

## Master-Thesis

zur Erlangung des akademischen Grades M.A.

Luka Fehrmann

Matr.-Nr. XXXXXXXXXX

# Entwicklung und Optimierung eines binauralen Faltungs-Plugins zur Simulation einer mehrkanaligen Lautsprecheranordnung im Kopfhörer-Mix

Erstprüfer: Thomas Görne

Zweitprüfer: Stefan Troschka

Eingereicht am 10. Oktober 2025

# Kurzfassung

Die realistische räumlich Wiedergabe von Audiosignalen über Kopfhörer erfordert Verfahren, die die akustischen Eigenschaften realer Lautsprecherumgebungen präzise simulieren. Ziel dieser Arbeit ist die Entwicklung eines binauralen Faltungs-Plugins, das monaurale Raumimpulsantworten in Kombination mit Head-Related Transfer Functions (HRTFs) zu einem immersiven Kopfhörer-Mix verarbeitet.

Das unter dem Namen *HAW Binaural Lab* entwickelte Audio-Plugin integriert eine modulare Audio-Engine auf Basis des JUCE-Frameworks, welche die Faltung von Eingangssignalen mit gemessenen Raumimpulsantworten und deren anschließende Binauralisierung unter Verwendung externer C++-Bibliotheken ermöglicht. Für die Untersuchung wurden zwei Studioumgebungen mit unterschiedlicher Lautsprecheranordnung erfasst: ein Studio mit drei Lautsprecherpaaren sowie ein 33.2-Lautsprecher-Array für Ambisonics-Anwendungen. Die Lautsprecher-Layouts und zugehörigen Parameter wie Position, Gain und Delay werden über externe JSON-Dateien definiert, wodurch das Plugin flexibel erweiterbar bleibt.

Zur Optimierung wurden Effizienz, Latenz und Klangkonsistenz in verschiedenen Konfigurationen analysiert. Die Ergebnisse zeigen, dass die Signalverarbeitung auch bei komplexen Mehrkanal-Layouts stabil arbeitet und eine konsistente binaurale Wiedergabe gewährleistet.

Insgesamt bestätigt die Arbeit, dass das entwickelte Plugin eine praxisnahe und leicht erweiterbare Lösung zur Simulation mehrkanaliger Lautsprecheranordnungen über Kopfhörer bietet. Es schafft die Grundlage für zukünftige Erweiterungen wie Head-Tracking, individuelle HRTF-Profile und zusätzliche Raumkonfigurationen.

# Abstract

The realistic spatial reproduction of audio signals over headphones requires techniques that accurately simulate the acoustic characteristics of real loudspeaker environments. The objective of this thesis is the development of a binaural convolution plugin that processes monaural room impulse responses in combination with head-related transfer functions (HRTFs) to create an immersive headphone mix.

As part of this work, the *HAW Binaural Lab* audio plugin was developed, integrating a modular audio engine based on the JUCE framework. It enables the convolution of input signals with measured room impulse responses and their subsequent binaural rendering using external C++ libraries. Two studio environments with different loudspeaker configurations were captured for evaluation: one studio equipped with three stereo loudspeaker pairs and another with a 33.2 loudspeaker array designed for Ambisonics applications. The loudspeaker layouts and associated parameters such as position, gain, and delay are defined via external JSON files, allowing for flexible extension without code modification.

Optimization focused on efficiency, latency, and acoustic consistency across configurations. The results demonstrate that the signal processing pipeline remains stable even for complex multichannel setups and ensures consistent binaural rendering quality.

Overall, the study confirms that the developed plugin provides a practical and easily extendable solution for simulating multichannel loudspeaker setups over headphones. It establishes a foundation for future extensions such as head tracking, personalized HRTF profiles, and additional room configurations.

# Danksagung

Mein besonderer Dank gilt meinen Betreuern Thomas Görne und Stefan Troschka für die Begleitung dieser Arbeit sowie für die freie und offene Gestaltung des Studiengangs. Ebenso danke ich für die aufschlussreichen und inspirierenden Gespräche, die meinen Studienverlauf bereichert haben.

Ich danke den Entwicklerinnen und Entwicklern des SONICOM-Projekts für die Bereitstellung der *Binaural Rendering Toolbox* sowie den Forschenden, die die verwendeten HRTF- und HpTF-Datensätze erstellt und zugänglich gemacht haben.

Darüber hinaus möchte ich mich herzlich bei meiner Familie und meiner Patentante für ihre bedingungslose Unterstützung und Liebe bedanken – ohne diesen Rückhalt hätte ich die Arbeit nicht in dieser Form umsetzen können.

# Lizenz

Diese Arbeit steht unter der Lizenz **Creative Commons Namensnennung – Nicht-kommerziell 4.0 International (CC BY-NC 4.0)**.

Sie dürfen:

- das Material in jedwedem Format oder Medium vervielfältigen und weiterverbreiten,
- das Material remixen, verändern und darauf aufbauen.

Unter folgenden Bedingungen:

- **Namensnennung** – Sie müssen angemessene Urheber- und Rechteangaben machen, einen Link zur Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.
- **Nicht kommerziell** – Sie dürfen das Material nicht für kommerzielle Zwecke nutzen.

Weitere Informationen zur Lizenz finden Sie unter:

<https://creativecommons.org/licenses/by-nc/4.0/deed.de>



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Motivation</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>3</b>
2.1	Impulsantwort . . . . .	3
2.2	Faltung . . . . .	5
2.3	Räumliche Wahrnehmung . . . . .	9
2.4	Binauralisierung . . . . .	10
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>14</b>
3.1	Anaglyph . . . . .	14
3.2	Personalized Virtual Studio Renderer . . . . .	16
<b>4</b>	<b>Methode und Implementierung</b>	<b>18</b>
4.1	Überblick über die Entwicklungsumgebung . . . . .	21
4.2	Aufbau des Plugins und Signalkette . . . . .	23
4.3	Effizienz und Latenz-Reduktion . . . . .	29
4.4	Erweiterbarkeit: Einbindung neuer Räume . . . . .	30
<b>5</b>	<b>Evaluation der Funktionsweise</b>	<b>31</b>
5.1	Faltung mit Impulsantworten . . . . .	31
5.2	Binauralisierung . . . . .	32
5.2.1	Interaurale Laufzeitdifferenz (ITD) . . . . .	32
5.2.2	Magnitude-Spektren . . . . .	33
5.3	Analyse des Debug-Logs . . . . .	33
5.4	Auditiver Funktionstest . . . . .	34

<b>6</b>	<b>Diskussion</b>	<b>35</b>
6.1	Bewertung des erreichten Funktionsumfangs . . . . .	35
6.2	Einschränkungen und Optimierungspotenzial . . . . .	36
6.3	Vergleich mit verwandten Arbeiten . . . . .	37
6.4	Ausblick auf mögliche Erweiterungen . . . . .	39
<b>7</b>	<b>Fazit</b>	<b>41</b>
	<b>Anhang</b>	<b>43</b>
A	Messung und Vorbereitung von Impulsantworten . . . . .	43
B	<i>HAW Binaural Lab</i> : Nutzung und Erweiterung . . . . .	50
C	Magnitude-Spektren . . . . .	57
	<b>Literatur</b>	<b>58</b>

# 1 Einleitung und Motivation

Die zunehmende Bedeutung immersiver Audioproduktion hat in den vergangenen Jahren dazu geführt, dass räumliche Wiedergabeformate nicht mehr ausschließlich in spezialisierten Tonstudios, sondern auch in kleineren Produktionsumgebungen eine Rolle spielen. Binaurale Audiowiedergabe, die mit handelsüblichen Kopfhörern ein räumliches Hörerlebnis ermöglicht, stellt hierbei eine besonders relevante Technologie dar. Sie erlaubt es, mehrkanalige Lautsprecher-Setups virtuell zu simulieren und somit unabhängig von räumlichen oder finanziellen Einschränkungen mit komplexen Wiedergabeformaten zu arbeiten.

In der praktischen Musik- und Audioproduktion ergibt sich häufig das Problem, dass ein Mix oder eine Produktion zwar für mehrkanalige Lautsprecheranordnungen (z. B. 5.1, 7.1.4 oder spezielle 3D-Setups) vorgesehen ist, jedoch nicht kontinuierlich in einem entsprechend ausgestatteten Studio überwacht werden kann. Insbesondere Studierende, kleinere Produktionsfirmen und unabhängige Produzenten verfügen oft nicht über den Zugang zu aufwendigen Abhörumgebungen. Kopfhörer sind dagegen leicht verfügbar, bieten aber ohne zusätzliche Signalverarbeitung keine überzeugende räumliche Wahrnehmung. Daraus ergibt sich die Motivation, Werkzeuge zu entwickeln, die eine realistische Simulation von Lautsprecheranordnungen im Kopfhörer-Mix ermöglichen.

Ziel der vorliegenden Arbeit ist die Entwicklung und Optimierung eines binauralen Faltungs-Plugins, das es erlaubt, gemessene Raumimpulsantworten (RIRs) und Head-Related Transfer Functions (HRTFs) in einer Digital Audio Workstation (DAW) zu kombinieren. Auf diese Weise soll die akustische Wirkung realer Lautsprecheranordnungen über Kopfhörer erfahrbar gemacht werden. Ein besonderes Augenmerk liegt auf einer möglichst transparenten und modularen Implementie-

rung, die es auch Anwendern ohne tiefere Programmierkenntnisse ermöglicht, das Plugin für andere Räume und Lautsprecher-Layouts zu erweitern. Daraus ergibt sich die zentrale Forschungsfrage dieser Arbeit:

**Wie kann ein binaurales Faltungs-Plugin gestaltet werden, das eine realistische und erweiterbare Simulation mehrkanaliger Lautsprecheranordnungen im Kopfhörer-Mix ermöglicht, ohne dass Anwender auf den Quellcode zugreifen müssen?**

Die Arbeit ist dabei klar abgegrenzt: Sie fokussiert sich auf die Implementierung eines funktionsfähigen, dokumentierten Software-Prototyps unter Verwendung des JUCE-Frameworks und externer C++-Bibliotheken. Die Evaluierung erfolgt primär über technische Tests und eine nachvollziehbare Darstellung der Signalverarbeitungskette. Subjektive Hörtests mit Probanden, die Entwicklung neuer Faltungs- oder Lokalisationsalgorithmen sowie ein umfassender Vergleich unterschiedlicher HRTF-Datenbanken sind hingegen nicht Gegenstand dieser Arbeit.

In Kapitel 2 werden für die Arbeit relevante theoretische Grundlagen erklärt, gefolgt von einer Vorstellung verwandter Arbeiten und bestehender Ansätze in Kapitel 3, die sich mit binauraler Audiowiedergabe und der Simulation von Lautsprecheranordnungen auf Kopfhörern befassen. Dadurch wird der aktuelle Stand der Technik umrissen und die Einordnung der vorliegenden Arbeit in den wissenschaftlichen Kontext ermöglicht. In Kapitel 4 wird das Vorgehen und die Umsetzung des Plugins erläutert, gefolgt von einer technischen Evaluation der Funktionsweise in Kapitel 5. Ein Bewertung und Diskussion der Ergebnisse erfolgt in Kapitel 6 mit einer anschließenden Zusammenfassung in Kapitel 7.

## 2 Theoretische Grundlagen

In diesem Kapitel werden die für diese Arbeit relevanten theoretischen Grundlagen der Akustik und digitalen Signalverarbeitung erläutert.

### 2.1 Impulsantwort

Eine Impulsantwort  $h(t)$  beschreibt das zeitabhängige Verhalten eines akustischen Systems als Übertragungsfunktion zwischen einer Schallquelle und einem Empfänger. Unter der Annahme von Linearität und Zeitinvarianz (LTI) genügt die Impulsantwort, um das System im Zeitbereich vollständig zu charakterisieren (Möser, 2012, 39 f.). In reflexionsfreien Umgebungen spricht man von einer Freifeldimpulsantwort, während man in räumlich begrenzten und reflektierenden Umgebungen von einer *Raumimpulsantwort* (engl. Room Impulse Responses, kurz: *RIRs*) spricht. Letztere ist insbesondere in der Raumakustik von zentraler Bedeutung, da sie wesentliche Eigenschaften der Schallausbreitung wie Laufzeit, Reflexionsstruktur, Nachhall und frequenzabhängige Dämpfung abbildet (Ahnert & Weinzierl, 2025, S. 215).

Mathematisch stellt die Übertragungsfunktion  $H(\omega)$  die Fourier-Transformierte der Impulsantwort dar:

$$H(\omega) = \int_{-\infty}^{\infty} h(t) \cdot e^{-j\omega t} dt \quad (2.1)$$

Sie liefert sowohl den Betrag  $|H(\omega)|$  als auch den Phasenverlauf von  $h(t)$ . Damit werden die Verstärkungs- und Dämpfungscharakteristika eines Systems im Frequenzbereich sichtbar, was eine zentrale Grundlage für die akustische Analyse und

die digitale Signalverarbeitung darstellt (Weinzierl, 2025, S. 19).

### Struktur der Raumimpulsantwort

Die zeitliche Abfolge einer RIR lässt sich in charakteristische Bereiche gliedern (Vorländer & Feistel, 2025, S. 292):

- **Direktschall:** Der erste Signalanteil, der ohne Reflexion am Empfänger eintrifft. Er bestimmt die Laufzeit zwischen Quelle und Empfänger und dient als Referenz für nachfolgende Schallanteile.
- **Frühe Reflexionen:** Kurz nach dem Direktschall eintreffende Signalanteile, die an Oberflächen reflektiert werden. Sie sind für die räumliche Wahrnehmung von großer Bedeutung, da sie Richtungseindrücke und Klarheit des Signals unterstützen (Fuchs, 2017, S. 185).
- **Später Nachhall:** Ab einer bestimmten Zeit überlagern sich so viele Reflexionen, dass sie nicht mehr einzeln unterscheidbar sind. Es entsteht ein diffuser, exponentiell abklingender Schallanteil, dessen Dauer in Form der Nachhallzeit (z. B.  $T_{30}$ ) beschrieben wird. Der Nachhall prägt maßgeblich die akustische Signatur eines Raumes.

Zusätzlich können aus dem Nachhall herausragende Pegelspitzen als wahrnehmbare Echos mit Verzögerungen  $> 50$  ms auftreten (Fuchs, 2017, S. 185). Abbildung 2.1 zeigt die Aufteilung einer Impulsantwort schematisch.

### Messung und Modellierung

Die Gewinnung von Raumimpulsantworten erfolgt typischerweise durch Anregung mit breitbandigen Testsignalen wie etwa exponentiellen Sinus-Sweeps und anschließender Dekonvolution (Möser, 2012, S. 40). Alternativ lassen sich Reflexionen geometrisch berechnen, etwa durch das Image-Source-Verfahren, bei dem Spiegelquellen an reflektierenden Flächen konstruiert werden (Borish, 1984). Bei diesen Verfahren wird jedoch oft ein idealisiertes Raum-Modell genutzt und der Einfluss

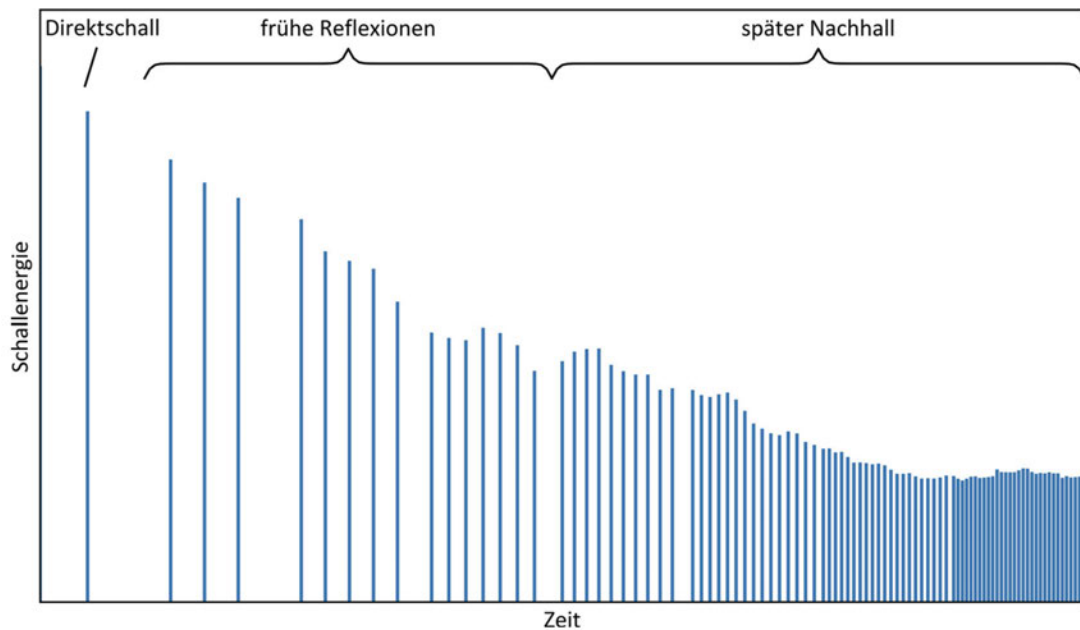


Abbildung 2.1. Typische Energie-Impulsantwort eines Raums mit Einteilung in a) Direktschall, b) frühe Reflexionen und c) Nachhall (Vorländer & Feistel, 2025, S. 292).

kleiner geometrischer Abweichungen wird nicht berücksichtigt (Kuttruff, 1995). RIRs hingegen bilden die akustischen Eigenschaften eines Raumes hinsichtlich der gemessenen Übertragungstrecke zwischen Quelle und Empfänger präzise ab (Ahnert & Weinzierl, 2025, S. 215). Neben Quelle- und Empfängerposition sind sie jedoch auch abhängig von der Raumgeometrie, sowie den spektralen Eigenschaften und der Richtwirkung von Schallquelle und Empfänger (Fuchs, 2017, S. 186).

## 2.2 Faltung

Die *Faltung* (engl. convolution) ist eine der zentralen Operationen in der Signalverarbeitung. Sie beschreibt die Reaktion eines linearen, zeitinvarianten Systems (LTI) auf ein Eingangssignal und kann anschaulich als Überlagerung zeitlich verschobener und gewichteter Kopien des Eingangssignals interpretiert werden. Die Gewichtungsfaktoren entsprechen den Abtastwerten der Impulsantwort (Bartetzki,

2025, S. 161).

Ein System wird als linear und zeitinvariant bezeichnet, wenn es das Superpositionsprinzip erfüllt und eine zeitliche Verschiebung des Eingangssignals  $x(t)$  ausschließlich zu derselben Verschiebung des Ausgangssignals  $y(t)$  führt (Mertins, 1996, S. 47).

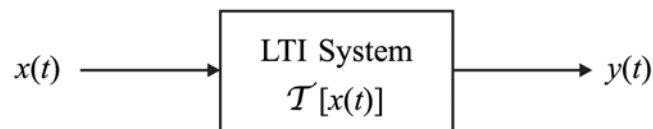


Abbildung 2.2. Transformation eines Signals  $x(t)$  mit dem Operator  $\mathcal{T}$  durch ein LTI-System (Mertins, 1996, S. 47).

In der Signaltheorie wird zudem zwischen zeitkontinuierlichen Signalen  $x(t)$  und zeitdiskreten Signalen  $x[n]$  unterschieden. Erstere beschreiben analoge Größen, wie elektrische Spannungen oder Druckverläufe von Schallwellen. Zeitdiskrete Signale entstehen hingegen durch Abtastung und werden zur Beschreibung digitaler Systeme genutzt und bilden daher die Grundlage für praktische Implementierungen in digitalen Audio-Plugins (Zölzer & Lerch, 2025, S. 641).

### Faltungsintegral

Wird ein LTI-System mit einem zeitkontinuierlichen Eingangssignal  $x(t)$  gespeist, so ergibt sich das Ausgangssignal  $y(t)$  als Faltung von  $x(t)$  mit der Impulsantwort  $h(t)$ :

$$y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau, \quad (2.2)$$

wobei  $*$  die Faltungsoperation bezeichnet (Vorländer & Feistel, 2025, S. 306). Eine besondere Rolle spielt dabei der Dirac-Impuls  $\delta(t)$ <sup>1</sup>. Wird ein System mit  $\delta(t)$  angeregt, so ist die Antwort unmittelbar die Impulsantwort  $h(t)$  dieses Systems (Vorländer, 2018, 3 f.).

---

<sup>1</sup>Auch als Einheitsimpuls bezeichnet

### Faltungssumme

Für digitale Systeme ergibt sich die zeitdiskrete Faltungssumme:

$$y[n] = (x * h)[n] = \sum_{k=0}^{N-1} h[k] x[n - k], \quad (2.3)$$

wobei  $h[n]$  die Impulsantwort und  $N$  ihre Länge bezeichnet. Sie kann als Antwort auf den Einheitsimpuls  $\delta[n]$  bestimmt werden und beschreibt das digitale System vollständig (Zölzer & Lerch, 2025, S. 643). Anschaulich lässt sich das Eingangssignal in Impulssegmente zerlegen, deren gewichtete und verschobene Antworten die Gesamtausgabe bilden (Smith et al., 1997, 108 ff.). Eine Impulsantwort  $h[n]$  mit einer endlichen Länge  $N$ , z.B.  $N = 512$ , wird auch als *FIR-Filter*<sup>2</sup> bezeichnet.

### Faltungstheorem

Faltungsprozesse können sowohl im Zeit- als auch im Frequenzbereich beschrieben werden. Im Zeitbereich werden Signale als Amplitudenverlauf über der Zeit dargestellt, etwa in Form einer Audiosignal-Wellenform. Im Frequenzbereich hingegen erfolgt eine Zerlegung in Sinus- und Kosinus-Schwingungen, die das Spektrum des Signals bilden. Das sogenannte *Faltungstheorem* besagt, dass eine Faltung im Zeitbereich einer Multiplikation im Frequenzbereich entspricht (Vorländer, 2018, 3 f.). Diese Dualität ist wesentlich für die effiziente numerische Umsetzung und lässt sich mathematisch wie folgt ausdrücken:

$$x(n) * h(n) \Leftrightarrow X(e^{j\omega}) H(e^{j\omega}), \quad (2.4)$$

(Mertins, 1996, S. 117).

---

<sup>2</sup>FIR steht für Finite Impulse Response

### Numerische Umsetzung in Audioplugins

Eine direkte Berechnung der Faltung im Zeitbereich ist für lange Impulsantworten, wie sie etwa bei Räumen mit langem Nachhall oder binauralen Raumimpulsantworten auftreten, rechenintensiv. Daher werden in der Praxis schnelle *FFT*<sup>3</sup>-basierte Verfahren eingesetzt. Das Eingangssignal und die Impulsantwort werden in den Frequenzbereich transformiert, dort punktweise multipliziert und durch inverse FFT zurückgeführt. Um lange laufende Eingangssignale zu verarbeiten, wird das Signal in Blöcke zerlegt und mit Zero-Padding auf die passende Länge gebracht. Unter Zero-Padding versteht man das Auffüllen der Signalblöcke mit Nullen, sodass deren Länge mit der für die FFT benötigten Blockgröße übereinstimmt. Anschließend werden die Segmente mit dem Overlap-Add-Verfahren zusammengesetzt (Bartetzki, 2025, S. 162).

Die entstehende Latenz kann durch *partitionierte Faltung* reduziert werden, bei der die Impulsantwort in Segmente zerlegt und blockweise verarbeitet wird (Zölzer, 2005). In Faltungsplugins, wie etwa für Faltungshall oder binaurales Rendering, werden oft hybride Verfahren eingesetzt: kurze Anfangssegmente werden direkt gefaltet, während lange Nachhallanteile im Frequenzbereich verarbeitet werden. Dies erlaubt eine effiziente Echtzeitverarbeitung bei gleichzeitig niedriger Latenz (Brinkmann & Pike, 2025, S. 625).

### Bezug zu binauralem Rendering

Im binauralen Kontext wird ein trockenes Signal entweder direkt mit binauralen Raumimpulsantworten (BRIRs) oder mit monauralen RIRs gefaltet, die anschließend über HRTFs binauralisiert werden. Die Qualität des räumlichen Eindrucks hängt dabei wesentlich von der Genauigkeit der HRTF-Daten, der zeitlichen Abstimmung interauraler Differenzen sowie von der Effizienz der numerischen Umsetzung ab (Brinkmann & Pike, 2025, S. 625).

---

<sup>3</sup>FFT steht für Fast Fourier Transformation

## 2.3 Räumliche Wahrnehmung

Die Fähigkeit des Menschen, Schallquellen räumlich zu lokalisieren, basiert auf einer Vielzahl auditiver Hinweisreize. Diese werden vom auditorischen System aus den Signalen beider Ohren extrahiert und erlauben Rückschlüsse auf Richtung, Entfernung und Ausdehnung einer Schallquelle (Carlini et al., 2024). Grundlegend wird zwischen Hinweisen in der Horizontalebene (Azimut), in der Medianebene (Elevation) und für die Distanz unterschieden (Maschke & Widmann, 2012). Da einzelne Mechanismen jeweils nur in bestimmten Frequenzbereichen oder Situationen effektiv sind, wirken sie komplementär zusammen (Rayleigh, 1907).

### ITD und ILD

Die beiden wichtigsten binauralen Lokalisationshinweise sind die interaurale Zeitdifferenz (engl. Interaural Time Difference, kurz *ITD*) und die interaurale Pegeldifferenz (engl. Interaural Level Difference, kurz *ILD*).

**Interaurale Zeitdifferenz (ITD):** Bei tiefen Frequenzen wird der Laufzeitunterschied zwischen linkem und rechtem Ohr ausgewertet. Dieser Unterschied entsteht, da eine Schallwelle das dem Schall abgewandte Ohr mit einer Verzögerung erreicht. Der kleinste noch wahrnehmbare ITD beträgt etwa  $10 \mu\text{s}$  (Mills, 1958). Physiologisch liegt die höchste Empfindlichkeit damit im Bereich unterhalb von ca. 1,5 kHz (Brughera et al., 2013).

**Interaurale Pegeldifferenz (ILD):** Bei höheren Frequenzen spielt der Kopf als akustischer Schallschatten eine größere Rolle, sodass Pegeldifferenzen zwischen den beiden Ohren entstehen. Bereits Unterschiede von ca. 1 dB können wahrgenommen werden (Gelfand, 2017). ILDs sind besonders oberhalb von 3–5 kHz relevant, da die Wellenlängen hier kürzer als der Kopfdurchmesser sind (Stevens & Newman, 1936).

Die kombinierte Wirksamkeit von ITD und ILD wird als Duplex-Theorie beschrieben: ITD ist primär für tiefe Frequenzen, ILD für hohe Frequenzen relevant (Rayleigh, 1907). Diese Mechanismen ermöglichen eine präzise Lokalisation in der Horizontalebene (Azimut), wobei Genauigkeiten von  $1\text{--}2^\circ$  im Frontalbereich möglich

sind (Makous & Middlebrooks, 1990).

Ein weiterer wichtiger Mechanismus ist das *Gesetz der ersten Wellenfront*, nach dem sich die Lokalisation mehrerer Schallereignisse an der Richtung des zuerst eintreffenden Signals orientiert (Cremer & Müller, 1948). Dieser Effekt trägt wesentlich zur Stabilität der Lokalisation in reflektierenden Umgebungen bei.

### **Filterung durch die Ohren**

Neben den binauralen Unterschieden tragen monaurale spektrale Hinweise entscheidend zur räumlichen Wahrnehmung bei. Die Ohrmuscheln, der Kopf und der Oberkörper formen richtungsabhängige Filter, die durch Interferenzeffekte zu Verstärkungen und Absenkungen im Frequenzspektrum führen. Diese spektralen Modifikationen werden in den Head-Related Transfer Functions (HRTFs) abgebildet und sind besonders für die Lokalisation in der Medianebene (Elevation) sowie für die vorn-hinten-Unterscheidung relevant (Carlini et al., 2024).

Während ITD und ILD zusammen eine hohe Genauigkeit in der Horizontalebene ermöglichen, sind für Elevationsunterscheidungen hochfrequente spektrale Hinweise entscheidend. Studien zeigen, dass die Lokalisationsschärfe hier stärker individuell variiert und mit physiologischen Merkmalen wie der Ohrgröße zusammenhängt (Otte et al., 2013).

Zusätzlich können Reflexionen und Nachhall (Raumakustik) sowie Bewegungsinformationen (aktive Kopfbewegungen) die Wahrnehmung verstärken oder verfälschen (Carlini et al., 2024). Insbesondere die Verfügbarkeit mehrerer Hinweisarten gleichzeitig trägt dazu bei, dass trotz der begrenzten Präzision einzelner Mechanismen insgesamt eine hohe Lokalisationsgenauigkeit erreicht wird (Hartmann et al., 2016).

## **2.4 Binauralisierung**

Unter Binauralisierung wird die Erzeugung kopfhörertauglicher Stereosignale verstanden, die beim Hören einen räumlichen Eindruck (Richtung, Entfernung, Ex-

ternalisation) erzeugen. Dies geschieht durch Filterung eines Eingangssignals mit Head-Related Transfer Functions (HRTFs) bzw. deren Zeitbereichsform (HRIRs) (Sun, 2021). Formal ist die Multiplikation des Signals mit den HRTFs im Frequenzbereich äquivalent zur Faltung mit den HRIRs im Zeitbereich (siehe Abschnitt 2.2).

### HRTFs im Rendering-Kontext

HRTFs charakterisieren die richtungs- und entfernungsabhängige Filterung durch Ohrmuscheln, Kopf und Oberkörper und bilden monaurale spektrale Hinweise ab (Wu et al., 2019). Da HRTFs typischerweise an diskreten Raumrichtungen vorliegen, sind für dynamische Szenen Interpolationsverfahren notwendig (zeit- oder frequenzbereichsbasiert), um beim Wechsel zwischen Stützpunkten Artefakte zu vermeiden (Cuevas-Rodríguez et al., 2019).

Individualisierte HRTFs liefern in der Regel eine deutlich verbesserte Lokalisation und Externalisation, da sie die individuellen anatomischen Eigenschaften berücksichtigen. Ihre Erhebung ist jedoch aufwendig, da Messungen für viele Einfallrichtungen durchgeführt werden müssen und die so gewonnenen diskreten Daten zusätzlich eine Interpolation erfordern, um kontinuierliche Bewegungen von Schallquellen abzubilden (Wu et al., 2019). Generische HRTFs sind hingegen praktisch, da sie ohne individuellen Messaufwand direkt einsetzbar sind und in Datenbanken leicht verfügbar gemacht werden können, auch wenn sie häufig zu front/back-Verwechslungen oder Fehlern in der Elevationswahrnehmung führen (Carlini et al., 2024).

HRTFs werden üblicherweise im *SOFA*-Format gespeichert, das neben den eigentlichen Impulsantworten auch die zugehörigen Informationen zur räumlichen Orientierung enthält (siehe AES69-2022). Als internationaler Standard erleichtert SOFA den Austausch und die Wiederverwendung von Datensätzen und wird von gängigen Binaural-Rendering-Libraries wie *libmysofa*<sup>4</sup> oder der *BRT*-Library direkt unterstützt (González-Toledo et al., 2023).

---

<sup>4</sup><https://github.com/hoene/libmysofa>

Head-Tracking kann die Externalisation durch dynamische HRTF-Interpolation bzw. Ambisonics-Rotation zusätzlich unterstützen (Zotkin et al., 2004). Die Kopfbewegung sorgt für eine Veränderung der auditiven Cues, wodurch das Lokalisieren von Schallquellen verbessert und Vorne-Hinten-Vertauschungen aufgelöst werden (McAnally & Martin, 2014).

Durch die Lage des Ohrs außerhalb des HRIR-Koordinatenzentrums verändern sich bei geringen Abständen die Einfallswinkel der Schallquelle deutlich. Dieses Phänomen wird als akustische Parallaxe oder Parallax-Effekt bezeichnet und führt zu einer Unterscheidung zwischen Nahfeld und Fernfeld (Brungart & Rabinowitz, 1999).

Mit abnehmender Distanz steigt der Pegel frequenzabhängig: Tiefe Frequenzen werden relativ verstärkt, während auf der Gegenseite höhere Frequenzen stärker abgeschwächt werden. Um diesen Nahfeld-Effekt auszugleichen, können Differenzfilter auf Basis eines Sphärenmodells in Kombination mit einem geometrisch präzisen HRTF-Lookup eingesetzt werden, sodass keine zusätzlichen Messungen erforderlich sind (Rombom & Cook, 2008). Eine solche Korrektur ist notwendig, da HRTFs üblicherweise in einem festgelegten Messradius erfasst werden (Jin et al., 2006).

### **Rendering-Pipelines und Effizienz**

Binaurale Renderer trennen aus Qualitäts- und Effizienzgründen häufig Direktschall und Raumanteil. Der Direktschallpfad nutzt HRTFs, während der Raumpfad über BRIR-Segmente oder Ambisonics realisiert und anschließend binaural addiert wird (Sun, 2021). Eine verbreitete Strategie ist die Pfadzerlegung mit skalierbarem Reverb, bei der lange BRIRs in Direktschall (anechoisch) und hallige Anteile zerlegt werden. Letztere können z. B. per FOA<sup>5</sup>-Binauralisierung oder über virtuelle Lautsprecher effizient gerendert werden (Politis et al., 2018). Für die Echtzeitimplementierung werden zudem FFT-basierte Verfahren (Overlap-Save/Overlap-Add, partitionierte Faltung) eingesetzt (Brinkmann & Pike, 2025, S. 625).

---

<sup>5</sup>FOA steht für First Order Ambisonics

### **Kopfhörerentzerrung**

Für glaubwürdige Lokalisation ist zudem eine Entzerrung der Kopfhörer wesentlich. Die sogenannten Kopfhörer-Übertragungsfunktion (engl. Headphone Transfer Function, kurz HpTF) wird einerseits durch den Frequenzgang des Kopfhörers und andererseits durch die Kopplung an das Ohr beeinflusst. Die HpTF eines Kopfhörers unterscheidet sich somit nicht nur durch das Modell, sondern ist für jede Person individuell (Møller et al., 1995). Durch einen geeigneten Filter kann die HpTF eines Kopfhörers kompensiert werden (Choadhry et al., 2019).

## 3 Verwandte Arbeiten

Software zur binauralen Wiedergabe von Audio ist in unterschiedlichen Ausführungen verfügbar - sei es als Plugin oder Standalone-Anwendung. Neben kommerziellen Lösungen etablierter Hersteller wie etwa von *Waves*<sup>1</sup> oder *Emboddy*<sup>2</sup> existieren auch nicht-kommerzielle Entwicklungen wie *Anaglyph*<sup>3</sup>.

In diesem Kapitel werden ausgewählte Anwendungen vorgestellt und die unterschiedlichen Herangehensweisen, um räumliche Klangwiedergabe zu ermöglichen, verdeutlicht. Abschnitt 6.3 liefert zudem einen Vergleich mit dem im Rahmen dieser Arbeit entwickelten Plugin.

### 3.1 Anaglyph

Das Plugin *Anaglyph* ist Teil eines langjährigen Forschungsprojekts zur Untersuchung der technischen und wahrnehmungsbezogenen Möglichkeiten binauraler Wiedergabe. Es basiert auf mehr als einem Jahrzehnt Forschung im Bereich räumliches Hören und Binauralsynthese und bündelt die Ergebnisse in Form eines VST-Plugins, das sowohl als Forschungsplattform wie auch als praxisnahes Werkzeug für Audioingenieure konzipiert ist (Poirier-Quinot & Katz, 2018). Die Zielsetzung von *Anaglyph* ist daher zweigeteilt. Einerseits dient es als Testumgebung für aktuelle Spatialisierungstechniken, andererseits soll es Anwendern in klassischen DAW-Umgebungen den Zugang zu diesen Forschungsergebnissen ermöglichen.

---

<sup>1</sup><https://www.waves.com/plugins/abbey-road-studio-3>

<sup>2</sup><https://embody.co/de/pages/ivs-alan-meyerson>

<sup>3</sup><https://anaglyph.dalembert.upmc.fr>

Zu den zentralen Features zählen ein personalisierbares ITD-Modell, Korrekturen für nahfeldabhängige ILD- und HRIR-Parallaxeffekte, ein *Localisation Enhancer*, ein *Externalisation Booster* sowie die Unterstützung des SOFA-Formats für HRIR-Daten. Darüber hinaus bietet Anaglyph eine flexible Distanzdämpfung auf Basis des Invers-Quadrat-Gesetzes sowie Optionen zur individuellen Anpassung der Parameter für Distanz, Pegel und Filterung.

Die grundlegende Architektur des Plugins umfasst drei Hauptkomponenten: einen HRTF-basierten Faltungskern, einen ITD-Modifizierer und einen ILD-Modifizierer. Zusammen bilden diese Module das Binaural-Encoding, welches aus einem einkanaligen Eingangssignal ein zweikanaliges binaurales Ausgangssignal erzeugt. Der ITD-Modifizierer erlaubt eine Personalisierung der interauralen Zeitdifferenzen auf Basis der Kopfgeometrie des Nutzers (Aussal et al., 2012), während der ILD-Modifizierer frequenzabhängige Pegelunterschiede unter Verwendung eines kugelförmigen Kopfmodells simuliert (Duda & Martens, 1998). Für Quellpositionen im Nahfeld implementiert Anaglyph eine HRIR-Parallaxkorrektur, um die Messentfernung klassischer HRIR-Datensätze zu kompensieren (Otani et al., 2009).

Parallel zur direkten binauralen Faltung wird das Eingangssignal an den sogenannten *Externalisation Booster* weitergeleitet. Dieser basiert auf einem modifizierten dreidimensionalen Raumhall-Simulator (Guastavino & Katz, 2004) und verwendet FIR-Faltung mit 2nd-Order-Ambisonics-Raumsignalen. Die resultierenden Ambisonics-Puffer werden über ein virtuelles Lautsprechersystem dekodiert (Noisternig et al., 2003) und anschließend mit dem binaural verarbeiteten Direktsignal summiert. Auf diese Weise werden frühe Reflexionsmuster erzeugt, die die externe Lokalisation der Schallquelle unterstützen und eine realistischere räumliche Wahrnehmung ermöglichen.

Ein weiteres Merkmal ist der *Localisation Enhancer*, der dem Audiosignal mikroskopische Bewegungen im Azimut- und Elevationsbereich überlagert. Diese Oszillationen imitieren die feinen Kopfbewegungen eines realen Hörers und verbessern dadurch die Lokalisationsschärfe in nicht getrackten Wiedergabeszenarien (Nykänen et al., 2013).

Das Laden und Interpolieren der HRIR-, ITD-, ILD- und Raumsignale erfolgt

über das SOFA-Format. Hierbei greift Anaglyph auf die *MySofa*-Bibliothek zurück, die sowohl für die zeitliche Ausrichtung als auch für die Interpolation zwischen benachbarten Positionen zuständig ist (Hoene et al., 2017).

Zusammengefasst bietet Anaglyph eine modulare und forschungsorientierte Plattform, die zahlreiche Erweiterungen klassischer binauraler Rendering-Methoden integriert und dadurch sowohl für wissenschaftliche Studien als auch für künstlerische Produktionen geeignet ist.

## 3.2 Personalized Virtual Studio Renderer

Der *Personalized Virtual Studio Renderer* (VSR) wurde entwickelt, um die wachsende Bedeutung von Kopfhörer-Workflows in Mischung und Mastering zu adressieren. Ziel ist es, Mehrkanal- und Atmos-Produktionen ohne aufwendige Lautsprecherinstallationen zuverlässig auf Kopfhörern beurteilen zu können. Der VSR kombiniert dazu personalisierte HRTFs mit einem akkuraten Modell der Studioakustik und ermöglicht so eine realistische, binaurale Reproduktion realer Abhörsituationen (Sunder et al., 2022).

Zu den zentralen Features zählen personalisierte HRTFs, die mithilfe eines ML<sup>4</sup>-Verfahrens innerhalb weniger Sekunden aus einem Ohrbild geschätzt werden können (Shahid et al., 2018), die Integration gemessener Raumimpulsantworten zur Abbildung des akustischen Fingerabdrucks eines Studios sowie eine Kopfhörerentzerrung, die über die Inversion der HpTF und eine zusätzliche perzeptive Feinabstimmung realisiert wird (Møller, 1992). Der Renderer ist als DAW-Plugin konzipiert und lässt sich nahtlos in bestehende Produktionspipelines integrieren (Sunder et al., 2022).

Die Architektur des VSR basiert auf einer Trennung zwischen Direktschall- und Raumpfad. Der Direktschall wird über anechoische, personalisierte HRTFs gefiltert, die für jede Lautsprecherposition (Azimut, Elevation, Distanz) angewendet werden (Sunder, 2021). Parallel dazu modelliert der Raumpfad die reflektierenden

---

<sup>4</sup>ML steht für Machine Learning

Anteile, indem BRIRs in frühe, mittlere und späte Reflexionen zerlegt und mit den bereits HRTF-gefilterten Signalen gefaltet werden. Dabei werden die individuellen Lautsprecherfrequenzgänge berücksichtigt, und relative Delays verhindern Phasenauslöschungen bei der Summation (Sunder et al., 2022).

Nach der Zusammenführung von Direktschall- und Raumpfad wird eine Kopfhörerentzerrung angewendet. Da Kopfhörer nicht akustisch transparent sind und ihre Übertragung stark von der individuellen Ohrkopplung abhängt (siehe Abschnitt 2.4), wird die Korrektur nicht allein durch die inverse HPTF erreicht, sondern zusätzlich perceptiv angepasst, um eine möglichst vertrauenswürdige Beurteilbarkeit zu gewährleisten (Møller, 1992).

Zusammengefasst kombiniert der VSR personalisierte HRTFs, gemessene Raumantworten und eine gezielte Kopfhörerentzerrung zu einer integrierten Rendering-Pipeline. Die Trennung von Direktschall und Reflexionen sowie die anschließende perceptiv Feinabstimmung stellen zentrale Bausteine dar, um eine glaubwürdige Emulation realer Studioabhören auf Kopfhörern zu ermöglichen.

## 4 Methode und Implementierung

In diesem Kapitel werden die Vorgehensweise und Methodik zur Entwicklung des binauralen Faltungs-Plugins beschrieben.

### Reale Umgebung

Um die Akustik realer Umgebungen simulieren zu können, wurden zwei Räume des Tonlabors der HAW Hamburg ausgewählt, die durch das Plugin binaural erfahrbar gemacht werden. Der entwickelte Prototyp trägt daher den Arbeitstitel *HAW Binaural Lab*.

Das Tonlabor umfasst mehrere Regie- und Aufnahmeräume, die sowohl für klassische Stereo-Produktionen als auch für Ambisonics-Mischungen genutzt werden. Für die Plugin-Entwicklung wurden die zwei Räume R1 und A1 ausgewählt:

In Regie **R1**, einem Studio mit drei Lautsprecherpaaren, wurden insgesamt sechs monaurale RIRs mit einem omnidirektionalen Mikrofon (*Schoeps MK2*) sowie binaurale BRIRs mit einem Kunstkopf (*Neumann KU 100*) gemessen – jeweils eine pro Lautsprecher.

Der Raum **A1** ist mit einer 33.2-Lautsprecheranordnung ausgestattet, deren Lautsprecher in mehreren Ebenen halbkugelförmig um die Hörposition angeordnet sind. Er wird vorwiegend für die Wiedergabe und Mischung von Spatial-Audio-Formaten wie Ambisonics genutzt. Das System ist über eine DSP-Vorstufe kalibriert und bietet mehrere Presets für unterschiedliche Abhörsituationen. Für jeden Lautsprecher wurden sowohl monaurale RIRs als auch binaurale BRIRs aufgenommen – jeweils für mehrere Presets.

Die Mikrofonposition blieb bei allen Messungen konstant und entsprach der typischen Hörposition des jeweiligen Raums. Über jeden Lautsprecher wurde separat ein Sinus-Sweep abgespielt, aus dem anschließend die jeweilige Impulsantwort berechnet wurde. Eine detaillierte Anleitung zur Messung und Aufbereitung der Impulsantworten für die Nutzung im *HAW Binaural Lab* findet sich in Anhang A.

### Software-Frameworks

Die Plugin-Entwicklung erfolgte mit dem C++-Framework *JUCE*<sup>1</sup> unter Verwendung der Entwicklungsumgebung *Xcode* auf einem MacBook Pro mit M2-Prozessor. Für die Binauralisierung kommt die *Binaural Rendering Toolbox (BRT)*<sup>2</sup> zum Einsatz, die effiziente Methoden zur HRTF-basierten Filterung bereitstellt (González-Toledo et al., 2023).

Ziel der Implementierung ist es, die gemessenen Lautsprecherpositionen und deren räumliche Anordnung durch digitale Signalverarbeitung nachzubilden, sodass Eingangssignale kanalbasiert mit den entsprechenden Impulsantworten gefaltet und anschließend binaural über Kopfhörer wiedergegeben werden können.

### Zentrale Anforderungen

Ein zentrales Entwicklungsziel ist ein klar strukturierter und nachvollziehbarer Aufbau der Signalkette, der auch von Dritten erweitert werden kann. Neue Räume und Lautsprecher-Layouts (z. B. 7.1.4) sollen ohne Änderungen am Quellcode integrierbar sein, indem die entsprechenden Impulsantworten sowie die Lautsprecherpositionen in einer Textdatei im JSON-Format hinterlegt werden, auf die das Plugin beim Start zugreift. Dadurch wird eine flexible Erweiterung ohne Programmierkenntnisse ermöglicht.

Die erste Implementierung erfolgte für den Raum R1, da die begrenzte Anzahl an Lautsprechern eine übersichtliche Struktur erlaubt und als Grundlage für komple-

---

<sup>1</sup><https://juce.com>

<sup>2</sup><https://github.com/GrupoDiana/BRTLlibrary>

xere Setups dient. Das Plugin ermöglicht hier den Wechsel zwischen drei Lautsprecherpaaren. Für Raum A1 ist vorgesehen, Mischungen bis zur siebten Ambisonics-Ordnung zu unterstützen, also die Verarbeitung von bis zu 64 Eingangskanälen, deren Dekodierung auf 33 Lautsprecher und zwei Subwoofer, die anschließende binaurale Reduktion auf ein Stereosignal, eine Kopfhörerentzerrung sowie perspektivisch eine Erweiterung um Head-Tracking.

Der im Rahmen dieser Arbeit entwickelte Prototyp verfügt derzeit über keinen integrierten Ambisonics-Dekoder. Um das *HAW Binaural Lab* dennoch für Ambisonics-Produktionen einsetzen zu können, ist daher ein externer Dekoder an einer vorgelegerten Stelle der Effektkette erforderlich. In der Entwicklungs- und Testphase wurde hierfür der *AllRADecoder* des IEM verwendet<sup>3</sup>.

### Vorgehensweise

Das methodische Vorgehen der Entwicklung lässt sich in fünf zentrale Schritte gliedern:

1. Messung und Aufbereitung der Raumimpulsantworten,
2. Literaturrecherche und Analyse bestehender binauraler Rendering-Plugins,
3. Konzeption der Signalkette und Entwicklung einer erweiterbaren Dateistruktur,
4. Implementierung der einzelnen Verarbeitungsschritte (siehe Abschnitt 4.2),
5. Validierung des Signalflusses sowie Feinanpassung der Signalverarbeitung.

Der Schwerpunkt der Entwicklung liegt dabei auf zwei zentralen Aspekten: Erstens einer benutzerfreundlichen Erweiterbarkeit und zweitens einer möglichst realistischen binauralen Wiedergabe der akustischen Umgebungen der beiden untersuchten Räume.

---

<sup>3</sup><https://plugins.iem.at/docs/allradecoder/>

### 4.1 Überblick über die Entwicklungsumgebung

Bereits in der Konzeptionsphase zeigte sich, dass eine vollständige Implementierung sämtlicher Signalverarbeitungsschritte auf Basisebene nicht nur ineffizient, sondern auch innerhalb des zeitlichen Rahmens dieser Arbeit nicht umsetzbar gewesen wäre.

Um die Entwicklung dennoch in angemessenem Umfang zu realisieren, wurde auf bestehende C++-Libraries zurückgegriffen. Eine *Library* bezeichnet im Kontext der Softwareentwicklung eine Sammlung vorgefertigter Methoden, Funktionen und Klassen, die wiederverwendbare Funktionalitäten bereitstellen und dadurch eine effiziente Umsetzung komplexer Programmstrukturen ermöglichen.

Auf diese Weise mussten zentrale Operationen, die für die Realisierung eines Faltungs-Plugins erforderlich sind, nicht eigenständig entwickelt werden. Stattdessen konnten erprobte Module integriert und gezielt an die spezifischen Anforderungen dieses Projekts angepasst werden.

#### JUCE

*JUCE* ist eine umfangreiche C++-Bibliothek, die vor allem für die Entwicklung von Audioanwendungen und grafischen Benutzeroberflächen (GUI<sup>4</sup>) eingesetzt wird. Sie wird in der Audioindustrie von zahlreichen Unternehmen wie Izotope, Steinberg, Focusrite oder Genelec verwendet.

Die Bibliothek ist modular aufgebaut und bietet eine Vielzahl spezialisierter Komponenten für unterschiedliche Anwendungsbereiche. Über die mitgelieferte Anwendung *Projuicer* können Projektvorlagen erstellt werden, in denen grundlegende Strukturen und Methoden bereits implementiert und die benötigten Module automatisch eingebunden sind.

Für die Entwicklung des Faltungs-Plugins wurde die Vorlage *Plugin Basic* verwendet. Diese umfasst unter anderem das Modul `juce_plugin_client`, das Klassen für die Erstellung von VST-, VST3- und AU-Plugins bereitstellt, sowie `juce_audio_formats`,

---

<sup>4</sup>engl. Graphical User Interface; dt. Benutzeroberfläche

das Funktionen zur Verarbeitung von Audio- und MIDI-Signalen enthält.

Von zentraler Bedeutung für dieses Projekt ist das Modul `juce_dsp`, das Werkzeuge für digitale Signalverarbeitung bereitstellt, darunter auch Faltungsalgorithmen. Da dieses Modul nicht Bestandteil der Standardvorlage ist, wurde es manuell in das Projekt integriert.

### **BRT-Library**

Die *Binaural Rendering Toolbox (BRT)* ist eine offene C++-Bibliothek für binaurales Rendering, die im Rahmen des europäischen SONICOM-Projekts entwickelt wird (González-Toledo et al., 2023). Sie basiert auf den Erfahrungen mit der *3D Tune-In Toolkit Library (3DTI-Toolkit)*<sup>5</sup>, die zwar effiziente Algorithmen bereitstellt, jedoch nur begrenzt erweiterbar ist. Im Gegensatz dazu verfolgt die BRT das Prinzip einer *Open Architecture*, bei der die internen Schnittstellen der Rendering-Kette modular und flexibel gestaltet sind. Dadurch wird sowohl die Wiederverwendbarkeit einzelner Komponenten als auch die Integration neuer Algorithmen erheblich vereinfacht.

Die Bibliothek ist als *header-only* implementiert, was ihre Einbindung in bestehende Projekte erleichtert. Ihr Aufbau gliedert sich in drei hierarchische Ebenen:

- Die **untere Ebene** implementiert ein template-basiertes System zur modularen Verknüpfung einzelner Komponenten.
- Die **mittlere Ebene** umfasst Signalverarbeitungs- und Servicemodule, darunter Faltungs- und Filtereinheiten sowie HRTF-Manager mit Unterstützung des SOFA-Formats.
- Die **obere Ebene** enthält sogenannte Meta-Module, die die Verarbeitungseinheiten zu vollständigen Modellen verbinden, etwa Listener-, Source- oder Environment-Modelle.

Diese Architektur ermöglicht sowohl die gezielte Verwendung einzelner Module in spezifischen Anwendungen als auch die Realisierung vollständiger binauraler Szenen.

---

<sup>5</sup>[https://github.com/3DTune-In/3dti\\_AudioToolkit](https://github.com/3DTune-In/3dti_AudioToolkit)

rien. Im Rahmen der Entwicklung von *HAW Binaural Lab* wurden ausschließlich Module der obersten Ebene eingesetzt und in einer auf das Projekt abgestimmten Konfiguration miteinander verknüpft.

### 4.2 Aufbau des Plugins und Signalkette

Das Plugin arbeitet kanalbasiert und unterstützt bis zu 64 Eingangssignale. Jeder Kanal wird bis zur abschließenden Stereo-Summierung in der Binauralisierung getrennt verarbeitet. Der Wechsel zwischen verschiedenen Lautsprecheranordnungen erfolgt über sogenannte Layout-Dateien im JSON-Format, in denen Positionsangaben der Lautsprecher und die zugehörigen Impulsantworten (RIRs) definiert sind. Durch die Auswahl eines Layouts werden diese Informationen aus der entsprechenden Datei geladen und an die einzelnen Verarbeitungsschritte des Plugins übergeben.

#### Layout-Definition über JSON-Datei

Die JSON-Datei enthält grundlegende Angaben wie die Sample-Rate der verwendeten RIRs sowie das gewählte Koordinatensystem (kartesisch oder polar). Für jede Quelle sind die folgenden Parameter spezifiziert:

- "host\_channel": Zuordnung der Quelle zu einem bestimmten Eingangskanal der DAW.
- "azimuth\_deg": Horizontalwinkel der Quelle in Grad.
- "elevation\_deg": Vertikalwinkel der Quelle in Grad.
- "radius\_m": Abstand der Quelle zur Hörposition; dient zugleich zur Berechnung relativer Laufzeitunterschiede zwischen den Quellen.
- "gain\_db": Pegelkorrektur der Impulsantwort zur Anpassung relativer Lautstärkeunterschiede zwischen den Lautsprechern.
- "ir": Pfad zur entsprechenden Impulsantwort-Datei der Quelle.

Die Laufzeitunterschiede zwischen den Quellen werden auf Grundlage der Distanzangaben berechnet und den jeweiligen Kanälen hinzugefügt. Alternativ kann eine Verzögerung pro Quelle direkt angegeben werden, was insbesondere dann sinnvoll ist, wenn das Lautsprechersystem bereits durch ein Time-Alignment mittels DSP kompensiert wurde. Die Quellpositionen können wahlweise in Polar- oder kartesischen Koordinaten angegeben werden.

Eine Anleitung zur Implementierung neuer Lautsprecher-Layouts befindet sich in Anhang B.

### Signalfluss und Verarbeitungsstufen

Der Signalfluss eines mehrkanaligen Eingangssignals sowie die einzelnen Stufen der Signalverarbeitung sind schematisch in Abbildung 4.1 dargestellt.

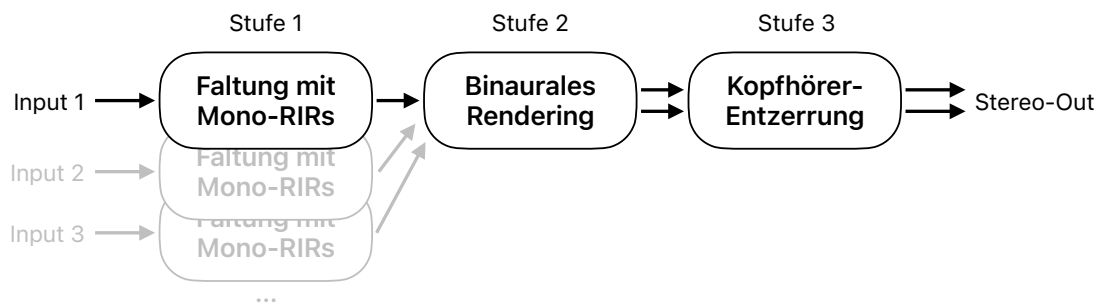


Abbildung 4.1. Schematische Darstellung der Signalkette des Plugins für mehrere Eingangssignale.

Das Plugin ist modular aufgebaut, wobei jede Datei klar definierte Aufgaben übernimmt und so eine strukturierte Trennung der Verarbeitungsschritte gewährleistet. Die zentrale Steuerung erfolgt im `PluginProcessor`, der die Verbindung zur DAW (Host) herstellt und alle relevanten Parameter verwaltet – darunter die Auswahl von HRTF und Lautsprecher-Layout, Bypass-Schalter, Feature-Flags und die Kopforientierung.

Für den zeitkritischen Echtzeitpfad (*Audio-Thread*, also den kontinuierlich ablaufenden Verarbeitungsthread für Audiodaten) werden zusätzlich sogenannte Spiegelwerte verwaltet, die atomar aktualisiert werden. *Atomar* bedeutet, dass diese Werte ohne Sperrmechanismen (*lock-free*) und somit jederzeit konsistent gelesen oder geschrieben werden können. Dadurch lassen sich Parameteränderungen zuverlässig in Echtzeit übernehmen, ohne den Audioprozess zu unterbrechen.

Rechenintensive Initialisierungen oder der Neuaufbau der Verarbeitungskette – etwa nach dem Wechsel einer HRTF oder eines Lautsprecher-Layouts – werden asynchron außerhalb des *Audio-Threads* ausgeführt. Auf diese Weise wird verhindert, dass während der Echtzeitverarbeitung Speicher dynamisch zugewiesen werden muss, was andernfalls zu hörbaren Aussetzern (*Dropouts*) führen könnte.

Die Audiodaten werden blockweise verarbeitet, das heißt in Blöcken von  $N$  Abtastwerten pro Durchlauf. Hierfür reserviert der `PluginProcessor` im Vorfeld zwei Puffertypen: einen Puffer der Größe  $S \times N$ , wobei  $S$  der Anzahl der quellenbezogenen Mono-Signale entspricht, sowie einen Puffer der Größe  $2 \times N$  für das binaurale Stereoergebnis. Die Latenzanteile aller Verarbeitungsstufen werden ermittelt und zu einer Gesamtlatenz zusammengeführt, die dem Host zur Kompensation gemeldet wird.

Definierte Bypass-Wege ermöglichen es, einzelne Verarbeitungsschritte gezielt zu deaktivieren. Gleichzeitig sorgen Fallback-Routen dafür, dass auch während kurzzeitiger Neukonfigurationen stets ein gültiges Stereosignal ausgegeben wird.

### **Stufe 1: Raumfaltung**

Die erste Verarbeitungsstufe führt die monaurale Faltung für jede Schallquelle separat durch. Hierzu stellt die Klasse `ConvolutionStage` für jede aktive Quelle eine eigene Faltungsinstanz der *JUCE*-Bibliothek (`juce::dsp::Convolution`) bereit. Diese lädt die zugehörige Mono-Impulsantwort aus dem im Layout definierten Dateipfad und wendet die Faltung auf das jeweilige Eingangssignal an. Die Verarbeitung erfolgt *blockweise in-place*, das heißt, die Signaldaten werden abschnittsweise verarbeitet und direkt im vorhandenen Speicherpuffer überschrieben, ohne dass

zusätzliche Zwischenspeicher angelegt werden müssen.

Optional definierte Korrekturen für Pegel und Laufzeit (Gain und Delay) werden während der Faltung berücksichtigt. Die Implementierung ist echtzeitfähig (*RT-sicher*) ausgelegt, was bedeutet, dass innerhalb des zeitkritischen Audio-Threads keine Operationen ausgeführt werden, die den Ablauf verzögern oder unterbrechen könnten. Aufgaben wie das Laden neuer Impulsantworten, Änderungen der Kanalanzahl oder Anpassungen der Partitionierungsparameter erfolgen dagegen außerhalb des Echtzeitpfads, um eine stabile und unterbrechungsfreie Signalverarbeitung sicherzustellen.

### **Stufe 2: Binaurales Rendering**

Die zweite Verarbeitungsstufe übernimmt das binaurale Rendering und wird durch die Klasse `Brtpipeline` realisiert. Sie nutzt die Module der *Binaural Rendering Toolbox (BRT)*, um aus den zuvor mit Mono-Raumimpulsantworten gefalteten Signalen eine räumlich korrekte Wiedergabe über Kopfhörer zu erzeugen. Dazu werden für jede Schallquelle zunächst omnidirektionale Quellenmodelle erzeugt, die die jeweiligen Eingangssignale repräsentieren. Diese Modelle werden anschließend mit einem Listener-Modell verknüpft, das in dieser Arbeit auf der *Direct HRTF Convolution* basiert. Das Listener-Modell führt die eigentliche binaurale Faltung mit den HRTFs durch und bildet die zentrale Verarbeitungseinheit der BRT.

Innerhalb des Listener-Modells werden die geometrischen Rahmenbedingungen definiert: Die Hörposition entspricht dem Ursprungspunkt des Koordinatensystems  $(0, 0, 0)$ , während die Kopforientierung (Yaw, Pitch, Roll) in Echtzeit aus den Plugin-Parametern übernommen wird. Dadurch kann die räumliche Wahrnehmung dynamisch an Änderungen der Kopfposition angepasst werden.

Der Signalfluss innerhalb dieser Stufe gestaltet sich wie folgt: Zunächst werden die vorbereiteten Monosignale aus der ersten Verarbeitungsstufe in die Quellpuffer der `Brtpipeline` übertragen. Anschließend berechnet die BRT den binauralen Mix, wobei die HRTFs für jede Quelle entsprechend ihrer Position im Lautsprecher-Layout angewendet werden. Das Ergebnis wird über ein Double-Buffering-System

ausgegeben, bei dem zwei Stereo-Puffer abwechselnd beschrieben und gelesen werden. Dieser Mechanismus ermöglicht eine kontinuierliche, echtzeitfähige Verarbeitung ohne Unterbrechungen oder Synchronisationskonflikte.

Darüber hinaus berücksichtigt die Pipeline geometrisch bedingte Laufzeitunterschiede zwischen den Lautsprecherpositionen sowie interne Latenzen der BRT. Alle Teillatenzen – einschließlich der Faltungslatenz aus der ersten Stufe – werden zusammengefasst und als konsistente Gesamtlatenz an den Host gemeldet. So bleibt das binaural gerenderte Ausgangssignal zeitlich korrekt mit der übrigen Audiokette synchronisiert.

Neben dem *Omnidirectional Source Model* stellt die BRT-Library auch ein *Directivity Source Model* bereit, mit dem sich die frequenzabhängige Richtcharakteristik von Schallquellen simulieren lässt. Da diese Richtwirkung bereits in der vorherigen Verarbeitungsstufe durch die Faltung mit den gemessenen Raumimpulsantworten abgebildet wird, genügt hier die Verwendung des omnidirektionalen Modells.

### Stufe 3: Kopfhörer-Entzerrung

Die dritte Verarbeitungsstufe führt eine Kopfhörerentzerrung (Headphone Transfer Function, kurz HpTF) auf das binaurale Stereosignal durch. Hierfür verwendet der `PluginProcessor` einen separaten `juce::dsp::Convolution`-Kern, der die entsprechenden inversen HpTF-Filter lädt und sie im Audiopfad blockweise auf das Signal anwendet. Das Nachladen neuer Filter sowie eventuelle Neukonfigurationen erfolgen außerhalb des Echtzeit-Audiothreads, um die Stabilität der Audiowiedergabe zu gewährleisten.

Durch diese Trennung bleibt die Verarbeitung latenzarm und echtzeitfähig (*RT-sicher*), das heißt frei von zeitkritischen Speicherzugriffen während der laufenden Audioverarbeitung. Der aktuelle Prototyp enthält inverse HpTFs für mehrere gängige Kopfhörermodelle und kann bei Bedarf unkompliziert um weitere Modelle erweitert werden. Die verwendeten HpTFs stammen aus der *FABIAN HRTF Database* der Technischen Universität Berlin<sup>6</sup>.

---

<sup>6</sup><https://depositonce.tu-berlin.de/items/3b423df7-a764-4ce1-9065-4e6034bba759>

### Konfiguration und Datenmanagement

Die Konfigurations- und Ressourcenebene wird durch die Klasse `BrConfiguration` realisiert. Sie ist für die Auflösung der Ressourcenpfade (im definierten Verzeichnis) verantwortlich, durchsucht verfügbare HRTF-Dateien, liest Lautsprecher-Layouts ein und generiert daraus strukturierte *SpeakerConfigs*. Diese Konfigurationen enthalten Informationen zu Lautsprecherposition (Pose), Host-Kanal-Zuordnung sowie Metadaten der zugehörigen Impulsantworten. Die Positionsangaben werden anschließend für die geometrische Modellierung innerhalb der BRT-Pipeline verwendet.

Darüber hinaus erstellt `BrConfiguration` vollständige *BrConfig*-Objekte, die Parameter wie Abtastrate und HRTF-Pfad enthalten und direkt zur Initialisierung der Rendering-Pipeline genutzt werden. Ergänzende Hilfsfunktionen in `AppUtils` unterstützen das Laden von SOFA-Dateien und überprüfen dabei die Konsistenz der Abtastrate mit den globalen BRT-Parametern. Dadurch wird sichergestellt, dass ausschließlich kompatible Datensätze in der zweiten Verarbeitungsstufe verwendet werden.

### Zusammenfassung

Der Signalfluss des Plugins erfolgt in mehreren eindeutig definierten Schritten. Zunächst werden die vom Host bereitgestellten Eingangskanäle anhand der im Layout-File spezifizierten Zuordnung den jeweiligen Quellen zugewiesen. In der ersten Verarbeitungsstufe wird jedes Eingangssignal mit der entsprechenden Mono-Impulsantwort gefaltet. Optional hinterlegte Pegelkorrekturen (IR-Gain) und Laufzeitverzögerungen (Distanz-Delays) werden dabei berücksichtigt.

Die resultierenden  $S$  Monosignale eines Blocks werden anschließend an den binauralen Renderer übergeben. Das berechnete Stereosignal wird über ein Double-Buffering-Verfahren echtzeitfähig in eine aktive Ausgabebank geschrieben. In der dritten Stufe erfolgt die Anwendung der Kopfhörerentzerrung (HpTF) auf die binaurale Stereosumme, um die frequenzabhängige Übertragungscharakteristik des verwendeten Kopfhörers zu kompensieren.

Abschließend überträgt der `PluginProcessor` das verarbeitete Signal in die vom Host bereitgestellten Ausgangskanäle, entfernt ungenutzte Signale und aktualisiert die gemeldete Gesamtlatenz. Während kurzfristiger Rekonfigurationen hält das System einen gültigen Stereo-Fallback bereit. Über definierte Bypass-Schalter können einzelne Stufen gezielt umgangen werden: Ein *Room-Bypass* überspringt Faltung und Distanzverzögerung, ein *BRT-Bypass* gibt definierte Stereo-Ersatzsignale aus den Stufe-1-Ausgaben oder dem zuletzt gültigen BRT-Block aus, und ein *HpTF-Bypass* führt die binaurale Summe ungefiltert an den Ausgang weiter.

### 4.3 Effizienz und Latenz-Reduktion

Die Architektur des Plugins ist auf eine effiziente Echtzeitverarbeitung bei minimaler Latenz ausgelegt. Anstelle absoluter Laufzeiten werden ausschließlich relative Zeitdifferenzen zwischen den Quellen berücksichtigt, wodurch ein globales Delay entfällt. Zusätzlich wurden die Impulsantworten in Abhängigkeit ihrer Nachhallzeit auf die energetisch relevanten Abschnitte gekürzt, was die Länge der Faltungsfiler und somit die Rechenlast deutlich reduziert (siehe Anhang A). Die Pegel der Impulsantworten werden bereits vor der Laufzeit analysiert und angepasst, sodass während der Echtzeitverarbeitung keine zusätzlichen Gain-Berechnungen erforderlich sind.

Die dreistufige Pipeline (Raumfaltung, BRT-Binauralisierung, Kopfhörerentzerrung) arbeitet blockbasiert und verzichtet vollständig auf dynamische Speicherallokationen im Audiothread. Unter Speicherallokation versteht man das Anfordern von Arbeitsspeicher während der Programmausführung – ein Vorgang, der zeitaufwendig sein und in Echtzeitanwendungen zu hörbaren Aussetzern (*Dropouts*) führen kann. Im *HAW Binaural Lab* wird der benötigte Speicher daher bereits im Vorfeld reserviert, sodass die Signalverarbeitung ohne Unterbrechungen und deterministisch in Echtzeit erfolgt.

Für die Faltung wird ein FFT-basiertes, nicht-uniform partitioniertes Verfahren eingesetzt, bei dem kurze Partitionen für frühe Reflexionen und längere für den Nachhall verwendet werden. Auf diese Weise bleibt die Latenz konstant, während

die Rechenlast der FFT signifikant verringert wird. Weitere Optimierungen – etwa Compiler-Flags für Release-Builds – tragen zusätzlich zur Laufzeiteffizienz bei. Insgesamt führt diese Kombination aus algorithmischen und architektonischen Optimierungen zu einer geringen Gesamtlatenz und einer nahezu linearen Skalierung der CPU-Last mit der Anzahl der verarbeiteten Quellen.

### 4.4 Erweiterbarkeit: Einbindung neuer Räume

Zur Erweiterung der Plugin-Architektur können neue Lautsprecher-Layouts über eine JSON-Datei definiert werden, in der sämtliche relevanten Parameter eines Raums hinterlegt sind. Diese Datei sowie eigene Impulsantworten, personalisierte HRTFs oder zusätzliche Kopfhörer-Transferfunktionen (HpTFs) können in einem Unterordner des VST3-Plugins abgelegt werden. Beim nächsten Laden des Plugins werden die neuen Ressourcen automatisch erkannt und stehen anschließend im Interface zur Auswahl bereit.

Die Struktur der JSON-Datei ist bewusst flexibel gehalten. Die Positionen der Lautsprecherquellen können entweder in kartesischen Koordinaten ( $x, y, z$ ) oder in sphärischer Form (`azimuth_deg`, `elevation_deg`, `radius_m`) angegeben werden. Laufzeitunterschiede zwischen den Quellen werden dabei entweder automatisch anhand der Entfernung zur Hörposition berechnet oder direkt als Verzögerungswerte in Millisekunden beziehungsweise Samples angegeben.

Zur Erstellung neuer Layouts genügt somit ein einfacher Texteditor und gegebenenfalls eigene, gemessene oder aus Datensätzen stammende Raumimpulsantworten. Auf diese Weise lässt sich das Plugin um beliebige akustische Szenarien erweitern, ohne dass Änderungen am Quellcode erforderlich sind.

# 5 Evaluation der Funktionsweise

Zur Validierung der korrekten Funktionsweise des entwickelten Plugins wurden mehrere technische und auditive Tests durchgeführt. Dieses Kapitel präsentiert die erzielten Ergebnisse und bewertet die einzelnen Verarbeitungsschritte.

## 5.1 Faltung mit Impulsantworten

Das Ergebnis der Faltung einer Impulsantwort mit einem Dirac-Impuls  $\delta_0(t)$  entspricht exakt der Impulsantwort  $h(t)$  (siehe Abbildung 5.1).

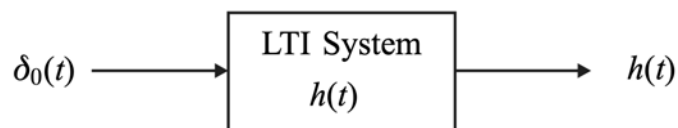


Abbildung 5.1. Impulsantwort eines linearen zeitinvarianten (LTI-)Systems (Mertins, 1996, S. 48)

Ausgehend von diesem Zusammenhang wurde ein Einheitsimpuls als Testsignal für die korrekte Funktionsweise der Faltungskomponente genutzt. Die Faltung mit dem Testsignal sollte als Ausgangssignal der gemessenen Impulsantwort entsprechen, sofern alle weiteren Verarbeitungsschritte – das binaurale Rendering sowie die Kopfhörerentzerrung – deaktiviert (*Bypass*-Modus) sind.

Um die spektrale Übereinstimmung zweier Signale zu quantifizieren wurde die *Log Spectral Distance* (LSD) bestimmt. Diese Metrik beschreibt die Abweichung zwischen den Amplitudenspektren zweier Signale. Bei identischen Signalen beträgt der Wert null, größere Werte weisen auf stärkere Unterschiede hin.

Nach Anwendung eines Gain-Matchings zur Pegelanpassung zwischen Plugin-Ausgabe und gemessener Impulsantwort ergab sich für alle getesteten Impulsantworten eine mittlere Abweichung von lediglich 0,09 dB. Dies entspricht einer nahezu exakten Übereinstimmung und bestätigt die fehlerfreie Implementierung der Faltungsstufe.

Der Funktionstest der Faltungskomponente kann somit als erfolgreich abgeschlossen betrachtet werden.

## 5.2 Binauralisierung

Zur Überprüfung der korrekten Funktion der Binauralisierung wurde ein Testsignal in Ambisonics 5. Ordnung enkodiert und auf vier Positionen der Hörsphäre verteilt:

- $-90^\circ$  Azimut (Signal von links)
- $0^\circ$  Azimut (Signal von vorne)
- $+90^\circ$  Azimut (Signal von rechts)
- $90^\circ$  Elevation (Signal von oben)

Das Signal wurde anschließend mit dem *AllRADecoder* auf das Lautsprecher-Layout des Studios A1 dekodiert. Dieses Layout wurde in *HAW Binaural Lab* geladen, und für jede der vier Quellpositionen wurde das resultierende Stereosignal separat exportiert und analysiert. Im Rahmen der Untersuchung wurden die interaurale Laufzeitdifferenz (ITD) zwischen linkem und rechtem Kanal sowie die Magnitudenspektren der resultierenden Signale ausgewertet.

### 5.2.1 Interaurale Laufzeitdifferenz (ITD)

Die gemessenen ITD-Werte der vier Testsignale sind in Tabelle 5.1 dargestellt.

Die bei den seitlichen Positionen gemessene Verzögerung von  $\pm 708,3 \mu\text{s}$  entspricht einer Laufzeitdifferenz von rund 24,3 cm und liegt damit im Bereich des typischen

<b>Position</b>	<b>ITD (<math>\mu\text{s}</math>)</b>
Rechts	+708.3
Oben	0.0
Links	-708.3
Vorne	0.0

*Tabelle 5.1.* Gemessene ITD für vier Lautsprecherpositionen

interauren Abstands des menschlichen Kopfes (“Sengpiel”, 2025). Für die frontale und obere Position wurde keine signifikante Verzögerung festgestellt, was auf einen symmetrischen Abstand beider Ohren zur Schallquelle hinweist.

### 5.2.2 Magnitude-Spektren

Die Magnitudenspektren der vier Testpositionen zeigen ebenfalls plausible Unterschiede. Die entsprechenden Abbildungen sind in Anhang C dargestellt. Für die Positionen *links* und *rechts* weist die jeweils gegenüberliegende Ohrseite einen deutlichen Abfall im Hochfrequenzbereich auf. Dies lässt sich durch die durch den Kopf verursachte Schallabschattung erklären. Auch für die Position *oben* ist im Vergleich zur Frontalposition eine leichte Absenkung der hohen Frequenzen zu beobachten, was auf den Einfluss der Kopfgeometrie und der Ohrmuscheln zurückzuführen ist.

## 5.3 Analyse des Debug-Logs

In der Entwicklungsumgebung *Xcode* wurde zusätzlich der Debug-Log ausgewertet. Hierzu wurden mehrere Log-Ausgaben implementiert, mit deren Hilfe die internen Prozesse des Plugins in Echtzeit überwacht werden können.

Die Analyse des Logs zeigt, dass alle benötigten Dateien – darunter HRTFs, HpTFs und monaurale Impulsantworten – korrekt geladen werden. Darüber hinaus konnte die fehlerfreie Reaktion auf Parameteränderungen, beispielsweise bei der Anpassung der Kopforientierung, bestätigt werden.

## 5.4 Auditiver Funktionstest

Zur subjektiven Überprüfung der Funktionsweise des Plugins wurde ein auditiver Funktionstest mit zwei unterschiedlichen Kopfhörermodellen durchgeführt (*Sennheiser HD 650* und *Sennheiser HD 25-II*). Als Testsignal diente eine Musikaufnahme, die in Ambisonics 5. Ordnung enkodiert, über den *AllRADecoder* dekodiert und über *HAW Binaural Lab* binaural wiedergegeben wurde. Während der Wiedergabe wurde zwischen *Bypass ON/OFF* geschaltet, um die einzelnen Verarbeitungsstufen – Faltung, Binauralisierung und Kopfhörerentzerrung – gezielt zu evaluieren.

Die Filterung durch die monauralen Raumimpulsantworten ist dabei deutlich wahrnehmbar und führt zu einer klanglichen Veränderung, die der jeweiligen Raumcharakteristik entspricht. Auch der räumliche Klangeindruck bestätigt die korrekte Funktion der binauralen Rendering-Stufe.

Ein auffälliger Aspekt ist die Variation des Ausgangspegels beim Wechsel zwischen verschiedenen Lautsprecher-Layouts. Diese Unterschiede lassen sich darauf zurückführen, dass beim Layout-Wechsel jeweils unterschiedliche Audiodateien für die Faltung geladen werden, die voneinander abweichende Pegel aufweisen.

Die integrierte Kopfhörerentzerrung zeigte bei beiden getesteten Modellen eine subjektiv positive Wirkung. Die Wiedergabe wirkt dadurch neutraler und transparenter, was auf eine funktionierende Implementierung der entsprechenden Filter hindeutet.

## 6 Diskussion

In diesem Kapitel werden die Ergebnisse der Evaluation bewertet, sowie Einschränkungen und Optimierungsbedarf des Plugins aufgezeigt. Das Plugin wird zudem mit verwandten Arbeiten aus Kapitel 3 verglichen und ein Ausblick für weiterführende Forschung und Entwicklung gegeben.

### 6.1 Bewertung des erreichten Funktionsumfangs

Die durchgeführten Tests zeigen, dass die wesentlichen Funktionskomponenten des entwickelten Plugins korrekt implementiert wurden. Die spektrale Übereinstimmung zwischen gemessenen und gefalteten Impulsantworten bestätigt die fehlerfreie Arbeitsweise der Faltungskomponente. Auch die Ergebnisse der binauralen Tests belegen eine konsistente interaurale Laufzeitdifferenz sowie plausible spektrale Verteilungen, die mit typischen akustischen Eigenschaften des menschlichen Kopfes übereinstimmen.

Die subjektive Hörkontrolle unterstreicht zudem, dass die Raumabbildung und die durch die monauralen Impulsantworten verursachte Filterung wahrnehmbar und realistisch sind. Die integrierte Kopfhörerentzerrung führt bei beiden getesteten Modellen zu einer klanglich ausgewogeneren Wiedergabe, was auf eine funktionierende Implementierung hinweist.

Gleichzeitig wurde deutlich, dass der Ausgangspegel zwischen verschiedenen Lautsprecher-Layouts und Kopfhörer-Transferfunktionen (HpTFs) variiert. Diese Unterschiede resultieren aus den jeweils unterschiedlichen Impulsantworten und könnten durch eine automatisierte Pegelanpassung reduziert werden. Auf eine Normali-

sierung der Impulsantworten auf 0 dB wurde bewusst verzichtet, um die realen Lautstärkeunterschiede der verwendeten Lautsprecher beizubehalten.

Insgesamt erfüllt das Plugin den im Rahmen dieser Arbeit angestrebten Funktionsumfang. Die erzielten Ergebnisse bestätigen eine stabile technische Umsetzung und eine akustisch konsistente Signalverarbeitung.

## 6.2 Einschränkungen und Optimierungspotenzial

Der aktuelle Stand des *HAW Binaural Lab* stellt einen funktionsfähigen, jedoch noch prototypischen Entwicklungsstand dar. Die wesentlichen Signalverarbeitungsschritte sind implementiert und arbeiten zuverlässig in Echtzeit, dennoch besteht in mehreren Bereichen Optimierungsbedarf.

Das Plugin wurde bislang ausschließlich auf macOS-Systemen entwickelt und getestet. Daher kann nicht abschließend beurteilt werden, ob die aktuelle Implementierung ohne Anpassungen auf anderen Betriebssystemen wie Windows oder Linux lauffähig ist. Insbesondere Unterschiede in der Audio-Host-Integration oder in Dateipfadstrukturen könnten Anpassungen erforderlich machen.

Die Benutzeroberfläche ist bislang vorwiegend funktional gestaltet und bietet noch keine umfassende visuelle Rückmeldung oder intuitive Steuerung der Parameter. Eine erweiterte grafische Darstellung der Lautsprecheranordnung oder der Kopforientierung würde die Benutzerfreundlichkeit erheblich verbessern.

Darüber hinaus wurde im Rahmen dieser Arbeit primär die technische Funktionalität des Plugins überprüft. Für eine aussagekräftige Beurteilung der räumlichen Klangabbildung ist jedoch eine weiterführende Evaluation, beispielsweise durch kontrollierte Hörversuche, erforderlich.

Auch auf konzeptioneller Ebene bestehen Erweiterungsmöglichkeiten. Der aktuelle Prototyp basiert auf der Faltung monauraler RIRs und der nachgelagerten Binauralisierung. Eine Erweiterung um vollständige binaurale Raumimpulsantworten (BRIRs) unter Nutzung entsprechender Module der BRT-Library könnte eine prä-

zisere Modellierung des Nachhalls und damit eine realistischere Raumabbildung ermöglichen.

Schließlich erfolgt die Dekodierung szenenbasierter Formate wie Ambisonics bislang extern. Eine direkte Integration dieser Funktion in das Plugin – beispielsweise über eine interne Dekodierung auf das gewählte Lautsprecher-Layout – würde den Workflow vereinfachen und den praktischen Nutzen des Systems deutlich erhöhen.

### 6.3 Vergleich mit verwandten Arbeiten

Der Vergleich des *HAW Binaural Lab* mit den in Kapitel 3 vorgestellten Plugins zeigt, dass sich die Ansätze zwar in ihrer Zielsetzung ähneln, die technische Umsetzung jedoch in wesentlichen Punkten unterscheidet. Während sowohl *Anaglyph* als auch der *Personalized Virtual Studio Renderer* (VSR) vollständige binaurale Wiedergabeketten abbilden, verfolgt das *HAW Binaural Lab* einen stärker modularen Ansatz, der insbesondere auf Nachvollziehbarkeit und Erweiterbarkeit ausgelegt ist.

*Anaglyph* zeichnet sich durch die umfassende Integration psychoakustischer Modelle aus. Mit personalisierbaren ITD- und ILD-Algorithmen, Parallax-Korrekturen sowie Erweiterungen zur Verbesserung von Lokalisation und Externalisation (z. B. *Localisation Enhancer*, *Externalisation Booster*) deckt es ein breites Spektrum räumlicher Wahrnehmungseffekte ab und nutzt dabei auch Ambisonics-basierte Verfahren.

Im Gegensatz dazu verfolgt das *HAW Binaural Lab* einen stärker physikalisch fundierten Ansatz. Es basiert auf real gemessenen Impulsantworten und verwendet die BRT-Library für das binaurale Rendering. Psychoakustische Effekte wie ITD, ILD, akustische Parallaxe und Nahfeldeffekte werden ebenfalls berücksichtigt, lassen sich jedoch nicht anpassen.

Einige Komponenten überschneiden sich jedoch auch mit *Anaglyph*, etwa die Nutzung der *MySofa*-Library zum Laden und Verarbeiten von SOFA-Dateien (Hoene et al., 2017). Darüber hinaus wurde *Anaglyph* im Rahmen des 3D Tune-In-Projekts entwickelt, dessen Forschungsergebnisse ebenfalls in die BRT-Library

eingeflossen sind (Cuevas-Rodríguez et al., 2019). Der wesentliche Unterschied liegt in den zusätzlichen psychoakustischen Erweiterungen von *Anaglyph*, die gezielt auf eine Verbesserung der Lokalisations- und Externalisationswahrnehmung abzielen.

Der *Personalized Virtual Studio Renderer* (VSR) hingegen ist auf die realistische Nachbildung von Studioumgebungen spezialisiert und verfolgt damit ein ähnliches Ziel wie das *HAW Binaural Lab*. Durch die Kombination personalisierter HRTFs mit gemessenen binauralen Impulsantworten und einer perzeptiv abgestimmten Kopfhörerentzerrung erreicht er eine besonders hohe räumliche und klangliche Authentizität. Die Nutzung von BRIRs erlaubt eine präzise Simulation von Nachhallstrukturen und stellt einen zentralen Unterschied zur Signalverarbeitung im *HAW Binaural Lab* dar. Da die Lokalisation von Quellen vorwiegend durch Direkt-schall und frühe Reflexionen bestimmt wird, kann im VSR auf die Berechnung von Kopfbewegungen in den BRIRs verzichtet werden, was die Rechenlast deutlich reduziert.

Darüber hinaus bieten sowohl *Anaglyph* als auch der *VSR* eine benutzerfreundlichere grafische Oberfläche als der aktuelle Prototyp des *HAW Binaural Lab*, der bisher primär auf funktionale Nachvollziehbarkeit ausgelegt ist.

Zusammenfassend lässt sich festhalten, dass *Anaglyph* insbesondere durch seine tiefgehende Integration psychoakustischer Modelle und Zusatzmodule zur Verbesserung der räumlichen Wahrnehmung überzeugt, während der *VSR* durch die Kombination personalisierter HRTFs, gemessener BRIRs und perzeptiv optimierter Kopfhörerentzerrung eine hohe klangliche Authentizität erreicht. Das im Rahmen dieser Arbeit entwickelte *HAW Binaural Lab* verbindet einen physikalisch fundierten Signalfluss mit der Flexibilität der BRT-Library, die sowohl geometrische als auch psychoakustische Parameter (ITD, ILD, Parallaxe, Nahfeldeffekte) integriert. Damit positioniert sich der Prototyp zwischen den beiden Ansätzen als offene und leicht erweiterbare Forschungsplattform.

## 6.4 Ausblick auf mögliche Erweiterungen

Die modulare und offene Architektur des entwickelten Plugins ermöglicht zahlreiche Weiterentwicklungen. Besonders im Hinblick auf zukünftige Versionen bestehen verschiedene Ansatzpunkte, um sowohl die Funktionalität als auch die Benutzerfreundlichkeit weiter zu verbessern.

Ein naheliegender Entwicklungsschritt betrifft die Integration von Head-Tracking-Systemen. Durch die Erfassung von Kopfbewegungen könnten HRTFs dynamisch angepasst werden, wodurch realitätsnahe Lokalisationsverläufe entstehen. Diese Erweiterung würde insbesondere bei der Arbeit mit virtuellen Lautsprecheranordnungen die Externalisation und Richtungswahrnehmung deutlich verbessern.

Auch die grafische Benutzeroberfläche bietet Potenzial für Erweiterungen. Eine interaktive Visualisierung der Lautsprecherpositionen, kombiniert mit der direkten Steuerung zentraler Parameter wie HRTF-Auswahl oder Raumkonfiguration, würde die Bedienung intuitiver gestalten und das akustische Szenario visuell nachvollziehbarer machen.

Darüber hinaus ist die Integration zusätzlicher Räume und Lautsprecher-Layouts vorgesehen. Das modulare, JSON-basierte Layout-System ermöglicht es, neue Konfigurationen ohne Programmierkenntnisse hinzuzufügen und das Plugin flexibel an unterschiedliche Studioumgebungen oder Forschungsszenarien anzupassen.

Da das Projekt unter einer Open-Source-Lizenz veröffentlicht ist<sup>1</sup>, kann seine Weiterentwicklung perspektivisch auch gemeinschaftlich erfolgen. Beiträge in Form neuer Räume, HRTF-Datensätze, Visualisierungstools oder algorithmischer Optimierungen können die Plattform langfristig erweitern und sie zu einem offenen Werkzeug für binaurales Mischen und räumliche Audioproduktion weiterentwickeln.

Insgesamt wird deutlich, dass das *HAW Binaural Lab* mit seiner modularen Struktur und klaren Signalarchitektur eine solide Grundlage für zukünftige Forschung und Weiterentwicklung im Bereich binauraler Audiowiedergabe bietet. Die identifizierten Optimierungspotenziale und Erweiterungsmöglichkeiten zeigen zugleich, in welche

---

<sup>1</sup><https://github.com/lfehrmann/HAW-Binaural-Lab>

Richtung sich das System weiterentwickeln kann, um als praxisorientiertes und wissenschaftlich fundiertes Werkzeug einen nachhaltigen Beitrag zur räumlichen Audioproduktion zu leisten.

## 7 Fazit

Ziel dieser Arbeit war die Konzeption und Umsetzung eines binauralen Faltungs-Plugins, das reale Lautsprecheranordnungen virtuell auf Kopfhörern erlebbar macht. Mit dem im Rahmen dieser Arbeit entwickelten *HAW Binaural Lab* konnte ein funktionaler Prototyp realisiert werden, der eine modulare und erweiterbare Architektur für binaurales Rendering bereitstellt.

Das Plugin kombiniert gemessene monaurale Raumimpulsantworten mit der *Binaural Rendering Toolbox (BRT)* und ermöglicht dadurch die physikalisch fundierte Simulation komplexer Lautsprecher-Setups. Durch den Einsatz eines dreistufigen Verarbeitungsschemas – bestehend aus Raumfaltung, binauraler Rendering-Pipeline und Kopfhörerentzerrung – wird eine klare Trennung der Signalpfade und eine effiziente Echtzeitverarbeitung gewährleistet. Die entwickelte Struktur erlaubt zudem eine einfache Integration neuer Räume und Lautsprecher-Layouts über ein flexibles JSON-basiertes Dateisystem, ohne dass Programmierkenntnisse erforderlich sind.

Die erzielten Ergebnisse zeigen, dass eine realitätsnahe binaurale Wiedergabe auf Grundlage realer Messdaten mit vertretbarem Rechenaufwand und geringer Latenz möglich ist. Damit eröffnet das *HAW Binaural Lab* neue Möglichkeiten für die realistische Beurteilung und Gestaltung von Kopfhörer-Mischungen in virtuellen Studio- und Produktionsumgebungen.

Gleichzeitig wurden bestehende Einschränkungen deutlich: Der aktuelle Entwicklungsstand ist noch prototypisch, die Benutzeroberfläche funktional, und psychoakustische Erweiterungen wie Head-Tracking, Ambisonics-Dekodierung oder BRIR-basierte Nachhallmodellierung sind bislang nicht implementiert. Dennoch bildet die modulare Struktur eine solide Basis für zukünftige Weiterentwicklungen.

gen, etwa durch Integration zusätzlicher Räume, Optimierung der Latenz oder die Erweiterung um adaptive Rendering-Verfahren.

Abschließend lässt sich festhalten, dass das *HAW Binaural Lab* sowohl als Forschungsplattform als auch als praxisorientiertes Werkzeug ein hohes Potenzial besitzt. Es vereint einen nachvollziehbaren Aufbau mit einer modularen Struktur und bietet damit eine solide Grundlage für weiterführende Entwicklungen im Bereich der binauralen Audiowiedergabe und virtuellen Akustiksimulation.

# Anhang

## A Messung und Vorbereitung von Impulsantworten

Die Messung und Vorbereitung der Raumimpulsantworten in den Räumen R1 und A1 des Tonlabors der HAW Hamburg erfolgte im Rahmen des Forschungsprojekts M7. In diesem Abschnitt werden die Methoden zur Messung der monauralen und binauralen Raumimpulsantworten beschrieben, die anschließende Aufbereitung dieser Daten für das Faltungs-Plugin *HAW Binaural Lab* erläutert und der daraus resultierende Datensatz vorgestellt.

### Messung

Für die Aufnahme von Impulsantworten in einem Raum sind drei Komponenten erforderlich:

- Schallempfänger (Mikrofon)
- Schallquelle (Lautsprecher)
- Mess-Software

### Wahl des Mikrofons

Für die unverfälschte Messung einer Raumimpulsantwort ist es von entscheidender Bedeutung, ein geeignetes Messmikrofon zu verwenden. Geeignet sind Mikrofone

mit Kugelcharakteristik und linearem Frequenzgang, da sie den eintreffenden Schall aus allen Richtungen gleichmäßig erfassen.

Im Rahmen des Forschungsprojekts wurde eine MK2-Mikrofonkapsel in Kombination mit dem CMC 6 Mikrofonverstärker von Schoeps eingesetzt. Die MK2-Kapsel besitzt eine Kugelcharakteristik und einen konstanten Frequenzgang bei frontal einfallendem Schall (siehe Abbildung 7.1) (“Schoeps MK2”, 2025).

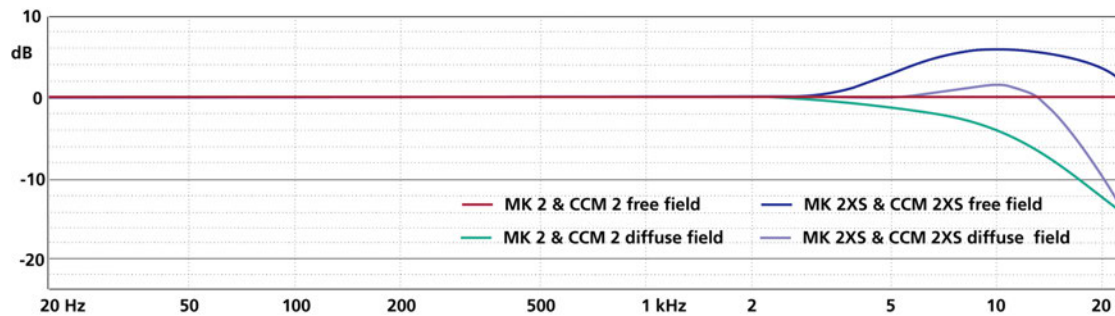


Abbildung 7.1. Vergleich der Frequenzgänge unterschiedlicher freifeld- und diffusfeldentzerrter Schoeps-Mikrofonkapseln (“Schoeps”, 2025)

Das Mikrofon arbeitet nach dem Prinzip des Druckempfängers. Um den sogenannten Druckstauereffekt auszugleichen, der bei hohen Frequenzen zu Pegelabsenkungen führen kann, ist es freifeldentzerrt ausgeführt (Schneider, 2025).

Die binauralen Messungen wurden mit einem Kunstkopf *KU 100* von Neumann durchgeführt<sup>1</sup>. In jedem Ohr ist ein Mikrofon integriert, wodurch binaurale Aufnahmen ermöglicht werden. Auch diese Mikrofone sind freifeldentzerrt.

## Lautsprecher

Um die Impulsantwort eines Raumes präzise zu bestimmen, ist es ebenso wichtig, dass die verwendeten Lautsprecher das Messergebnis nicht verfälschen. Daher werden für derartige Messungen üblicherweise lineare oder entzerrte Lautsprecher eingesetzt (Möser, 2009, S. 138).

Die im Rahmen dieser Arbeit aufgezeichneten Impulsantworten sollen jedoch die

<sup>1</sup><https://www.neumann.com/en-gb/products/microphones/ku-100>

realen Wiedergabebedingungen abbilden. Der Anwendungsfall unterscheidet sich somit von der klassischen raumakustischen Messpraxis. Vielmehr war es hier erwünscht, dass der Frequenzgang der Lautsprecher in den Messungen enthalten ist. Die aufgezeichneten Impulsantworten repräsentieren somit eine Kombination aus der akustischen Antwort des Raumes und der elektroakustischen Übertragungseigenschaften der Lautsprecher.

### **Mess-Software**

Zur Messung der Impulsantworten wurde die Software “Room EQ Wizard” (2025) (kurz: REW) verwendet. Alternativ können auch andere Programme wie beispielsweise *EASERA*<sup>2</sup> genutzt werden.

REW ermöglicht die Messung von Impulsantworten mit dem Sweep-Verfahren, also unter Verwendung eines Sinus-Sweeps als Anregungssignal. Über den Menüpunkt *Measure* öffnet sich ein Fenster, in dem der Messvorgang konfiguriert werden kann (siehe Abbildung 7.2).

### **Messaufbau**

Die Raumimpulsantwort ist abhängig von der Position der Schallquelle und des Mikrofons. Um die gemessenen monauralen Impulsantworten möglichst realistisch durch Faltung mit HRTFs binauralisieren zu können, wurde darauf geachtet, dass die Position der Mikrophonkapsel bei allen Messungen identisch bleibt.

Das Mikrofon beziehungsweise der Kunstkopf wurde in einer Höhe von 120 cm mittig zwischen den Lautsprechern positioniert. Die Platzierung im Raum orientierte sich an der typischen Hörposition: In R1 befand sie sich etwa 20 cm von der Kante des Mischpults entfernt, während das Mikrofon in A1 im gekennzeichneten *Sweet Spot* aufgestellt wurde.

Für die Messung wurde ein Sinus-Sweep mit einer Dauer von etwa 11 s separat über

---

<sup>2</sup><https://www.afmg.eu/en/afmg-easera>

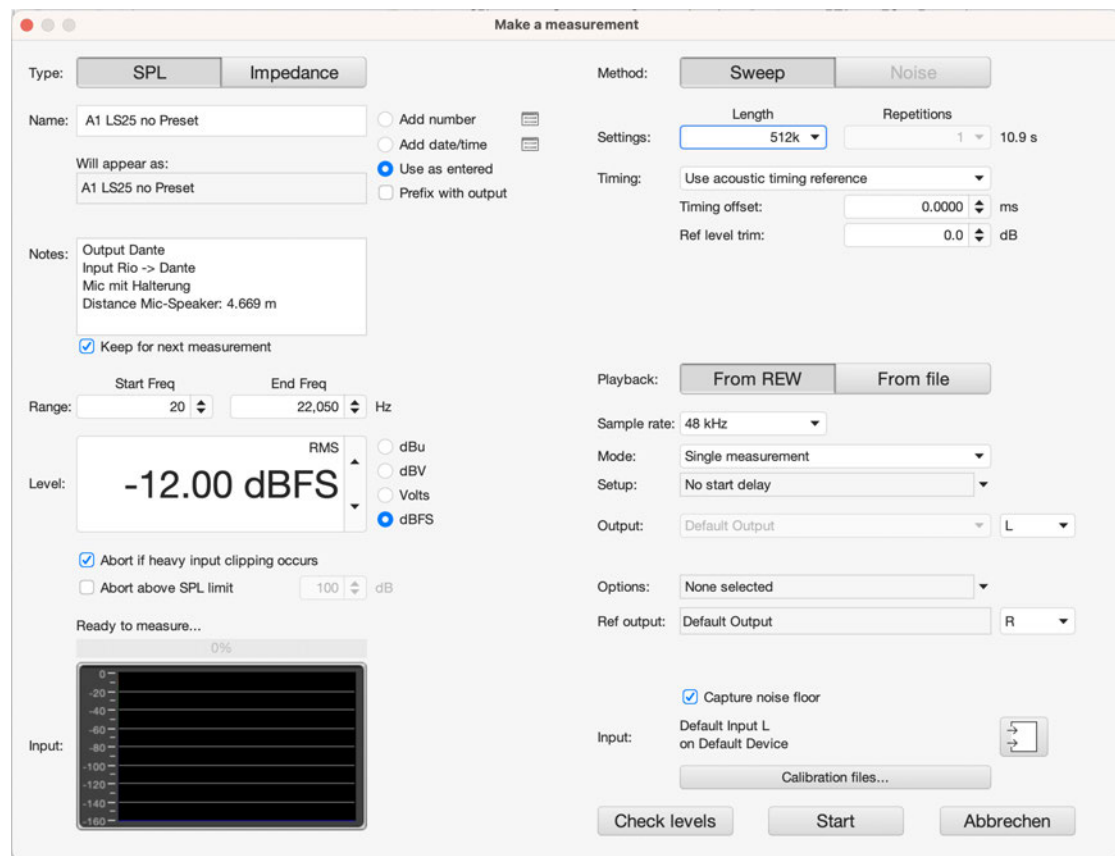


Abbildung 7.2. Measure-Fenster mit den verwendeten Einstellungen in der Software REW

jeden Lautsprecher wiedergegeben. Das Messmikrofon wurde vor jeder Messung präzise auf den jeweiligen Lautsprecher ausgerichtet und die Impulsantwort mit 24 Bit und einer Abtastrate von 48 kHz gemessen.

In Raum A1 wurden für jeden Lautsprecher mehrere Impulsantworten aufgenommen. Über eine DSP-Vorstufe können Frequenzgang und Laufzeit der Lautsprecher angepasst werden, beispielsweise zur Entzerrung des Systems oder zum Ausgleich von Laufzeitunterschieden. Insgesamt ergaben sich dadurch vier Messungen pro Lautsprecher mit unterschiedlichen DSP-Presets. Die Messung mit dem Kunstkopf wurden lediglich mit zwei unterschiedlichen Presets durchgeführt.

Mit REW wurde zudem die relative Verzögerung zwischen den Lautsprechern bestimmt. Hierzu wird im *Measure*-Fenster die Option *Use acoustic timing reference*

aktiviert. Als *Output* wird beispielsweise *L* und als *Reference Output* *R* gewählt. Bevor der eigentliche Sinus-Sweep über den *Output* abgespielt wird, sendet REW einen kurzen Referenzimpuls über den *Reference Output*. Wenn die Ausgänge mit unterschiedlichen Lautsprechern verbunden sind, lässt sich aus dem Zeitunterschied dieser beiden Signale anschließend die Laufzeitdifferenz bestimmen.

Zusätzlich wurden die Entfernungen der Lautsprecher relativ zur Hörposition sowie die Abstände der Lautsprecher zueinander mit einem Laser-Entfernungsmessgerät bestimmt. Aus diesen Messwerten wurden die relativen Raumkoordinaten der Lautsprecher berechnet und anschließend für die Binauralisierung mit dem Plugin *HAW Binaural Lab* verwendet.

### Vorbereitung für die Faltung

Damit die gemessenen Impulsantworten für die Faltung mit dem Plugin *HAW Binaural Lab* genutzt werden können, müssen sie zunächst angepasst werden.

Im ersten Schritt werden die Impulsantworten zeitlich angeglichen. Dazu wird in der Kopfleiste von *REW* das Fenster *All SPL* ausgewählt und anschließend über *Actions* die Funktion *Align IR start* auf alle ausgewählten Impulsantworten angewendet. Durch diesen Vorgang wird der Startzeitpunkt jeder Impulsantwort auf  $t = 0$  gesetzt.

Die gemessenen Laufzeitunterschiede bleiben jedoch weiterhin im Notizfenster der jeweiligen Impulsantworten gespeichert und können dem Faltungs-Plugin über die JSON-Datei des Lautsprecher-Layouts übermittelt werden.

Anschließend wird in das Fenster *Filtered IR* gewechselt, in dem die Impulsantwort im Zeitbereich dargestellt ist. Durch die zeitliche Angleichung wurde der Peak jeder Impulsantwort – also die lauteste Stelle – als Startpunkt definiert.

Dadurch wird jedoch der tatsächliche Beginn des Impulses teilweise abgeschnitten. Aus diesem Grund muss im Menü der Punkt *IR Windows* ausgewählt werden, mit dem sich ein Ausschnitt der Impulsantwort festlegen lässt (siehe Abbildung 7.3).

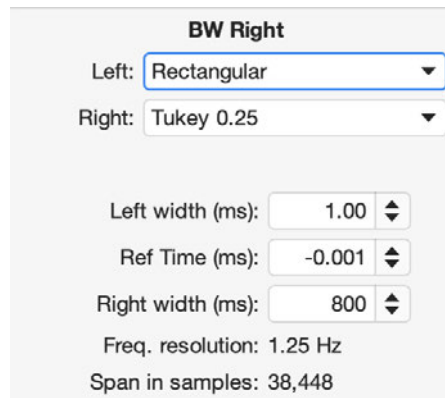


Abbildung 7.3. Fensterung der Impulsantwort mit *IR Windows*

Für die linke Seite wird *Rectangular* und für die rechte Seite *Tukey 0.25* gewählt. Dadurch entsteht ein Start ohne Fade-In und ein sanfter Fade-Out.

Der Startpunkt des Fensters wird über *Left width* definiert und auf 1 ms festgelegt. Dieser kurze Vorlauf ist ausreichend, um den Beginn der Impulsantwort zu erfassen, und vermeidet gleichzeitig unnötige Stille im Signal.

Die Länge des Fensters wird über *Right width* bestimmt. Zur korrekten Einstellung ist die Nachhallzeit (RT60) des gemessenen Raumes entscheidend. Im Fenster *Filtered IR* kann RT60 über das Schroeder-Integral abgeschätzt werden. Für Raum R1 lag die Nachhallzeit bei etwa 330 ms, weshalb *Right width* mit etwas Puffer auf 800 ms festgelegt wurde.

Im Fenster *IR Windows* kann dieses Zeitfenster über *Apply to all, keep reference time* auf alle gemessenen Impulsantworten angewendet werden.

Abschließend werden die Impulsantworten im Fenster *All SPL* über *Actions* exportiert. Bei den Exporteinstellungen muss das zuvor definierte Fenster berücksichtigt werden (siehe Abbildung 7.4).

## Ergebnis der Messungen

Im Rahmen der Messungen wurden insgesamt 140 Raumimpulsantworten (RIRs) in Raum A1 und sechs monaurale Impulsantworten in Raum R1 aufgenommen.

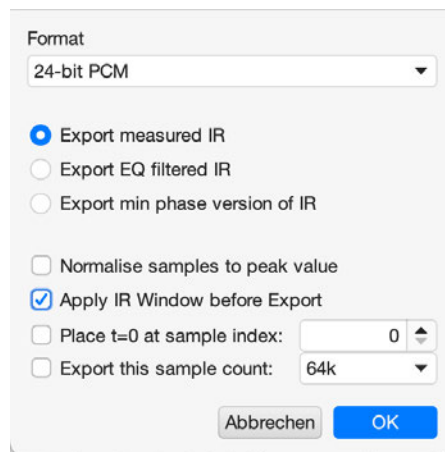


Abbildung 7.4. Einstellungen beim Export der Impulsantworten

Mit dem Kunstkopf *KU 100* wurden darüber hinaus 70 binaurale Impulsantworten (BRIRs) in A1 sowie sechs BRIRs in R1 gemessen. Die Lautsprecheranlage in A1 wurde dabei sowohl ohne DSP-Vorstufe, als auch mit dem DSP-Preset *A* betrieben, wodurch das System entzerrt war. Für die monauralen Messungen wurde zusätzlich mit Preset *C* und *E* aufgenommen. In Raum R1 wurden drei binaurale RIRs in einem Abstand von 20 cm und drei weitere in einem Abstand von 50 cm zum Mischpult aufgenommen.

Die Impulsantworten sind auf Zenodo<sup>3</sup> veröffentlicht und können dort als Datensatz für verschiedene Anwendungen heruntergeladen und genutzt werden.

<sup>3</sup><https://doi.org/10.5281/zenodo.17296510>

## B *HAW Binaural Lab*: Nutzung und Erweiterung

### Einleitung

*HAW Binaural Lab* ist ein auf Faltung basierender binauraler Renderer im VST3-Format, der die Wiedergabe mehrkanaliger Lautsprecheranordnungen über Kopfhörer ermöglicht. Das Plugin kombiniert gemessene Raumimpulsantworten (RIRs) mit Head-Related Transfer Functions (HRTFs), um eine realistische Simulation räumlicher Wiedergabesituationen zu erzeugen.

Die Software ist unter der Lizenz *GPL-3.0* auf GitHub veröffentlicht und modular aufgebaut, sodass neue Lautsprecherlayouts, Impulsantworten und HRTF-Datensätze auch ohne direkten Eingriff in den Quellcode hinzugefügt werden können. Für die binaurale Faltung nutzt das Plugin Komponenten der *Binaural Rendering Toolbox*<sup>4</sup> (González-Toledo et al., 2023).

Die integrierten HRTF-Dateien stammen aus verschiedenen öffentlichen Datensätzen und bilden sowohl HRTFs von verschiedenen Kunstkopf-Modellen, als auch von echten Menschen ab. Genutzt werden die *SADIE II Database* (Armstrong et al., 2018), der *ITA HRTF Database* (Bomhardt et al., 2016), sowie Nearfield- und Farfield-Messungen eines Kunstkopfes KU 100 (Arend et al., 2016; Bernschütz, 2020). Die enthaltenen Headphone-Transfer-Functions (HpTFs) sind dem Datensatz von Brinkmann et al. (2017) entnommen sind.

Bei dem Plugin handelt es sich um einen funktionalen Prototypen, der kontinuierlich weiterentwickelt wird. Die jeweils aktuelle Version des Quellcodes und der kompilierte VST3-Build stehen auf GitHub zur Verfügung:

<https://github.com/lfehrmann/HAW-Binaural-Lab>

---

<sup>4</sup><https://github.com/GrupoDiana/BRTLlibrary>

## Systemanforderungen

Zum Zeitpunkt der Veröffentlichung dieser Arbeit ist das Plugin für macOS mit Apple-Silicon- und Intel-Prozessoren entwickelt und getestet. Eine Windows-Version ist in Vorbereitung.

- **Betriebssystem:** macOS 12.0 oder neuer
- **Prozessor:** Apple M1/M2 oder Intel i5 (oder höher)
- **RAM:** mindestens 8 GB, empfohlen 16 GB
- **Samplerate:** bis 96 kHz
- **Plugin-Format:** VST3
- **Empfohlene DAWs:** Reaper, Ableton Live

## Installation und Setup

**1. Download und Installation** Die aktuelle VST3-Version kann aus dem GitHub-Repository heruntergeladen und in den lokalen VST3-Ordner verschoben werden. Dieser lautet typischerweise:

```
/Users/[Benutzername]/Library/Audio/Plug-Ins/VST3/
```

Beim nächsten Öffnen der Digital Audio Workstation (DAW) sollte das Plugin automatisch erkannt werden. Alternativ kann ein erneuter Scan des Plugin-Ordners manuell durchgeführt werden.

**2. Laden in der DAW** Nach erfolgreicher Installation kann das Plugin wie ein herkömmlicher Audioeffekt geladen werden. Es empfiehlt sich, das Plugin auf eine *Stereo-Audiospur* zu legen, um die binaurale Ausgabe über Kopfhörer zu erhalten. Alternativ kann es auch auf Mehrkanal-Spuren eingesetzt werden, beispielsweise zur Mischung von Ambisonics-Produktionen.

### 3. Erstes Setup

Nach dem ersten Start des Plugins:

1. In Stage 1 das gewünschte Lautsprecher-Layout über das Dropdown-Menü auswählen.
2. In Stage 2 über das Dropdown-Menü das gewünschte HRTF-Profil wählen.
3. In Stage 3 die HpTF des verwendeten Kopfhörer-Modells laden (falls vorhanden).

Ab diesem Punkt ist das Plugin einsatzbereit. Im Folgenden werden die internen Signalpfade, Parameter und Möglichkeiten zur Integration eigener Räume, Impulsantworten und HRTF-Datensätze erläutert.

## Aufbau und Parameter

Der interne Signalfluss von *HAW Binaural Lab* ist in drei aufeinanderfolgende Stufen gegliedert: *Room Convolution*, *Binaural Rendering* und *Headphone Compensation*. Für jede dieser Stufen stehen in der grafischen Benutzeroberfläche (GUI) spezifische Parameter zur Verfügung, über die sich die Signalverarbeitung gezielt anpassen lässt. Darüber hinaus existieren globale Parameter zur Steuerung der Kopforientierung sowie des Master-Outputs.

**Room Convolution** In dieser Stufe wird das Eingangssignal mit den gemessenen Raumimpulsantworten gefaltet, um die akustische Charakteristik der gewählten Lautsprecheranordnung zu simulieren.

- **Room Layout:** Dropdown-Menü zur Auswahl der verfügbaren Lautsprecher-Layouts bzw. Räume.
- **Add Delay:** Aktiviert die relativen Laufzeitunterschiede zwischen den einzelnen Lautsprechern, um realistische Laufzeitdifferenzen zu berücksichtigen.
- **Bypass:** Schaltet die gesamte Stufe ein oder aus.

**Binaural Rendering** Diese Stufe überführt das zuvor gefaltete Signal mithilfe von HRTFs in eine binaurale Wiedergabe, sodass der Raumklang über Kopfhörer dreidimensional wahrgenommen werden kann.

- **HRTF:** Dropdown-Menü zur Auswahl der verwendeten Head-Related Transfer Function (z. B. generisches KU100 oder benutzerdefiniertes SOFA-File).
- **Interpolation:** Aktiviert eine Interpolation zwischen benachbarten HRTF-Messpunkten, um Positionsübergänge – insbesondere bei Kopfbewegung – zu glätten.
- **ITD:** Simuliert interaurale Laufzeitdifferenzen (Interaural Time Difference). Funktioniert nur bei HRTFs, die Delay-Informationen in den Metadaten enthalten.
- **Parallax:** Modelliert den Effekt der akustischen Parallaxe bei Kopfbewegungen, wodurch sich die relative Position der Schallquellen dynamisch verändert.
- **Nearfield:** Aktiviert eine Nahfeld-Simulation, die den Pegel- und Spektralverlauf für nahe Schallquellen anpasst.
- **Bypass:** Schaltet die binaurale Verarbeitung ein oder aus.

**Headphone Compensation** In dieser Stufe wird das binaural gerenderte Signal durch eine Kopfhörer-Transferfunktion (HpTF) entzerrt, um die frequenzabhängige Färbung des verwendeten Kopfhörers zu kompensieren.

- **HpTF:** Dropdown-Menü zur Auswahl des entsprechenden Kopfhörermodells.
- **Bypass:** Schaltet die Kopfhörerkompensation ein oder aus.

**Weitere Parameter** Neben den Stages stehen zwei globale Parametergruppen zur Verfügung:

- **Kopforientierung (Yaw, Pitch, Roll):** Simuliert Drehungen und Neigungen des Kopfes und verändert entsprechend die räumliche Wahrnehmung der Schallquellen.

- **Master Gain:** Regelt den Ausgangspegel des Plugins.

## Verwendung mit Ambisonics

Um das Plugin für die Wiedergabe szenenbasierter Mehrkanalaufnahmen nutzen zu können, muss die Audiodatei zunächst dekodiert werden. Hierfür kann beispielsweise der *AllRADecoder*<sup>5</sup> verwendet werden. Die Audiospur in der DAW muss dabei mindestens die Kanalanzahl der Ambisonics-Aufnahme aufweisen. Im *AllRADecoder* sind anschließend die Lautsprecherpositionen gemäß den Angaben in der entsprechenden JSON-Datei einzutragen. Dabei ist unbedingt darauf zu achten, dass die Reihenfolge der Lautsprecher mit jener in der Layout-Datei übereinstimmt, damit das Routing korrekt funktioniert. Das Lautsprecher-Layout des Raums A1 ist bereits vorkonfiguriert und kann als *Preset* aus dem GitHub-Repository von *HAW Binaural Lab* heruntergeladen werden.

## Erweiterung durch eigene Layouts

*HAW Binaural Lab* ist modular aufgebaut und ermöglicht es, neue Lautsprecher-Layouts, Impulsantworten und Kopfhörerprofile hinzuzufügen, ohne Änderungen am Quellcode vorzunehmen. Alle Konfigurationsparameter werden in externen JSON- und Audiodateien definiert, die dem Plugin Informationen zu den jeweiligen Lautsprecherpositionen, den zugehörigen Impulsantworten sowie deren Gain- und Delay-Werten bereitstellen. Dieses Konzept gewährleistet eine einfache Erweiterbarkeit und erlaubt es, benutzerdefinierte Setups zu integrieren.

**Dateistruktur** Die konfigurierbaren Daten des Plugins sind in mehreren Unterordnern organisiert. Um die interne Struktur einzusehen, kann auf die VST3-Datei des Plugins mit der rechten Maustaste geklickt und „*Paketinhalt zeigen*“ gewählt werden. Im geöffneten Fenster führt der Pfad **Contents/Resources/** zum Unterordner **resources**. Darin liegen alle Konfigurationsdateien und Messdaten. Neue

---

<sup>5</sup><https://plugins.iem.at/docs/allradecoder/>

Dateien - etwa eigene Impulsantworten, Lautsprecher-Layouts oder personalisierte HRTFs - können dort direkt abgelegt werden. Um die korrekte Funktionsweise des Plugins zu gewährleisten müssen Audio-Dateien mit einer Abtastrate von 48 kHz vorliegen.

Die Verzeichnisstruktur ist in Listing 7.1 dargestellt:

```
HAW Binaural Lab.vst3/  
  '-- Contents/  
    '-- Resources/  
      '-- resources/  
        |-- layouts/  
          | '-- example_layout.json (kommentierte Vorlage)  
        |-- hrtf/  
        |-- hptf/  
        |-- irs/  
        |-- sos/
```

*Listing 7.1.* Dateistruktur des Plugins

**Ordner-Inhalte** Im Unterordner `layouts/` befinden sich die Lautsprecherkonfigurationen. Jede JSON-Datei beschreibt ein spezifisches Setup mit Positionsdaten, referenzierten Impulsantworten, Gain-Werten und optionalen Laufzeitdifferenzen. Eine kommentierte Beispieldatei (`example_layout.json`) dient als Vorlage und Anleitung für die Erstellung eigener Layouts.

Die Unterordner und ihr Inhalt im Überblick:

- **layouts/** – JSON-Dateien mit Lautsprecheranordnungen.
- **hrtf/** – Head-Related Transfer Functions (SOFA-Dateien).
- **hptf/** – Headphone-Transfer-Functions zur frequenzabhängigen Entzerrung.
- **irs/** – Gemessene Raumimpulsantworten, die in den Layout-Dateien referen-

ziert werden.

- **sos/** – Zusatzdaten zur Simulation des Nearfield-Effekts.

**Laden und Integration der Dateien** Beim Start durchsucht das Plugin die relevanten Verzeichnisse und lädt die dort abgelegten **JSON**- und Audiodateien. Neu hinzugefügte Dateien werden beim nächsten Start automatisch erkannt und in den Dropdown-Menüs unter *Room Layout*, *HRTF* bzw. *HpTF* zur Auswahl angeboten.

## C Magnitude-Spektren

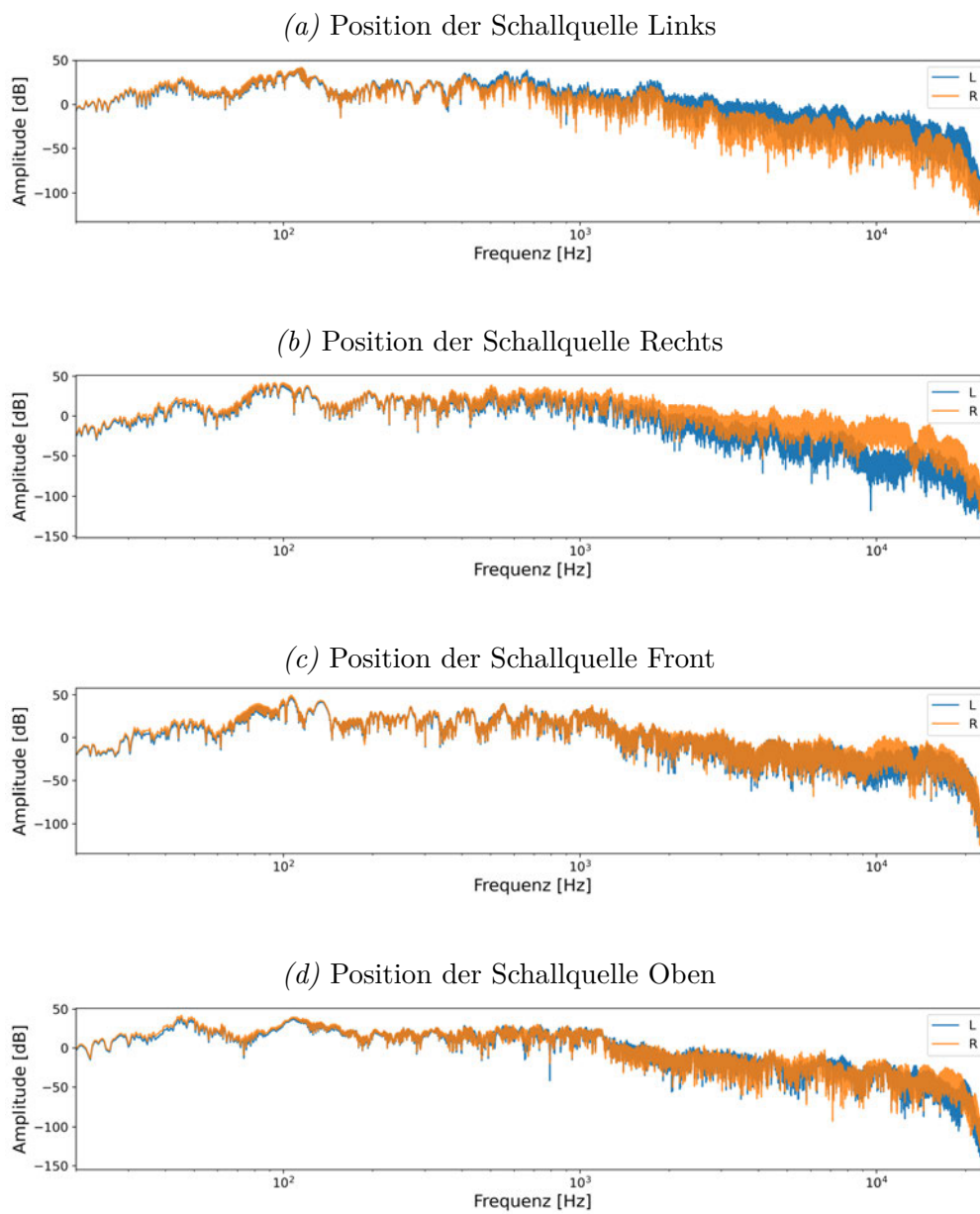


Abbildung 7.5. Darstellung der Magnitudenverläufe der binaural gemessenen Impulsantworten für vier unterschiedliche Lautsprecherpositionen im Raum.

# Literatur

- Ahnert, W., & Weinzierl, S. (2025). Raumakustik. In *Handbuch der audiotechnik* (S. 215–260). Springer.
- Arend, J. M., Neidhardt, A., & Pörschmann, C. (2016). Measurement and Perceptual Evaluation of a Spherical Near-Field HRTF Set (Messung und perzeptive Evaluierung eines sphärischen Satzes von Nahfeld-HRTFs). *29th Tonmeistertagung*.
- Armstrong, C., Thresh, L., Murphy, D., & Kearney, G. (2018). A perceptual evaluation of individual and non-individual HRTFs: A case study of the SADIE II database. *Applied Sciences*, 8(11), 2029.
- Aussal, M., Alouges, F., & Katz, B. (2012). HRTF interpolation and ITD personalization for binaural synthesis using spherical harmonics. *Audio Engineering Society Conference: UK 25th Conference: Spatial Audio in Today's 3D World*.
- Bartetzki, A. (2025). Klangsynthese. In *Handbuch der Audiotechnik* (S. 149–177). Springer.
- Bernschütz, B. (2020). Spherical far-field HRIR compilation of the Neumann KU100. *Proceedings of the 39th Fortschritte der Akustik (DAGA)*, 592–595.
- Bomhardt, R., de la Fuente Klein, M., & Fels, J. (2016). A high-resolution head-related transfer function and three-dimensional ear model database. *Proceedings of Meetings on Acoustics*, 29(1), 050002.
- Borish, J. (1984). Extension of the image model to arbitrary polyhedra. *The Journal of the Acoustical Society of America*, 75(6), 1827–1836.
- Brinkmann, F., Lindau, A., Weinzierl, S., Geissler, G., van de Par, S., Müller-Trapet, M., Opdam, R., & Vorländer, M. (2017). The FABIAN head-related transfer function data base. *Berlin: Technische Universität Berlin*.

- Brinkmann, F., & Pike, C. (2025). Binauraltechnik. In *Handbuch der Audiotechnik* (S. 615–637). Springer.
- Brughera, A., Dunai, L., & Hartmann, W. M. (2013). Human interaural time difference thresholds for sine tones: The high-frequency limit. *The Journal of the Acoustical Society of America*, *133*(5), 2839–2855.
- Brungart, D. S., & Rabinowitz, W. M. (1999). Auditory localization of nearby sources. Head-related transfer functions. *The Journal of the Acoustical Society of America*, *106*(3), 1465–1479.
- Carlini, A., Bordeau, C., & Ambard, M. (2024). Auditory localization: a comprehensive practical review. *Frontiers in Psychology*, *15*, 1408073.
- Choadhry, M., Eadie, J., & Kokaram, A. (2019). Headphone Filtering in Spectral Domain. *2019 30th Irish Signals and Systems Conference (ISSC)*, 1–6.
- Cremer, L., & Müller, H. A. (1948). Die wissenschaftlichen grundlagen der raumakustik.
- Cuevas-Rodríguez, M., Picinali, L., González-Toledo, D., Garre, C., de la Rubia-Cuestas, E., Molina-Tanco, L., & Reyes-Lecuona, A. (2019). 3D Tune-In Toolkit: An open-source library for real-time binaural spatialisation. *PloS one*, *14*(3), e0211899.
- Duda, R. O., & Martens, W. L. (1998). Range dependence of the response of a spherical head model. *The Journal of the Acoustical Society of America*, *104*(5), 3048–3058.
- Fuchs, H. V. (2017). *Raum-akustik und Lärm-minderung: Konzepte mit Innovativen Schallabsorbieren und-dämpfern*. Springer-Verlag.
- Gelfand, S. A. (2017). *Hearing: An introduction to psychological and physiological acoustics*. CRC Press.
- González-Toledo, D., Molina-Tanco, L., Cuevas-Rodríguez, M., Majdak, P., & Reyes-Lecuona, A. (2023). The Binaural Rendering Toolbox. A virtual laboratory for reproducible research in psychoacoustics. *Forum Acusticum*.
- Guastavino, C., & Katz, B. F. (2004). Perceptual evaluation of multi-dimensional spatial audio reproduction. *The Journal of the Acoustical Society of America*, *116*(2), 1105–1115.

- Hartmann, W. M., Rakerd, B., Crawford, Z. D., & Zhang, P. X. (2016). Transaural experiments and a revised duplex theory for the localization of low-frequency tones. *The Journal of the Acoustical Society of America*, *139*(2), 968–985.
- Hoene, C., Patino Mejia, I. C., & Cacerovski, A. (2017). MySofa—Design Your Personal HRTF. *Audio Engineering Society Convention 142*.
- Jin, C., Kan, A., & van Schaik, A. (2006). Psychoacoustic evaluation of a new method for simulating near-field virtual auditory space. *Audio Engineering Society Convention 120*.
- Kuttruff, H. (1995). A simple iteration scheme for the computation of decay constants in enclosures with diffusely reflecting boundaries. *The Journal of the Acoustical Society of America*, *98*(1), 288–293.
- Makous, J. C., & Middlebrooks, J. C. (1990). Two-dimensional sound localization by human listeners. *The journal of the Acoustical Society of America*, *87*(5), 2188–2200.
- Maschke, C., & Widmann, U. (2012). The effects of sound on humans. In *Handbook of engineering acoustics* (S. 69–86). Springer.
- McAnally, K. I., & Martin, R. L. (2014). Sound localization with head movement: implications for 3-d audio displays. *Frontiers in neuroscience*, *8*, 79254.
- Mertins, A. (1996). *Signaltheorie*. Springer.
- Mills, A. W. (1958). On the minimum audible angle. *The Journal of the Acoustical Society of America*, *30*(4), 237–246.
- Møller, H. (1992). Fundamentals of binaural technology. *Applied acoustics*, *36*(3-4), 171–218.
- Møller, H., Sørensen, M. F., Hammershøi, D., & Jensen, C. B. (1995). Head-related transfer functions of human subjects. *Journal of the Audio Engineering Society*, *43*(5), 300–321.
- Möser, M. (2009). *Messtechnik der Akustik*. Springer-Verlag.
- Möser, M. (2012). *Technische Akustik*. Springer-Verlag.
- Noisternig, M., Musil, T., Sontacchi, A., & Holdrich, R. (2003). 3D binaural sound reproduction using a virtual ambisonic approach. *IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS'03. 2003*, 174–178.

- Nykänen, A., Zedigh, A., & Mohlin, P. (2013). Effects on localization performance from moving the sources in binaural reproductions. *International Congress and Exposition on Noise Control Engineering: 15/09/2013-18/09/2013*, 4, 3193–3201.
- Otani, M., Hirahara, T., & Ise, S. (2009). Numerical study on source-distance dependency of head-related transfer functions. *The Journal of the Acoustical Society of America*, 125(5), 3253–3261.
- Otte, R. J., Agterberg, M. J., Van Wanrooij, M. M., Snik, A. F., & Van Opstal, A. J. (2013). Age-related hearing loss and ear morphology affect vertical but not horizontal sound-localization performance. *Journal of the Association for Research in Otolaryngology*, 14(2), 261–273.
- Poirier-Quinot, D., & Katz, B. F. (2018). The Anaglyph binaural audio engine. *Audio Engineering Society Convention 144*.
- Politis, A., Tervo, S., & Pulkki, V. (2018). COMPASS: Coding and multidirectional parameterization of ambisonic sound scenes. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6802–6806.
- Rayleigh, L. (1907). XII. On our perception of sound direction. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(74), 214–232.
- Romblom, D., & Cook, B. (2008). Near-field compensation for HRTF processing. *Audio Engineering Society Convention 125*.
- Room EQ Wizard*. (2025). John Mulcahy. Verfügbar 8. Oktober 2025 unter <https://www.roomeqwizard.com>
- Schneider, M. (2025). Mikrofone: Wandler und Bauformen. In *Handbuch der Audiotechnik* (S. 313–341). Springer.
- Schoeps*. (2025). Schoeps GmbH. Verfügbar 8. Oktober 2025 unter <https://schoeps.de/wissen/knowledge-base/technische-grundlagen/diffusfeldentzerrung.html>
- Schoeps MK2*. (2025). Schoeps GmbH. Verfügbar 8. Oktober 2025 unter <https://schoeps.de/produkte/colette/kapseln/kugeln/mk-2.html>
- Sengpiel*. (2025). Eberhard Sengpiel. Verfügbar 8. Oktober 2025 unter <https://sengpielaudio.com/DerOhrabstand-Welcher.pdf>

- Shahid, F., Javeri, N., Jain, K., & Badhwar, S. (2018). AI DevOps for Large-Scale HRTF Prediction and Evaluation: An End to End Pipeline. *Audio Engineering Society Conference: 2018 AES International Conference on Audio for Virtual and Augmented Reality*.
- Smith, S. W., et al. (1997). The scientist and engineer's guide to digital signal processing.
- Stevens, S. S., & Newman, E. B. (1936). The localization of actual sources of sound. *The American journal of psychology*, 48(2), 297–306.
- Sun, X. (2021). Immersive audio, capture, transport, and rendering: A review. *APSIPA Transactions on Signal and Information Processing*, 10, e13.
- Sunder, K. (2021). Binaural audio engineering. In *3D Audio* (S. 130–159). Routledge.
- Sunder, K., Jain, K., Cohen, R., & Lurssen, G. (2022). Reliable and Trustworthy Virtual Production Workflow for Surround and Atmos. *Audio Engineering Society Convention 153*.
- Vorländer, M. (2018). Digitale Signalverarbeitung in der Messtechnik. In *Digitale Signalverarbeitung in der Messtechnik* (S. 1–28). Springer.
- Vorländer, M., & Feistel, S. (2025). Raumakustische Simulation und Auralisation. In *Handbuch der Audiotechnik* (S. 289–309). Springer.
- Weinzierl, S. (2025). Audiosignale und -systeme. In *Handbuch der Audiotechnik* (S. 3–24). Springer.
- Wu, Y.-H., Murakami, S., & Roginska, A. (2019). The Effect of Transitioning Between Individualized and Generic HRTFs on Localization Performance in a Virtual Environment. *Audio Engineering Society Conference: 2019 AES International Conference on Immersive and Interactive Audio*.
- Zölzer, U. (2005). *Digitale Audiosignalverarbeitung*. B. G. Teubner.
- Zölzer, U., & Lerch, A. (2025). Digitale Audio-Effekte. In *Handbuch der Audiotechnik* (S. 641–662). Springer.
- Zotkin, D. N., Duraiswami, R., & Davis, L. S. (2004). Rendering localized spatial audio in a virtual auditory space. *IEEE Transactions on multimedia*, 6(4), 553–564.


## Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit eigenständig und ohne unerlaubte Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und alle wörtlich oder sinngemäß übernommenen Inhalte als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

---

(Ort, Datum)

---



(Unterschrift)