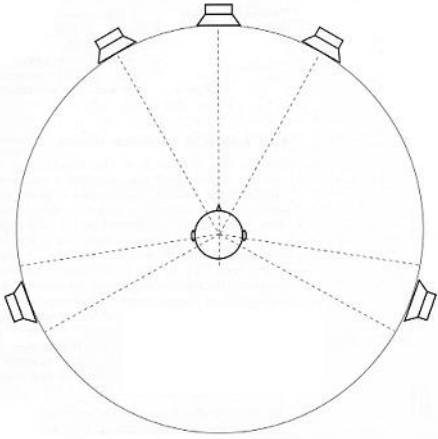
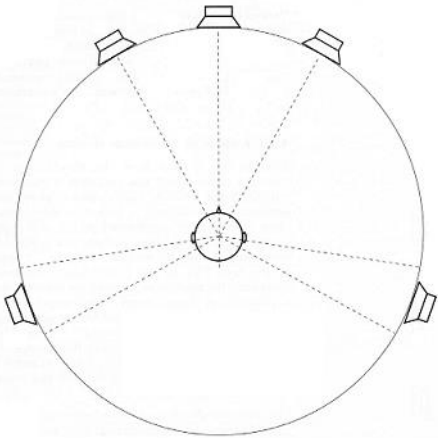
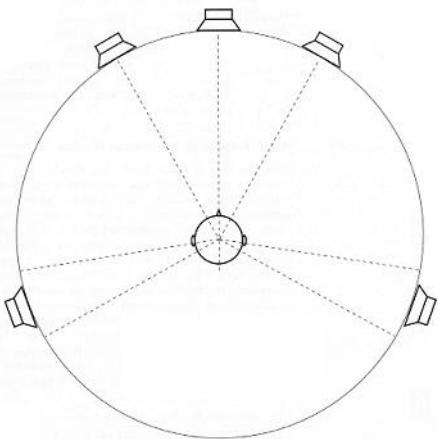
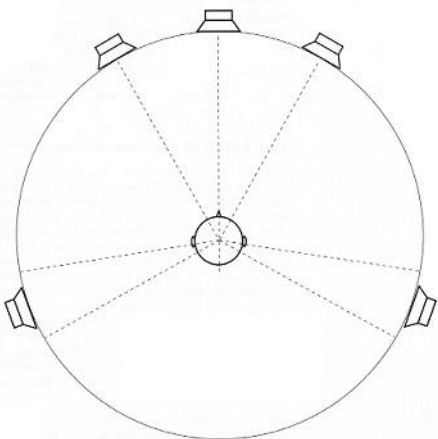
Teil der Bachelorarbeit von Robert Johns

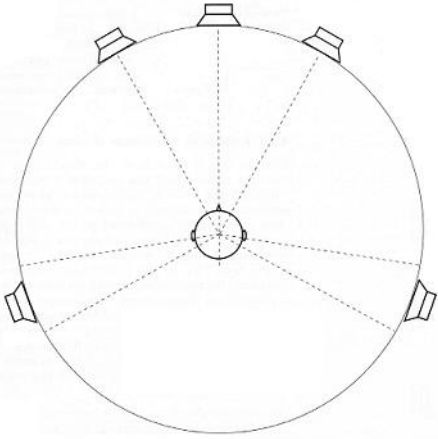
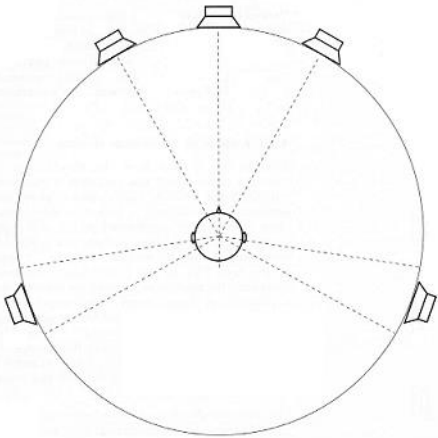
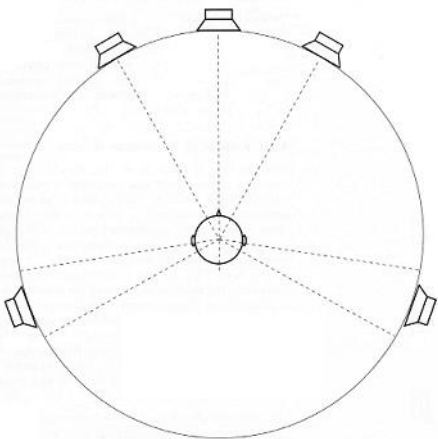
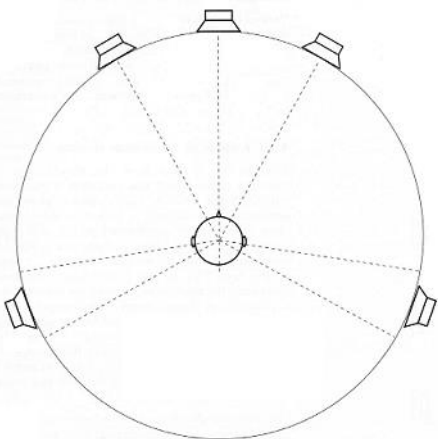
Datum:

Lfd.-Nr.:

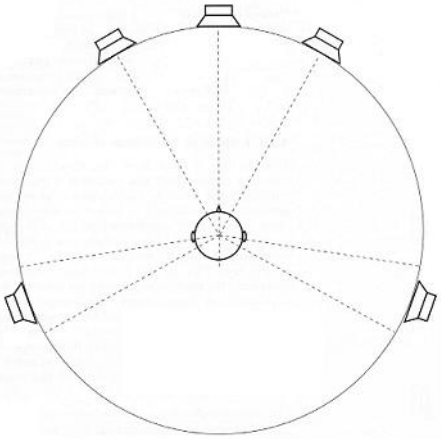
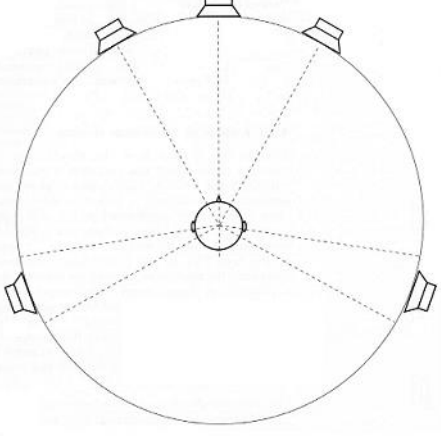
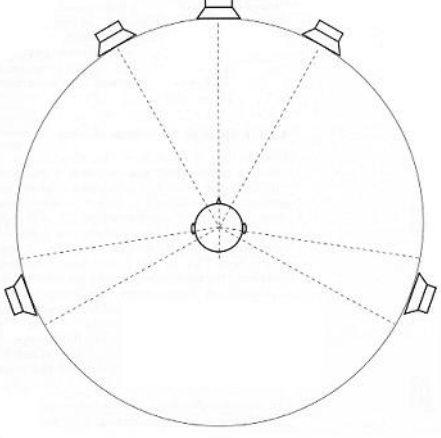
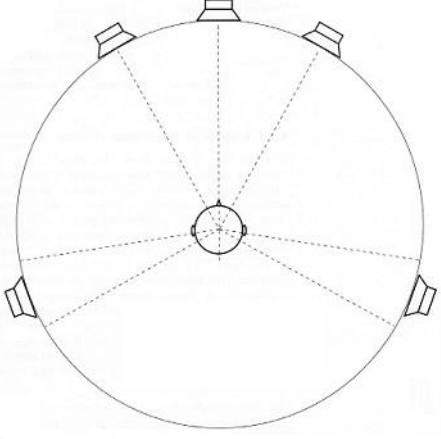
## TEST 1

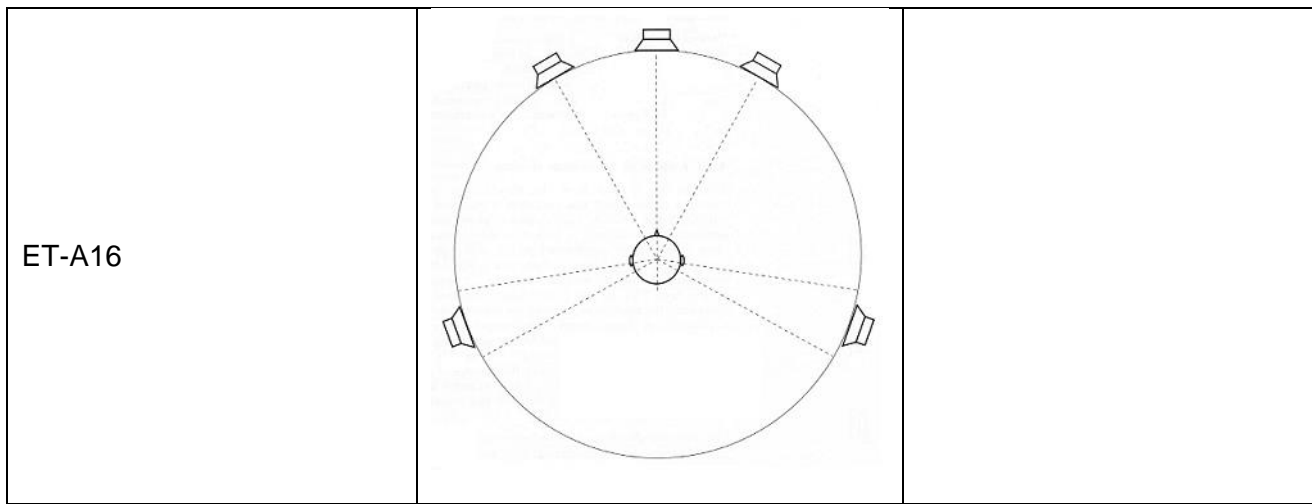
| <b>Test 1<br/>Lokalisation</b> | In diesem Hörtest werden nacheinander einzelne Klangquellen abgespielt. Die Entfernung beträgt 1 und die Klangquelle ist unbewegt. Der Proband soll die Position der Quelle lokalisieren und in dem dafür vorgesehenen Bild einzeichnen. |   |
|--------------------------------|--|---|
| <b>Testbezeichnung</b>         | <b>Wahrgenommene Richtung</b>  | <b>Auswertung</b><br><small>(nicht durch den Probanden auszufüllen)</small> |
| ET-A1                          |   |   |
| ET-A2                          |    |   |
| ET-A3                          |    |   |

|       |  |  |
|-------|--|--|
| ET-A4 |    |  |
| ET-A5 |   |  |
| ET-A6 |  |  |
| ET-A7 |  |  |

|        |  |  |
|--------|--|--|
| ET-A8  |  A technical drawing of a circular component. It features a central circular hole with a smaller concentric hole inside. Six dashed lines radiate from the center to the outer edge, dividing the circle into six sectors. Six trapezoidal tabs are attached to the outer edge, one in each sector. The tabs are positioned at approximately the 12, 2, 4, 6, 8, and 10 o'clock positions.                       |  |
| ET-A9  |  A technical drawing of a circular component, identical to ET-A8. It features a central circular hole with a smaller concentric hole inside. Six dashed lines radiate from the center to the outer edge, dividing the circle into six sectors. Six trapezoidal tabs are attached to the outer edge, one in each sector. The tabs are positioned at approximately the 12, 2, 4, 6, 8, and 10 o'clock positions.  |  |
| ET-A10 |  A technical drawing of a circular component, identical to ET-A8. It features a central circular hole with a smaller concentric hole inside. Six dashed lines radiate from the center to the outer edge, dividing the circle into six sectors. Six trapezoidal tabs are attached to the outer edge, one in each sector. The tabs are positioned at approximately the 12, 2, 4, 6, 8, and 10 o'clock positions. |  |
| ET-A11 |  A technical drawing of a circular component, identical to ET-A8. It features a central circular hole with a smaller concentric hole inside. Six dashed lines radiate from the center to the outer edge, dividing the circle into six sectors. Six trapezoidal tabs are attached to the outer edge, one in each sector. The tabs are positioned at approximately the 12, 2, 4, 6, 8, and 10 o'clock positions. |  |



|        |  |  |
|--------|--|--|
| ET-A12 |  A technical drawing of a circular object, possibly a lid or a cap, shown from a top-down perspective. The object is a large circle with a smaller concentric circle in the center. Six dashed lines radiate from the center to the outer edge, dividing the circle into six equal sectors. On the outer edge, there are six small, trapezoidal tabs, one in each sector. The tabs are positioned at approximately the 12, 2, 4, 6, 8, and 10 o'clock positions. |  |
| ET-A13 |  A technical drawing of a circular object, identical to ET-A12, shown from a top-down perspective. It features a central circle, six radial dashed lines, and six trapezoidal tabs on the outer edge.   |  |
| ET-A14 |  A technical drawing of a circular object, identical to ET-A12, shown from a top-down perspective. It features a central circle, six radial dashed lines, and six trapezoidal tabs on the outer edge.  |  |
| ET-A15 |  A technical drawing of a circular object, identical to ET-A12, shown from a top-down perspective. It features a central circle, six radial dashed lines, and six trapezoidal tabs on the outer edge.  |  |



(Tabelle 1: Test 1 Lokalisation)

## TEST 2

|  |   |   |  |  |  |
|--|---|---|--|--|--|
| <p align="center"><b>Test 2<br/>Entfernungen</b></p> | <p>In diesem Test wird eine Klangquelle nacheinander aus einer Richtung mit unterschiedlichen Entfernungen abgespielt. Die ersten beiden Entfernungen sind immer 1m und 2m damit der Proband eine räumliche Relation zur Klangquelle aufnehmen kann. Die Richtung ist dem Probanden bekannt</p> |   |  |  |  |
| <p align="center"><b>Testbezeichnung</b></p>         | <p align="center"><b>Wahrgenommene Entfernung (in Meter)</b></p>  |   |  |  |  |
| ET-B1 (vorne)  | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B2 (links)  | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B3 (rechts)                                       | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B4 (hinten)                                       | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B5 (vorne)  | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B6 (links)  | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B7 (rechts)                                       | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |
| ET-B8 (hinten)                                       | 1   | 2 |  |  |  |
| Auswertung   |   |   |  |  |  |

(Tabelle 2: Test 2 Entfernungen)

## TEST 3

| <b>Test 3<br/>Bewegte Szenen</b> | In diesem Test werden mehrere Klangquellen gleichzeitig abgespielt. Innerhalb der abgespielten Szene bewegen sich die Klangquellen im Raum. Jeder Test wird hintereinander auf der Wellenfeldsynthese- und der 5.1 Surround-Anlage abgespielt. Es ist nicht bekannt welche Anlage die wiedergebende ist. Der Proband soll entscheiden welche Variante ihm besser gefallen hat und warum. |  |
|----------------------------------|--|--|
| <b>Testbezeichnung</b>           | <b>Welche Variante hat besser gefallen?</b>  | <b>Warum?</b><br>(Stichworte / Kurztext) |
| MT-A1                            | <ul style="list-style-type: none"><li><input type="radio"/> Variante 1</li><li><input type="radio"/> Variante 2</li></ul>  |  |
| MT-A2                            | <ul style="list-style-type: none"><li><input type="radio"/> Variante 1</li><li><input type="radio"/> Variante 2</li></ul>  |  |
| MT-A3                            | <ul style="list-style-type: none"><li><input type="radio"/> Variante 1</li><li><input type="radio"/> Variante 2</li></ul>  |  |

(Tabelle 3: Test 3 Bewegte Szenen)

ENDE

## B. Glossar

**DAW** Digital Audio Workstation, computergestütztes System für Tonaufnahme, Musikproduktion, Abmischung und Mastering.

**GUI** Graphical user interface, grafische Benutzeroberfläche eines Computers.

**ITU** International Telecommunication Union, Sonderorganisation der Vereinten Nationen die sich mit technischen Aspekten der Telekommunikation beschäftigt.

**OSC** Open sound control, Netzwerk-Protokoll zur Kommunikation zwischen Computern und Multimedia-Geräten.

**SDK** Software development Kit, Sammlung von Anwendungen und Werkzeugen für die Softwareherstellung.

**SPAOP** Spatial Audio Object Positioner, Open Source Audioplugin für DAWs um virtuelle Klangquellen einer WFS-Anlage zu steuern.

**VST** Virtual Studio Technology, Software-Protokoll der Firma Steinberg.

# Abbildungsverzeichnis

|   |    |
|---|----|
| 2.1. Kopfbezogenes Koordinatensystem . . . . .  | 4  |
| 2.2. Laufzeitunterschied bei seitlichem Schallereignis . . . . .  | 5  |
| 2.3. Cone of Confusion . . . . .  | 7  |
| 2.4. Ablauf der Synthese einer Schallwelle . . . . .  | 8  |
| 2.5. Skizze der Wellenfeldsyntheseanlage im Labor der HAW . . . . .   | 9  |
| 2.6. Phantomschallquelle zwischen zwei Lautsprechern . . . . .  | 10 |
| 2.7. Stereohörfläche bei Aufstellung im Stereodreieck . . . . .   | 12 |
| 2.8. Aufstellung von drei Lautsprechern im Kreisbogen oder einer Linie . . . . .  | 13 |
| 2.9. Anordnung für 5-Kanal und 7-Kanal Surround nach ITU Recommendation<br>BS.775-3 . . . . .   | 14 |
| 2.10. Zweikanal-Anordnung . . . . .   | 16 |
| 2.11. Lineares Gesetz . . . . .   | 18 |
| 2.12. Tangens-Gesetz . . . . .  | 19 |
| 2.13. Screenshot der DAW Software Cubase 7.5 . . . . .  | 21 |
|   |    |
| 3.1. 5.1 Surround Panner . . . . .  | 23 |
| 3.2. Benutzeroberfläche von SPAOP . . . . .   | 24 |
| 3.3. Mapping der Klangquellen auf die Frontlautsprecher . . . . .   | 28 |
| 3.4. Architektur von SPAOP . . . . .  | 29 |
| 3.5. Architektur von SPAOP-MFC . . . . .  | 30 |
| 3.6. Dargestellt ist die Beziehung zwischen Funktionszeiger und dessen Aufruf über<br>die Methode processingToSelectedFormat. . . . .             | 33 |
| 3.7. Ablauf der Übersetzung eines Audioblocks. . . . .  | 34 |
| 3.8. Verschiebung der Klangquelle auf der Y-Achse um den Betrag des Sweet Spots.<br>Links ist die Darstellung ohne Korrektur, rechts mit. . . . . | 35 |
| 3.9. Berechnung der Polarkoordinate $\varphi$ anhand der kartesischen Koordinaten $x$ und $y$ . . . . .   | 35 |
| 3.10. Fünfkanal-Anordnung . . . . .   | 37 |
| 3.11. Abnahme des Schalldrucks zur Entfernung . . . . .   | 39 |

|  |    |
|--|----|
| 3.12. Linearquelle . . . . .                                     | 40 |
| 3.13. Vereinfachtes Klassendiagramm von SPAOP-MFC. . . . .       | 43 |
| 3.14. Mockup der grafischen Erweiterung im Introjucer . . . . .  | 44 |
| 4.1. Audio-Arbeitsplatz im WFS-Labor der HAW. . . . .            | 50 |
| 4.2. Aufbau des Hörtests im WFS-Labor der HAW. . . . .           | 51 |
| 4.3. Mittelwert der Angaben aus Test 1. . . . .                  | 54 |
| 4.4. Mittelwert der Abweichungen aus Test 1. . . . .             | 55 |
| 4.5. Median der Angaben aus Test 1. . . . .                      | 55 |
| 4.6. Median der Abweichungen aus Test 1. . . . .                 | 56 |
| 4.7. Standardabweichungen der Abweichungen aus Test 1. . . . .   | 56 |
| 4.8. Ergebnisse der 3/2/1 Surround-Anlage aus Test 2. . . . .    | 57 |
| 4.9. Ergebnisse der WFS Surround-Anlage aus Test 2. . . . .      | 58 |
| 4.10. Ergebnisse beider Anlagen im Vergleich aus Test 2. . . . . | 58 |
| 4.11. Stimmenverteilung beim Szenenvergleich in Test 3. . . . .  | 59 |

# Tabellenverzeichnis

|   |    |
|---|----|
| 3.1. Globale Einstellungen des SPAOP-MFC . . . . .                                | 27 |
| 3.2. Formeln für die Berechnung der Kanalanteile bei den Formaten 3/0/0 und 3/0/1 | 38 |
| 3.3. Formeln für die Berechnung der Kanalanteile bei den Formaten 3/2/0 und 3/2/1 | 38 |
| 3.4. Bedingungen für die Konvertierung . . . . .                                  | 46 |
| 4.1. Einzeltests von Test 1 . . . . .   | 52 |
| 4.2. Einzeltests von Test 2 . . . . .   | 53 |
| 4.3. Einzeltests von Test 2 . . . . .   | 53 |

## Literaturverzeichnis

- [Ber88] BERKHOUT, Augustinus J.: A Holographic Approach to Acoustic Control. In: *J. Audio Eng. Soc* 36 (1988), Nr. 12, S. 977–995. – online verfügbar unter <http://www.aes.org/e-lib/browse.cfm?elib=5117> - Abruf: 26.10.14
- [BH89] BOYE, Günther ; HERRMANN, Urbi F.: *Handbuch der Elektroakustik – Technik und praktische Anwendung*. 3. überarbeitete und erweiterte Auflage. Heidelberg : Hüthig Buch Verlag, 1989. – ISBN 3–7785–1575–6. – Bis 2. Aufl. u.d.T.: Herrmann, Urbi F.: *Handbuch der Elektroakustik*
- [Bla74] BLAUERT, Jens: *Räumliches Hören*. Stuttgart : Hirzel, 1974 (Monographien der Nachrichtentechnik). – ISBN 978–3777602509
- [FHB<sup>+</sup>14] FÜG, Simone ; HÖLZER, Andreas ; BORSS, Christian ; ERTEL, Christian ; KRATSCHEMER, Michael ; PLOGSTIES, Jan: Design, Coding and Processing of Metadata for Object-Based Interactive Audio. In: *Audio Engineering Society Convention 137*, 2014
- [Foh13] FOHL, Wolfgang: The Wave Field Synthesis Lab at the HAW Hamburg. In: BADER, Rolf (Hrsg.): *Sound - Perception - Performance*. Heidelberg : Springer International Publishing, 2013 (Current Research in Systematic Musicology Vol. 1). – ISBN 978–3–319–00106–7, S. 243–255. – Online verfügbar unter: [http://link.springer.com/chapter/10.1007/978-3-319-00107-4\\_10](http://link.springer.com/chapter/10.1007/978-3-319-00107-4_10)- Abruf: 2014-06-20
- [Gri96] GRIGOLEIT, Stefan: *Mehrkanaltonsysteme unter besonderer Berücksichtigung von Dolby Pro Logic*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 1996
- [Gör08] GÖRNE, Thomas: *Tontechnik*. 2. aktualisierte Auflage. München : Hanser, 2008. – ISBN 978–3–446–41591–1
- [Han14] HANSEN, Martin: *Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines Audio-Plugins*. Hamburg, Hochschule für Angewandte Wissenschaften



- ten Hamburg, Bachelorarbeit, 2014. – online verfügbar unter <http://edoc.sub.uni-hamburg.de/haw/volltexte/2014/2616> - Abruf: 24.10.14
- [HHKP14] HERRE, Jürgen ; HILPERT, Johannes ; KUNTZ, Achim ; PLOGSTIES, Jan: MPEG-H Audio?The New Standard for Universal Spatial / 3D Audio Coding. In: *Audio Engineering Society Convention 137*, 2014
- [ITU09] ITU, Radiocommunication S.: Report ITU-R BS.2159: Multichannel sound technology in home and broadcasting applications / International Telecommunication Union. 2009. – Forschungsbericht
- [ITU12] ITU, Radiocommunication S.: Recommendation BS.775-3: Multichannel stereophonic sound system with and without accompanying picture / International Telecommunication Union. 2012. – Forschungsbericht
- [Kre14] KREMER, Martina: *Ars Auditis: Grundlagen der Akustik*. 2014. – online verfügbar unter [http://web.fbe.uni-wuppertal.de/fbe0014/ars\\_auditus/ueberfunktion/funk24.htm](http://web.fbe.uni-wuppertal.de/fbe0014/ars_auditus/ueberfunktion/funk24.htm) - Abruf: 24.10.14
- [Nog12] NOGALSKI, Malte: *Gestengesteuerte Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines kamerabasierten 3D-Tracking-Systems*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2012. – online verfügbar unter <https://users.informatik.haw-hamburg.de/ubi-comp/arbeiten/bachelor/nogalski.pdf> - Abruf: 04.11.14
- [Oel13] OELLERS, Helmut: *Holophony - the path to true spatial audio reproduction : Wave Field Synthesis*. 2013. – online verfügbar unter <http://www.syntheticwave.de/Wavefieldsynthesis.htm> - Abruf: 19.11.14
- [Pau10] PAUSER, Martin: *Ambisonic Decoding für reguläre Lautsprecheraufstellungen und Möglichkeiten der Optimierung für die 5.1 Lautsprecheraufstellungen nach dem ITU Standard*. Österreich, Universität für Musik und Darstellende Kunst Graz, Diss., 2010. – online verfügbar unter <http://iem.kug.ac.at/fileadmin/media/iem/projects/2010/pauser.pdf> - Abruf: 26.10.14
- [Sau05] SAUERLAND, Frank: *Positionierung von Schallquellen in Mehrkanal-Audiosystemen: Grundlagenuntersuchung und Anwendungsentwicklung*. Hamburg, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2005

- [The92] THEILE, Günther: The New Sound Format "3/2-Stereo". In: *Vortrag 17. Tonmeister-tagung* (1992), Nr. 92023
- [Web07] WEBERS, Johannes: *Handbuch der Tonstudioteknik für Film, Funk und Fernsehen*. Poing : Franzis, 2007 (Franzis Film- und Videotechnik). – ISBN 978-3-7723-5429-8
- [Wei08] WEINZIERL, Stefan: *Handbuch der Audiotechnik*. Berlin : Springer, 2008 (VDI-Buch). – ISBN 978-3-540-34300-4
- [Wes98] WEST, James R.: *Five-Channel Panning Laws: An Analytical and Experimental Comparison*. Miami, FL, University of Miami, Masterarbeit, 1998. – online verfügbar unter <http://www.music.miami.edu/programs/mue/Research/jwest/Master.html> - Abruf: 23.10.14
- [Zim04] ZIMMERMANN, Angela: *Eigenschaften des Richtungshörens beim Menschen*. 2004. – online verfügbar unter <http://www.informatik.uni-ulm.de/ni/Lehre/SS04/HSSH/pdfs/richtungshoeren1.pdf> - Abruf: 19.11.14

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 15. Dezember 2014

---

Robert Johns