Hochschule der Medien Stuttgart

Fakultät Electronic Media

Entwurf und prototypische Implementierung eines Systems zur binauralen Simulation von Regieräumen

Masterarbeit im Studiengang Audiovisuelle Medien
zur Erlangung des akademischen Grades
Master of Engineering

Autor: Massimo Ehrhard
MatNr. 35429

vorgelegt am: 16.04.2021

Erstprüfer: Prof. Dr. Frank Melchior
Zweitprüfer: Prof. Oliver Curdt

Ehrenwörtliche Erklärung

Hiermit versichere ich, Massimo Ehrhard, ehrenwörtlich, dass ich die vorliegende Masterarbeit mit dem Titel: „Entwurf und prototypische Implementierung eines Systems zur binauralen Simulation von Regieräumen“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen

$Unterschrift$: $Ort, Datum$:

Zusammenfassung

In dieser Arbeit wird ein funktionsfähiges Gesamtsystem entworfen, dass die binaurale Simulation von Regieräumen über Kopfhörer ermöglicht. Nach einer kurzen Darstellung der begrifflichen und technischen Grundlagen der Arbeit werden zunächst die Funktionsweisen des räumlichen Hörens dargestellt, welche die Simulation erfassen muss. Auf deren Grundlage werden verschiedene Ansätze des binauralen Renderings diskutiert und schlussendlich das im System gewählte binaurale Rendering definiert. Im weiteren Verlauf werden die nötigen mathematischen Grundlagen und Systemkomponenten methodisch unter Zuhilfenahme wissenschaftlicher Literatur dargestellt und analysiert, ehe das Gesamtsystem in drei Systemmodulen mit Hilfe von Anforderungsanalysen entworfen wird. Die Systemmodule werden im Folgenden mit den Entwicklungsumgebungen Max/MSP 8 mit Spat-5 und MATLAB prototypisch umgesetzt und die Funktionsfähigkeit des Gesamtsystems anhand der beispielhaften Simulation eines Regieraums geprüft. Die erreichten Parameter des Systems werden schlussendlich dokumentiert, mit den Anforderungen verglichen und kritisch hinterfragt.

Abstract

In this thesis, a functional system is designed that enables the binaural simulation of control rooms via headphones. After a short presentation of the conceptual and technical basics of the work, first the functionalities of spatial hearing are presented, which the simulation has to capture. Based on these, different approaches of binaural rendering are discussed and finally the binaural rendering chosen in the system is defined. In the following, the necessary mathematical basics and system components are methodically presented and analyzed with the help of scientific literature, before the overall system is designed in three system modules with the help of requirements analysis. The system modules are then prototypically implemented using the development environments Max/MSP 8 with Spat-5 and MATLAB, and the functionality of the system is tested using the exemplary simulation of a control room. The achieved parameters of the system are finally documented, compared with the requirements and critically scrutinized.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

BRS	Binaural Room Scanning
IRT	Institut für Rundfunktechnik
IKL	Im-Kopf-Lokalisation
ITD	Interaurale Zeitdifferenz
ILD	Interaurale Pegeldifferenz
ILDs	Interaurale Pegeldifferenzen
TSL	Gesamtsystemlatenz
BRIR	Binaurale Raumimpulsantwort
BRIRs	Binaurale Raumimpulsantworten
GA	Geometrische Akustik
SNR	Signal-Rausch-Verhältnis
HRTF	Kopfbezogene Übertragungsfunktion
HRTFs	Kopfbezogene Übertragungsfunktionen
HRIR	Kopfbezogene Impulsantwort
HRIRs	Kopfbezogene Impulsantworten
ITDG	Anfangszeitlücke
EDT	Anfangsnachhallzeit
EDC	Frühe Abklingkurve
OSC	Open Sound Control
SMPTE	Society of Motion Picture and Television Engineers
JND	Differentielle Wahrnehmbarkeitsschwelle (Just Noticeable Difference)
mTSL	Minimale Gesamtsystemlatenz
IMU	Inertiale Messeinheit
VAE	Virtuelle auditorische Umgebung
HATS	Kopf- und Rumpfsimulator
SNR	Signal-Rausch-Abstand
SNRs	Signal-Rausch-Abstände
PNR	Spitzen-Rausch-Abstand

1 Einleitung

Virtual reality, *augmented reality* und *mixed reality* sind im Jahr 2021 Thematiken, denen weit über den Medienbereich Aufmerksamkeit geschenkt wird; sie sind in der marktwirtschaftlichen Realität von Unternehmen aus verschiedenen Wirtschaftszweigen angekommen. Sie sind neben Unterhaltungs- auch zu Marketinginstrumenten geworden. Als audiovisuelle Medientechnologien sind diese Verfahren nicht nur auf visuelle Reize beschränkt, sondern funktionieren auch auf auditiver Ebene, wobei die Letztere oftmals in Form von binauraler Wiedergabe umgesetzt wird, welche die Realitätsnähe durch die prinzipielle Nachbildung des natürlichen Hörens deutlich erhöht und somit die Qualität der virtuellen Umgebung verbessern kann.

Ebendiese im vorangegangenen Absatz genannten Gründe der Entstehung neuer medialer Darstellungsformen von audiovisuellen Inhalten in Verbindung mit dem mooreschen Gesetz, welches besagt, dass sich die Anzahl an Transistoren, die in einen integrierten Schaltkreis festgelegter Größe passen, etwa alle zwei Jahre verdoppelt (Schanze, 2016), haben dazu geführt, dass die computergestützte Binauralsynthese in den vergangenen fünfzehn Jahren stark an Relevanz in der Gesellschaft gewonnen und dadurch auch verstärkte Aufmerksamkeit in Forschung und Wissenschaft erhalten hat. Dies führte zu vielfältigen Untersuchungen bezüglich der wahrgenommenen Qualität binauraler Auralisation; wobei es an dieser Stelle auch nicht immer um die bestmöglich zu erreichende perzeptive Qualität geht, sondern vielmehr um eine Abwägung zwischen perzeptiver Qualität und benötigter Rechenleistung.

Neben diesen eher multimedial zu verstehenden Medienformen virtueller Realitäten ist es aber auch *nur* die einfache binaurale Simulation und Auralisation von akustischen Umgebungen, welche mit den heutigen Rechenkapazitäten problemlos sogar auf mobilen Endgeräten umgesetzt werden kann; zum Beispiel über den mobilen Kopfhörer der Firma *Steven Slate Audio*, welcher im Oktober 2020 veröffentlicht wurde, und eine Darstellung der Lautsprecherwiedergabe in Tonregieräumen allein über Kopfhörer ohne weitere Anbindung an einen externen Prozessor ermöglicht (Slate, 2020).

All dies zusammen kann dem professionellen Anwender als auch Consumer einen echten Mehrwert bieten, sodass in dieser Arbeit eine Umgebung zur Simulation bzw. Auralisation von Tonregieräumen und Mischkinos entworfen und prototypisch implementiert werden soll. Dabei steht eine Anbindung an die hochschuleigene Infrastruktur zu Lehr- und Forschungszwecken im Vordergrund. Die Hochschule der Medien Stuttgart besitzt eine Vielzahl an Räumen zur Audioproduktion, welche raumakustisch optimiert worden sind und den Studenten eine optimale Umgebung für die Tonproduktion und -reproduktion bieten. Dies sind

unter anderem Tonregieräume in Stereo und Surround, Räume für die Lautsprecher-basierte 3D-Audioproduktion, ein Vorführ- bzw. Mischkino sowie diverse weitere Produktionsräume. Diese Räume sind den Studenten jedoch nicht jederzeit zugänglich, was vor allem im Zuge der Corona-Pandemie in den Jahren 2020 und 2021 zu einem deutlichen Einschnitt der Nutzungsmöglichkeiten geführt und den Studenten und Mitarbeitern der Hochschule die Wichtigkeit dieser Räumlichkeiten nochmals dargestellt hat. Des Weiteren sind die Kapazitäten der für Audioproduktionen nutzbaren Räumlichkeiten zum Ende des Semesters aufgrund von Abgabefristen regelmäßig erschöpft. Eine Simulation und Auralisation dieser Räumlichkeiten über Kopfhörer kann den Studenten somit auch in den genannten Phasen noch einen virtuellen akustischen Zugang gewähren. Neben diesen eher produktionsbezogenen praktischen Aspekten ist es auch die Nutzbarmachung zu Forschungszwecken im Bereich der Raumakustik und Psychoakustik, welche als Motivation dieser Arbeit gilt.

2 Zielsetzungen

An der Hochschule der Medien gibt es eine Vielzahl von Räumen für die Tonproduktion und -reproduktion. Dies sind unter anderem Regieräume für die Tonproduktion in Stereo und Surround, Räume für die Lautsprecher-basierte 3D-Audio-Produktion, ein Vorführ- und Mischkino sowie diverse weitere Produktionsräume.

Im Rahmen dieser Arbeit soll ein funktionsfähiges Gesamtsystem entworfen werden, das die binaurale Simulation von typischen Tonregieräumen und Mischkinos - wie sie an der Hochschule der Medien Stuttgart zu finden sind - über Kopfhörer ermöglicht. Dieses System soll im Rahmen der Forschung und Lehre an der Hochschule nutzbar sein und den Ansprüchen an ein solches System genügen. Des Weiteren ist eine Skalierbarkeit auf Lautsprechersetups mit vielen Lautsprechern wünschenswert.

Das System soll mit den Entwicklungsumgebungen Max/MSP 8 mit Spat-5 sowie MATLAB prototypisch umgesetzt werden und die Funktionsfähigkeit des Gesamtsystems anhand einer beispielhaften Simulation eines Regieraums geprüft werden. Die erreichten Parameter des Systems sollen schlussendlich dokumentiert, mit den Anforderungen verglichen und kritisch hinterfragt werden.

.

3 Grundlagen

Das 3. Kapitel dieser Arbeit enthält Grundlagen für das Verständnis der weiteren Kapitel. Diese haben nicht direkt etwas mit der binauralen Erfassung sowie binauraler Wiedergabe zu tun, welche im Kapitel 4 sowie bei der Vorstellung der Methodik in Kapitel 5 ausführlicher behandelt werden und folglich abgesetzt von den Grundlagen dieses Kapitels sind.

3.1 Kopfbezogenes Koordinatensystem

In der Abbildung 3.1 ist das kopfbezogene Koordinatensystem inklusive der Ebenen, welche es aufspannt, dargestellt; dies wird im Rahmen dieser Arbeit genutzt. Es verläuft entlang der sogenannten *interauralen Achse* (y-Achse) zwischen den beiden Gehörkanaleingängen, sodass sich der Ursprung des Koordinatensystems in der Mitte des Kopfes befindet und die positive x-Achse frontal Richtung Nase den Kopf verlässt. Die Horizontalebene wird durch die interaurale Achse und die Unterkanten der Augenhöhlen aufgespannt, die Frontalebene enthält die interaurale Achse und verläuft orthogonal zur Horizontalebene, wohingegen die Medianebene orthogonal sowohl zur Horizontalebene als auch zur Frontalebene ist. Die Position einer Schallquelle im dreidimensionalen Raum wird mit Hilfe der mathematisch wohldefinierten Kugelkoordinaten Azimut ϕ, Elevation θ, und Radius bzw. Entfernung r dargestellt. Zu beachten ist, dass im Bereich der Tontechnik der Azimut ϕ oft statt im mathematisch positiven Drehsinn im Uhrzeigersinn gezählt wird; dies wird in dieser Arbeit jedoch vermieden und der mathematisch positive Drehsinn wird angewendet. (Weinzierl, 2008)

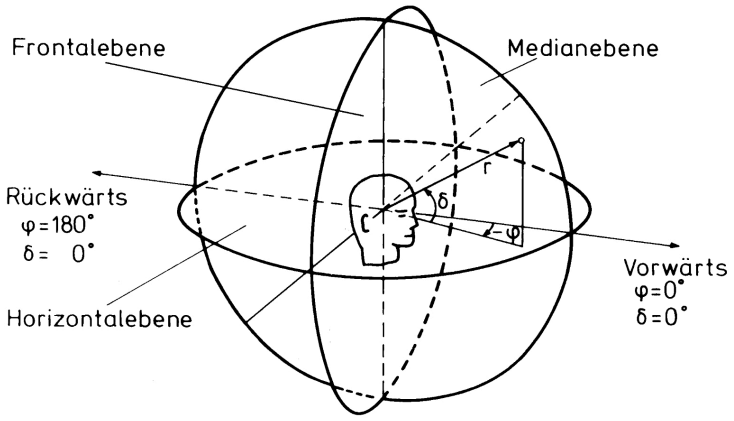

Abbildung 3.1: Kopfbezogenes Koordinatensystem (Weinzierl, 2008)

3.2 Freiheitsgrade

Die unabhängig voneinander möglichen Bewegungen eines starren Körpers im dreidimensionalen Raum werden durch sechs Freiheitsgrade beschrieben. Dabei kann in *translatorische Bewegung* (alle Punkte erfahren die gleiche Verschiebung) und *rotatorische Bewegung* (alle Punkte bewegen sich mit gleicher Winkelgeschwindigkeit um eine gemeinsame Achse) unterteilt werden, dabei gilt (Goldstein et al., 2006):

Translation

- vor/zurück entlang der x-Achse (Längsachse)
- links/rechts entlang der y-Achse (Querachse)
- auf/ab entlang der z-Achse (Vertikalachse)

Rotation

- von Seite zu Seite kippen (Rollen) an der x-Achse (Längsachse)
- vor und zurück kippen (Nicken) an der y-Achse (Querachse)
- links und rechts drehen (Gieren) an der z-Achse (Vertikalachse)

3.3 Position und Lage

In räumlichen Bezugssystemen ist zwischen der Position und der Lage zu unterscheiden, jedoch wird häufig zwischen diesen Begriffen nicht genau differenziert. Es soll Folgendes gelten (Goldstein et al., 2006):

- Die Position beschreibt den Ort im Raum, der durch Translation geändert werden kann.
- Die Lage beschreibt die Orientierung im Raum, die durch Rotation geändert werden.

3.4 Head-Tracking

Head-Tracking beschreibt technische Einrichtungen der Erfassung der Lage (Orientierung) des Kopfes. In diesem Kapitel sollen die mathematischen Grundlagen der Lagebestimmung beschrieben werden, dies unter dem besonderen Aspekt der Lagebestimmung des Kopfes. In diesem Zusammenhang gibt es verschiedene mathematische Modelle, wobei die Euler-/Kardan-Winkel sowie Quaternionen die wesentlichen - auch im Zusammenhang dieser Arbeit genutzten - Beschreibungen sind und folglich im Folgenden vorgestellt werden.

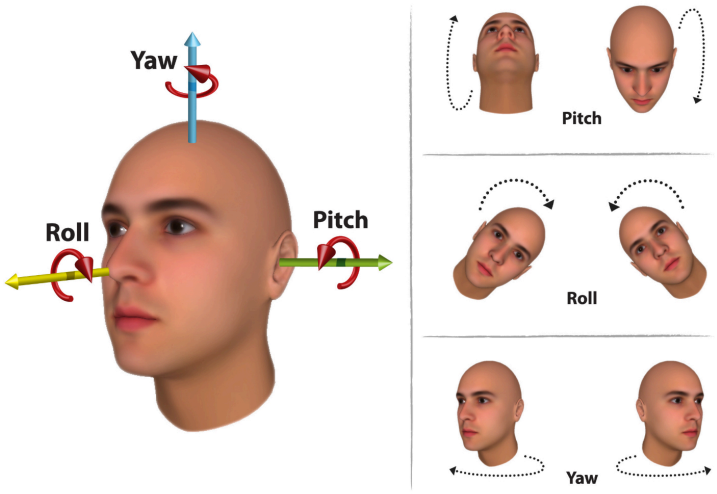

Abbildung 3.2: Gieren (engl. *yaw*), Nicken (engl. *pitch*) und Rollen (engl. *roll*) eines menschlichen Kopfes (Arcoverde Neto et al., 2014)

3.4.1 Euler- und Kardan-Winkel

Die Euler-Winkel (eulerschen Winkel) oder Kardan-Winkel sind ein Satz dreier unabhängiger Winkel, mit denen die Lage eines festen Körpers im dreidimensionalen Raum beschrieben werden kann; Letztere wird aus einer beliebigen anderen durch eine Abfolge dreier Drehungen um spezielle Achsen erzeugt (Goldstein et al., 2006). Die Drehung eines Körpers wird als aktive Drehung bezeichnet und stellt hierbei eine Abbildung über eine geometrische Transformation dar; wird hingegen ein ganzes Koordinatensystem gedreht, spricht man von einer Koordinatentransformation und einer passiven Drehung. Es wird unterschieden, ob es sich um sogenannte extrinsische Drehungen um die festen Raumkoordinatenachsen des kartesischen Koordinatensystems oder intrinsische Drehungen um die Koordinatenachsen des körperfesten kartesischen Koordinatensystems handelt, welche sich bei Drehung des Körpers verändern. Des Weiteren wird zwischen zwei verschiedenen Konventionen bei der Drehreihenfolge unterschieden, die die Unterscheidung in Euler- und Kardan-Winkel mit sich bringt: Bei den Euler-Winkeln findet die erste und die dritte Drehung um die gleiche Koordinatenachse statt (z. B. Drehung um z-Achse, x-Achse, z-Achse), bei den Kardan-Winkeln wird jede der drei Drehungen um eine andere Koordinatenachse durchgeführt (z.B. Drehung um z-Achse, y-Achse, x-Achse). Im Rahmen dieser Arbeit werden die Kardan-Winkel und deren Drehreihenfolge aus der Konvention in der Luft- und Raumfahrttechnik (DIN 9300) bzw. aus dem Automobilbau (DIN ISO 8855) genutzt; dabei wird das zu Beginn der Gesamtrotation raumbezogene Koordinatensystem bei jeder Drehung in ein anderes körperbezogenes Koordinatensystem überführt, somit findet bei dieser intrinsischen Betrachtung nur die erste Drehung an einer Koordinatenachse den Raumkoordinatensystems statt. Die Euler-/Kardan-Winkel definieren sich wie folgt:

- Gier-Winkel ψ (engl. *yaw*): Drehung um die z-Achse (Vertikalachse)
- Nick-Winkel ϑ (engl. *pitch*): Drehung um die y-Achse (Querachse)

- Roll-Winkel φ (engl. *roll*): Drehung um die x-Achse (Längsachse)

Wie diese Winkel an einem menschlichen Kopf angewendet werden können, zeigt Abbildung Abbildung 3.2. Sie repräsentieren dabei exakt die drei rotatorischen Freiheitsgrade, mit Gier-, Nick- und Roll-Winkel kann die Lage eindeutig beschrieben werden; und bilden somit eine sehr anschauliche Verbindung zwischen der Realität und der mathematischen Darstellung.

Die Drehungen um die Euler-/Kardan-Winkel können mit Hilfe von einfachen elementaren Drehmatrizen, deren Einträge Sinus- und Kosinus-Werte der Letzteren sind, beschrieben werden; dabei spricht man bei aktiver Drehung eines Körpers von Abbildungsmatrizen, bei Drehungen des Koordinatensystems bzw. passiven Drehungen eines Körpers von Koordinatentransformationsmatrizen. Im Folgenden sind beispielhaft die Drehmatrizen um die drei Koordinatenachsen des kartesischen Koordinatensystems dargestellt (Gleichung 3.4, Gleichung 3.2 und Gleichung 3.3):

$$D_{\mathrm{x}}(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{3.1}$$

$$D_{\mathrm{y}}(\vartheta) = \begin{bmatrix} cos(\vartheta) & 0 & sin(\vartheta) \\ 0 & 1 & 0 \\ -sin(\vartheta) & 0 & cos(\vartheta)) \end{bmatrix} \tag{3.2}$$

$$D_{\mathrm{z}}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Diese Drehmatrizen können je nach Drehreihenfolge in eine entsprechende Rotationsmatrix via Matrixmultiplikation überführt werden, welche die Gesamtrotation beschreibt; so auch um eine Gesamtrotation mit Hilfe der Kardan-Winkel an den mitdrehenden körperbezogenen Achsen (momentane Achsen) in der intrinsischen Drehreihenfolge z-y'-x" (Gier-Nick-Roll-Winkel) zu beschreiben (Eberhard & Ziegler, 2021):

$$R = D_{\mathrm{z}}(\psi) \cdot D_{\mathrm{y}}(\vartheta) \cdot D_{\mathrm{x}}(\varphi) \tag{3.4}$$

Computer können diese trivialen Matrizenrechnungen zwar sehr schnell durchführen, jedoch kommt es aufgrund der begrenzten Genauigkeit bei der Darstellung von Gleitkommazahlen zu Rundungsfehlern, welche sich als Fehler bei wiederholten Matrix-Multiplikationen aufsummieren und folglich den Gesamtfehler vergrößern (Hofmann, 2009). Betrachtet man die Theorie hinter den Berechnungen mit den Euler-/Kardan-Winkeln kommt es zu zwei kritischen Zuständen, die jedoch bei dem im Rahmen dieser Arbeit betrachteten Head-Tracking keine Relevanz haben, trotzdem aus Gründen der Vollständigkeit an dieser Stelle erwähnt sein sollen: Die kardanische Blockade (engl. *Gimbal Lock*) tritt auf, wenn durch Drehung zwei Drehachsen parallel zueinander liegen und das System damit einen Freiheitsgrad verliert. Des Weiteren existiert eine Ambiguität, denn eine Endposition kann durch unterschiedliche Ro-

tationen erreicht werden; ein Rollen um 180° wäre z. B. auch durch ein Gieren um 180° in Verbindung mit einem Nicken um 180° darstellbar.

3.4.2 Quanternionen und Achsenwinkel

Neben der Berechnung der Rotation eines Körpers mit Hilfe der Eulerschen Geometrie im reellen dreidimensionalen Koordinatenraum $\mathbb{R}^3$ ist dies auch durch die sogenannten Quanternionen möglich; diese stellen eine Erweiterung der komplexen Zahlen dar und spielen eine wichtige Rolle für die Darstellung von Drehungen im $\mathbb{R}^3$ (Lust, 2001).

Der Begriff *Quaternione* stammt vom lateinischen Wort *quattuor* ab und heißt übersetzt *vier*. Die Quaternionen $\mathbb{H}$, der erste Buchstabe des Namens ihres Entdeckers William Rowan Hamilton, stellen einen vierdimensionalen Vektorraum dar (Ledoux, 2011):

$$\mathbb{H} = \{q_w + q_x i + q_y j + q_z k \mid q_w, q_x, q_y, q_z \in \mathbb{R}\} \tag{3.5}$$

Dieser Vektorraum wird von der Basis $\{1, i, j, k\}$ aufgespannt. Eine Quaternion $q = q_w + q_x i + q_y j + q_z k$ ist ähnlich konstruiert wie eine komplexe Zahl $a + ib$ mit $a, b \in \mathbb{R}$ und $i^2 = -1$. Sie besteht ebenso aus einem eindimensionalen Realteil $q_w \in \mathbb{R}$, jedoch ist der Imaginärteil nicht nur eindimensional, sondern dreidimensional. Die drei Basiselemente i, j, k erfüllen die sogenannten Hamilton-Regeln für ihre Multiplikation, welche an dieser Stelle jedoch nicht weiter betrachtet werden sollen.

Drehungen im $\mathbb{R}^3$ werden mit Hilfe von Quaternion-Multiplikationen durchgeführt. Eine Quaternion, welche lediglich eine Drehung darstellen soll, muss normiert werden, sodass gilt:

$$q \cdot \overline{q} = \overline{q} \cdot q = 1 \tag{3.6}$$

Die Drehung mit Hilfe eines solchen normierten Quaternions q multipliziert mit einem Punkt p im Raum und dem konjungierten Quaternion $\overline{q}$ ergibt die neue Position für den Punkt p'. Bei dieser Art der Drehung werden keine Matrizen benötigt:

$$p' = q \cdot p \cdot \overline{q} \tag{3.7}$$

In der sogenannten Achsenwinkeldarstellung kann der Drehwinkel konkret beschrieben werden, wobei für ein normalisiertes Quaternion mit dem normalisierten Drehvektor (x, y, z), der die Drehachse darstellt, in dieser Darstellung gilt (Hofmann, 2009):

$$q_r = cos(\frac{\alpha}{2}) + ixsin(\frac{\alpha}{2}) + jysin(\frac{\alpha}{2}) + kzsin(\frac{\alpha}{2}) \tag{3.8}$$

Diese Berechnung ist genauer als die Berechnung mit Matrizen in der Eulerschen Geometrie. Des Weiteren existiert weder die kardanische Blockade, noch sind sie mehrdeutig wie die Euler-Winkel (Lust, 2001).

3.4.3 Sensorbasierte Lagebestimmung und IMU

Head-Tracking-Verfahren können optisch oder sensorbasiert umgesetzt werden. Im Rahmen dieser Arbeit wird auf ein sensorbasiertes Head-Tracking-System unter Nutzung einer inertialen Messeinheit (*engl.* Inertial Meassurement Unit, IMU) zurückgegriffen, welches in Unterunterabschnitt 5.6.2.2 konkret beschrieben wird.

Sensorbasierte Lagebestimmung finden mit Sensoren statt, welche Beschleunigungen oder Winkelgeschwindigkeiten messen. Oft werden zum Tracking verschiedene Sensoren in einer IMU kombiniert. Sensorbasierte Tracking-Systeme können prinzipiell die Lage von sich bewegenden Objekten in allen Richtungen messen und sind damit sehr gut für ein Head-Tracking geeignet. Die genutzten Sensoren sind Beschleunigungssensoren (*engl.* accelerometer) und Drehratensensoren (*engl.* gyroscope), welche zur Gruppe der Inertialsensoren (*lat.* inertia für Trägheit) gehören. Diese arbeiten - wie der Name es andeutet - mithilfe der Trägheit der Masse. Um eine noch genauere Informationsverarbeitung als mit einzelnen Sensoren möglich zu machen, können unterschiedliche Sensorinformationen kombiniert werden. Die Nachteile von Drehratensensoren (vor allem Drift) und Beschleunigungssensoren (vor allem Rauschen) sind dabei komplementär und werden durch Sensorfusion sehr gut ausgeglichen. *BoschSensortec2020*

3.5 Digitale Audiosignalverarbeitung

Die digitale Audiosignalverarbeitung beschreibt jene Verarbeitung, die am digitalen Signal in einem digitalen System vorgenommen wird. Dabei handelt es sich um die Verarbeitung von zeit- und wertediskreten digitalen Signalen, die nach der Wandlung aus den zeit- und wertekontinuierlichen analogen Signalen entstehen.

Digitale Audiosignalverarbeitung kann viele Prozesse und mathematische Operationen auf dem Audiosignal sehr schnell und effizient ausführen. Des Weiteren kommt es aufgrund der Verarbeitung ohne analoge elektrische Schaltkreise, welche bestimmte Fehlertoleranzen aufweisen, verschiedene Formen des Rauschens einbringen, sowie aufgrund von Alterungsprozessen zunehmend an einwandfreier Funktionalität einbüßen können, zu genaueren Ergebnissen der Verarbeitung. Zu beachten ist jedoch an dieser Stelle, dass auch digitale Signale nur in begrenzter Genauigkeit (Abtatsrate, Bittiefe sowie Festkomma-/Gleitkommadarstellungen) vorliegen und es bei vielen Verarbeitungsschritten zu sich fortpflanzenden Ungenauigkeiten in Form von Rundungsfehlern kommen kann. Des Weiteren ist auch die Analog-Digital-, sowie Digital-Analog-Wandlung, sofern das bei der Betrachtung der digitalen Verarbeitung eine Rolle spielt, nicht fehlerlos. (Weinzierl, 2008)

Die digitale Audiosignalverarbeitung kann im Zeitbereich als auch im Frequenzbereich erfolgen, da das Signal über die Fouriertransformation von der einen in die andere Darstellung ohne Einschränkungen überführbar ist. Dabei findet die Verarbeitung je nach Anwendung samplegenau oder blockweise statt.

3.6 Mikrocontroller

Mikrocontroller vereinen in einem einzelnen Chip einen Prozessorkern, Speicher sowie Ein-/Ausgabeschnittstellen, mit dem Ziel Steuerungs- und Kommunikationsaufgaben möglichst einfach zu lösen. Die Leistung ist dabei meist relativ gering und an den Verwendungszweck angepasst; dadurch sind sie recht preisgünstig. Der Speicher des Mikrocontrollers wird mit einem bestimmten Code bespielt, den der Chip anschließend ausführt. Entweder Nur-Lese-Speicher (ROM) oder programmierbare Nur-Lese-Speicher (EPROM, EEPROM) kommen als Speicher zum Einsatz, wobei selten auch Flash-Speicher Verwendung finden. Mikrocontroller werden sehr nah an der Hardware programmiert (meist speziell für einen Controller bzw. eine Architektur). Je nach Hardware ist ein spezieller Programmer (Kombination aus Hardware und Software, die die Programmierung des Microcontrollers ermöglicht) notwendig. Aufgrund der Tatsache, dass die Systeme in sich geschlossen sind, gibt es praktisch keine Möglichkeit für direktes Debugging des Codes und die Ausgabe von Fehlermeldungen ist ebenfalls sehr eingeschränkt. Zumeist wird der Quellcode für Mikrocontroller in der Programmiersprache C geschrieben und die Programmierung erfolgt sequentiell, da die Hardware in den meisten Fällen kein Multi-Threading unterstützt. Es gibt in der Regel eine Hauptfunktion, die wiederholt ausgeführt wird. (Brinkschulte & Ungerer, 2010)

3.6.1 Arduino

Als Open-Source-Projekt, welches einen leichten Einstieg in die Programmierung von Mikrocontrollern anbietet, eignet sich Arduino. Es umfasst sowohl Hard- als auch Software. Ein spezieller Bootloader der Hardware erlaubt die weitere Programmierung ohne speziellen Programmer über eine Universal Asynchronous Receiver Transmitter (UART) Verbindung, in den meisten Fällen erfolgt dies über USB. Der Bootloader ist ein vorinstallierter Code im Speicher des Mikrocontrollers. Die integrierte Arduino-Entwicklungsumgebung (IDE) vereinheitlicht und vereinfacht zudem die Programmierung, da der Code automatisch hardwarespezifisch kompiliert wird und die IDE bereits eine große Zahl an Funktionen bereitstellt.

3.7 Auralisation

Unter Auralisation („Hörbarmachung“ von lat. auris = dt. ’Ohr’ = aurikular) versteht man ein Verfahren zur künstlichen Hörbarmachung einer akustischen Situation. Diese Hörbarmachung kann entweder modellbasiert anhand einer Akustik-Simulation unter Verwendung von Spiegelschallquellen, Raytracing und der Errechnung des Diffusschalls erfolgen oder datenbasiert unter Zuhilfenahme von Messdaten. Die modellbasierte Auralisation wurde in den 1960er-Jahren zur Nutzung im Akustikbau entwickelt und löste zu dieser Zeit das gängige Modellmessverfahren in der akustischen Planung ab, bei dem im Maßstab 1:20 Modelle gefertigt wurden, um die akustische Situation des geplanten Raums zu messen. (Kuttruff, 2000)

4 Räumliches Hören und Binauraltechnik

Im 4. Kapitel dieser Arbeit sollen die Grundlagen des räumlichen Hörens dargestellt werden, ehe im weiteren Verlauf ein Anriss der Geschichte der Binauraltechnik bis hin zum heutigen Stand dargelegt wird; dabei sollen vor allem die Grundlagen für das binaurale System dieser Arbeit gelegt werden. Da die Inhalte dieses Kapitels für das System von hoher Wichtigkeit sind, wurden sie nicht im vorangegangenen Kapitel behandelt, sondern es wird ihnen ein eigenes Kapitel gewidmet.

4.1 Binaurales und räumliches Hören

Binaurales Hören (binaural = *lat.* für „mit beiden Ohren“ oder auch „beidohrig“) bezieht sich im grundlegenden Sinne auf den menschlichen Hörmechanismus, der zwei Rezeptoren verwendet und Informationen über ein Schallereignis durch Auswertung verschiedener im Schallfeld vorhandener Hinweise ableitet (Møller, 1992). Das ultimative Ziel binauraler Wiedergabetechniken ist es, ein Hörerlebnis zu schaffen, das die Unterschiede zwischen der natürlichen Wahrnehmung der Klangszene (und deren Hörereignisse) zu der mit Hilfe von Technologie reproduzierten oder synthetisierten Szene minimiert. Um dies zu erreichen, ist ein gutes Verständnis des natürlichen Hörsystems bzw. der psychoakustischen Reizverarbeitung zwischen Schalleregnis und Hörereignis notwendig.

Das räumliche Hören des menschlichen Hörorgans besteht aus dem Richtungs- und Entfernungshören, sowie der Fähigkeit bei mehreren räumlich verteilten Schallquellen, die Aufmerksamkeit auf einzelne Quellen zu lenken und diese separat zu bewerten. Die räumliche Information, die in den Ohrsignalen enthalten ist, entsteht prinzipiell durch Laufzeitunterschiede, sowie Beugungs- und Reflexionseffekte der auf das menschliche Trommelfell eintreffenden Schallwelle beim Richtungshören und leitet sich aus dem Direktschall-/Difussschallverhältnis und Pegelunterschieden bei der Entfernungswahrnehmung ab (Weinzierl, 2008). Gebeugt wird der Schall dabei maßgeblich durch die Körperstrukturen von Torso/Rumpf bzw. Schulter, Kopf und Ohrmuschel, deren Beugungs- und Reflexionsmuster für jede Schalleinfallsrichtung spezifisch und unterschiedlich sind. Nachrichtentechnisch lassen sich diese Effekte als lineare Verzerrungen interpretieren, die mit Hilfe der sogenannten kopfbezogenen Übertragungsfunktionen beschrieben werden (siehe Unterabschnitt 4.1.3).

Die folgenden Betrachtungen sind Darstellungen der grundlegenden akustisch-auditiven Aspekte des räumlichen Hörens, um ein Verständnis für die in dieser Arbeit genutzten Methoden zu erreichen; ohne Einschränkungen der Gültigkeit kann diese Auswahl getätigt werden. Zur Analyse komplexer Schallsignale bzw. überlagerter Schallfelder oder gar multimodaler

Wahrnehmung und einer Auswertung der Letzteren, welche unserem (Hör-)Empfinden möglichst exakt entspricht, sind deutlich komplexere Signalverarbeitungsschritte notwendig, die beispielsweise zeitliche Aspekte (Feinstruktur, Hüllkurve) als auch die Verarbeitung auditiver Filter (parallele Bandpassfilter) berücksichtigen. Weiterhin spielt die emotionale und kognitive Situation des Hörenden bei der Hörereignisbildung ebenso eine Rolle. (Weinzierl, 2008) Diese Schritte sind jedoch wahrnehmungsbestimmende Vorgänge, die innerhalb des Ohrs und Gehirns stattfinden und folglich für die grundlegende Methodik dieser Arbeit bei der Bestimmung, Bearbeitung und Simulation der Übertragungswege bis zum Trommelfell irrelevant sind; sie sind weiterhin Gegenstand der Forschung auditiver Wahrnehmung.

Die spezifische Positionierung der Ohren auf beiden Seiten des menschlichen Kopfes, die spezifische Form der Ohrmuscheln und der beiden Gehörgänge als auch die spezifische Form der restlichen Anatomie des menschlichen Rumpfes erzeugt also eine Reihe von Unterschieden in der an den beiden Trommelfellen ankommenden Schallwelle, die Rayleigh (1907) in seiner Duplex-Theorie des räumlichen Hörens als Interaurale Pegeldifferenz (ILD), verursacht durch Abschattungen und Interferenzen an Kopf, Rumpf und Ohrmuscheln, und die Interaurale Zeitdifferenz (ITD), die Ankunftszeitdifferenz des Schalls am linken bzw. rechten Ohr beschreibt. Neben diesen sogenannten *interauralen Cues* sind auch *monoaurale Cues* vorhanden, die die frequenzspezifische Filterung beim Hören mit nur einem Ohr unabhängig von der Beziehung der beiden Ohren zueinander, beschreiben. Des Weiteren soll an dieser Stelle auch der *dynamische Cue* genannt werden, welcher beschreibt, dass Unterschiede in der Ankunftszeit und des Schalldruckpegels bzw. Frequenzgangs aufgrund von Kopfbewegungen (Peilbewegungen) verglichen und daraus - besonders im Falle von Mehrdeutigkeiten - Rückschlüsse auf das tatsächliche Schallereignis gezogen werden (Weinzierl, 2008). Es können also zusammenfassend folgende Hauptkategorien genannt werden:

1. ILD - beschreibt den Unterschied im Schalldruckpegel, der an den beiden Ohren ankommt
2. ITD - bezieht sich auf den Unterschied in der Ankunftszeit der Schallwelle an jedem Ohr
3. Monoaurale Cues - beziehen sich auf die spezifischen Filterungen, die ohne interaurale Beziehungen auftreten
4. Dynamische Cues - beziehen sich auf den Vergleich und das Abtasten der vorherigen Cues bei Kopfbewegung

Die interauralen Cues (ILD und ITD) bestimmen das Richtungshören in der Horizontalebene, während die monoauralen Cues für das Richtungshören in der Medianebene verantwortlich sind. Dies ist vor allem dadurch zu begründen, dass aufgrund der symmetrischen Anordnung der Ohren auf beiden Seiten des Kopfes entlang der interauralen Achse (y-Achse) des kopfbezogenen Koordinatensystems nur bei Verschiebung eines Schallereignisses entlang dieser Achse auch ein Unterschied in ILD und ITD zu generieren ist. Verschiebt man ein Schallereignis lediglich in der Medianebene, so kommt es zu keinen interauralen Differenzen. Jedoch sind hier spezifische positionsgebundene spektrale Veränderungen, welche vor allem durch

die sogenannten *Pinnae-Cues*, die die Interferenzeffekte durch die Form der Ohrmuscheln beschreiben, als auch durch Schulter- und Torsoreflexionen hervorgerufen werden, zu beobachten. Diese Cues wirken - wie bereits genannt - nicht interaural oder binaural und werden folglich auch als monoaurale Cues bezeichnet. Hieraus leitet sich auch die Theorie der richtungsbestimmenden Bänder der Lokalisation in der Medianebene nach Blauert ab (Daniel et al., 2007).

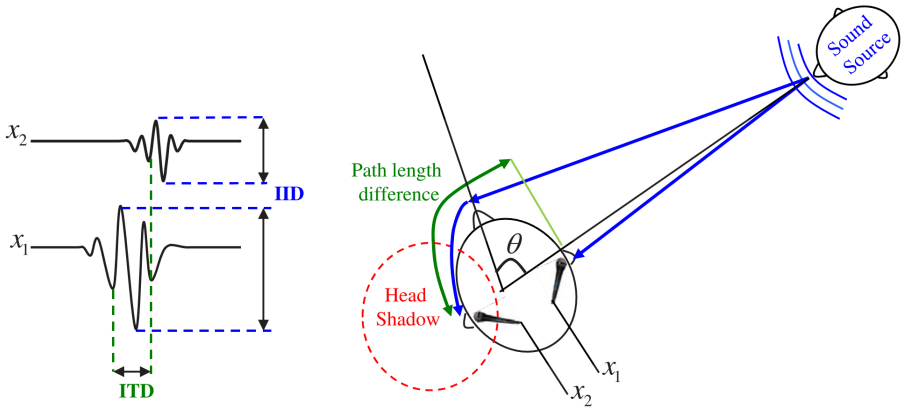

Abbildung 4.1: Interaurale Zeitdifferenz (ITD) und interaurale Pegeldifferenz (ILD)(Keyrouz, 2008)

4.1.1 Richtungshören

4.1.1.1 Interaurale Pegeldifferenzen (ILDs)

Interaurale Pegeldifferenzen (ILDs) werden durch Dämpfung des Schalldruckpegels zwischen den beiden Ohren verursacht und können durch Auswertung des Betragsfrequenzgangs der interauralen Übertragungsfunktion (ITF) analysiert werden (Blauert, 1974). ILD entstehen hauptsächlich durch Abschattungseffekte des Kopfes und sind frequenzabhängig. Die Frequenzkomponenten mit einer Wellenlänge, die mit dem Durchmesser des Kopfes vergleichbar oder kleiner ist, werden reflektiert oder teilweise absorbiert, wodurch ein signifikanter Schalldruckpegelunterschied zwischen den beiden Ohren entsteht. Die Frequenz, ab der ILDs signifikant werden, kann näherungsweise bestimmt werden, indem man den wirksamen Ohrabstand und die größte vergleichbare Wellenlänge betrachtet. Dies wird in Gleichung 4.1 mit der Wahl des wirksamen Ohrabstandes bzw. einer Wellenlänge $\lambda = 21{,}6\ cm$ und der Schallgeschwindigkeit $c = 343\ \frac{m}{s}$ veranschaulicht (Sengpiel, 1995):

$$f_{\text{minimum für signifikante ILD}} = \frac{1}{0.216\,cm} \cdot 343\,\frac{m}{s} \approx 1.6\,kHz \tag{4.1}$$

4.1.1.2 Interaurale Laufzeitdifferenzen (ITDs)

Wenn sich der Schall einer Schallquelle an einem bestimmten seitlichen Ort, welcher mathematisch durch das kopfbezogene polare Koordinatensystem mit Hilfe des Azimutwinkels ϕ (der Radius r des Schallereignisortes sei in diesem relativen Zusammenhang irrelevant)

beschrieben werden kann, in Richtung des Zuhörers ausbreitet, ist die Weglänge zwischen der Schallquelle und den beiden Ohren unterschiedlich, abhängig vom Azimutwinkel ϕ und der Größe des Kopfes. Dies verursacht einen Unterschied in der Ankunftszeit des Signals an jedem Ohr, was die interaurale Zeitdifferenz oder interaurale Verzögerung (ITD) verursacht (Blauert, 1974).

Der Verlauf der ITD kann mit Kopfmodellen wie dem Kugelkopfmodell angenähert werden (Romblom & Bahu, 2018). Wenn eine Schallquelle komplett seitlich des Hörers positioniert ist (bei $\pm$ 90°), werden die ITD-Werte maximiert. Die Auswertung der ITDs erfolgt dabei zumeist über den Phasenversatz der Ohrsignale, die mithilfe der interauralen Phasendifferenz (IPD) bestimmt wird. Hierbei wird deutlich, dass nur ein begrenzter Frequenzbereich aufgrund des Ohrabstands auf diese Weise sinnvoll ausgewertet werden kann: Sehr tiefe Frequenzen erzeugen aufgrund ihrer großen Wellenlänge nahezu keinen Phasenversatz, während hohe Frequenzen einen Versatz von mehr als 180° verursachen, wodurch es zu zu Phasenmehrdeutigkeiten kommt und keine eindeutige Richtungszuweisung mehr stattfinden kann. Dieser Grenzwert kann bei Betrachtung einer reinen Sinusschwingung, die sich in der horizontalen Ebene bei $\phi = 90°$ befindet abgeschätzt werden: Die maximale Laufzeit zwischen den beiden Ohren entspricht 0,63 ms, sodass eine Sinusschwingung mit 1,59 kHz während dieser Zeitspanne eine volle Periode durchläuft, was zur Phasengleichheit und folglich einer Fehlokalisation bei $\phi = 0°$ führt. Das Gehör kann bis zu dieser Frequenz Phasendifferenzen der Feinstruktur beider Ohrsignale erkennen, für höhere Frequenzen bedarf es eines Vergleichs der Hüllkurven beider Ohrsignale. (Weinzierl, 2008)

Es lässt sich jedoch festhalten, dass ITDs (bzw. IPDs) vor allem im Bereich unter 1,6 kHz für die Lokalisation zuständig sind, während ILDs das Richtungshören von Frequenzen über 1,6 kHz bestimmen. Die Übergänge und damit Zusammenhänge der menschlichen Auswertung aller Cues bei Bildung des Hörereignisortes gestalten sich jedoch fließend und sind des weiteren abhängig von der Signalbeschaffenheit. (Ahrens et al., 2020) Eine genauere Analyse dieser Zusammenhänge soll nicht Inhalt dieser Arbeit sein.

4.1.1.3 Monoaurale Cues

Die Streuung/Reflexion sowie Beugung der Schallwellen durch die Ohrmuschel, den Kopf sowie den Rumpf und die daraus resultierenden konstruktiven und destruktiven Interferenzen aufgrund von Überlagerung der Schallwellen erzeugen je nach individueller Körperanatomie spektrale Anhebungen und Einbrüche, die für jede Einfallsrichtung der Schallwelle gemessen, berechnet bzw. kodiert werden können (Zhong & Xie, 2014)(Kudo et al., 2005). Diese spektralen monoauralen Cues sind besonders nützlich für die Lokalisation von Schallquellen in Medianrichtung, wo die *interauralen Cues* nicht vorhanden bzw. minimiert sind und die spektrale Information somit zuverlässiger ist (Carty & Lazzarini, 2009).

4.1.1.4 Dynamische Cues

Dynamische Cues beziehen sich auf Pendel- bzw. Mikrobewegungen des Kopfes, welche genutzt werden, um potenziell mehrdeutige räumliche Informationen aufzulösen, indem alle anderen Cues in Echtzeit aktualisiert werden (Pöntynen et al., 2016). Diese Peilbewegungen des Kopfes, welche zu einer Art räumlicher Abtastung führen, sind für die Qualität von binauralem Rendering - wie in 4.2.1 zu lesen - ausschlaggebend. Sie werden vorrangig verwendet, um Phänomene wie die Vorne-Hinten-Vertauschung, Oben-Unten-Konfusion bzw. den damit einhergehenden sogenannten *Cone of Confusion* aufzulösen (Pöntynen et al., 2016). Der *Cone of Confusion* bezeichnet dabei jene *Kegelscheiben* rechts- bzw. linksseitig des Kopfes mit der Kegelachse entlang der interauralen Achse, auf denen Schallereignisse, welche entlang der Kegelscheiben verlaufen, zu gleichen ITD und ILD führen. Somit ist es nicht möglich Schallereignisse entlang einer solchen Kegelscheibe eindeutig zu lokalisieren und es kommt zu den genannten Vorne-Hinten, sowie Oben-Unten-Vertauschungen. Der für die Lokalisation ebenfalls bedeutsame spektrale Cue, welcher sich in dem betrachteten Fall doch ändert, ist offensichtlich nicht stark genug, um eine eindeutige Lokalisation zu erreichen. Sollte keine visuelle Information vorliegen, so ist dieser Effekt noch stärker. (Weinzierl, 2008)

4.1.2 Entfernungshören

Das räumliche Hören wird neben dem Richtungshören auch durch das Entfernungshören charakterisiert. Dabei tragen vor allem das Frequenzspektrum des Hörereignissen als auch die Lautstärke, welche sich auf dem Weg zum Ohr abhängig von der Entfernung ändert, zum Entfernungshören bei. Sehr bedeutend für eine Beurteilung anhand dieser komplexen Merkmale, ist die Erfahrung des Gehörs und die Bekanntheit des Signals. Mit der Verdoppelung der Entfernung halbiert sich der Schallpegel bei gleichmäßiger Abstrahlung; bei gerichteter Schallemission ist der Pegelabfall geringer. Aufgrund der ungerichteten Abstrahlung der tiefen Frequenzen kommt es mit zunehmender Entfernung auch zu Klangfarbenveränderungen; das Gehör wird zudem bei geringen Pegeln für tiefe Frequenzen unempfindlicher, was diesen Effekt zusätzlich verstärkt. Das Verhältnis von direktem und diffusem Schall (Nachhall) trägt in geschlossenen Räumen zum Entfernungshören bei. Auch das setzt eine gewisse Hörerfahrung voraus, da das Verhältnis von Direktschall und Nachhall vom Raum abhängt. (Weinzierl, 2008)

4.1.3 Kopfbezogene Übertragungsfunktion (HRTF), interaurale Übertragungsfunktion (ITF) und binaurale Raumimpulsantwort (BRIR)

Wenn sowohl die Schallquelle als auch der Hörer fixiert sind, kann die akustische Übertragung von einer Punktquelle zu den beiden Ohren als ein linear-zeitinvariabler (LTI) Prozess betrachtet werden. Mathematisch kann dies als Übertragungsfunktion zwischen dem Schalldruck P_{L} bzw. P_{R} an jedem Trommelfell und dem Schalldruck in der Mitte der Ohren (ohne den Kopf) P_0 dargestellt werden. Der Schalldruckpegel am Trommelfell ist eine Funktion

der Position zwischen Quelle und Empfänger (Ohr) - üblicherweise in einem kopfbezogenen Kugelkoordinatensystem mit den Dimensionen Radius r und den Winkeln Azimut ϕ und Elevation θ dargestellt - sowie der Frequenz f und einem speziellen Faktor a der Individualität, um die Vielzahl von Unterschieden in den Übertragungsfunktionen aufgrund unterschiedlicher Anatomie zu berücksichtigen. Die sogenannten kopfbezogenen Übertragungsfunktionen (HRTFs) H_{L} bzw. H_{R} sind als die akustische Übertragungsfunktion dieses LTI-Systems folgendermaßen definiert (Zhong & Xie, 2014):

$$H_{\mathrm{L}}(r,\theta,\phi,f,a) = \frac{P_{\mathrm{L}}(r,\theta,\phi,f,a)}{P_0(r,f)}, \quad H_{\mathrm{R}}(r,\theta,\phi,f,a) = \frac{P_{\mathrm{R}}(r,\theta,\phi,f,a)}{P_0(r,f)} \tag{4.2}$$

mit

P_{L}, P_{R} - Schalldruck am linken und rechten Trommelfell

P_0 - Freifeld-Schalldruck in der Mitte des Kopfes bzw. in der Mitte zwischen beiden Ohren (ohne den Kopf)

r, θ, ϕ - Schallquellposition in kopfbezogenen Kugelkoordinaten

f - Frequenz

a - Faktor der Individualität

Die in den vorherigen Abschnitten vorgestellten interauralen bzw. binauralen Cues können mit Hilfe der kopfbezogenen Übertragungsfunktionen in Form der interauralen kopfbezogenen Übertragungsfunktionen (ITF) H_{I}, welche für das räumliche Hören wichtige Unterschiede beider Ohrsignale beschreiben, dargestellt werden. Sie ergeben sich aus der Divison der beiden kopfbezogenen Übertragungsfunktionen H_{L} bzw. H_{R} (Weinzierl, 2008):

$$H_{\mathrm{I}}(r,\theta,\phi,f,a) = \frac{H_{\mathrm{R}}(r,\theta,\phi,f,a)}{H_{\mathrm{L}}(r,\theta,\phi,f,a)} \tag{4.3}$$

In der Praxis werden kopfbezogene Übertragungsfunktionen (HRTFs) in Form von Impulsantworten an jedem Ohr für einen bestimmten Schallquellort aufgezeichnet (HRIRs). Das Mikrofon, welches die Impulsantworten aufzeichnet, wird - je nachdem, ob der Einfluss des Gehörgangs in der HRTF inkludiert sein soll - direkt vor dem Trommelfell platziert oder vor dem geschlossen bzw. geöffneten Gehörgang. Da eine Impulsantwort verwendet werden kann, um ein LTI-System vollständig zu beschreiben, können die kopfbezogenen Impulsantworten verwendet werden, um die Eigenschaften des statischen binauralen Hörsystems für eine bestimmte Position zu synthetisieren. Die HRIR kann als Filterkern eines Filters mit endlicher Impulsantwort (FIR-Filter) verwendet werden, um eine beliebige Quelle an einer bestimmten Position binaural zu synthetisieren. In dieser Arbeit werden die Begrifflichkeiten der Übertragungsfunktion (z.B. HRTF) und Impulsantwort (z.B. HRIR), welche die Systemantwort des LTI-Systems einmal im Frequenzbereich und einmal im Zeitbereich vollständig beschreiben und über Fouriertransformation ineinander überführbar sind, bedeutungsgleich verwendet.

Neben der Nutzung von HRIRs, welche die räumliche kopfbezogene Übertragung unter Freifeldbedingungen beschreiben, ist es auch möglich, diese kopfbezogenen Cues mit der LTI-Impulsantwort eines raumakustischen Umfelds zu verbinden; dies führt zur Definition der binauralen Raumimpulsantwort (BRIR). Eine BRIR kann als eine Überlagerung zahlreicher akustischer Reflexionen betrachtet werden, die jeweils mit einer bestimmten kopfbezogenen Übertragungsfunktion gewichtet und um ihre Ankunftszeit verzögert sind. Somit kann eine BRIR das räumliche Hören in einem reverberanten akustischen Umfeld direkt binaural simulieren. Neben der direkten Messung von BRIRs kann diese Raumdarstellung jedoch auch erreicht werden, in dem monaural berechnete Raumübertragungsfunktionen, in der noch die Einfallsrichtungen aller Einzelreflexionen enthalten sind, mit HRTFs für jede Einfallsrichtung multipliziert werden (Weinzierl, 2008). Diese beiden Konzepte stellen die Grundlage für den in Unterunterabschnitt 4.2.2.2 durchgeführten Vergleich von datenbasierten (gemessene BRIRs) und modellbasierten (Multiplikation einer monoaural besrechneten Raumübertragungsfunktion mit einem HRTF-Datensatz) dar.

4.1.4 Seperation der interauralen Cues aus der komplexwertigen Übertragungsfunktion

In der LTI-Systemtheorie kann der komplexe Frequenzgang einer Übertragungsfunktion auch in Form von Betragsfrequenzgang und Phasengang ausgedrückt werden. Im Fall von gemischtphasigen HRTFs kann die Phase in eine minimalphasige Komponente und eine Exzessphasenkomponente aufgeteilt werden (Lindau, 2014):

$$H(j\omega) = |H(\omega)| \cdot e^{j\varphi_{\min}(\omega)} \cdot e^{j\varphi_{\text{Exzess}}(\omega)} \tag{4.4}$$

Die Exzessphasenkomponente kann des Weiteren auch in linearphasige und Allpass-Komponenten zerlegt werden:

$$H(j\omega) = |H(\omega)| \cdot e^{j\varphi_{\min}(\omega)} \cdot e^{j\varphi_{\text{lin}}(\omega)} \cdot e^{j\varphi_{\text{allpass}}(\omega)} \tag{4.5}$$

Der Mensch ist nicht in der Lage, Phase absolut zu hören und die Empfindlichkeit für das Phasenspektrum ist demnach gering (Preis, 1982). Es wurde für HRIRs des Weiteren gezeigt, dass die enthaltene Allpasskomponente für die meisten Schalleinfallsrichtungen unhörbar ist und somit kann sie ohne Einschränkungen der Allgemeinheit vernachlässigt werden, ohne die räumliche Wahrnehmung zu stören (Minnaar et al., 1999):

$$H(j\omega) = |H(\omega)| \cdot e^{j\varphi_{\min}(\omega)} \cdot e^{j\varphi_{\text{lin}}(\omega)} \tag{4.6}$$

Des Weiteren kann die linearphasige Komponente in Gleichung 4.6 ohne hörbare Folgen durch eine Zeitverzögerung bzw. ein Delay ersetzt werden, solange das Letztere die ITD adäquat approximiert, sei es durch eine Bestimmung aus den Messdaten oder die Nutzung eines Kopfmodells (Kulkarni et al., 1999).

An dieser Stelle sei im Zusammenhang dieser Arbeit ausdrücklich erwähnt, dass diese Herleitung unter Angabe der entsprechenden Literatur sich auf eine Betrachtung von HRIRs bezieht, welche im Freifeld bzw. unter reflexionsarmen Bedingungen gemessen wurden und somit keine gemischtphasigen räumlichen Signalkomponenten beinhalten. Damit lässt sich eine direkte Extraktion der ITD des Direktschalls wie im Falle der Analyse einer HRIR nicht durchführen. Lindau (2014) untersucht die Nutzung einer Delay-Approximation für BRIRs hinreichend, unter der Fragestellung, ob sich diese Trennung bei BRIRs ohne Artefakte und mit einer ausreichenden Genauigkeit der Richtungswahrnehmung unter Zuhilfenahme entsprechender Methoden *trotzdem* durchführen lässt. Im Rahmen dieser Arbeit wird eine solche Trennung jedoch nicht genauer betrachtet oder umgesetzt; trotzdem soll dieses Konzept aufgrund der Nutzung in binauralen Rendering-Cores (wie auch dem in dieser Arbeit benutzten) vorgestellt werden. Eine Nutzung dieser Trennung hätte neben der Stabilisierung der Lokalisierung durch Anpassungsmöglichkeiten auf die individuelle Kopfform auch eine Eliminierung von möglichen Überblendkammfiltern durch die Verwendung von quasi-minimalphasigen Audiosignalen, sowie Möglichkeiten der unterschiedliche Raster-Auflösung und/oder Interpolation für zeitliche und spektrale Cues zur Folge (Lindau, 2014).

4.1.5 Cue-Fehler und deren Auswirkungen auf das Hörereignis

Da ILD, ITD und die spektralen Cues eng mit den physiologischen Merkmalen des Menschen verbunden sind, hat die Verwendung von nicht-individuellen HRTFs einen erheblichen Einfluss auf die Authentizität der Auralisation.

Die Verwendung von nicht-individuellen ILD wirkt sich vor allem auf die Klangfarbe (das Timbre) aus. Da das absolute Gedächtnis des Menschen für Klangfarben jedoch schwach ist, tritt dieses Problem typischerweise nur im direkten Vergleich mit realen Schallfeldern auf (Estrella, 2010). Jedoch können statische spektrale Verfärbungen in bestimmten Blauertschen Bändern zu konstanten Fehllokalisationen in eine bestimmte Richtung (Vorne-Hinten, Oben-Unten) führen, die vor allem bei nicht-dynamischer Binauralsynthese nicht aufgelöst werden können. Eine nicht-individuelle ITD erzeugt einen offensichtlicheren Effekt der Instabilität der Schallquellen und konstante Lokalisierungsfehler in der Horizontalebene. In diesem Fall findet keine Adaption statt und dieses Artefakt fällt auch ohne direkten Vergleich mit realen Schallquellen auf (Estrella, 2010).

Bei der dynamischen Binauralsynthese führt eine falsche ITD zu folgenden Phänomenen: Eine Verschiebung der Schallquellen in die gleiche Richtung der Kopfbewegung wird wahrgenommen, wenn der Kopf des Modells für die Datenerfassung kleiner war als der des Benutzers; eine Verschiebung in die entgegengesetzte Richtung wird wahrgenommen, wenn der Kopf des Modells größer war (Algazi et al., 2001).

4.2 Binauraltechnik

4.2.1 Binaurales Rendering allgemein

Die Reproduktion von räumlichem Audio über Kopfhörer erfordert die Filterung des Quellmaterials mit den HRTFs für die spezifischen Positionen der Quellobjekte. Dazu muss die genaue Position jeder räumlichen Quelle bekannt sein und das Quellmaterial muss so gefiltert werden, dass die relevanten Cues, die in der HRTF des Hörers für die betreffende Position identifiziert wurden, reproduziert werden.

Bevor die Möglichkeiten diskutiert werden, wie binaurale Inhalte dem Hörer präsentiert werden können, ist es wichtig, zwischen den Begriffen *authentisch* und *plausibel* in Bezug auf binaurale Signale zu unterscheiden. *Authentizität* ist untrennbar mit einer bestehenden Referenz oder einem Original verbunden, von dem das System, welches die binauralen Signale berechnet und präsentiert, nicht zu unterscheiden ist. Betrachtet man die Wahrnehmung, muss also eine Identitätsbeziehung zwischen einer authentischen binauralen Wiedergabe eines Systems und dem Originalsystem bestehen. Oft ist es jedoch so, dass eine externe Referenz nicht existiert, sodass der Vergleich nur mit der internen Referenz des Hörers durchgeführt wird. In diesem Fall wird der Begriff *plausibel* verwendet, um „eine Simulation in Übereinstimmung mit der Erwartung des Hörers gegenüber einem äquivalenten realen akustischen Ereignis “ zu beschreiben ((Lindau & Weinzierl, 2012), S. 1187). Damit entfällt der Zwang zur *Authentizität* und das binaurale System muss lediglich glaubwürdige Eigenschaften aufweisen, die vom Hörer im gegebenen Kontext korrekt identifiziert werden können.

Das ideale Endziel eines binauralen Renderers für ein akustisches System ist eine authentische Wiedergabe des Systems. Dabei bleibt es jedoch bei modellbasierten Verfahren wie in 4.2.2.2.2 beschrieben schwierig virtuelle Umgebungen mit den vorhandenen Rendering-Techniken zu erzeugen. Daher wird zur Bewertung der Leistung solcher Systeme häufiger die Plausibilität herangezogen (Blauert, 1974). Datenbasierte Verfahren wie das in 4.2.2 beispielhaft beschriebene *Binaural Room Scanning* sind prinzipiell überlegen und können auch authentische Ergebnisse der binauralen Wiedergabe liefern; neuere Vergleiche raumakustischer Simulationen belegen dies (Brinkmann et al., 2019).

Es gibt prinzipiell drei Hauptmöglichkeiten, eine räumliche Darstellung über Kopfhörer zu erzeugen, welche im Folgenden kurz dargestellt werden sollen:

1. Binaurale Aufnahme des Materials
2. Direkte binaurale Synthese des Quellmaterials
3. Ansatz mit virtuellen Lautsprechern

4.2.1.1 Binaurale Aufnahme des Materials

Die für eine räumliche Darstellung erforderlichen Übertragungsfunktionen können durch Aufnahme des Inhalts mit einem binauralen Mikrofon nachgebildet werden. Eine Möglichkeit

besteht darin, Mikrofone im oder vor dem Gehörgang einer menschlichen Testperson zu platzieren. In diesem Fall stimmen die binauralen Cues mit den HRTFs dieser Person überein. Eine andere Möglichkeit ist die Verwendung eines Kunstkopfmikrofons - ein Modell eines menschlichen Kopfes, teilweise mit Nachbildung des Torsos - mit Mikrofonen an der Position des Trommelfells bzw. am Eingang des Gehörgangs(Hammershoi & Møller, 2002).

Dies ist eine der einfachsten Methoden, um eine binaurale Darstellung zu generieren; jedoch gibt es eine Reihe von Nachteilen, die damit verbunden sind:

- Die Aufnahme ist die Einzelperspektive des Hörers/Kunstkopfmikrofons. Jegliche Bewegung des Letzteren wird zum Bestandteil der Aufnahme.
- Das darzustellende Tonmaterial muss in Echtzeit abgespielt werden, in einem physischen Raum, in dem die Aufnahme stattfindet.
- Alle Positionsänderungen müssen an der physischen Quelle vorgenommen werden, und es sind keine Änderungen mehr möglich, sobald der Inhalt aufgenommen wurde.

Im Falle eines binauralen Renderers zur Simulation von Regieräumen ist dies unpraktisch, da die Produktion von Audiomaterial auf die Möglichkeit angewiesen ist, eine Bearbeitung und Modifikation des Quellmaterials jederzeit zu ermöglichen.

4.2.1.2 Direkte binaurale Synthese des Quellmaterials

Eine wesentlich flexiblere Methode zur Erstellung binauraler Inhalte ist die sogenannte *binaurale Synthese* der Positionen der Quellen mit HRIRs. Vorausgesetzt, dass eine HRIR für jede gewünschte Position der wiederzugebenden Schallquellen verfügbar ist, kann der Effekt des natürlichen Hörens durch FIR-Filterung der Quelle mit den entsprechenden HRIRs und Summierung der resultierenden Signale in einen linken und rechten Kanal für die Wiedergabe über Kopfhörer erzielt werden. Diese Methode verwendet entweder reale bzw. gemessene oder mit Hilfe von anthropometrische Daten numerisch berechnete Impulsantworten als Filterkerne und kann daher - je nach Übereinstimmung der binauralen Cues des Renderings mit denen des Hörers - sehr gute Ergebnisse für die Reproduktion des binauralen Hörens liefern. Da die Klangszene künstlich konstruiert ist, können außerdem Positionsanpassungen an den einzelnen Quellsignalen vorgenommen werden, ohne eine physische Anpassung vornehmen zu müssen. Anpassungen der Positionsparameter können auch in Bezug auf Tracking-Geräte vorgenommen werden, um eine konsistente Klangszene auch bei Bewegungen des Hörers zu erhalten (Møller, 1992) (Völk, 2011).

Ein wichtiger Faktor ist die Wahl des HRIR-Datensatzes: Bei der reinen binauralen Synthese wird die positionsrichtige Darstellung einer Quelle durch HRIR-Filterung für diese Richtung mit dem Signal der Quelle erreicht. Wenn eine große Anzahl von Quellen an verschiedenen Positionen erforderlich ist, wird ein größerer HRIR-Satz benötigt, der eine HRIR für jede benötigte Richtung enthält. Wird im Weiteren auch ein Head-Tracking-System einbezogen, muss der HRIR-Satz eine solch hohe Auflösung haben, dass alle neuen Positionen bei Kopfbewegung in ausreichender Genauigkeit berücksichtigt bzw. gefiltert werden können. Wenn

neben der Kopfdrehung (Gierwinkel, yaw) auch Kopfneigungen (Rollwinkel, roll bzw. Nickwinkel, pitch) gewünscht sind, ist es damit auch notwendig die benötigten Cues für diese Kopfbewegungen im Datensatz wiederzugeben; damit kann der Datensatz je nach Länge der inkludierten Impulsantworten sehr groß werden. Eine gängige Lösung für dieses Problem ist die Verwendung eines niedriger aufgelösten Datensatzes und die Berechnung der nicht im Datensatz direkt vorhandenen Impulsantworten durch Interpolation (Carty & Lazzarini, 2009). Rechnerisch ist für die reine binaurale Synthese eine Faltung für jede räumliche Quelle in der Klangszene erforderlich. Je komplexer die Szene wird, desto höher sind damit auch die Anforderungen an die Verarbeitung.

Der beschriebene Filterungsprozess kann mit HRIRs erfolgen, welche unter Freifeldbedingungen gemessen wurden bzw. auf Grundlage von anthropometrischen Daten numerisch generiert wurden; diese Daten enthalten neben den Cues, die dem Direktschall durch die menschliche Gestalt aufgeprägt werden und richtungsbestimmend sind, keine Informationen, die durch Reflexionen von Schallwellen in reverberanten Räumen entstehen und vom Menschen ebenso im natürlichen Hörprozess für das räumliche Hören aufgelöst werden. HRIR-Datensätze, welche in unterschiedlichen Entfernungen zum Probanden oder Kunstkopf gemessen werden, spiegeln jedoch Effekte des Hörens im Nah- und Fernfeld wieder. Wie in 4.2.2 nochmals näher und unter Aspekten der Wiedergabe von virtuellen Lautsprechern betrachtet wird, können kopfbezogene Übertragungsfunktionen auch in einem diffusen Schallfeld gemessen werden; die so gewonnenen Impulsantworten werden binaurale Raumimpulsantworten (BRIRs) genannt und enthalten alle Anteile der natürlichen Wahrnehmung des Raumes. Diese Impulsantworten können ebenso in Datensätzen gespeichert und in einem zur Verarbeitung der HRIRs strukturell gleichen Filterungsprozess genutzt werden; es kommt neben der akkuraten Wahrnehmung der Richtung von Schallquellen auch zu einer Entfernungswahrnehmung bzw. zu einem *Raumeffekt*. Da die BRIRs aufgrund der Erfassung und Speicherung der zeitlichen Abfolge von verschiedenen Signalanteilen (Direktschall, erste Reflexionen, Nachhall) deutlich länger als HRIRs sind, werden die entsprechenden Datensätze größer und auch die Verarbeitung wird rechenintensiver.

4.2.1.3 Ansatz mit virtuellen Lautsprechern

Ein weiterer Ansatz zur Wiedergabe von räumlichen Inhalten über Kopfhörer ist die Verwendung von sogenannten virtuellen Lautsprechern. Der Ursprung dieser Technologie findet sich im Binaural Room Scanning (BRS), welches in 4.2.2 ausführlich beschrieben wird.

Im Vergleich zur direkten binauralen Synthese des Quellmaterials wie in 4.2.1.2 beschrieben, werden bei diesem Ansatz die Schallquellen wie bei einer lautsprecherbasierten Audioproduktion zunächst über die Letzteren wiedergegeben - und das virtuell. Das heißt, die Schallquellen werden nicht direkt mit der entsprechenden positionsbezogenen HRIR gefaltet, sondern lediglich die die Schallquellen wiedergebenden Lautsprecher, welche sich an festen Positionen befinden, werden mit den HRIRs gefaltet. Die Verteilung der Schallquellen findet also nicht frei im Raum statt, sondern ist an die Lautsprecher und einen auf diese Lautsprecher angewendeten Panning-Algorithmus gebunden. Finden die Faltungen mit einem einzigen

HRIR-Datensatz statt, so wird die Anzahl der Faltungen auf die gewählte Anzahl der virtuellen Lautsprecher begrenzt - auch wenn mehrere Quellen nicht positionsgebunden über die virtuellen Lautsprecher synthetisiert werden. Somit ist bei statischer Binauralsynthese eine Reduktion des HRIR-Datensatzes auf die gewählten festen Positionen der virtuellen Lautsprecher möglich, jedoch muss bei einer dynamische Anpassung der Synthese der Einfluss der Positionsänderungen der Lautsprecher bei Kopfdrehung genauso berücksichtigt werden wie bei der direkten binauralen Synthese des Quellmaterials.

Der Ansatz mit virtuellen Lautsprechern hat jedoch auch alle Nachteile, welche mit dem jeweils genutzten Panning-Algorithmus bzw. der genutzten räumlichen Kodierung zusammenhängen; dabei sind diverse klangliche und räumliche Abbildungsfehler bekannt, die wiederum auch in Wechselwirkung mit dem binauralen Rendering treten können (C. Pike et al., 2016).

4.2.1.4 Herausforderungen der binauralen Wiedergabe

Wenn keine binaurale Aufnahmetechnik verwendet wird, ist ein Datensatz erforderlich, der die positionsgebundenen Cues des binauralen Hörens beinhaltet und auf Signale von Schallquellen anwenden kann. Da die physischen Eigenheiten der Ohrmuschel, des Kopfes und des Torsos für jede Person spezifisch sind, ergibt die Verwendung eines Datensatzes, der mit einem Kunstkopfmikrofon oder mit Hilfe des Kopfes einer anderen Person gemessen wurde nur eine Annäherung an die individuellen Übertragungsfunktionen. Die Verwendung individueller HRTFs führt im Allgemeinen zu einer besseren Wahrnehmung der Position von lateralen Schallquellen (*Cone of Confusion*), reduziert die Verwirrung zwischen Vorne und Hinten und ermöglicht eine genauere Wahrnehmung von elevierten Quellen (Hoene et al., n. d.).

Neben der Nutzung von möglichst individualisierten HRTF-Datensätzen ist auch die Auflösung des Datensatzes für die Qualität der Wiedergabe wichtig. Während verschiedene Interpolationstechniken eine Lösung für das Rendering von Quellen zwischen zwei im Datensatz vorhandenen HRIRs bieten, wird ein hoch aufgelöster HRIR-Datensatz die binaurelen Cues genauer abbilden als ein Datensatz mit niedrigerer Auflösung und anschließender Interpolation (Christensen et al., 1999) (Kearney & Doyle, 2015). Das Auflösungsvermögen des menschlichen Gehörs in den verschiedenen Richtungen ist in diesem Fall der ausschlaggebende Faktor für ein bestmöglich aufgelöstes Grid und die daraus beste zu erzielende Qualität. Die Messung eines Datensatzes mit sehr genauen 1°-Auflösung in horizontaler und vertikaler Richtung ist aufwendig und - wenn man das anschließende Rendering betrachtet - oft nicht praktikabel für Anwendungen, bei denen der Speicherplatz begrenzt ist.

Dynamische Hinweise spielen ebenfalls eine wichtige Rolle bei der korrekten binauralen Darstellung eines Schallfelds. Diese können beispielsweise reproduziert werden, indem ein Head-Tracking-System am Kopf des Zuhörers eingesetzt wird und die HRTF-Filterung entsprechend der Kopfausrichtung aktualisiert wird. Es hat sich gezeigt, dass Head-Tracking die Lokalisierungsgenauigkeit in allen Ebenen verbessert und Effekte wie die Umkehrung von vorne nach hinten aufhebt (Parks et al., 2013) (Ashby et al., 2013). Bei Nutzung dieser dynamischen

Cues muss die Latenz beachtet werden, die das Head-Tracking, die komplette Schnittstelle zwischen Head-Tracker und darauffolgender Signalverarbeitung, sowie die Signalverarbeitung selbst zur Anpassung der Filterung benötigt. In der Literatur werden verschiedene Schwellwerte für die sogenannte totale Systemlatenz (TSL) genannt, welche nicht überschritten werden sollte, um eine plausible Wahrnehmung zu erreichen.

Individualisierte HRIRs und vor allem akkurates Headtracking sind Schlüsselelemente bei der Schaffung einer natürlichen Darstellung des räumlichen Hörens. Da HRIR-Messungen unter Freifeldbedingungen in reflexionsarmen Räumen durchgeführt werden, fehlt diesen die Information, welche durch Reflexionspattern und Nachhall beim natürlichen Hören in gewöhnlichen Umgebungen gewonnen wird; es kommt aus diesen Gründen zu einer nur unzureichenden Externalisierung, die vor allem durch die räumliche Information und das daraus bestimmte Entfernungshören erreicht wird. In vielen binauralen Renderern werden unter Zuhilfenahme von bekannten Räumlichkeiten frühe Reflexionen und/oder Nachhall durch entsprechende Algorithmen generiert, die wiederum auch durch den HRIR-Datensatz gefiltert werden, um einen Raumeindruck nachzubilden. Minimaler Nachhall oder frühe Reflexionen haben einen dramatischen Effekt bei der Erhöhung der wahrgenommenen Externalisierung von Quellen; des Weiteren wird die Lokalisierungsgenauigkeit von Schallquellen besser, welche aufgrund dieser Externalisierung nicht mehr im Kopf (IKL, Im-Kopf-Lokalisation), sondern außerhalb des Kopfes (AKL, Außer-Kopf-Lokalisation) gehört werden (Begault et al., 2000).

Binaurale Raumimpulsantworten (BRIRs) können auch verwendet werden, um die akustischen Eigenschaften des Raumes, in dem sie gemessen wurden, zu einer binauralen Simulation hinzuzufügen und so eine virtuelle akustische Umgebung zu schaffen. In diesem Fall werden die binauralen Impulsantworten in einem halligen Raum aufgenommen, im Gegensatz zu den Freifeldbedingungen im reflexionsarmen Raum. Fehlende Externalisierung, die den Unterschied zwischen der Wahrnehmung der Schallquelle bei Kopfhörerwiedergabe im Kopf (IKL) oder außerhalb des Kopfes (AKL) beschreibt, ist ein Kernproblem der Kopfhörerwiedergabe, welches durch binaurale Wiedergabe über Kopfhörer gelöst werden soll. Oft unterscheidet sich die virtuelle akustische Umgebung, die binaural reproduziert wird, jedoch erheblich von der realen Umgebung, in der sich der Hörer zur gleichen Zeit physisch befindet. Dadurch entsteht ein Konflikt zwischen den binauralen Cues der virtuellen Umgebung und anderen Wahrnehmungsreizen bzw. den realen akustischen Eigenschaften des Raumes, in dem sich der Hörer befindet; dies führt wiederum zu einem erhöhten Lokalisierungsfehler und einer geringeren Externalisierung (Werner et al., 2016) (Werner et al., 2017).

4.2.2 Gewähltes binaurales Rendering: Simulation virtueller Lautsprecher

Wie im Laufe des letzten Kapitels beschrieben gibt es mehrere Möglichkeiten binaurale Inhalte zu präsentieren. In Kapitel 2 sind die Ziele des im Rahmen dieser Masterarbeit entwickelten Systems zur binauralen Simulation genannt und anhand dieser soll nun kurz diskutiert werden, welche Art des binauralen Renderings für das System am geeignetsten erscheint.

Eine einfache binaurale Aufnahme kann für das System nicht nutzbar sein, da sie in keinster Weise den Anforderungen der zeitlichen und räumlichen Unabhängigkeit bei der Nutzung des Systems entspricht.

Ein direkter binauraler Synthesealgorithmus deckt die Eigenheiten als auch Einschränkungen bei der Wiedergabe von Audio über Lautsprechersysteme, die mit Formen des Amplituden- und Delaypannings arbeiten, nicht hinlänglich ab, sodass dieser nicht für eine Simulation von Räumen mit solchen Lautsprechersetups (wie z.B. Regieräume in Tonstudios) genutzt werden kann. Die Implementierung all dieser spezifischen Parameter in einem reinen binauralen Synthese-Algorithmus würde die Komplexität des Algorithmus und die Anforderungen an die Rechenleistung des Anwenders deutlich erhöhen.

Die meisten der beschriebenen Probleme lassen sich durch einen Ansatz mit virtuellen Lautsprechern wie in Unterunterabschnitt 4.2.1.3 beschrieben überwinden. In diesem Fall ist die räumliche Präsentation bzw. Kodierung des Schallfelds beliebig wählbar und wird von einem System übernommen, welches vor dem eigentlichen System des binauralen Renderings liegt. Das Letztere empfängt die Lautsprechersignale und berechnet daraus die binauralen Signale, welche dem Anwender über Kopfhörer präsentiert werden. In einem solchen Fall wird sowohl für das physische Lautsprechersystem als auch für die Kopfhörersimulation derselbe Algorithmus für die Verteilung von Schallquellen verwendet, was somit auch alle spezifischen Eigenschaften dieses Algorithmus überträgt. Die Verarbeitungsanforderungen innerhalb eines Simulationssystems bleiben unabhängig von der Anzahl der Schallquellen konstant, da die Anzahl der erforderlichen Operationen nur von der Anzahl der akustischen Quellen (d.h. der Anzahl Lautsprecher) abhängt.

4.2.2.1 Binaural Room Scanning

Das sogenannte BRS wurde Mitte der 1990er-Jahre am Institut für Rundfunktechnik (IRT) in München entwickelt. Es ermöglicht die virtuelle Wiedergabe eines Lautsprecheraufbaus in einem bestehenden Raum über Kopfhörer (Mackensen, Felderhof et al., 1999). Im Gegensatz zu modellbasierten raumakustischen Auralisationsverfahren, die simulierte erste Reflexionen sowie eine Berechnung des Diffusschalls verwenden, basiert dieses Verfahren auf real gemessenen Daten und kann folglich als der Ursprung der datenbasierten Binauralsynthese zur Darstellung von virtuellen Lautsprechern verstanden werden. Bereits in diesem ersten solchen System wurde die Faltung der Eingangsdaten mit den Raumimpulsantworten dynamisch mit der Ausrichtung des Kopfes durch einen Head-Tracker gesteuert, um die Fähigkeiten des Hörsinns in Form von dynamischen Cues (Pendelbewegungen des Kopfes) zu erfassen.

Das große, weitverbreitete Problem bei falschem Gebrauch von jeglichen Surround-Lautsprechersetups mit vielen Lautsprechern - man denke dabei auch an 3D-Lautsprechersetups - ist neben den hohen Anschaffungskosten aufgrund der hohen Anzahl von Lautsprechern vor allem die korrekte Aufstellung der Lautsprecher, welche im häuslichen Bereich in den wenigsten Fällen gewährleistet werden kann und auch unter schwierigen Bedingungen im professionellen Umfeld teilweise nicht erreicht wird. Somit kommt es zu einer starken Verzerrung der Wiedergabe,

welche nicht mehr das wiedergibt, was bei der Produktion im Studio oder Mischkino zu hören war. Des Weiteren bedeutet eine Audiowiedergabe mit vielen Lautsprechern eine multidimensionale akustische Anregung, welche eine akustische Anpassung an diese Gegebenheiten erfordert. Auf diese Weise kann der Vorteil von Lautsprechersetups mit vielen Lautsprechern, welche bei richtiger Anwendung im kontrollierten Umfeld dazu in der Lage sind, den Sweetspot zu einer Sweetarea zu vergrößern, als auch das Klangerlebnis mit Hilfe geringer Korrelationen zwischen den Lautsprechersignalen *immersiver* werden zu lassen, sehr oft nicht gewinnbringend beim Konsumenten eingebracht werden.

Diese heute noch gültige Problematik führte dazu, dass mit dem BRS am IRT ein System entworfen wurde, welches eine möglichst originalgetreue Wiedergabe von 3/2-Stereosetups über Kopfhörer ermöglicht. Dabei sollte die typische Im-Kopf-Lokalisation (IKL) vermieden werden, welche den Raumeindruck bereits bei der Wiedergabe von 2.0-Stereo über Kopfhörer verfälscht bzw. von der Wiedergabe über 2.0-Lautsprechersetups deutlich verändert erscheinen lässt. Es ist so möglich, die Vorteile beider Abhörverfahren (Lautsprecher bzw. Kopfhörer) zu kombinieren und so dem Tonmeister ein Werkzeug an die Hand zu geben, das ihm das Abhören von Surround-Sound-Material selbst unter akustisch ungünstigen Bedingungen (Ü-Wagen) über Kopfhörer ohne störende Artefakte ermöglicht. (Mackensen, Theile et al., 1999)

Dieses System ermöglicht es mit Hilfe eines Kunstkopfmikrofons (genutzt wurde zumeist ein *Neumann KU 100*) einen realen Abhörraum zu vermessen, in dem für verschiedene Orientierungen des Kunstkopfmikrofons BRIRs bzw. BRIR-Datensätze gemessen werden, welche neben der eigentlichen kopfbezogenen Übertragungsfunktion (HRTF) einer oder mehrerer im Raum vorhandener realer Quellen (in diesem Fall der Lautsprecher) auch die akustischen Eigenschaften des Raums beinhalten. Mithilfe einer zeitlich variierenden Filterung der Lautsprecher-Quellsignale unter Anpassung der Filter an die momentane Kopforientierung lassen sich so die Ohrsignale für alle Lautsprecher generieren, welche bei Wiedergabe über Kopfhörer den gleichen Raumeindruck vermitteln. So lässt sich ein virtueller Abhörraum synthetisieren.

Horbach et al. (1999) beschreibt die anwendungsbezogenen Fähigkeiten des BRS-Systems folgendermaßen:

- Tontechniker kann einen hochwertigen Abhörraum sowie seine gewohnten Lautsprecher mitnehmen (z.B. in einen Ü-Wagen).
- Verschiedene Abhörsituationen, die in der Datenbank gespeichert sind, können ausgewählt und verglichen werden (beispielsweise ein professionelles Tonstudio, ein großes Mischkino oder auch Wiedergabegeräte beim Verbraucher usw.).
- Verschiedene Wiedergabeformate können direkt miteinander verglichen werden.
- Musikproduktion unter standardisierten Bedingungen, z. B. in einem synthetisch erzeugten Referenz-Abhörraum mit idealisierten Lautsprechern.

Daraus werden vor allem die Aspekte der Mobilität deutlich, welche das System bietet, welche vorrangig eine Verbesserung der Produktionsbedingungen bieten können. Des Weiteren

können, wie Mackensen, Theile et al. (1999) schreibt, damit jedoch auch Anwendungen im Bereich der Akustik und psychoakustischen Forschung umgesetzt werden, wie beispielsweise Lautsprecher- oder Raumvergleichstests per Knopfdruck oder Untersuchungen bestimmter Übertragungsparameter (beispielsweise unter deren Auswirkungen auf die Lokalisation), indem eine gezielte Beeinflussung der Letzteren vorgenommen wird und diese auch wieder exakt reproduziert werden können. Dies hat Mackensen, Theile et al. (1999) selbst in diversen psychoakustischen Tests mit dem BRS-System durchgeführt.

Das BRS-System wurde zunächst in einer 19-Zoll Hardware-Version entwickelt, ehe es im Verlauf der Produktgeschichte auch ein VST-Plugn gab. Die maximale Lautsprecheranzahl in der Hardware-Version wurde auf 5 beschränkt (Horbach et al., 1999), während bis zu 8 Räume gleichzeitig gespeichert werden konnten (Rathbone, 2000). Die BRIRs wurden in 5°-Schritten in einem Bereich von -45° bis +45° bei frontaler Kopforientierung (0°) gemessen, wobei es auf einem separaten DSP mittels eines Interpolationsalgorithmus im Frequenzbereich (Real- und Imaginärteil getrennt, einfache Interpolation der Magnitude und Phase ist aufgrund der zeitlichen Versetzung benachbarter BRIRs zueinander nicht möglich(Horbach et al., 1999)) zu einer Interpolation auf ein 1°-aufgelöstes Raster kam, welches unhörbare Übergänge zwischen der Filtern beim Head-Tracking ermöglichte (Horbach & Pellegrini, 1998). Die Länge der komplett dynamisch verarbeiteten Filter lag in üblichen Bereichen der Nachhallzeit von Tonstudios (z.b. 0,3 ms), wobei eine partionierte Faltung im Frequenzbereich umgesetzt wurde, um eine Latenzminimierung des zeitlich variierenden Filterupdates zu erreichen (Horbach et al., 1999). Eine Gesamtsystemlatenz von 50 ms konnte erreicht werden (**Rathbone200**). Diese hier genannten Kenndaten des Systems werden im Verlauf dieser Arbeit zum Teil erneut aufgegriffen und diskutiert.

4.2.2.2 Datenbasierte und modellbasierte Verfahren der Raumsimulation

In diesem Kapitel soll die binaurale Simulation sowohl im Bezug auf modellbasierte als auch datenbasierte Verfahren der Raumsimulation kurz dargestellt und diskutiert werden, um eine Abschätzung auf eine erreichbare Performance der Letzteren bei Nutzung der Simulation von virtuellen Lautsprechern zu geben.

4.2.2.2.1 Datenbasierte Verfahren Datenbasierte Verfahren binauraler Raumsimulationen - besonders unter dem Aspekt der Darstellung virtueller Lautsprecher - finden im Jahr 2021 weiterhin hauptsächlich nach dem Vorbild des BRS durch das *Scanning* eines Raumes mithilfe eines sich bewegenden Kunstkopfmikrofons und der anschließenden dynamischen Faltung dieser gemessenen Daten auf Grundlage der Orientierungsdaten des Kopfes statt. Hierbei wird aus den umfangreichen Untersuchungen und Ergebnissen des BRS für neuere Systeme Nutzen gezogen. So wird in der Regel aus Gründen der Komplexität und des Rechenaufwands auf das Abtasten des Raumes in vertikaler Richtung verzichtet, da dies wie Mackensen (2004) zeigt, keinen signifikanten Einfluss auf die Lokalisation hat. Auch eine laterale Kopfbewegung wird zumeist verzichtet. Erweiterungen der Systematik des BRS werden vor allem in der Hinsicht getroffen, dass diverse Abhörpunkte innerhalb eines Lautsprechersetups für ein Scanning

mithilfe des Kunstkopfes gewählt werden, welche dann für eine Auralisation genau dieses Abhörpunkts genutzt werden können (Satongar et al., 2014) (Melchior et al., 2014). Des Weiteren ist es auch möglich diese BRIR-Datensätze, welche an verschiedenen Abhörpunkten innerhalb des Lautsprechersetups gemessen wurden, für eine Erweiterung des System hinsichtlich drei weiterer Freiheitsgrade der translatorischen Bewegung zu nutzen - diese translatorischen Bewegungen müssen dann ebenfalls mit einem Head-Tracking an einen entsprechenen binauralen Renderer übergeben werden. Auch Einflüsse von akustischen Absorbern oder Diffusoren auf die empfunden Raumakustik können durch die Messung der BRIR-Datensätze an meheren Abhörpunkten auf Hörtests, welche nicht in situ stattfinden müssen, verschoben werden (Erbes et al., 2015).

Das in Unterunterabschnitt 5.5.2.3 (vor allem hinsichtlich der im System dieser Arbeit genutzten *SimpleFreeFieldHRIR*-Convention) ausführlich beschriebene SOFA bietet hinsichtlich der Speicherung von BRIR-Datensätzen neue Möglichkeiten: So können Listener-Positionen und -orientierungen als auch die Positionen und Orientierungen der Lautsprecher (Source bzw. Emitter) direkt zugehörig zu den jeweils gemessenen BRIRs in beispielsweise ein SOFA-File der *MultiSpeakerBRIR*-Convention geschrieben werden (Audio Engineering Society, 2015) (Audio Engineering Society, 2020).

4.2.2.2.2 Modellbasierte Verfahren Modellbasierte Verfahren zur Simulation eines Raumes gehorchen in den meisten praktischen Umsetzungen den Gesetzen der Geometrische Akustik (GA) und treffen damit deutliche Vereinfachungen in der Berechnung der Schallausbreitung. Die verfügbaren Rechenkapazitäten steigen nach dem Mooreschen Gesetz an, sodass auch für akustische Berechnungen schnellere und effizientere deterministische Berechnungen des emulierten physikalischen Verhaltens in Computermodellen möglich werden. Auch wenn bahnbrechende Verbesserungen in wellen-basierten Berechnungen gemacht werden, die schnellere und effizientere Simulationen ermöglichen, stehen heutige akustische Computermodellierungen immer noch weitgehend unter dem Einfluss der geometrischen Approximation der Schallausbreitung. Zu der stark vereinfachten Beschreibungsweise der geometrische Akustik als Analogon zur geometrischen Optik gelangt man durch die Annahme des Grenzfalls verschwindend kleiner Wellenlängen, d. h. des Grenzfalls sehr hoher Frequenzen. Beugungsphänomene werden in der geometrischen Raumakustik vernachlässigt, da die Ausbreitung in geraden Linien das Hauptpostulat ist. Ebenso werden Interferenzen nicht berücksichtigt, d.h. bei Überlagerung mehrerer Schallfeldkomponenten werden deren gegenseitige Phasenbeziehungen nicht in die Berechnung einbezogen, sondern lediglich deren Energiedichten bzw. deren Intensitäten addiert. Dieses vereinfachte Verfahren ist zulässig, wenn die verschiedenen Komponenten zueinander nicht kohärent sind. Diese dargestellten Vereinfachungen zeugen davon, dass die geometrische Raumakustik nur einen Teilaspekt der in einem Raum auftretenden akustischen Phänomene erfasst und wiedergibt (Kuttruff, 2000).

In Brinkmann et al. (2019) werden die Ergebnisse eines Round-Robin-Experiments aus dem Jahr 2019 zur modellbasierten raumakustischen Simulation und Auralisation anschaulich dargestellt; diese zeigen den Stand der Forschung in ebendiesem Bereich. Verschiedene Algorithmen wurden neben der physikalischen Genauigkeit der Simulationen und den daraus

gewonnenen raumakustischen Parametern gemäß ISO 3382-1 auch bezüglich der perzeptiven Beurteilung der Auralisationen untersucht. Als Vergleich dienten hierzu gemessene Impulsantworten der gleichen Szenen. Stärken und Schwächen der untersuchten Algorithmen konnten aufgedeckt werden. Die Ergebnisse zeigen, dass die meisten getesteten derzeitigen Simulationsalgorithmen, die auf der Theorie der geometrischen Akustik basieren, offensichtliche Modellfehler erzeugen, sobald ihre Annahmen nicht mehr erfüllt sind. Infolgedessen sind sie weder in der Lage, ein exaktes Muster der frühen Reflexionen, noch eine exakte Vorhersage der raumakustischen Parameter außerhalb eines mittleren Frequenzbereichs von 500 Hz bis 2 kHz zu liefern. Neben diesen physikalischen Modellabweichungen liefern die Algorithmen meist plausible, aber nicht authentische Auralisationen, d.h. der Unterschied zwischen simulierten und gemessenen Impulsantworten der gleichen Szene war immer deutlich hörbar. Vor allem Abweichungen in der Klangfarbe und der Wahrnehmung von Schallquellpositionen zwischen Messung und Simulation traten auf, die zu einem großen Teil auf Fehler bei der Simulation von frühen Reflexionen zurückzuführen sind, bedingt durch die vereinfachte Verwendung von Absorptions- und Streukoeffizienten zufällig einfallenden Schalls und die fehlende oder unzureichende Modellierung von Beugung. Daher sind raumakustische Simulationen und deren Auralisationen, anders als die datenbasierten Verfahren auf Grundlage von Messungen, nach dem derzeitigen Stand der Technik noch nicht geeignet, die wahrnehmbaren Eigenschaften von Schallquellen in virtuellen akustischen Umgebungen genau vorherzusagen.

Des Weiteren gibt es auch modellbasierte Ansätze mit Feedback-Delay-Networks zur Umsetzung von Erstreflexionen und Nachhall bei der binauralen Simulation (Menzer, 2011)(Carty & Lazzarini, 2010), welche jedoch zumeist nicht das Ziel einer möglichst plausiblen Darstellung einer echten Räumlichkeit verfolgen, sondern vielmehr auf einen prinzipiell als natürlich empfundenen Nachhall und eine möglichst geringe Rechenlast achten.

Diese von Brinkmann et al. (2019) durchgeführten Versuche zur wahrgenommenen Qualität der untersuchten algorithmischen raumakustischen Simulationen sind immer in direktem Zusammenhang mit der durchgeführten Auralisation zu bewerten. Im Falle eines komplett modellbasierten Ansatzes muss also auch immer der Einfluss der HRIRs, die für die kopfbezogene Filterung genutzt wurden, in den direkten Vergleich mit den kopfbezogenen Übertragungsfiltern der direkten datenbasierten Form bei Messung von BRIRs gestellt werden; wird hier z.B. das gleiche Kunstkopfmikrofon genutzt, so ist der Einfluss quasi nicht vorhanden. Bei unterschiedlichen kopfbezogenen Filtern ist ein direkter Vergleich eigentlich nicht möglich. Des Weiteren ist wichtig zu erwähnen, dass in einem modellbasierten Fall auch alle Anteile der Übertragungskette, vor allem das Übertragungsverhalten des Lautsprechers simuliert werden müssen. Dieser Einfluss ist in einem vollständig datenbasierten Verfahren bereits direkt in den BRIRs enthalten.

4.2.2.3 Schlussfolgerung an das System

Neben diesen Verfahren der vollständig datenbasierten und vollständig modellbasierten binauralen Raumsimulation, gibt es auch Mischformen, welche anteilig auf verschiedene Verfah-

ren setzen und so beispielsweise die für den Raumeindruck wichtigen Teil des Direktschalls und der frühen Reflexionen datenbasiert umsetzten, während modellbasiert die mit besserem Ergebnis zu erwartenden Anteile des diffusen Schallfelds umgesetzt werden (Stade & Arend, 2016).

Diese dargestellten Ergebnisse zur Leistungsfähigkeit von modellbasierten Simulationen bzw. Auralisationen unter Beachtung der (auch in Abschnitt 5.2) diskutierten praktischen Umsetzungsmöglichkeiten an der Hochschule der Medien Stuttgart führen dazu, dass das System der vorliegenden Arbeit auf Grundlage der rein datenbasierten Form entwickelt worden ist, welche das Ziel einer möglichst authentischen Auralisation von real existierenden Räumen bzw. akustischen Umgebungen verfolgt.

Im Verlauf des folgenden Kapitels soll im Zuge der Vorstellung einzelner Systemkomponenten und der dahinter liegenden Methodik der datenbasierte Ansatz näher betrachtet werden und - falls möglich - direkte Anforderungen an die Systemkomponenten gestellt bzw. die Qualität der Letzteren eingeschätzt werden.

5 System zur binauralen Simulation von Regieräumen

5.1 Hauptanforderungen an das Gesamtsystem

In Abschnitt 2 sind die Ziele und Hauptanforderungen an das Gesamtsystem zur binauralen Simulation bzw. Auralisation von Regieräumen und Mischkinos genannt; aus diesen ergeben sich diverse funktionale Anforderungen sowie Benutzeranforderungen an die einzelnen Systemmodule. Ehe in den folgenden Kapiteln die einzelnen Konzepte und Entwürfe, sowie auch konkrete Implementierungsschritte dargestellt werden, soll für jedes Systemmodul zunächst eine getrennte Anforderungsanalyse durchgeführt werden. Diese stellt jeweils eine Zielsetzung an die Systemmodule dar, die jedoch nicht für alle Module im Rahmen dieser Arbeit praktisch vollständig erreicht wird; es sollen somit auch Ausblicke auf noch bevorstehende Schritte der Systementwicklung, sowie bereits umgesetzt Schritte kritisch hinterfragt werden. Dabei wird dieses System auch als ein *Work in Progess* für weitere hochschulinterne Entwicklungen gesehen. Bevor ein Einstieg in die Konzeption und Entwicklung der Systemmodule getätigt wird, soll noch eine Motivation des gewählten binauralen Renderings erfolgen und die methodischen Konzepte des Vorgehens beschrieben werden. Diese Inhalte werden aufgrund ihrer Wichtigkeit abgesetzt von den restlichen Grundlagen behandelt.

-

5.2 Methodik

In diesem Kapitel werden die wichtigsten algorithmischen und mathematischen Methoden vorgestellt, die für die Grundfunktionalität eines binauralen Renderers erforderlich sind. Danach werden schrittweise alle benötigten Komponenten und Konzepte vorgestellt und diskutiert, die bei der Umsetzung des Gesamtsystems Anwendung finden.

5.2.1 Faltungstheorem

Die kopfbezogenen Übertragungsfunktionen werden in den meisten Fällen im Zeitbereich als Impulsantworten gespeichert. Eine Impulsantwort ist die Ausgabe eines Systems, wenn es mit einer Deltafunktion angeregt wird. Für lineare zeit-invariante Systeme (LTI-Systeme) liefert die Impulsantwort eine vollständige Darstellung der Systemeigenschaften. Die Kenntnis der

Impulsantwort ermöglicht die Berechnung des Systemausgangs für jedes mögliche Eingangssignal. Die Beziehung zwischen dem Eingang $x(n)$, der Systemimpulsantwort $h(n)$ und dem Ausgang $y(n)$ wird durch die Operation der Faltung beschrieben (Smith, 1999).

Mathematisch wird die diskrete (digitale) lineare Faltung von zwei Sequenzen unbestimmter Länge beschrieben als:

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k), \qquad n, k \in \mathbb{Z} \tag{5.1}$$

In praktischen Anwendungen stellen die Sequenzen ein Eingangssignal $x(n)$ und einen Filterkern $h(n)$ dar. Während das Eingangssignal eine unbestimmte Länge haben kann, was bei der Echtzeit-Audioverarbeitung der Fall ist, hat der Filterkern im Falle eines Filters mit endlicher Impulsantwort (FIR-Filter) - wie auch bei der Nutzung von endlichen HRIRs oder BRIRs - eine bestimmte Länge. Obwohl das Eingangssignal $x(n)$ ein kontinuierlicher Strom von Audiosamples ist, wird die Verarbeitung auf diskreten Teilen des unendlichen Signals durchgeführt, die als *Frames* (oder *Blöcke*) mit einer endlichen Anzahl von Samples dargestellt werden (Wefers, 2014). Die Faltung wird also an einem Eingangssignal $x(n)$ mit der Länge N Samples und mit einer *finiten* Impulsantwort $h(n)$ mit der Länge M Samples durchgeführt, was zu einem Ausgangssignal $y(n)$ mit der Länge $N+M-1$ Samples führt (Smith, 1999). Da sich die vorliegende Arbeit mit der Echtzeitverarbeitung befasst, beziehen sich alle Beschreibungen und Diskussionen über die Audioverarbeitung immer auf ein unbestimmtes Eingangssignal, das in Eingangsblöcken verarbeitet wird. Die Begriffe *Block* und *Frame* werden im weiteren Verlauf austauschbar verwendet.

Der im letzten Absatz beschriebene Fall mit einem Filterkern der Länge M Samples bringt Gleichung 5.1 in folgende Form:

$$y(n) = x(n) * h(n) = \sum_{k=1}^{M} x(k) \cdot h(n-k), \qquad n, k \in \mathbb{Z} \tag{5.2}$$

Wenn $X(k)$ die digitale Fouriertransformierte des diskreten Eingangssignals $x(n)$ ist und die Transformierte der Impulsantwort $h(n)$ $H(k)$ ist, führt die Eigenschaft des Faltungstheorems zu folgendem Zusammenhang (Brigham, 1997):

$$x(n) * h(n) \circ\!\!-\!\!\bullet X(k) \cdot H(k), \qquad n, k \in \mathbb{Z} \tag{5.3}$$

Es gibt verschiedene Implementierungen der Faltung; eine detaillierte Betrachtung aller verschiedenen Arten würde den Rahmen dieser Arbeit übersteigen und ist in Wefers (2014) zu finden. Die nachfolgenden Erläuterungen konzentrieren sich auf die direkte und die spektrale Faltung, da sie zwei der am häufigsten verwendeten Algorithmen in der modernen digitalen Signalverarbeitung sind. Eine gängige Implementierungstechnik ist die partitionierte Faltung.

5.2.1.1 Direkte Faltung im Zeitbereich

In ihrer einfachsten Form kann die lineare Faltung durch Auswertung von Gleichung 5.2 berechnet werden; dies wird *direkte Faltung* im Zeitbereich genannt. Mit dieser Form können gute Leistungen für sehr kurze Filterkerne mit $M \leq 32$ erreicht werden, da so die Anzahl der Multiplikationen und Additionen, welche vom Algorithmus berechnet werden, relativ klein ist. In der Praxis wird die Leistung der direkten Faltung auch durch weitere Faktoren wie Speicherzugriffsmuster, Vektorisierung oder Cache-Nutzung beeinflusst (Wefers, 2014).

5.2.1.2 Schnelle Faltung im Frequenzbereich

In Audioanwendungen ist es nicht ungewöhnlich, dass die genutzten Filterkerne bzw. Impulsantworten deutlich länger als $M \leq 32$ Samples sind, und somit kann die Berechnungsgeschwindigkeit der direkten Faltung im Zeitbereich bei Echtzeitanwendungen zu einem Problem werden. Das Problem kann gelöst werden, indem die Operation in den Frequenzbereich übertragen wird, da die Faltung im Zeitbereich einer Spektrenmultiplikation im Frequenzbereich entspricht (Smith, 1999). Auf modernen Computern kann die diskrete Fouriertransformation (DFT) mit Hilfe des Algorithmus der schnelle Fouriertransformation (FFT) sehr schnell berechnet werden. Die DFT zerlegt das Signal in periodische Funktionen und das resultierende Signal ist eine Repräsentation des Signals im Frequenzbereich. Unter Beachtung des Faltungstheorems kann die Faltung durch punktweise Multiplikation der DFT-Koeffizienten für die beiden Signale durchgeführt werden. Die inverse DFT wird verwendet, um das resultierende Signal zurück in den Zeitbereich zu transformieren. Die Multiplikation im Frequenzbereich wird als *schnelle Faltung* im Frequenzbereich oder *spektrale Faltung* bezeichnet (Wefers, 2014). An dieser Stelle ist zu erwähnen, dass die FFT eine schnelle Implementierung der DFT ist und die spezifischen Besonderheiten der DFT und FFT außerhalb des Rahmens dieser Arbeit liegen und nicht näher behandelt werden.

In einer praktischen Implementierung müssen bei der Durchführung der spektralen Faltung zwei Aspekte berücksichtigt werden:

1. **Das Eingangssignal und die Impulsantwort müssen die gleiche Anzahl an DFT-Koeffizienten haben.** Da die Multiplikation elementweise erfolgt, müssen die Signale im Frequenzbereich $H(n)$ und $X(n)$ die gleiche Länge haben. Für eine korrekte Darstellung der Zeitbereichssignale im Frequenzbereich muss die Anzahl der durch die DFT gegebenen Frequenz-Bins größer oder gleich der Länge des Zeitbereichsignals sein. Andernfalls wird das Ausgangssignal abgeschnitten. In den meisten Fällen werden die Längen der beiden Sequenzen im Zeitbereich unterschiedlich sein. Betrachten wir den Fall, dass ein 1024-Sample Eingangsframe mit einer 512-Tap langen Impulsantwort gefaltet wird. Eine 512-Punkte FFT der Impulsantwort ergibt ein Signal im Frequenzbereich mit 512 Frequenz-Bins. Das Problem lässt sich lösen, indem das kürzere Signal im Zeitbereich mit Zero-Padding versehen wird, damit es dem längeren der beiden Signale

entspricht. In dem beschriebenen Fall werden 512 0-Samples am Ende der Impulsantwort hinzugefügt, sodass beide Signale eine Länge von 1024 Samples haben, bevor die FFT durchgeführt wird.

2. **Die FFT-Länge muss lang genug sein, um die korrekte lineare Faltung zu erhalten.** Wie der Name schon sagt, ist die DFT eine diskrete Transformation, die von einem unbestimmten periodischen Signal ausgeht. Die lineare Faltung ist aperiodisch, und Echtzeit-Audiosignale sind ebenfalls aperiodisch. Die diskrete Faltung wird daher im Kontext der DFT als *zirkuläre Faltung* umformuliert (Wefers, 2014). Das bedeutet, dass die Indizes der resultierenden Sequenz einer K-Punkt-DFT-Spektralfaltung modulo K ausgewertet werden, wodurch die Linearität der Ausgabe verfälscht wird. Um die Integrität der linearen Faltung zu erhalten, muss die Auflösung der DFT $K \geq M+N-1$ beachten, wobei M die Länge des Eingangssignals und N die Länge der Impulsantwort ist.

Im Zusammenhang der binauralen Simulation wird die Faltung angewendet, um einen FIR-Filter, der die HRIR bzw. BRIR-Systemantwort enthält, auf ein beliebiges Eingangssignal anzuwenden. Wie bereits erwähnt, wird das Eingangssignal in Blöcken einer definierten Länge verarbeitet, und die Filterkerngröße der HRIRs bzw. BRIRs ist ebenfalls bekannt. Da der Ausgang einer linearen Faltung ein Signal ist, das länger als das Eingangssignal ist, erfordert die Verarbeitung aufeinanderfolgender Eingangsblöcke eine Methode zur Kombination der hinzugefügten Samples. In diesem Fall stehen zwei Methoden zur Verfügung, die im Folgenden beschrieben werden: *overlap-add* und *overlap-save* (Wefers, 2014). Für die nachfolgenden Betrachtungen sei $x(n)$ ein unendliches Eingangssignal, das in Blöcken der Größe B verarbeitet wird, wobei der erste Block $x_1(n)$ genannt wird. $h(n)$ sei eine Impulsantwort der Länge N. Das Ausgangsframe erfordert ein Zeitbereichssignal $y(n)$ der Länge B.

Overlap-Add-Verfahren

Der erste Block $x_1(n)$ des Eingangssignals wird wie folgt verarbeitet:

1. Eine Variable t mit $K-B$ 0-wertigen Samples wird initialisiert, wobei $K \geq B+N-1$.
2. Das Eingangssignal $x_1(n)$ und die Impulsantwort $h(n)$ werden auf die Länge K mit 0-wertigen Samples aufgefüllt.
3. Eine K-Punkt-FFT wird auf $x_1(n)$ und $h(n)$ angewendet
4. Die resultierenden Signale $H(n)$ und $X_1(n)$ der Länge K im Frequenzbereich werden elementweise multipliziert
5. Die resultierende Sequenz $Y(n)$ wird einer inversen FFT unterzogen, um das Ausgangssignal $y(n)$ der Länge K zu erhalten.
6. Die elementweise Addition wird an der in der Variablen t gespeicherten $K-B$-Sequenz und den ersten $K-B$-Elementen in $y(n)$ durchgeführt.
7. Die ersten B Abtastwerte des Ausgangssignals $y(n)$ werden in das Ausgangsframe übertragen.

8. Die letzten $K - B$ Abtastwerte des Ausgangssignals $y(n)$ werden in der Variablen t gespeichert

9. Der nächste Eingangsblock wird zur Verarbeitung abgerufen.

Overlap-Save-Verfahren

Das Overlap-Save-Verfahren verwendet ein K-Punkt großes gleitendes Fenster, in dem zunächst die ersten K Samples des Eingangssignals $x(n)$ gespeichert werden. Jedes Mal, wenn ein neues Eingangssignal $x_1(n)$ zur Verarbeitung abgerufen wird, werden die letzten B Samples im gleitenden Fenster nach links verschoben und die nächsten $K-B$ Samples werden rechts im gleitenden Fenster hinzugefügt.

Die Schritte 2-5 aus dem im letzten Abschnitt beschriebenen *Overlap-Add-Verfahren* werden auch beim *Overlap-Save-Verfahren* durchgeführt, wobei das gleitende Fenster die Eingabesequenz $x_1(n)$ darstellt. Dann wird wie folgt verfahren:

7. Die ersten $K - B$ Samples der Ausgabesignals $y(n)$ werden verworfen und die verbleibenden B Samples werden in das Ausgangsframe übertragen.

8. Der nächste Verarbeitungsblock wird abgerufen und das gleitende Fenster wird verschoben.

5.2.1.2.1 Partitionierte Faltung Eine gängige Implementierung des Faltungsalgorithmus ist die der gleichmäßig oder ungleichmäßig partitionierten Faltung, wobei die ungleichmäßig partionierte Faltung die Leistung in einigen Fällen verbessern kannWefers, 2014. Diese ist besonders nützlich, wenn die Filterkerne sehr lang sind und das Zero-Padding des Eingangssignals zur Anpassung an die Länge der Impulsantwort rechnerisch ineffizient wird. Die partitionierte Faltung hilft, die Latenzzeit zu minimieren, indem sie kleinere Blockgrößen zulässt. Des Weiteren ermöglicht sie die Faltung mit sehr langen Impulsantworten, die andernfalls möglicherweise zu Unter- oder Überläufen der Ein- bzw. Ausgabe führen könnte, wenn die Verarbeitung in Echtzeit nicht Schritt halten kann (Armelloni et al., 2003).

5.2.2 Kunstkopfmikrofon und Kopfhörer

Wie in Unterunterabschnitt 4.2.2.3 diskutiert, soll dieses System auf der Grundlage eines datenbasierten Ansatzes der Simulation von virtuellen Lautsprechern erfolgen. Da die Simulation zunächst nicht individualisiert werden soll, ist es nötig, die BRIR-Datensätze mithilfe eines Kunstkopfmikrofons zu erstellen. Dies entspricht einem nicht-individualisierten Ansatz. Aus dieser Motivation heraus sollen in diesem Kapitel die grundlegenden Bestandteile einer menschlichen HRTF und ihre anthropometrischen Ursachen geklärt und das Konzept eines Kunstkopfmikrofons vorgestellt werden. Des Weiteren wird aus diesen Erkenntnissen und zitierten Untersuchungen zur Qualität der Abbildung durch Kunstkopfmikrofone eine Abschätzung der Eignung für das gewählte Kunstkopfmikrofon unter Nutzung des in Absatz 5.4.2.2.1 beschrieben Drehaufbaus gegeben.

Ein Kunstkopfmikrofon ist eine modellierter künstlicher Kopf mit Druckempfängern an den Positionen, an welchen bei einem Menschen die Ohren bzw. Trommelfelle als Receiver (Schallwandler) sitzen. Die Idee eines solchen Mikrofons geht bereits auf die 1920/1930er-Jahre zurück, in denen Harvey Fletcher den 1933 als ersten Kunstkopf vorgestellten Typ entwickelte und patentierte (Weinzierl, 2008). Aufnahmen mit dem Kunstkopfmikrofon, welche somit die Ohrsignale nachbilden, können anschließend über Kopfhörer abgespielt werden und sollen so den natürlichen Höreindruck beim Hören über die Kopfhörerwiedergabe, welche nicht unserem Hören unter *natürlichen Bedingungen* entspricht, nachbilden; man spricht aus diesem Grund auch von kopfbezogener Stereofonie. Die Wiedergabe der Ohrsignale ist nicht zwingend an die Kopfhörerwiedergabe gebunden, jedoch kann durch sie die nötige Kanaltrennung leicht erreicht werden und es kommt nicht zu ungewollten Einflüssen des Abhörraums sowie des menschlichen Körpers des Hörerenden bei der Wiedergabe. An dieser Stelle wird deutlich, dass die Qualität des empfundenen natürlichen Höreindrucks eines jeden Probanden, dem die mit dem Kunstkopfmikrofon erstellten Inhalte präsentiert werden, davon abhängig ist, wie die in Abschnitt 4.1 vorgestellten interauralen sowie monoauralen Cues des Kunstkopfmikrofons mit denen des Probanden übereinstimmen. Des Weiteren ist auch der Einfluss des elektroakustischen Schallwandlers, welcher die Übertragungskette der binauralen Inhalte zu den Ohren des Probanden bestimmt, zu betrachten. An dieser Stelle gibt es viele Einflussfaktoren, welche wisssenschaftlich ausreichend untersucht worden sind, und im weiteren Verlauf dieses Abschnitt 5.2 schrittweise in der nötigen Relevanz für diese Arbeit vorgestellt werden.

Kunstkopfmikrofone bestehen laut Definition aus Kopfformen, welche *dem Normkopf* entsprechen und somit einen durchschnittlichen menschlichen Kopf abbilden. Des Weiteren ist hier auch die Form der Ohrmuscheln (*engl. Pinnae)* und des Gehörgangs, falls dieser überhaupt haptisch nachgebildet wird und nicht elektrisch entzerrt wird, durchschnittlich. Was bedeutet nun *durchschnittlich* für die wahrgenommene Qualität und unterscheiden sich bestimmte Kunstkopfmodelle in ihrer Qualität deutlich voneinander?

In Abschnitt 4.1 sind die Konzepte und mathematische Methoden zur Beschreibung des interauralen, sowie monoauralen menschlichen Hörens und der wirksamen Cues dargestellt. Wie schlagen sich diese jedoch in den Kopfbezogene Übertragungsfunktionen (HRTFs) konkret nieder und welche frequenzspezifischen Einflusfaktoren der menschlichen Anatomie sind ausmachbar? Dazu soll zunächst der Einfluss der menschlichen Anatomie auf die HRTFs grundlegend in Bereiche aufgeteilt werden, wobei sich prinzipiell zwei Kategorien definieren lassen: Komponenten, die unabhängig vom Richtungseinfall des Schalls sind und welche, die eine Richtungsabhängigkeit aufweisen.

Abbildung 5.1 zeigt, dass die richtungsabhängigen Komponenten der kopfbezogenen Übertragungsfunktion deutlich überwiegen. Diese setzen sich aus Einflüssen des Korpus/Torso, der Schulter, des Kopfes und der Ohrmuscheln zusammen und entstehen durch die bekannten Konzepte der Laufzeit, Abschattung, Beugung, Reflexion/Streuung, den daraus resultierenden konstruktiven und destruktiven Interferenzen bei Überlagerung und den wiederum daraus resultierenden Unterschieden beider Ohrsignale. Jedoch existieren auch zwei Komponenten, welche einen von der Schalleinfallsrichtung unabhängigen Einfluss auf die kopfbezogene Über-

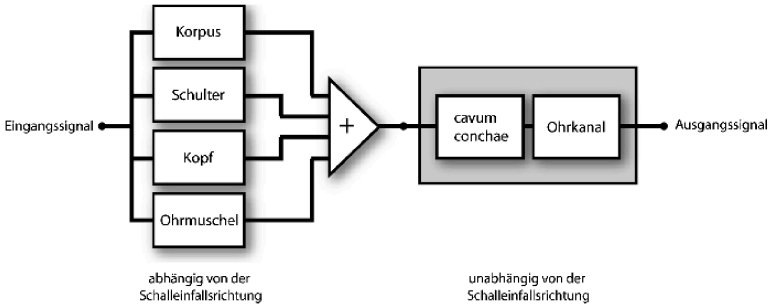

Abbildung 5.1: Richtungsabhaengige und richtungsunabhaengige Komponenten der kopfbezogenen Uebertragungsfunktion (Weinzierl, 2008)

tragungsfunktion haben: den Eingang in den Ohrkanal bzw. den Ohrkanaleingangstrichter (cavum conchae) und den Ohrkanal selbst.

Abbildung 5.2 zeigt den Verlauf einer kopfbezogenen Übertragungsfunktion (Außenohrübertragungsfunktion) eines Probanden bei frontalem Schalleinfall, also ohne Einflüsse der ITD und ILD. An ihr sind die von der Schalleinfallsrichtung unabhängigen Einflüsse abzulesen, des Weiteren können Bereiche, die von einer linearen Übertragungsfunktion (Messmikrofon) abweichen, dem Einfluss von bestimmten Körperteilen zugewiesen werden. Es sind vier Hauptbereiche in dieser Übertragungsfunktion auszumachen:

- **1**: Im tieffrequenten Bereich ($f < 200$ Hz) ist die HRTF weitestgehend linear, was sich mit durch Beugung des Schalls um die anatomischen Hindernisse (Kopf) herum aufgrund der großen Wellenlänge (λ » Abmessungen des Kopfes) begründen lässt. Danach beginnen Kopf und Schulter im mittleren Frequenzbereich bereits eine Wirkung auf der Verlauf der HRTF zu zeigen.
- **2**: Dies stellt den Einbruch in der Übertragungsfunktion aufgrund der charakteristischen Schulterreflexion dar; anhand der genauen Lage des Einbruchs lässt sich die Halslänge des Probanden oder des Kunstkopfmikrofons (*Head and Torso*) ablesen.
- **3**: Hier ist die $\lambda/4$-Resonanz des Ohrkanals abzulesen, sowie die ungeradzahligen Vielfachen der Letzteren aufgrund des einseitig offenen Gehörkanals; diese sind vollständig statisch zu betrachten und ändern sich auch bei Änderung des Schalleinfalls nicht.
- **4**: Im hochfrequenten Bereich (typischerweise $f > 3$ kHz) sind die deutlichen Einflüsse der Ohrmuschel (die sogenannten *Pinnae-Cues*) ablesbar; diese zeichnen sich durch zum Teil sehr starke Einbrüche sowie Überhöhungen aufgrund der komplexen Interferenzmuster innerhalb der Ohrmuschel aus und sind sehr individuell. Es können jedoch charakteristische Bereiche im Frequenzband für bestimmte Richtungen in der Medianebene, in der die *Pinnae-Cues* bei der Wahrnehmung und Lokalisation eine entscheidende Rolle spielen, ausgemacht werden, welche in den sogenannten *Blauertschen Bändern* beschrieben werden.

Betrachtet man den Einfluss der verschiedenen Körperteile auf die HRTF, so lässt sich feststellen, dass den größten Einfluss auf deren Verlauf der Kopf und die Ohrmuscheln haben; der Kopf selbst ist hierbei vor allem in der Horizontalebene wirksam, während die Ohrmuscheln hauptsächlich in der Medianebene Einfluss nehmen. Ein Experiment von Hofman et al.

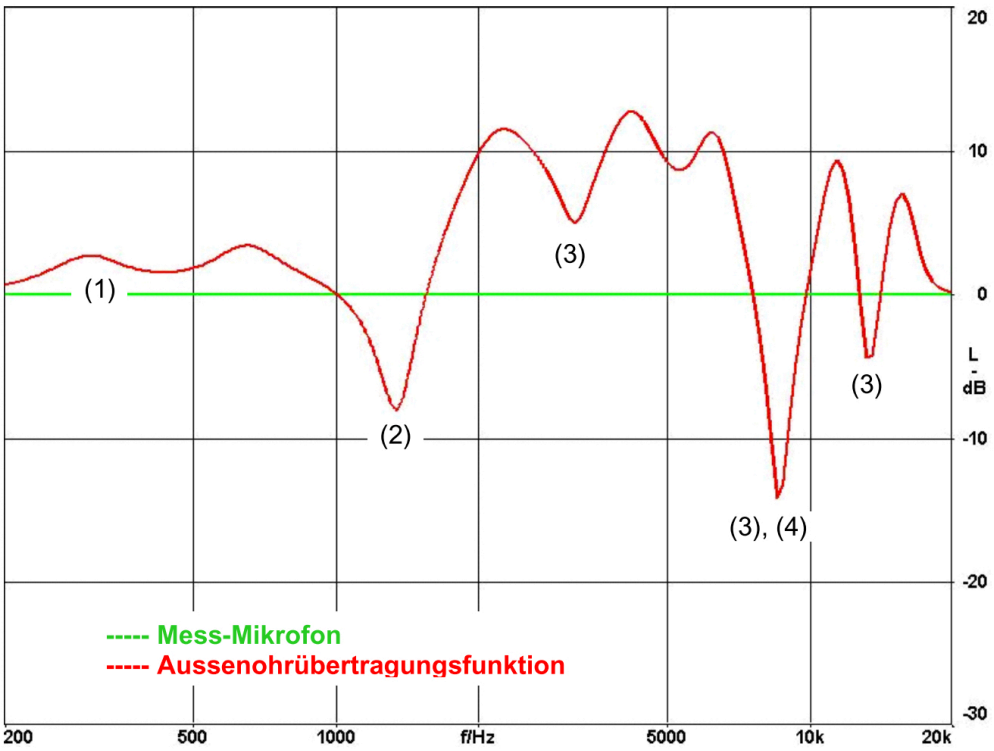

Abbildung 5.2: Beispiel einer gemessenen HRTF eines Probanden: linkes Ohr, frontale Beschallung, Messposition liegt 4 mm ohrkanaleinwaerts. (1): tieffrequenter Bereich (Wellenlänge λ » Abmessungen des Kopfes). (2): Schultereinbruch, destruktive Interferenz aufgrund der Schulterreflexion. (3): Einbrueche bei $(2n\text{-}1)\cdot f_0$, $n \in \mathbb{N}$ mit f_0: $\lambda/4$-Resonanz des Ohrkanals. (4): Interferenzen aufgrund der Ohrmuschel (*Pinnae-Cues*) und des Gehörgangseingangstrichters (cavum conchae) (Daniel et al., 2007)

(1998), in welchem die Interferenzmuster der Ohrmuscheln von Probanden durch den Einsatz einer teilweise anders geformten Ohrmuschel verändert wurden, zeigt, dass durch die Veränderungen der Form der Ohrmuscheln die Lokalisationsfähigkeit in der Medianebene zunächst extrem beeinträchtigt ist, während die Lokalisation in der Horizontalebene noch ausreichend gut funktioniert. Dieses Experiment zeigt des Weiteren, dass der Mensch in der Lage ist, die Lokalisation mit den neuen Ohrmuscheln zu lernen; so verbessert sich die Lokalisationsfähigkeit der Probanden mit den neuen Ohrmuscheln in der Medianebene von Tag zu Tag.

Die Schultern wirken bei bestimmten Frequenzen mit etwa ±5 dB und der Torso mit etwa ±3 dB auf den Frequenzgang der HRTF ein (Weinzierl, 2008). Trotz dieser Tatsache bieten nicht alle Kunstkopfmikrofone eine Nachbildung der Schultern und des Torso. Zur Bedeutung der Schulter und des Torso für die Lokalisationsleistung, welche mit binauralen Aufnahmen erreicht werden kann, wurden bereits Untersuchungen angestellt; so hat z.B. Minnaar et al. (2001) die Lokalisationsleistung von Kunstkopfmikrofonen in direkten Hörversuchen mit menschlichen Probanden in der Horizontal- und Medianebene verglichen. Der einzige Kunstkopf ohne eine Schulter/Torso-Nachbildung (namentlich der *Neumann KU 100*) zeigt hier die größten Lokalisationsfehler. Jedoch ist hier unbedingt anzumerken, dass diese vergleichenden Hörversuche mit direkter binauraler Wiedergabe des Materials über Kopfhörer erfolgte und keine dynamische Anpassung an die Kopfbewegungen durchgeführt wurde. Somit soll für einen kurzen Abgleich der ausreichenden Qualität von verschiedenen Kunstkopfmikrofonen

im speziellen Fall der Simulation von virtuellen Lautsprechern nach dem Prinzip des BRS die im folgenden Absatz beschriebenen Betrachtungen gelten.

Rathbone (2000) hat im Rahmen von Versuchen zur Verbesserung des BRS Systems verschiedene Kunstkopfmikrofone hinsichtlich der Lokalisationseigenschaften und Klangfarbe als Messvorrichtung für datenbasierte binaurale Raumsynthese evaluiert; Motivation hierfür war die Tatsache, dass in Hörversuchen mit dem BRS System einige Probanden die Lautsprecher stets eleviert wahrgenommen haben und auch die Einführung eines weiteren Head-Tracking-Freiheitsgrades neben der Kopfdrehung (Yaw), nämlich dem des Nickens (Pitch), keine Verbesserung hinsichtlich dieses Problems gebracht hat. Die untersuchten Kunstkopfmikrofone verschiedener Hersteller unterschieden sich in ihren geometrischen Abmessungen und konnten teilweise eine Schulter/Torso-Konstruktion aufweisen; für die Untersuchungen wurden alle Kunstkopfmikrofone als Schnittstelle zum genutzten diffusfeldentzerrten Kopfhörer exakt diffusfeldentzerrt. Die Ergebnisse zeigen im speziellen Fall der Simulation von virtuellen Lautsprechern, dass es hinsichtlich der Lokalisationsperformance keine signifikanten Unterschiede zwischen den getesteten Kunstkopfmikrofonen gibt; es zeigt sich in etwa die gleiche Lokalisationsunschärfe. Bezüglich der wahrgenommenen Elevation treten stark individuelle Unterschiede bei den Probanden auf, die deutlich von genutzten Kunstkopfmikrofon abhängen; hierbei konnten elevierte Abweichungen von etwa 10 Grad für einzelne Versuchspersonen bei verschiedenen Kunstkopfmikrofonen ermittelt werden. Hinsichtlich der Klangverfärbung kam es bei einem der untersuchten Kunstkopfmikrofone zu deutlichen, von 80 Prozent der Probanden wahrgenommenen Verfärbungen gegenüber der Lautsprecherwiedergabe. Ansonsten sind auch individuelle Schwankungen zu verzeichnen, denen das BRS Verfahren in seiner nicht individualisierten Form generell unterworfen zu sein scheint. Das *Neumann KU 100* Kunstkopfmikrofon ohne Schulter/Torso-Nachbildung zeigt in dieser Untersuchung trotz der fehlenden Schulter/Torso-Nachbildung keine schlechten Eigenschaften. Es schneidet gegen einen Aufbau des *Neumann KU 100* Kunstkopfmikrofons mit einer speziell dafür entworfenen Schulter/Torso-Nachbildung, welches für die Generierung der BRIR-Datensätze des BRS genutzt wurde, nur geringfügig schlechter bei der wahrgenommenen Klangfarbe und der horizontalen Lokalisationsleistung ab; die ungewünschten Elevationseffekte zeigen bei der Version mit Schulter/Torso-Nachbildung sogar eine größere Ausprägung. Somit kann vorausschauend für diese Arbeit das gewählte *Neumann KU 100* Kunstkopfmikrofon unter Nutzung des in Absatz 5.4.2.2.1 beschriebenen Drehtellers ohne Schulter/Torso-Nachbildung als sinnvolle Wahl für eine perzeptiv gute datenbasierte Simulation von Tonregieräumen und Mischkinos gelten.

Vergleicht man in Abbildung 5.3 den Verlauf der HRTFs von zwölf Personen, welche direkt vor dem Trommelfell, am offenen Eingang zum Ohrkanal und am geblockten Eingang zum Ohrkanal gemessen wurden, so fällt auf, dass der Verlauf oberhalb von 1 kHz stark von der Position der Messung abhängig ist; die Unterschiede bei der Messung am geblockten Ohrkanal werden jedoch kleiner. Wie in Abbildung 5.2 zu erkennen ist, sind die Einbrüche oberhalb von 2 kHz durch die Resonanz des Ohrkanals zu verantworten und verschwinden deshalb bei der Messung am geblockten Ohrkanal. Da der Einfluss des Ohrkanals nicht richtungsabhängig ist, kann er durch einen einfachen statischen Filter beschrieben werden. Die Schallwandler in heu-

tigen Kunstkopfmikrofonen befinden sich in der Regel am geschlossenen Ohrkanal bzw. einige Millimeter im Ohrkanal, sodass die Membran nahezu bündig mit der Kopfform abschließt; aufgrund der Tatsache der richtungsunabhängigen Filterung des Ohrkanals wird dadurch keine Richtungsinformation bei der Aufnahme unterschlagen. Vielmehr wird den Ohrsignalen bei der Kopfhörerwiedergabe so die individuelle Ohrkanalresonanz des Hörenden aufgeprägt, da der Ohrkanal auch bei der Kopfhörerwiedergabe unausweichlich durchlaufen werden muss. Die richtungsabhängigen Einflüsse der Ohrmuschel, welche zumindest bei ohrumschließenden Kopfhöreren ebenfalls beschallt wird, sind bei der Kopfhörerwiedergabe, bei der der Schallwandler nahe an der Ohrmuschel sitzt und Letztere zentral und frontal beschallt, zu vernachlässigen; gerade hier sei nochmals daran erinnert, dass die *Pinnae-Cues* vorrangig bei der Elevation von Schallquellen wirken. Die meisten Kunstkopfsysteme sind Köpfe, die sich an Mittelwerten aus anthropometrischen Datenbanken orientieren (DIN V 45608, IEC TR 60959, ANSI S3.36, ITU-T P.58); hierbei ist das Ziel, die Diskrepanzen zum einzelnen Individuum zu minimieren. Gerade diese über mehrere Freiheitsgrade der Kopf-, Ohr- und sonstigen Körperform gemittelten Daten führen jedoch, wie auch Rathbone (2000) in ihrem Vergleich verschiedener Kunstkopfmikrofone zur Umsetzung des BRS herausstellt, zu sehr individuellen Abweichungen bei der Hörempfindung von binauralen Inhalten, die mit unterschiedlichen Kunstkopfmikrofonen aufgenommen bzw. gerendert wurden. So können Fehler in den *Pinnae-Cues* hinsichtlich der individuellen Cues zum Beispiel dazu führen, dass es zu fehlerhafter Lokalisation in der Medianebene kommt (oben/unten bzw. vorne/hinten) oder das Klangbild als allgemein verfärbt und mit *unzureichender Klangfarbe* wahrgenommen wird. Head-Tracking und Lerneffekte bei *längerer Exposition* können diese Abweichungen minimieren.

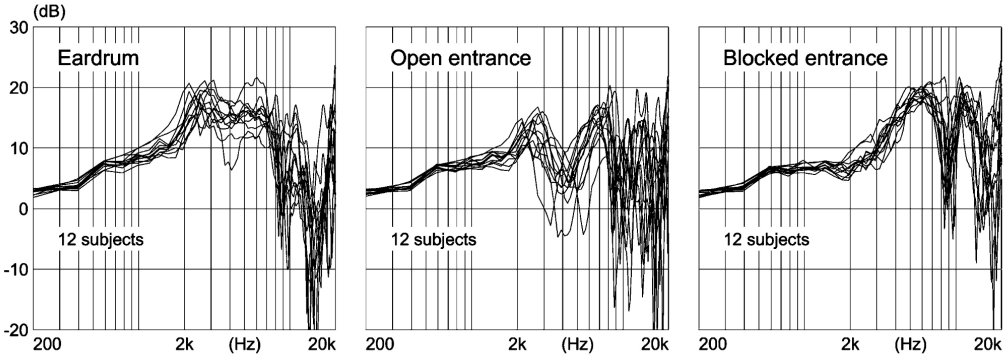

Abbildung 5.3: Kopfbezogene Übertragungsfunktion von zwölf Personen, gemessen an verschiedenen Positionen: Vor dem Trommelfell (links), am offenen Eingang zum Ohrkanal (Mitte) und am geblockten Eingang zum Ohrkanal (rechts) (Hammershoi, 2002)

Kunstkopfmikrofone können divers entzerrt sein; hierbei wird zumeist wie bei normalen Druckempfängern die bestmögliche Anbindung an das im Durchschnitt aufzunehmende Schallfeld gesucht: Eine Freifeldentzerrung bedeutet also, dass das Kunstkopfmikrofon bei frontaler Beschallung den gleichen Frequenzgang aufweist wie ein freifeldentzerrtes Messmikrofon. Dies bedeutet jedoch auch, dass sich für Schallsignale aus anderen Richtungen deutliche klangliche Verfärbungen ergeben, die für schmalbandige Signale 30 dB und mehr betragen können (Daniel et al., 2007). Da Kunstkopfmikrofone allerdings meist in überwiegend diffusem Schallfeld verwendet werden, in welchem Schall aus allen Richtungen mit gleicher Wahrscheinlichkeit

inkohärent auf das Mikrofon trifft, werden sie zumeist diffusfeldentzerrt. Diffussfeldentzerrungen können bestimmt werden, in dem das Kunstkopfmikrofon im Hallraum, welcher ein diffuses Schallfeld nach Norm darstellen kann, vermessen wird und dieses Ergebnis genutzt wird, um einen Korrekturfilter zu erstellen, der das Kunstkopfmikrofon bei difussem Schalleinfall im Frequenzgang linearisiert. Da Freifeld- als auch Diffusfeldbedingungen in der Praxis nie in idealer Weise vorkommen, existieren auch Formen der Entzerrung, welche nur die genannten richtungsunabhängigen Anteile der HRTF (die sich durch feste Resonanzen auszeichnen) kompensieren. Sie werden *ID-Entzerrung* (mit ID: Independent of Direction) genannt und haben sich vor allem bei Messungen von Fahrzeugen etabliert (Daniel et al., 2007). Des Weiteren ist es in einigen Systemen auch möglich, die Entzerrung entsprechend nach Anwendungsfall zu verändern.

Die mit einem Kunstkopfmikrofon gewonnenen Ohrsignale besitzen nun je nach benutztem Modell eine spezifische richtungsbezogene Filterung. Wie bekannt sein sollte, werden zur Nutzbarmachung dieser Ohrsignale vornehmlich Kopfhörer benutzt, da sie in einfacher Weise die erforderliche Kanaltrennung erreichen. Diese Kopfhörersysteme besitzen je nach Bauart und Modell eine Übertragungsfunktion, welche ebenfalls auf der Grundlage der Freifeld-/Diffusfeldentzerrung entstanden oder perzeptiv motiviert ist. Die perzeptiv motivierte Entzerrung bezieht sich vor allem darauf, durch spezielle Filterung einem gewissen Klangideal zu entsprechen, das der Kopfhörer erfüllen soll; dieses ist nicht zuletzt auch durch Marketing-Aspekte motiviert und definiert. So kommt es gerade im Consumer Bereich sehr häufig nicht zu einer verfärbungsfreien Darstellung, sondern z.B. zur relativen Senkung der mittleren Frequenzanteile oder Erweiterungen der Stereobühne durch entsprechende Eingriffe in den Frequenz- und Phasengang des akustischen Systems.

Betrachtet man nun jedoch die Wiedergabeseite der gewonnen Ohrsignale unter dem Aspekt der bestenfalls vollständig *naturgetreuen Darstellung*, so ist durch geeignete Korrekturfilter sicherzustellen, dass die Wiedergabe an der ursprünglichen Position des Aufnahmemikrofons vor oder im Ohrkanal die gleichen Ohrsignale hervorbringt wie in der Originalsituation. Dies kann mit einer Messung der Übertragungsfunktion des genutzten Kopfhörers auf dem Kunstkopf oder Individuum, mit Hilfe dessen die Ohrsignale generiert wurden, erfolgen, aus der die entsprechenden Korrekturfilter gewonnen werden. Dies ist als eine *Schnittstellenübertragungsfunktion* zu betrachten. Da diese *Schnittstellenübertragungsfunktion* jedoch nicht richtungsabhängig ist, ist sie für das Richtungshören quasi nicht von Relevanz. Trotzdem kann es aufgrund nicht vorhandener Anpassungen dieser Schnittstelle aufgrund deutlicher Überhöhungen und Einbrüche im Frequenzgang der Kopfhörerübertragsfunktion zu Fehlempfindungen kommen, die dann, neben der Fehlempfindung der Klangfarbe, zu Lokalisationsverzerrungen vor allem in der Medianebene führen können. Somit sollte gerade bei Systemen, welche einen Austausch der Einzelkomponenten Kunstkopfmikrofon und - was anwendungsbezogen von mehr Relevanz ist - Kopfhörer möglich machen, eine Anpassung dieser Schnittstelle stattfinden, um eine gleichbleibende Qualität zu gewährleisten.

Aus diesem Grund gibt es hinsichtlich der Plausibilität und Authentizität der dynamischen Binauralsynthese diverse Untersuchungen dieser *Schnittstellenanpassung*. Zunächst soll nochmals erläutert werden, was Freifeld- und Diffusfeldentzerrung im Zusammenhang mit Kopf-

hörern prinzipiell bedeutet (Daniel et al., 2007): Ein freifeldentzerrter Kopfhörer erweckt bei Wiedergabe eines bestimmten Schallsignals genau den gleichen Höreindruck, den ein Proband hätte, wenn er im freien Schallfeld mit dem selben Schallsignal über einen idealen Lautsprecher exakt von vorne beschallt werden würde. Sitzt er jedoch in einem diffusen Schallfeld und wird das Schallsignal über einen idealen Lautsprecher (bei frequenzunabhängiger Nachhallzeit) abgespielt, so entsteht näherungsweise der gleiche Höreindruck wie bei Darbietung über einen diffusfeldentzerrten Kopfhörer. Vergleicht man diese Tatsachen mit den Definitionen der Freifeld- bzw. Diffusfeldentzerrung bei einem Kunstkopfmikrofon, fällt auf, dass die Kombinationen aus freifeldentzerrtem Kunstkopfmikrofon mit freifeldentzerrtem Kopfhörer sowie diffusfeldentzerrtem Kunstkopfmikrofon mit diffusfeldentzerrtem Kopfhörer ideale Schnittstellen bilden. In Abbildung 5.4 ist der Diffusfeldfrequenzgang diverser laut Herstellerangabe diffusfeldentzerrter Kopfhörer dargestellt. Es ist zu erkennen, wie deutlich die Kopfhörer teilweise von dieser Vorgabe abweichen; somit ist es neben der Tatsache, dass auch die Entzerrungen der Kunstkopfmikrofone nicht immer vollends dem Standard genügen, sinnvoll, die Schnittstellenfilterung für jedes Kunstkopfmikrofon in Kombination mit einem bestimmten Kopfhörer eigens zu bestimmen.

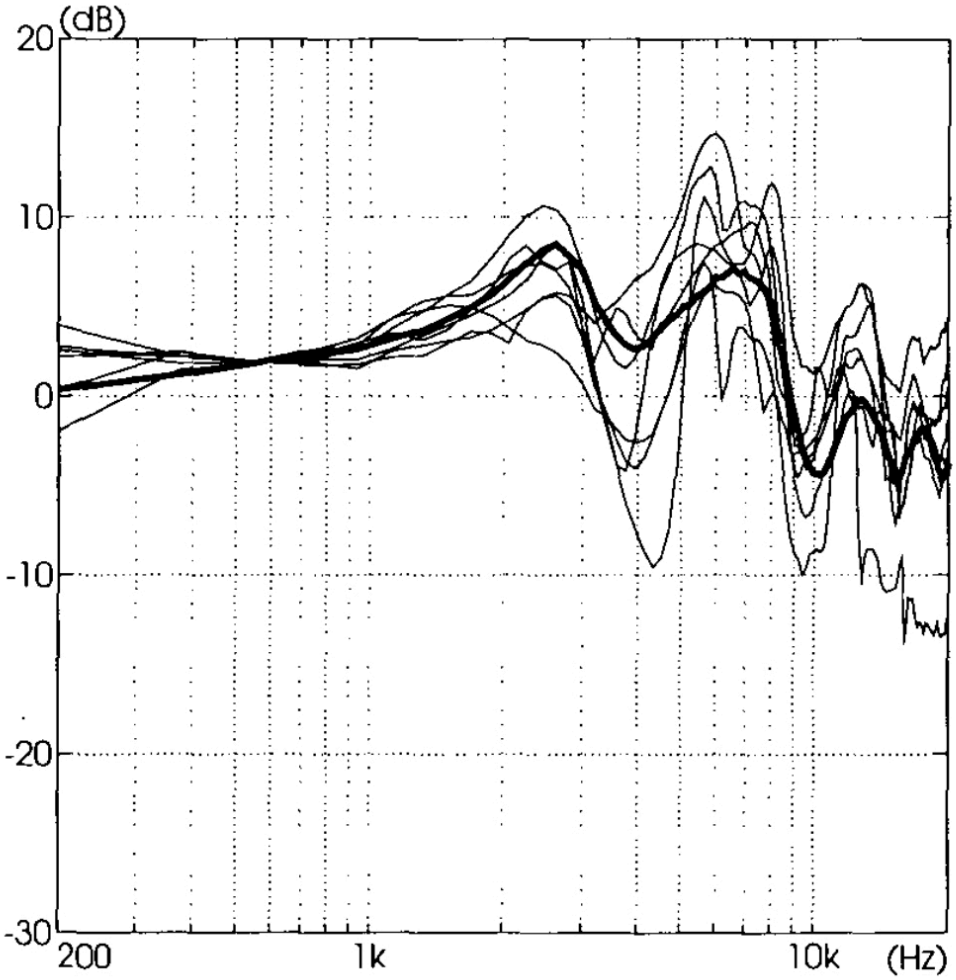

Abbildung 5.4: Diffusfeldentzerrungs-Vorgabe (dicke Linie) und mittlere realisierte Übertragungsfunktionen von sieben kommerziell erhältlichen und laut Herstellerangabe diffusfeldentzerrten Kopfhörern (Messungen wurden am Eingang des geöffnetem Gehörgangs durchgeführt, die $\lambda/4$-Resonanz des Gehörgangs wurde folglich berücksichtigt) (Møller et al., 1995)

Diverse wissenschaftliche Literatur befasst sich mit der Messung und der anschließenden Anpassung der genannten Schnittstelle zwischen Mikrofonen am geblockten Gehöreingang bzw. Kunstkopfmikrofon und Kopfhörer; dabei wird neben Einflüssen der Individualisierung auch die Filterform bzw. die algorithmische Erstellung der Korrekturfilter in Betracht gezogen (Schärer & Lindau, 2009)(Kirkeby & Nelson, 1999). Es zeigt sich bei Lindau (2014), dass individualisierte Entzerrung prinzipiell zwar zu besserer Wahrnehmung führen kann, jedoch nur, wenn die Schnittstelle auch exakt passend gewählt wird. So erreicht z.B. eine individualisierte Entzerrung, die am Ohr des Probanden bestimmt wurde, oder auch gemittelte Filter von mehreren Probanden bei der Nutzung mit nicht-individualisiertem binauralem Rendering (Datensatz wurde mithilfe eines Kunstkopfmikrofons generiert) keine Vorteile; im Gegenteil kommt es zu einer Fehlanpassung, sodass die Schnittstellenkorrektur, welche direkt am Kunstkopfmikrofon bestimmt wurde, mit welchem auch die BRIR-Datensätze gemessen wurden, von Probanden besser bewertet wird. Die Schnittstellenfilter in Lindau (2014) wurden des Weiteren auch minimalphasig approximiert und in Hörtests gegen Filter mit unveränderter Phase erprobt, wobei sich keine der beiden Filterformen als signifikant überlegen herausgestellt hat. Um die Latenz des späteren dynamischen Renderings zu minimieren, fällt im Rahmen dieser Arbeit die Wahl auf einen nicht-individualiersten Schnittstellenfilter in minimalphasiger Filterform. Die von Lindau (2014) genutzten Filter wurden mit einer sogenannten *Highpass-Regularization* bestimmt (genauer wurde eine Hochpass-regulierte Least-Mean-Square (LMS)-Inversion zur Filtererstellung genutzt), wobei bestimmte Bereiche im hohen Frequenzbereich der Messung für die anschließende Generierung der Filtertaps ausgespart wurden. So konnte erreicht werden, dass die teilweise sehr schmalbandigen und starken Einbrüche und Überhöhungen im Bereich des Einflusses der Ohrmuschel nicht durch den Filter verändert wurden, was sich in Hörversuchen als vorteilhaft herausgestellt hat (Lindau, 2014).

Aus den in den letzten Absätzen getätigten Betrachtungen lässt sich festhalten, dass ein *Schnittstellenkorrekturfilter* zwischen der nicht-individualisierten Messung der BRIRs mit einem *Neumann KU 100* Kunstkopfmikrofon und der auf Grundlage der Motivation dieses Gesamtsystems abzusehenden Wiedergabe über diverse Kopfhörer, welche im Besitz der Studierenden sind, sinnvoll ist. Dieser Filter soll zur Latenzminimierung eine minimalphasige Form haben. Im Rahmen dieser Arbeit wurden keine eigenen Filter dieser Form erstellt, jedoch sollen Filter, welche an der *TH Köln* bestimmt wurden, zur Verfügung gestellt werden[1]. Diese Filter wurden mithilfe dem für die Messung der BRIRs dieser Arbeit genutzten *Neumann KU 100* unter Nutzung von hochwertigem I/O-Equipment erstellt, sodass keine nichtkontrollierbaren Einflüsse bei der Nutzung zu erwarten sind. Dabei stehen minimalphasige Filter für 20 verschiedene gängige Kopfhörermodelle zur Verfügung, die unter 12-maligem Auf- und Absetzten der Kopfhörer und Messen der Übertragungsfunktion unter Nutzung eines halbautomatischen Log-Spline-Inversions Algorithmus bestimmt wurden (Bernschütz, 2013). Dabei wurden die Kopfhörer 12 Mal auf- und wieder abgesetzt, um eine stabile und repräsentative Übertragungsfunktion zu erfassen, denn aufgrund nicht gleichbleibender Platzierungen des Kopfhörers auf bzw. über den Ohrmuscheln kommt es immer wieder zu Veränderungen der Übertragungsfunktion. Die Nutzung dieser Daten stellt eine Abweichung zu

[1] http://audiogroup.web.th-koeln.de/wdr_irc.html

der von Schärer und Lindau (2009) und Lindau (2014) als perzeptiv vorteilhaft ermittelten Inversionsmethode dar; dies soll im Rahmen dieser Arbeit jedoch als hinlänglich ausreichend angenommen werden, um überhaupt eine Kopfhörer- bzw. Schnittstellenkorrektur zu ermöglichen.

5.2.3 Diskretisierung sowie Interpolation kopfbezogener Übertragungsfunktionen

Neben einer möglichst großen Übereinstimmung der kopfbezogenen Übertragungsfunktionen des Renderings mit denen des Hörers ist es die zeitlich-räumliche Auflösung des Systems der binauralen Simulation, welche die Abbildungsleistung des Letzteren definiert. Hierbei ist - anders als unsere spezifischen Körperformen bei der Darstellung der kopfbezogenen Übertragungsfunktionen - unser Gehör und dessen Auflösungsvermögen im Zeit- und Frequenzbereich verantwortlich. Prinzipbedingt lassen sich in der digitalen Signalverarbeitung keine werte-kontinuierlichen Daten darstellen, sodass es bei der binauralen Synthese immer zu Diskretisierungen kommt, die vor allem einen Einfluss in ihrer zeitlichen Abfolge bei der dynamischen Anpassung der Kopforientierung (oder auch der Position des Hörers) haben.

Die räumliche Auflösung des menschlichen Gehörs wurde mit verschiedenen Maßen operationalisiert: Dazu gehören die Lokalisationsunschärfe, d.h. der mittlere Fehler bei der Identifizierung der Position einer Schallquelle, und der minimale hörbare Winkel (MAA), d.h. die minimal erkennbare Verschiebung einer Schallquelle. Diese Maße sind unter Freifeldbedingungen bestimmt und beziehen sich folglich allein auf die Richtungslokalisation. MAAs im Bereich von 1° bis 10° können abhängig von der Frequenz und der Richtung des Schalleinfalls festgestellt werden (Mills, 1958). Diese Maße liefern zwar Informationen zum Auflösungsvermögen des Gehörs, jedoch lassen sich daraus keine direkten Rückschlüsse auf die nötige Raster-Auflösung eines Datensatzes von kopfbezogenen Übertragungsfunktionen für die dynamische Binauralsynthese ziehen, bei der der Hörer seinen Kopf frei und in beliebiger Geschwindigkeit ausrichten kann. Des Weiteren ist an dieser Stelle auch eine Unterscheidung zwischen HRTFs, welche unter Freifeldbedingungen gemessen bzw. simuliert wurden, und BRIRs, welche neben dem Einfluss des Direktschalls auch noch diffuse Signalanteile des Raumes in das binaurale Rendering einbeziehen, zu treffen.

Lindau (2014) untersuchte den Einfluss der Raster-Auflösung von BRIR-Datensätzen (auch im Vergleich zu HRIRs). Dazu wurden mit einem HATS in drei charakteristischen akustischen Umgebungen (Tonstudio, zwei Vorlesungssäle sowie in einem reflexionsarmen Raum unter Freifeldbedingungen) Datensätze einer frontal vor dem Kunstkopfmikrofon positionierten Schallquelle in allen verfügbaren Freiheitsgraden der Kopfbewegung mit einer Winkelauflösung von jeweils 1° gemessen (Gier-Winkel bzw. horizontale Kopfbewegung im Bereich ±80°, Nick-Winkel bzw. vertikale Kopfbewegung im Bereich ±35°, Roll-Winkel bzw. laterale Kopfbewegung im Bereich ±60°). Die Einschränkungen der Bewegungsbereiche bei der Binaurale Raumimpulsantwort (BRIR)-Erfassung wurden unter Nennung von Untersuchungen zu typischen, für das natürliche Hören beobachteten Werten (Thurlow et al. (1967),

zitiert nach Lindau (2014)) als auch physiologisch motivierten *Komfort*- (33408-1 (1987), zitiert nach Lindau (2014)) und *Maximal*-Werten (et al. Morgan (1963), zitiert nach Lindau (2014)) motiviert. Unter Freifeldbedingungen wurden die Messungen im Fernfeld durchgeführt, unter reverberanten Bedingungen in etwa zwei Mal der kritischen Distanz, sodass unter Beachtung der Abstrahlcharakteristik der anregenden Schallquelle ein ziemlich ausbalanciertes Direktschall/Difussschall-Verhältnis erreicht wurde. Das binaurale Rendering wurde mit einer Block-partionierten FFT-basierten Faltungs-Engine und einem Head-Tracking durchgeführt, welche eine ausreichend geringe Gesamtsystemlatenz (TSL) garantierte; die Schnittstelle zum genutzten Kopfhörer wurde korrigiert. Dabei wurden die ersten 16384 Samples bei neuer Kopforientierung und dem damit verbundenen Event der Signalverarbeitung geändert. Die Partitionsgröße des anfänglichen BRIR-Teils lag bei 256 Samples, während für den diffusen Nachhall eine Blockgröße von 8192 Samples genutzt wurde. Ein linearer Crossfade mit 5,8 ms Dauer (entsprechend der kleineren Blockgröße) fand im Zeitbereich statt und verhinderte *Switching*-Artefakte. Somit wurden die ersten Ergebnisse eines Filteraustauschs eine Blockgröße nach Erkennen eines Trigger-Events ausgegeben. Durch die Wahl der Crossfadelänge wurde sichergestellt, dass es nicht zu einer hörbaren linearen Interpolation aufgrund des linearen Crossfades kommt, *Switching*-Artefakte jedoch trotzdem unterdrückt werden. Diese grundlegenden Systemparameter stellten sicher, dass ausschließlich der Einfluss der Raster-Auflösung untersucht wurde. Die Untersuchung nutzte zwei verschiedene Stimuli: ein breitbandiges Rosa-Rauschen (5 Sekunden Länge, jeweils 20 ms Fade-In und Fade-Out), welches vor allem spektrale Einflüsse aufdecken sollte, sowie auch ein musikalisches Signal einer akustischen Gitarre (bourrée von J. S. Bach), welches sowohl harmonische Passagen als auch transiente Komponenten enthält. Dabei wurde zunächst mit der vollen 1°-Raster-Auflösung begonnen ehe die Letztere auf Grundlage der Aussagen der Testperson zur Erkennbarkeit der Raster-Auflösung reduziert wurde. Es zeigten sich gemittelt über alle Versuchspersonen und alle akustischen Umgebungen (also sowohl BRIRs als auch HRIRs) die in Tabelle Abbildung 5.5 dargestellten gerade erkennbaren Raster-Auflösungen bei horizontaler (*hor*), vertikaler (*ver*) sowie lateraler (*lat1* und *lat2*) Änderung der Kopforientierung (zu unterscheiden ist zwischen *lat1* und *lat2*, wobei *lat1* die Ergebnisse bei direkter frontaler Beschallung und *lat2* Ergebnisse einer Untersuchung derselben Methodik, jedoch unter Beschallung aus der Richtung direkt oberhalb des Kopfes wiedergibt):

…was audible for	**Noise**	**Guitar**
	hor/ver/lat1/lat2	hor/ver/lat1/lat2
50%	6° x 5° x 16° x 4°	9° x 12° x 16° x 7°
25%	4° x 4° x 12° x 3°	7° x 9° x 12° x 6°
5%	4° x 3° x 8° x 2°	5° x 4° x 8 x 5°
0%	2° x 1° x 3° x 1°	3° x 2° x 3° x 4°

Abbildung 5.5: Gerade noch hörbare Raster-Auflösungen für horizontale, vertikale und laterale (Schallquelle frontal vor Kopf/Schallquelle über Kopf) Kopfbewegungen (Lindau, 2014)

Auffällig ist, dass die gerade erkennbare Raster-Auflösung bei lateraler Kopfbewegung und

Beschallung von oben (angegeben unter *lat2*) als auch bei vertikaler Kopfbewegung und Beschallung von vorne (angegeben unter *vert*) durchschnittlich kleiner ist als bei horizontaler Kopfbewegung; besonders bei der Untersuchung mithilfe des Rauschsignals sind die Ergebnisse auffällig. Dies lässt - wie auch die weitere Versuchsauswertung von Lindau (2014) beschreibt - die Aussage zu, dass die Kammfiltereffekte, welche durch Interferenzen an der Ohrmuschel bei vertikaler Kopfbewegung als auch jene welche, die bei lateraler Kopfbewegung und Beschallung von oben durch die Schultern und den Torso verursacht werden, eine höhere Raster-Auflösung erfordern als Diskretisierungseffekte der ITD und ILD bei horizontaler Kopfbewegung; Lindau (2014) fand im Weiteren Verlauf seiner Arbeit heraus, dass Kammfiltermodulationen bei Messungen ohne Schulter und Torso deutlich weniger ausgeprägt sind. Somit ist zu erwarten, dass Schulter- und Torsoreflexionen - vor allem bei Schalleinfall von oben - bei räumlichen Unterscheidungsaufgaben und damit auch für die Authentizität virtueller akustischer Umgebungen im Allgemeinen eine nicht zu vernachlässigende Rolle spielen. Die entstehenden schmalbandigen Kammfiltereffekte und deren Auswertung durch das Gehör sind kritisch hinsichtlich der Raster-Auflösung des BRIR- oder HRIR-Datensatzes.

Die Tatsache, dass die Untersuchung keine signifikanten Unterschiede der gerade erkennbaren Raster-Auflösung unter Freifeldbedingungen als auch in einem typischen Tonstudio in doppeltem kritischen Abstand hervorbringen konnte, lässt eine Übertragbarkeit der Versuchsergebnisse auf ein System der binauralen Simulation von virtuellen Lautsprechern im Umfeld von Tonregieräumen und Mischkinos zu. Eine Raster-Auflösung der BRIR-Datensätze von 2° für horizontale, 1° für vertikale als auch 1° für laterale Kopfbewegungen bietet also eine räumliche Darstellung, die auch für kritische Hörer, kritische Audioinhalte und diverse Lautsprecherpositionen ausreichend ist. An dieser Stelle sollte bei Betrachtung der Ziele dieser Arbeit herausgestellt werden, dass für musikalische Inhalte eine Auflösung von 5° für horizontale, 4° für vertikale und 5° für seitliche Kopfbewegungen für 95 Prozent der Hörer bereits für eine plausible Auralisation ausreichend ist.

In aktuelleren BRIR-Datensätzen, die beispielsweise Binaurale Raumimpulsantworten (BRIRs) einer Konzerthalle enthalten (Băcilă & Lee, 2019), sowie vor allem in jenen zur Darstellung von virtuellen Lautsprechern (C. Pike & Romanov, 2017a)(Erbes et al., 2015)(Satongar et al., 2014)(Melchior et al., 2014) wird eine Beschränkung der Freiheitsgrade der Kopforientierung nicht mehr getroffen und gleichmäßig aufgelöste Datensätze, jedoch zumeist lediglich im horizontalen Freiheitsgrad der Kopfdrehung gemessen. In den letztgenannten Datensätzen werden allerdings auch Schallquellen in Form von virtuellen Lautsprecher dargestellt, die bei normaler Orientierung des Kopfes im Lautsprechersetup nach vorne den Hörer von hinten beschallen. Damit dem Hörer gleichmäßig entlang des verfügbaren Freiheitsgrads der horizontalen Orientierung ermöglicht wird, sich in gleichbleibender Wahrnehmungsqualität auch den rückwärtigen Schallquellen zuzuwenden, scheint hier eine den kompletten Freiheitsgrad erfassendes Raster der BRIRs sinnvoll und auch den zu erwartenden Bewegungen des Hörers angemessen. Aufgrund der von Lindau (2014) getroffenen Einschränkungen der Bewegungsfreiheit sind Bereiche der Wahrnehmung des Hörers nicht erfasst worden, welche ebenfalls einen Einfluss auf die Hörbarkeit der BRIR- bzw. Kopfbezogene Impulsantwort (HRIR)-Rasterauflösung haben

könnten. Allgemein bekannt ist jedoch, dass zum Beispiel die nicht erfasste Hörwahrnehmung im Rückraum des Menschen nicht besser als im frontalen Bereich ist, sodass dadurch keine höhere Erkennbarkeit von BRIR-Rastern zu erwarten ist. Eine mögliche Abstufung hin zu einer größeren Rasterauflösung in diesen Bereichen erschwert die Messung sowie das binaurale Rendering massiv, da es nicht mehr für alle Lautsprecherpositionen und Kopforientierungen gleichförmig erfolgen kann. Der gedachte Nutzen einer möglichen Datenreduktion ist somit schlussendlich nicht vorhanden.

Unter Aspekten der Praktikabilität und Performance des binauralen Renderings zu betrachten, dass höhere Auflösungen von HRIR- bzw. BRIR-Datensätzen eine längere Messung verursachen und die höhere Datenmenge wiederum zu einem höheren Rechenaufwand sowie einer größeren benötigten Speicherkapazität bei der Auralisation führt. Daher sind gemessene Schwellenwerte von gerade noch wahrnehmbarer Raster-Auflösung entscheidend, um den Aufwand für die Messung und Auralisation binauraler Daten zu optimieren, ohne dabei Wahrnehmungsartefakte zu verursachen. Interessant ist neben der in den letzten Absätzen betrachteten tatsächlichen gerade noch erkennbaren Raster-Auflösung die Wahrnehmung von Interpolationen, die zwischen gemessenen bzw. vollständig simulierten Daten, durchgeführt werden. Da dies eine Untersuchung und deren Interpretation zu diversen Interpolationsalgorithmen unter verschiedenen charakteristischen Bedingungen in verschiedenen akustischen Umfeldern erfordert, sei auf eine weitere Betrachtung dieses Feldes im Rahmen dieser Arbeit verzichtet.

Die mithilfe des in dieser Arbeit gebauten Drehtellers gemessenen BRIR-Datensätze in einer horizontalen Raster-Auflösung (auch *Schrittweite* genannt) von 1,8° erreichen die benötigten 2° Raster-Auflösung in ebendieser Dimension und sind folglich für eine plausible bzw. authentische Auralisation ausreichend. Somit ist die Betrachtung von Interpolationsalgorithmen und deren Einfluss auf die Wahrnehmung bei einer durchgeführten Auralisation mit den Letzteren im Rahmen dieser Arbeit nicht notwendig. Konstruktionsdetails, sowie eine Diskussion des konzipierten Drehtellers sind in Absatz 5.4.2.2.1 beschrieben.

5.2.4 Richtungsspezifische Messung

Kopfbezogene Impulsantworten (HRIRs) werden unter reflexionsarmen Bedingungen gemessen. Dabei werden zumeist Drehsysteme verwendet, welche die zu vermessende Person relativ zu einem Lautsprecherarray drehen und/oder das Array wird selbst um die Person gedreht, sodass es möglich ist, die Schallquellen an den gewünschten Stellen des Datensatz-Rasters um den Kopf herum zu positionieren (Warusfel et al., 2018)(Algazi et al., 2001). Möglichkeiten bestehen auch in der Nutzung eines kompletten Lautsprecher-Domes um die Person herum (Chevalier et al., 2018). Bei Vermessungen von Menschen ist hierbei die ständige Überprüfung der Kopfposition und -orientierung besonders wichtig, um Abweichungen in der Messung feststellen zu können; jedoch werden auch Positionsdaten von Lautsprechern geprüft (Warusfel et al., 2018). Auch die Vermessung mit nur einem Lautsprecher ist theoretisch möglich (Li & Peissig, 2020); sofern sich der Lautsprecher nicht frei im Raum bewegen kann, muss um auf diese Art und Weise jedoch eine HRIR für jede gewünschte Richtung

zu erhalten, ein Drehsystem entworfen werden, welches den Kopf in den gewünschten Freiheitsgraden der Messung positioniert. Da ein echter menschlicher Kopf auf diese Weise nicht vermessen werden kann, kann eine solche Messung lediglich mit einem Kunstkopfmikrofon wie bei Bernschütz (2013) erfolgen. Des Weiteren werden solche Drehsysteme benutzt, um BRIRs zu messen, welche eine spezifische räumliche Antwort für jede relative Quelle-/Hörer-Positionierung bzw. Orientierung zueinander haben und folglich nicht uneingeschränkt in der Wahl der Drehung des Lautsprechersystems oder der zu vermessenden Person bzw. des Kunstkopfmikrofons sind. Dabei können Systeme in allen drei Freiheitsgraden der Kopforientierung umgesetzt werden wie z.B. beim *FABIAN* genannten System von Lindau (2014); häufig kommt es jedoch zur Vereinfachung hin zu nur einem Freiheitsgrad der Kopforientierung um die z-Achse (Vertikalachse) wie in Systemen zur Messung von BRIR-Datensätzen zur Simulation von virtuellen Lautsprechern von C. Pike und Romanov (2017a), Satongar et al. (n. d. a), Erbes et al. (2015), Melchior et al. (2014) und Stade et al. (2012) oder auch bei der Erstellung von BRIR-Datensätzen virtueller akustischer Umgebungen (Băcilă & Lee, 2019).

Bei diesen richtungsspezifischen Messungen kann neben den bekannten Einflüssen des Kopfes und der Ohrmuscheln der Einfluss der Schulter und des Torso in verschiedener Hinsicht einbezogen werden. Bei der Vermessung von Menschen ist es konstruktionsbedingt durch die Drehung der Person mit Hilfe eines Drehsystems so, dass die Schulter und der Torso mitgedreht und folglich Einflüsse bei alleiniger Kopfdrehung nicht mit einbezogen werden; eine Bewegung nur des Kopfes wäre zwar möglich, ist allerdings in der gewünschten Genauigkeit schwer umzusetzen, während die Überprüfung nur der nach vorne gerichteten Kopforientierung leichter umsetzbar ist. Bei der Nutzung von Kunstkopfmikrofonen ist dies anders: Sie werden bei der Messung alleinig durch Motoren gesteuert und können folglich in der Genauigkeit, die diese Motoren bieten, positioniert werden. Statische sogenannte Kopf- und Rumpfsimulator (HATS) wie beispielsweise die kommerziell erhältlichen der Head Acoustics HMS Serie oder der Brüel & Kjaer 4100 sind jedoch so gebaut, dass der Kopf nicht um den Torso gedreht werden kann. Folglich ist es mit solchen Systemen ebenfalls nur möglich den Kopf abhängig vom Torso zu drehen. Da die Orientierung des Kopfes über dem Torso beim natürlichen Hören jedoch eine wichtigere Rolle einnimmt als die Veränderung der Orientierung des kompletten Torso, wurden solche Systeme bei der Entwicklung des BRS(Rathbone, 2000), von Lindau (2014) und auch in BRIR-Datensätzen zur Darstellung von virtuellen Lautsprechern genutzt (Erbes et al., 2015) (Satongar et al., n. d. a). Mit mehreren Motoren ist es des Weiteren möglich, sowohl den Einfluss bei Drehung um die komplette Körperachse, als auch nur die Drehung des Kopfes über einem festen Torso zu berücksichtigen.

Da sich bei Rathbone (2000) der Einfluss eines Schulter/Torso-Nachbaus bei Nutzung des BRS-Systems (Simulation von bis zu 5.0-Surround nach ITU-R Rec. BS 775–1(Horbach et al., 1999)) als nicht signifikant für die Lokalisationsleistung und wahrgenommene Klangfarbe herausgestellt hat und sich bei Lindau (2014) der Einfluss der Schultern und des Torsos vor allem bei vertikaler und lateraler Bewegung des Kopfes herausstellt, welche in diesem System zunächst nicht implementiert werden soll, wird auf die Nachbildung einer Schulter und eines Torsos zunächst verzichtet. Das in Absatz 5.4.2.2.1 beschriebene Drehsystem

nutzt das *Neumann KU 100* Kunstkopfmikrofon in der Form wie es von *Neumann* entwickelt wurde und dessen Nutzung auf einem Mikrofonstativ vorgesehen ist, ohne eine Schulter- und/oder Torsokonstruktion nach anthropometrischen Daten umzusetzen. Dies ist bei Erweiterung des Systems hin zu Kopforientierungen entlang der x-Achse (Längsachse) und y-Achse (Querachse), als auch Einbindung von Lautsprechern, welche eleviert sind, jedoch näher zu betrachten.

5.2.5 Rendering-Engine

Die Struktur der typischen Faltungsalgorithmen, welche in der dynamischen Binauralsynthese genutzt werden, setzt sich folgendermaßen zusammen: Aufgrund der Länge der BRIRs, welche je nach auralisiertem Raum bis zu mehrere Sekunden betragen kann, wird in der Regel auf eine schnelle Faltung im Frequenzbereich gesetzt. Dabei wird eine ungleichmäßig partionierte FFT-basierte Faltung (Wefers, 2014) so implementiert, dass die Terminierung der Filterwechsel durch das Head-Tracking im Zusammenwirken mit den gewählten Partionierungsgrößen noch eine ausreichende Minimale Gesamtsystemlatenz (mTSL) (siehe weiterer Verlauf dieses Kapitels) erreicht; eine Abschätzung zur Audioblockgröße bzw. zum genutzten Signalvektor der Entwicklungsumgebung muss jedoch auch bei einer Implementierung mit einer direkten linearen Faltung im Zeitbereich erfolgen, die jedoch prinzipiell zunächst keine Latenz verursacht(Wefers & Vorländer, 2011). In Abbildung 5.6 ist eine FIR-Filterstruktur eines kurzen unpartionierten 9-Tap-FIR-Filters in direkter Form dargestellt wie sie bei der direkten Faltung im Zeitbereich umgesetzt wird. Abbildung 5.7 zeigt beispielhaft eine ungleichmäßige Partionierung eines FIR-Filters, wie sie bei der ungleichmäßig partionierten FFT-basierten Faltung genutzt wird.

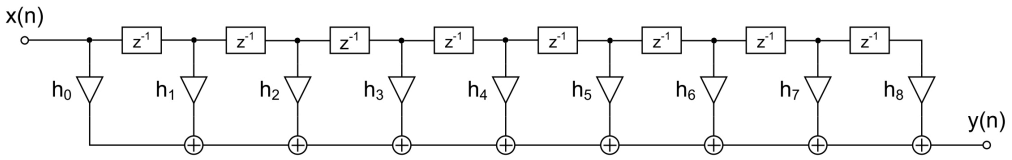

Abbildung 5.6: Unpartitionierter 9-Tap-FIR-Filter in direkter Form (Wefers, 2014)

Eine solch zeitlich variierende Faltungs- bzw. Filterstruktur kann wie in **??** dargestellt mit zwei parallelen Filtern umgesetzt werden. So ist es möglich zwischen zwei Filtern bei entsprechender Kopforientierung und dem daraus resultierenden Event in der Rendering-Engine umzuschalten bzw. sie ineinander überzublenden, was durch einen entsprechend implementierten Crossfade umgesetzt wird. Die FIR-Filterkerne werden bei entsprechender Terminierung in die Faltungs-Engines geladen und das *Switching* zwischen den Filtern kann aufgrund einer entsprechend klein gewählten minimalen Partionierungsgröße zu Beginn der ungleichmäßig partionierten FFT-basierten Faltung bzw. einer klein gewählten Audioblockgröße bei der direkten linearen Faltung im Zeitbereich latenzfrei innerhalb eines Blocks erfolgen(Wefers, 2014). Neben dieser Struktur zweier paralleler FIR-Filter, ist es auch möglich FIR-Filterkoeffizienten einzeln anzupassen bzw. zu verändern und auf diese Weise eine inkrementelle Filterumschaltung zu erreichen (Brandtsegg et al., 2018); Letzteres soll in dieser Arbeit jedoch nicht

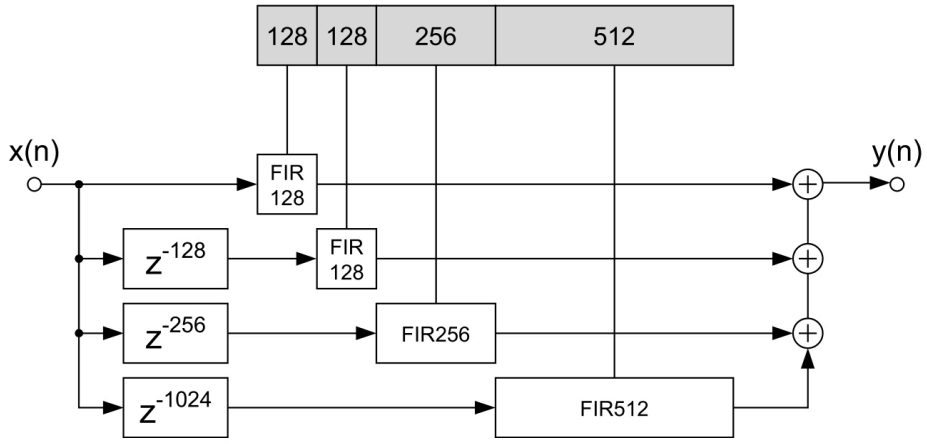

Abbildung 5.7: Beispiel einer FIR-Filterung mit ungleichmäßig partitionierter Impulsantwort (Wefers, 2014)

betrachtet werden. Die Implementierung des Crossfades zwischen den beiden Filtern kann entweder im Zeitbereich oder im Frequenzbereich erfolgen und die Crossfadelänge ist im binauralen Rendering in der Regel zwischen 8 und 32 Samples gewählt, kann jedoch länger sein und bei bei der Nutzung von BRIRs bis zu einer Blockgröße betragen(Wefers, 2014). Sollte es bei der Verarbeitung von HRTFs zu einer minimalphasigen Approximation und dadurch bedingten Trennung der linearphasigen Anteile der Übertragungsfunktion in ein echtes Delay kommen, so kann dieses interaurale Delay in einer fraktionierten Delayline für jeden Filterstrang umgesetzt werden . Die Nutzung einer fraktionierten Delayline ist hierbei zu nutzen, um die nötige Feinstruktur der Berechnung zu erreichen. Aufgrund der Nennung und Beschreibung von BRIRs im System dieser Arbeit ist diese Delayline nicht in Abbildung 5.8 eingezeichnet.

Die intensive Auseinandersetzung mit Implementierungen und der daraus resultierenden Latenzen von FFT-basierten ungleichmäßig partionierten Faltungalgorithmen ist nicht Bestandteil dieser Arbeit. Jedoch sollen Benchmark-Daten aus vergleichbaren Systemen genannt werden, um eine Vergleichbarkeit des konzipierten Systems zu erreichen. Betrachtet man hierbei die in Horbach et al. (1999) genannten Werte, so wird eine beispielhafte Partionierungsgröße von 128 Samples und eine resultierende Latenz von 5,3 ms genannt. Lindau (2014) nutzt innerhalb seiner vielfältigen Untersuchungen minimale Partionierungsgrößen von 128 und 256 Samples, je nach Experiment. Diese führen bei der genutzten Abtastrate von 44100 Hz zu Latenzen von 2,9 ms bzw. 5,8 ms. Dabei argumentiert er des Weiteren, dass diese minimale Partitionierungsgröße so gewählt wird, dass sie der Audioblockgröße des hinter der Implementierung liegenden JACK Audio-Servers entspricht und somit die Latenz des dynamisch angepassten Signals aufgrund der Overlap-Add implementierten Faltung genau einer Audioblockgröße entspricht. Es ist auffällig, dass Horbach et al. (1999) die Latenz des Systems auf Grundlage der zu Beginn der Verarbeitung stehenden FFT und deren FFT-Länge beschreibt, welche im Falle des BRS der in der Literatur standardmäßig genutzten und als sinnvoll erachteten Länge der doppelten Blocklänge bzw. doppelten Länge der Partionierung entspricht(Wefers & Vorländer, 2011). Er bezieht folglich die vollständige Eingangslatenz in Betracht. Lindau (2014) dagegen argumentiert lediglich mit der gewählten minimalen Parti-

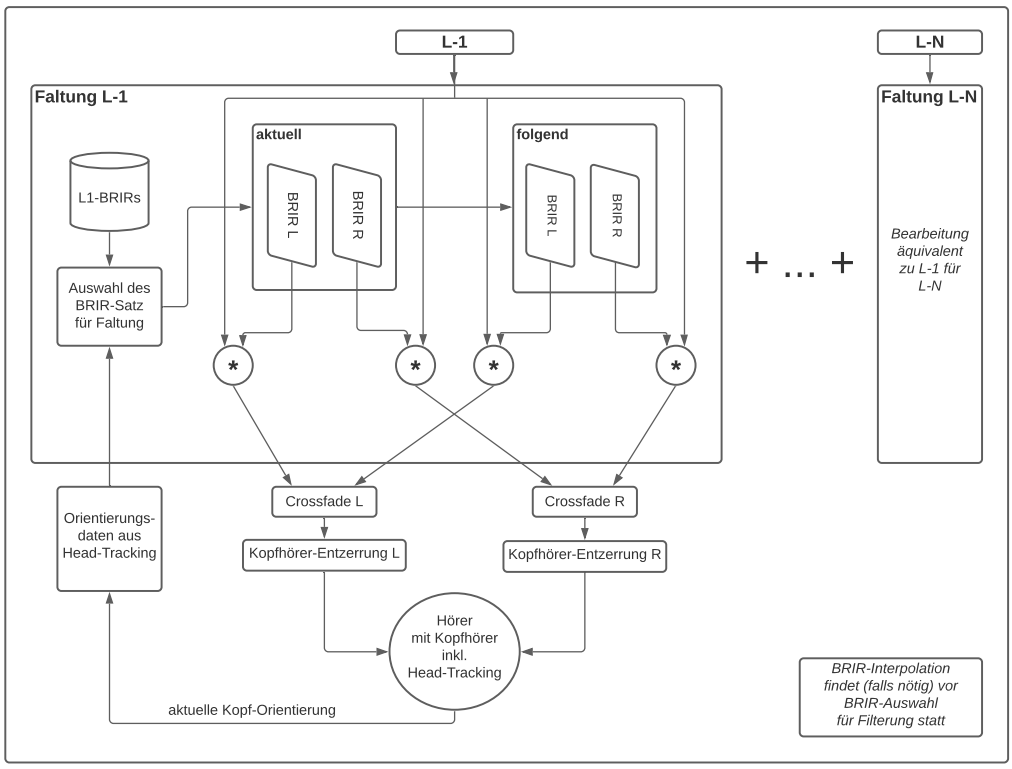

Abbildung 5.8: Struktur der dynamischen Binauralsynthese (eigene Darstellung)

tionierungsgröße, die am Anfang der ungleichmäßigen Partionierung steht. Diese ist für das dynamische Update der Filter ausschlaggebend und kann folglich als eine Art *Filter-Update-Latenz* betrachtet werden. Trotz der Tatsache, dass sich an dieser Stelle die Argumentationsketten etwas unterscheiden, lässt sich festhalten, dass in diesen System minimale Partionierungsgrößen bzw. Audioblockgrößen von 128 resp. 256 Samples gewählt werden, welche die dynamische Anpassung der Filter charakterisieren. Dabei sind Latenzen von bis zu 5,9 ms zu verzeichnen, welche allein durch die dynamische Anpassung der digitalen Signalverarbeitung des Faltungsalogrithmus hervorgerufen werden. Wie Lindau (2014) allerdings darstellt, ist diese minimal erreichte Latenz von 2,9 ms bei Gegenüberstellung zu den konstanten Latenzen, die ebenfalls zur dynamischen Anpassung beitragen wie die Update-Rate des Head-Trackings und die serielle Übertragung der Head-Tracker-Daten, jedoch ausreichend gering und eine weitere Reduzierung wäre weitgehend wirkungslos. Diese Latenzbetrachtungen werden im Verlauf dieses Kapitels erneut aufgegriffen.

Wie Wefers (2014) schreibt, wird bei einem binauralen Rendering von BRIRs die Crossfadelänge bis zu einer Länge der dynamischen angepassten Blockgröße gewählt. Diese Aussage bestätigt sich bei der Betrachtung der gewählten Crossfadelängen in Lindau (2014), welche der minimalen Partionierungsgröße und damit kleinsten Blockgröße des Systems entsprechen und somit zumeist bei 5,8 ms liegen. Letztere werden in linearer Form implementiert, was aufgrund der genutzten Raster-Auflösung der BRIR-Datensätze von 1° sinnvoll erscheint. Die gewählte Crossfade-Länge passt sich in diesem System also an die Head-Tracker-Terminierung und an die kleinste Blockgröße an und garantiert so, dass das Filter-Update eine Blockgröße später zu Verfügung steht. An dieser Stelle muss beachtet werden, dass in der beschriebenen

Implementierung von Lindau (2014) die weiteren Partionierungsgrößen für den diffusen Anteil der BRIRs hin bis zu 8192 Samples reichen. Damit kann es zu zeitlichen Verschiebungen der Überblendung bei den größeren Blockgrößen kommen, welche den Crossfading-Prozess zueinander verschieben und verlängern (Muller-Tomfelde, 2001).

Es soll zum Ende dieses Kapitels nochmals darauf hingewiesen werden, dass sich die erwähnten Zeiten, welche zwischen Filter-Updates liegen, in diesem Kapitel lediglich auf die Audiosignalverarbeitung beziehen und den Einfluss der Übertragungskette zwischen Kopf bzw. Headtracking-System und der Renderingumgebung nicht mit einbeziehen.

5.2.6 Systemlatenz und Head-Tracking

Die simple Signallaufzeit durch einen digitalen binauralen Renderer (und damit dessen algorithmische Struktur) wird zumeist durch die Audioblockgröße der genutzten Systemarchitektur bestimmt; dieser Audioblock oder auch Signalvektor muss zunächst mit entsprechend vielen Samples befüllt werden, ehe eine blockweise Verarbeitung startet. Da die dynamische binaurale Verarbeitung - wie aus Unterabschnitt 5.2.5 bekannt - ebenfalls blockweise bzw. partitionsweise erfolgt, ist es sinnvoll, Letztere an die Audioblockgröße der Systemarchitektur anzupassen; so werden keine weiteren Verzögerungen durch den Zusatz der dynamischen Verarbeitung an sich in der Signallaufzeit verursacht. Diese reine Signallaufzeit wird entsprechend der gewählten Blockgrößen der Systemarchitektur des rein Software-basierten Systems bei Lindau (2014) mit zumeist 256 Samples gewählt, was bei der genutzten Abtastrate von 44100 Hz zu 5,8 ms Latenz führt (eine Blockgröße von 128 Samples und 2,9 ms Latenz wird bei kritischen Hörversuchen genutzt). Horbach et al. (1999) beschreibt eine Signallaufzeit des Hardware-basierten BRS-Systems von weniger als 6 ms, die vorrangig durch die gewählten FFT-Ein- bzw. IFFT-Ausgangspuffer bestimmt wird. Hierbei ist neben der Wahl der Blockgrößen unter Gesichtspunkten der daraus resultierenden Latenz, auch zu beachten, dass FFT-basierte Faltungen kleinerer Audioblocks zu einer geringeren Frequenzauflösung führen und damit die Genauigkeit der Filterberechnungen abnimmt (Horbach et al., 1999).

Im System von Lindau (2014) erfolgt die Terminierung des Head-Tracker-Events innerhalb der Zeitspanne der genutzten kleinsten Block- bzw. Partionierungsgrößen, sodass aufgrund des entsprechend gleichsinnigen Crossfade-Scheduling (siehe Unterabschnitt 5.2.5) die neuen Filter genau einen Audioblock später zur Verfügung stehen. Somit die eigentliche Signallaufzeit auch bei Nutzung des Head-Tracking gleichwertig.

Veränderungen der Kopforientierung werden zunächst auf den Head-Tracker übertragen, welcher mithilfe eines Lagesensors oder eines ganzen Sensorsystems (siehe Inertiale Messeinheit (IMU)) Lagedaten generiert, die über eine serielle Schnittstelle an das Rendering-System übergeben werden, welches diese seriellen Daten ausliest und an die Rendering-Engine weitergibt. Die eigentliche mathematische Beschreibung der Veränderungen der Lage des kopfbezogenen Koordinatensystems des Hörers wird dabei mit Hilfe von Quaternionen und/oder Euler/Kardan-Winkeln umgesetzt (siehe Abschnitt 3.4). Die Verarbeitungsgeschwindigkeit von Sensoren bzw. Sensor-Systemen wird über die Update-Rate bestimmt, in welcher

das Sensor-System neue Daten zur aktuellen Kopforientierung ausgibt; diese beträgt beim BRS-System 120 Hz (Horbach et al., 1999) und auch Lindau (2014) nutzt einen Head-Tracker mit 120 Hz Update-Rate. Dies führt zu neuen Lagedaten etwa alle 8,3 ms, was ebenso als Latenz des Head-Trackers bzw. dessen Sensors bezeichnet werden kann. Lindau (2014) spricht von typischen Head-Tracker-Update-Raten im Bereich von 120 bis 180 Hz.

Diese durch die Update-Rate des Sensors verursachte Latenz beschreibt jedoch nicht die vollständige Latenz, die dem Head-Tracker-System zuzuordnen ist. Die Sensordaten werden typischerweise durch einen Mikrocontroller ausgelesen, so kann auch diese Systemkomponente eine Latenz bei der Verarbeitung der Sensordaten verursachen. Des Weiteren kommt es zu möglicher Asynchronizität der Datenverarbeitung zwischen Sensor, Mikrocontroller und der darauf folgenden seriellen Übertragung, die wiederum auch eine Laufzeitlatenz verursacht. Wenzel (1997) gibt folgendes Beispiel: Bei einem Polhemus Fastrak Head-Tracking System kommt es im Taktzyklus des Systems von 8,3 ms (120 Hz Aktualisierungsrate) zur Verarbeitung von neuen Lagedaten, während die benötigte Zeit für die Übertragung des 17-Byte-Datenpakets 8,85 ms bei Verwendung einer seriellen RS-232-Schnittstelle mit 19,2 kBaud beträgt. Die Auswirkungen dieser 0,5 ms Diskrepanz akkumulieren sich, bis das Head-Tracking-System die Übertragung stoppt und Daten, die sich bereits in der seriellen Warteschlange befinden, verwirft und sich dann neu synchronisiert, indem es bis zu seinem nächsten internen Taktzyklus wartet, um neue Daten zu übertragen (Jacoby et al., 1996). Folglich gehen an dieser Stelle Datenframes verloren, was zu einer langsameren effektiven Aktualisierungsrate führt. Eine weitere Unsicherheit in der *Gesamt-Aktualisierungsrate* wird im Weiteren durch die Interaktion des Head-Trackers mit der dynamischen Anpassung des Renderings eingeführt. Dieses beispielhaft erwähnte Polhemus Fastrak Head-Tracking System wurde auch vom BRS-System (Mackensen, 2004), sowie von Lindau (2014) genutzt.

Um eine plausible oder gar authentische Wahrnehmung der dynamischen Binauralsynthese zu erreichen, ist die absolute Latenz der dynamischen digitalen Signalverarbeitung ausschlaggebend. Dynamisch bedeutet in diesem Fall, dass die Interaktion der Audiosignalverarbeitung mit dem Head-Tracking vollständig betrachtet werden muss. An dieser Stelle sind Wahrnehmungsschwellen der dynamischen Anpassung der Kopforientierung an das Rendering definiert worden, welche vom Rendering eingehalten werden müssen. Betrachtet man alle Anteile des Systems, welche an der dynamischen Verarbeitung beteiligt sind, können für jede einzelne Systemkomponente, die an der Signalgebung und -übertragung beteiligt ist Größen von Datenpaketen, Längen der gewählten Verarbeitungsblocks oder Update-Raten und somit zeitkritische Faktoren ausgemacht und diese auch analysiert werden. Wichtig für das Gesamtsystem ist jedoch das zeitliche Zusammenwirken der einzelnen Systemkomponenten und die daraus resultierende mTSL. Diese mTSL ist die vollständige Systemlatenz, die das System mit den gewählten Komponenten minimal erreichen kann und sich neben den bereits genannten Aspekten nach Wenzel (1997) und Miller et al. (2003) aus folgenden Komponenten zusammensetzt: die Aktualisierungs- bzw. Updaterate des Head-Trackers (bzw. genauer des Head-Tracker-Sensors), die Latenz der vollständigen seriellen Schnittstelle, die Latenz möglicher Head-Tracker-Bibliotheken, die - falls das System in ein Netzwerk integriert ist -

Netzwerklatenz (welche im System dieser Arbeit nicht von Relevanz ist), die Terminierung der Filterwechsel in der Faltungs-Engine, die in der Faltungs-Engine verwendete Audioblockgröße (bzw. minimale Partionierungsgröße), die Verzögerung, die durch die Laufzeit des Schall vom Lautsprecher zum Mikrofon in den BRIR-Datensätzen vorhanden ist (falls Letztere nicht wie im System dieser Arbeit kompensiert wurde) und die Latenz, welche durch die Kopfhörer-Entzerrungsfilter verursacht wird sowie die Latenz des Audio-Ausgabegerätes, welche durch dessen Treiber und die genutzte Hardware verursacht wird. Diese mTSL kann durch die analytische Betrachtung aller Systemkomponenten zwar abgeschätzt werden, jedoch kann nur eine Messung über alle Systemkomponeten hinweg - wie sie beispielsweise Lindau (2014) vorschlägt - die tatsächliche Latenz des Systems beim Zusammenwirken aller Komponenten aufdecken. Dabei ist jedoch unbedingt eine Mittelung aus mehreren Messungen durchzuführen, da es durch das Zusammenspiel der Aktualisierungsraten bzw. Verarbeitungslatenzen verschiedener Komponenten zu einer Verteilung der Messwerte kommt.

Neben der mTSL, welche vom System erreicht werden kann, wurde in diversen Untersuchungen der Schwellwert der gerade erkennbaren Systemlatenz TSL untersucht. Während das BRS-System eine mTSL von 50 ms erreicht (Rathbone, 2000), hat sich in Untersuchungen derselben Forschungsgruppe eine maximal mögliche TSL von 85 ms herausgestellt (Horbach et al., 1999). Somit liegt die Wahrnehmungsschwelle deutlich höher als die systembedingt mögliche Performance. Dies stellt sich auch in vielen anderen Untersuchungen und Messungen dieser Art heraus wie beispielsweise die von Lindau (2014) angeführten Quellen im Rahmen der binauralen Darstellung von VAEs, welche mTSL von 12, 50 und 9,9 ms erreichen und dabei gerade erkennbare TSL von 60, 75 und 70 ms in Hörtests bestimmen. Die benötigte TSL scheint mithilfe gängiger Systemkomponenten unter vertretbarer Rechenlast also einfach zu erfüllen.

Lindau (2014) hat in eigenen Untersuchungen ebenfalls Grenzen für die gerade erkennbaren und somit maximalen TSL gefunden, wobei er neben der Nutzung eines Renderings mit BRIRs, die in einem großen Vorlesungssaal ($V = 8600m^3$, $RT60 = 2,1s$) aufgenommen wurden, auch welche in den Test einbezogen wurden, die unter reflexionsarmen Bedingungen gemessen wurden: Das 95 % Konfidenzintervall der Ergebnisse liegt bei 101,3 bis 114 ms mit einer jedoch deutlichen Standardabweichung von 30,39 ms; drei mal wurde eine TSL von $\leq$ 64 ms festgestellt, was jedoch nur 3,4 % aller Messwerte entspricht. Die mTSL wurde dabei zuvor als Mittelwert aus 60 gemessenen Werten ermittelt und liegt bei 43 ms.

5.2.7 Länge der BRIRs: Raumakustik, SNR und Mixing Time

BRIRs tragen wie Raumimpulsantworten die spezifische akustische Information eines Raumes bei einer definierten Anregung und Messung an festgelegten Positionen als LTI-System in sich; hinzu kommt die richtungsspezifische Information der kopfbezogenen Übertragung zwischen Lautsprecher und Kunstkopmikrofon, welche sich durch die Position und Orientierung der beiden zueinander definiert. Mögliche zeitvariante Komponenten eines realen Raums wie die Schallgeschwindigkeit in der Luft, die sich bei tagesabhängiger Temperaturänderung im Raum

ebenfalls ändert (Weinzierl, 2008) werden ohne Einschränkungen im Rahmen dieser Arbeit nicht näher betrachtet. Die raumakustische Information bestimmt sich aus dem Reflektionspattern des Raumes, welches sich aus der zeitlichen Abfolge des Direktschalls und aller Reflexionen zusammensetzt. Letztere ist in ihrer Zeitstruktur abhängig von der Geometrie und dem Absorptions- und Reflexiongrad aller Begrenzungsflächen des Raumes. Die Kombination aus frühen und späten Reflexionen mit deren fortschreitender Absorption und Streuung an allen Oberflächen des Raumes ergibt eine zunehmend komplexe gemischtphasige Überlagerung, die zu einem diffusen Schallfeld ohne Richtungsinformation führt.

Es ist möglich, das Reflektionspattern eines Raumes in einem sogenannten Reflektogramm darzustellen; dies ist eine schematische Darstellung der zeitlichen Abfolge von Direktschall und allen Reflexionen des Raumes, von der ersten initialen Reflexion über weitere erste Reflexionen hin zu späten Reflexionen, die aufgrund von Überlagerungen zunehmend ihre Richtungsinformation verlieren und den diffusen Nachhall des Raumes prägen. Der Direktschall, die Anfangszeitlücke (ITDG) (Zeit zwischen dem Eintreffen des Direktschalls und dem Eintreffen der ersten starken initialen Reflexion) und die frühen Reflexionen sind für das Richtungs- und Entfernungshören einer Schallquelle im Raum essentiell, während die späten diffusen Reflexionen keine Richtungsinformationen mehr tragen. Betrachtet man eine Raumimpulsantwort, so erkennt man, dass diese prinzipielle Ähnlichkeit mit einem schematischen Reflektogramm hat und auch in ihr ein früher und später Teil charakterisiert werden kann.

Einer der gebräuchlichsten Parameter, um das raumakustische Verhalten bei Anregung zu beschreiben, ist die Nachhallzeit. Gewöhnlich mit *RT60* bezeichnet, gibt dieser Parameter die Zeit an, die die Schallenergie benötigt, um im Raum um 60 dB abzufallen. Dieser Parameter kann aus einer Raumimpulsantwort mithilfe der Frühe Abklingkurve (EDC) bestimmt werden, die mit der Schroeder-Integrationsmethode (Schroeder, 1965) ermittelt wird. Die EDC ist definiert als das Schwanzintegral der quadrierten Impulsantwort h zum Zeitpunkt t:

$$EDC(t) \triangleq \int_t^\infty h^2(\tau)\, d\tau \tag{5.4}$$

Dies stellt die gesamte Signalenergie dar, welche zum Zeitpunkt t in der Impulsantwort verbleibt. Somit kann man die *RT60* direkt aus der logarithmierten (dekadischer Logarithmus) EDC (*engl.* Early Deacy Curve und damit *EDC*) extrahieren, als den Zeitpunkt, bei dem der Abklingvorgang -60 dB erreicht. In der Praxis führt dieses Vorgehen jedoch oft zu falschen Ergebnissen, da es eine Aufnahme mit einem ausreichenden Signal-Rausch-Abstand (SNR) voraussetzt, der vor allem im tieffrequenten Bereich aufgrund geringer abgestrahlter Schallleistung der verwendeten anregenden Lautsprecher oft schwer zu erreichen ist. Um diese Problematik zu beheben, gibt es Metriken wie die *T30*, die *T20* (als auch die *EDT*), welche über einen kleineren Abklingbereich berechnet werden, was mit einer höheren Wahrscheinlichkeit zu richtigen Ergebnissen der Nachhallzeit *RT60* führt. Mithilfe einer Regressionsgeraden wird die Nachhallzeit *RT60* entlang der *EDC* aus diesen Metriken extrapoliert (Hak et al., 2012). Die Norm für raumakustische Messungen ISO 3382 legt einen Mindestabklingbereich und damit SNR von 45 dB für *T30* (mit einem Regressionsbereich von -5 dB bis -35 dB) und einen Mindestabklingbereich/SNR von 35 dB für *T20* (mit einem Regressionsbereich von

-5 dB bis -25 dB) fest (Weinzierl, 2008). Dabei reduziert ein Regressionsbereich, der bei -5 dB beginnt, den störenden Einfluss von frühen Signalschwankungen, die durch Reflexionen verursacht werden, und liefert so eine bessere Schätzung für die *RT60* (Rossing, 2007). Die Anfangsnachhallzeit (EDT) (*engl.* Early Deacy Time und damit *EDT*) wird analog aus der EDC berechnet, wobei eine Regressionslinie von 0 dB bis -10 dB verwendet wird. Die EDT (welche keinen Estimator für die *RT60* darstellt) ist eine psychoakustisch relevante Metrik, die für das raumakustische Design nützlich ist. Da die Wahrnehmung von Nachhall nicht linear ist, sind starke Abnahmen des frühen Energieabfalls oft die Ursache dafür, dass der Raum als *akustisch trockener* empfunden wird als er tatsächlich ist (Howard & Angus, 2017). Somit stimmt die EDT mit der subjektiv empfundenen Nachhalldauer - besonders bei kleinen Lautstärken und Programmmaterial (Weinzierl, 2008) - meist besser überein. Sie ist jedoch deutlich stärker von der Position im Raum abhängig, der Einfluss des Verhaltens der frühen Reflexionen ist in der *EDT* höher.

Aus einer entprechenden Bestimmung der Nachhallzeit oder vergleichbarer raumakustischer Kriterien kann eine Festlegung der benötigten Gesamtlänge der BRIRs für die Faltung getroffen werden. Bei kritischen Hörversuchen ist es jedoch ratsam, das komplette Abklingverhalten bis zum Erreichen des Rauschpegels, welcher neben allgemeinem Hintergrundrauschen zumeist durch die akustische Messkette bestimmt wird, zu nutzen. Mit Hilfe exponentieller Sinus-Sweeps können mit vertretbarem Aufwand Signal-Rausch-Abstände (SNRs) bei BRIR-Messungen von bis zu 80 bis 90 dB erreicht werden (Erbes et al., 2015)(C. Pike & Romanov, 2017a)(Melchior et al., 2014); mit Hilfe von 576 Messiterationen wurde sogar ein Spitzen-Rausch-Abstand (PNR) einer einzelnen BRIR von 101,9 dB erreicht (Hahne et al., 2019). Es ist eine Abwägung zu treffen, da BRIRs, die einen hohen SNR erreichen und somit die Feinstruktur des Abklingverhalten des Raumes prinzipiell übertragen können, länger sind und folglich im Faltungsprozess die Anforderungen erhöhen. Hahne et al. (2019) hat den Schwellenwert für die Erkennung von Rauschen in einer BRIR eines akustisch behandelten Raumes (5,75 m x 5 m × 3 m, RT60 = 0,3 s) untersucht: Bei den Ergebnissen von 15 Probanden lag die mittlere Schwelle bei 59,2 dB mit einer Standardabweichung von ± 3,4 dB. Zu beachten ist dabei jedoch, dass diese Ergebnisse bei einem Abhörpegel von 58,4 dB(A) ermittelt wurden und es eine mögliche Abhängigkeit dieser Schwelle vom Abhörpegel gibt. Bezieht man diese Ergebnisse jedoch auf die Bestimmung einer ausreichenden Länge der BRIRs, so scheint die Nachhallzeit *RT60* und somit ein SNR von 60 dB als ausreichend.

Aufgrund des Ziels dieser Arbeit einer binauralen Lautsprecher-Simulation in Tonregieräumen und Mischkinos ist es essenziell, die raumakustischen Eigenschaften dieser Räume zu betrachten; Normen für diese Räume existieren, sodass die Betrachtungen hier prinzipiell übertragbar erscheinen. Letztere beschreiben jedoch lediglich maximal zulässige Nachhallzeiten und liefert keine konkreten Informationen zum Reflexionspattern, was folglich keine Rückschlüsse auf die genaue Abfolge und Verteilung der Reflexionen mit zunehmender Diffusität zulässt. In den genannten Räumen werden frühe Reflexionen durch entsprechend sinnvoll absorbierende und reflektierende Begrenzungsflächen und die daraus resultierende gewählte Raumgeometrie im Sinne eines unverfälschten Höreindrucks des Direktschalls am Hörplatz zwar vermieden,

jedoch sind sie nicht vermeidbar. Typische Werte für die Dauer des Eintreffens von frühen Reflexionen werden zumeist ohne weitere Angaben zur Raumbeschaffenheit gegeben: Weinzierl (2008) spricht innerhalb der gleichen Quelle von bis zu 15 ms, als auch 50 ms oder 80 ms. Die beiden letztgenannten Werte sind am ehesten durch das Diskriminierungsvermögen unseres Gehörs motiviert; ähnliche Werte findet man auch in akustischen Kriterien wie beispielsweise dem des Deutlichkeitsmaßes C_{50} und des Klarheitsmaßes C_{80} (Weinzierl, 2008). Es lässt sich jedoch festhalten, dass die Dauer dieses Diffusionsvorgangs mit der Raumgröße zunimmt, da sich bedingt durch größere freie Weglängen die Zeitabstände zwischen den Einzelreflexionen vergrößern. Dieser Effekt ist noch ausgeprägter, wenn im Raum keine den Schall diffus streuenden Begrenzungsflächen oder Gegenstände vorhanden sind. In Räumen mit ungleichmäßiger Verteilung großer Flächen mit stark variierenden Absorptionskoeffizienten (z. B. wenn große Fensterscheiben mit stark absorbierenden Zuschauersitzen kombiniert werden) kann der Prozess der zunehmenden diffusen Überlagerung gestört werden. Absorbierende Räume können des Weiteren nie perfekt diffus sein, da immer ein Netto-Energiefluß in Richtung der Schallenergieverluste (d.h. zu den absorbierenden Wänden hin) bleibt. Auch Raum-in-Raum-Konstruktionen, stark gedämpfte Räume und sehr kleine Räume können in ihrem Abklingverhalten eine fehlende echte Diffussität aufweisen. Des Weiteren ist modales Verhalten im tieffrequenten Bereich zu beachten, wie auch die Nähe zu Raumbegrenzungen (Seitenwände, Böden) deutliche Unterschiede hervorruft. In diesem Fall können Reflexionen deutliche Kammfilter bilden, deren Spektren von der genauen Empfangsposition abhängen und damit die Annahme der Ortsunabhängigkeit des diffusen Schallfeldes verletzen. (Lindau, 2014) -

Der Begriff der *Mixing Time* bezeichnet den Zeitpunkt in Raumimpulsantworten, an dem die diffuse Hallfahne beginnt. Ein diffuses Schallfeld kann dabei physikalisch definiert werden durch die Gleichverteilung der akustischen Energie und einen gleichmäßigen akustischen Energiefluß über den gesamten Raumwinkel. Diese entspricht also einer *physikalischen Mixing Time* aufgrund der Beschaffenheiten des Raumes und hat zunächst nichts mit der vom Gehör *wahrgenommenen Mixing Time* zu tun, welche als der Moment angesehen werden kann, in dem der diffuse Nachhall nicht mehr von dem einer anderen Position oder Hörerorientierung im Raum unterschieden werden kann. Aufgrund von auditiven und kognitiven Diskriminierungseffekten, welche auch von den Eigenschaften des Audioinhalts abhängig sind, ist zu erwarten, dass die *wahrgenommenen Mixing Time* gleich oder kleiner als die *physikalische Mixing Time* ist. (Lindau, 2014)

Auf dieser Grundlage wurden in diverser wissenschaftlicher Literatur Untersuchungen zur Bestimmung der *wahrgenommenen Mixing Time* mit dem Wunsch der Trennung der Signalverarbeitung zwischen einem dynamisch zu verarbeitenden Teil des Direktschalls und der ersten Reflexionen und einem statisch zu verarbeitenden Teil des diffusen Schalls, der sich aus späten Reflexionen ohne Richtungsinformation zusammensetzt, durchgeführt. Bereits bei der Entwicklung des BRS wurden zur Verbesserung der Performance des Systems Untersuchungen zur *wahrgenommene Mixing Time* angestellt und als sinnvolle Obergrenze für das System Mischkinos wie das der *Bavaria Filmstudios* in München (Dimensionen: $11x15x6m$, Volumen: $990m^3$, RT60 = $300ms(bei500Hz)$) gewählt; hierbei wurde eine *wahrgenommene*

Mixing Time von 30 ms in psychoakustischen Tests für diesen Raum gefunden (Mackensen, Theile et al., 1999)(Fruhmann et al., 2002). Die Verbesserung der Performance des binauralen Renderings bei einer Verkürzung des dynamisch zu verarbeitenden Teils ist nicht komplex: Die Länge des FIR-Filterkerns (bzw. die Anzahl der FIR-Filterkoeffizienten), welcher bei Änderung der Kopforientierung neu gesetzt und mit dem aperiodischen Schallsignal entsprechend dieser Länge gefaltet werden muss, verringert sich deutlich. Auf diese Weise kommt es zu (je nach Implementierung) deutlich weniger Multiplikationen, die durch die Veränderung der Kopforientierung und der damit verbundenen dynamischen Anpassung und Überlagerung der Filterstränge begründet sind. Eine statische Faltung bleibt ungeachtet der Kopforientierung dauerhaft gleich und somit auch gleichmäßig *signalverarbeitungstechnisch teuer*. So ist es ein anerkannte Topologie von Systemen, die künstliche Umgebungen synthetisieren, dass eine *ideale* Simulation des Direktschalls, eine mehr oder weniger ideale Simulation der frühen bzw. ersten Reflexion und eine ungefähre Simulation des diffusen Nachhalls, die sich entweder durch Nachhallalgorithmen mit parametrischer Eingabe in Bezug auf den Raum oder durch eine Faltung eines späten Teils einer einzelnen BRIR zusammensetzt (Meesawat & Hammershøi, 2003).

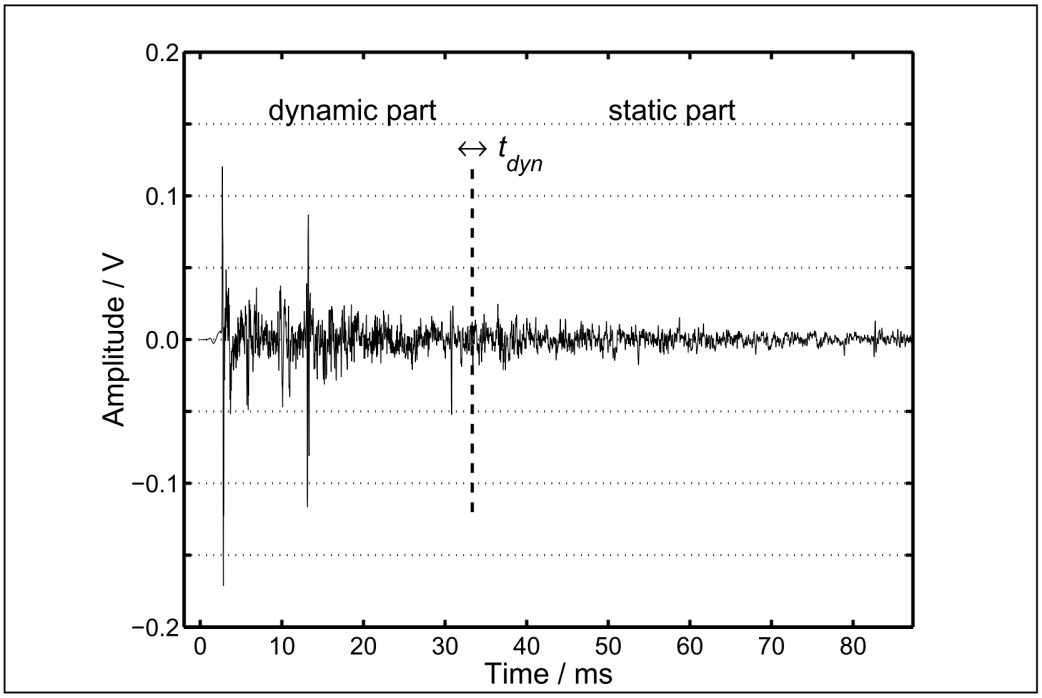

Abbildung 5.9: Impulsantwort mit ausgeprägten Erstreflexionen: Wahl der dynamisch/statischen Trennung anhand der Mixing Time t_{dyn}((Fruhmann et al., 2002))

Lindau (2014) hat im Rahmen seiner Dissertation zur Plausibilität der binauralen Resynthese von akustischen Umgebungen mehrere signal- und modellbasierte Prädiktoren zur Bestimmung der *physikalischen Mixing Time* perzeptiv untersucht. Seine Ergebnisse zeigen, dass diese Prädiktoren für unterschiedliche charakteristische Räume teils deutlich unterschiedliche Ergebnisse liefern, die wiederum deutlich von den bestimmten Werten der *wahrgenommenen Mixing Time* aus den psychoakustischen Hörversuchen abweichen. Da im Rahmen dieser Arbeit nicht die Beurteilung dieser Prädiktoren ausgiebig diskutiert, wohl aber eine Abschätzung der benötigten *gerade bzw. gerade nicht mehr wahrgenommenen Mixing Time* für

Tonregieräume und Mischkinos gegeben werden soll, sind die psychoakustisch bestimmten Ergebnisse seine Untersuchung für Räume dieser Art interessant: Dabei wurde ein rechteckiger Studioraum für elektronische Musik untersucht, welcher mit Dimensionen von $10mx5m$, einem Volumen von $216m^3$ und einem gemittelten Absorptionsgrad seiner Begrenzugsflächen von $\alpha = 0,36$ eine Nachhallzeit von $RT60 = 0,39s$ erreicht. Innerhalb der Untersuchung wurden nur rechteckige Räume ausgewählt, da diese aufgrund ihrer Form mit parallelen Wänden und den daraus resultierenden langen ungehinderten Weglängen vermuten lassen, dass sie in Bezug auf eine spät zu findende *Mixing Time* am kritischten sind; Hidaka et al. (2005) kommt im Rahmen von Untersuchungen der *physikalischen Mixing Time* in großen Konzertsälen zu Ergebnissen, die diese Vermutung unterstützen. Das 95-prozentigen Konfidenzintervall der Ergbenisse der *gerade noch wahrgenommenen Mixing Time* dieses Raums reicht von 25 bis 50 ms. Bei weiteren getesteten Räumen ähnlicher Größe, jedoch mit kleiner werdendem gemittelten Absorptionsgrad verläuft dieses Intervall zwischen 25 und 40 ms ($\alpha = 0,26$) sowie zwischen 15 und 35 ms ($\alpha = 0,17$). Mit zunehmender Raumgröße und damit verbundener Verlängerung der Nachhallzeit *RT60* werden die *gerade noch wahrgenommenen Mixing Times* größer, wobei eine Abnahme bei kleiner werdenden gemittelten Absorptionsgraden bei ähnlicher Raumgröße auch hier zu beobachten ist.

Neben der Darstellung seiner Ergebnisse zur *gerade noch wahrgenommenen Mixing Time* berichtet Lindau (2014) auch davon, dass unabhängig von der Trennung der dynamischen hin zu statischer Verarbeitung der BRIRs, Effekte eines anders gearteten Rauschens (vor allem in Form eines anders gearteten Hintergrundrauschens) zwischen den genutzten BRIRs auftreten können, welche die Wahrnehmung stören. Des Weiteren ist die Messposition des Raumes kritisch, d.h. die BRIRs können zwar zwischen verschiedenen Kopforientierungen bei gleicher Anregungsposition getauscht werden, jedoch sind Wechsel der BRIRs hinsichtlich der dynamischen und statischen Verarbeitung bei unterschiedlich gewählten Messpositionen hörbar trotz Einhaltung der *gerade noch wahrgenommenen Mixing Time*. Dies ist für die im Rahmen dieser Arbeit durchgeführten Darstellung von virtuellen Lautsprechern interessant, bei der die Anregung des Raumes für jeden Lautsprecher an einer anderen Position erfolgt. Tieffrequente modale Eigenschaften oder kammfilterartige Veränderungen des Schallfelds aufgrund der Nähe zu Begrenzungsflächen des Raumes erhöhen die *gerade noch wahrgenommene Mixing Time*, sodass ein Übergang zwischen der dynamischen Verarbeitung hin zu einer gewählten (zumeist bei frontaler Kopfausrichtung) statischen Verarbeitung immer mit der gleichen Hörer-/Schallquelle-Positionierung erfolgen sollte, um diese Schallfeldeigenschaften bei der Trennung der Verarbeitung nicht hervorzuheben.

Die Verbindung zwischen der dynamischen und der statischen Verarbeitung erfolgt bei Lindau (2014) in zwei getrennten Experimenten sowohl mit Hilfe einer linearen Überblendung innerhalb der kleinsten Blockgröße der dynamischen Verarbeitung, welche für die kleinen von ihm untersuchten Räume einer Dauer von 5,8 ms entspricht und auf 11,6 ms für die mittleren und großen Räume anwächst, also auch in Form eines energieerhaltenden kosinusförmigen Crossfades der Länge 20 ms. Die Blockgröße von 5,8 ms entspricht auch der Schrittweite, mit der die *Mixing Time* in diesem Hörversuch verändert werden konnte; dies erleichtert die schrittweise Anpassung der Faltungsoperationen. Im Hörversuch, der mit dem

20 ms langen energie-erhaltenden kosinusförmigen Crossfade durchgeführt wurde, lag diese Schrittweite bei 10 ms. Die Untersuchungen und Entwicklungen im Rahmen des BRS-Systems erfolgten ein wenig anders: Hier kann die Trennung zwischen der dynamischen und statischen Verarbeitung in Schrittweiten von 4 ms vorgenommen werden, während die Überblendung kosinusförmig innerhalb von 1,3 ms (resp. 64 samples) vorgenommen wird (Fruhmann et al., 2002).

5.3 Gesamtsystem und Systemmodule

Das Gesamtsystem dieser Arbeit besteht aus drei Systemmodulen, welche sich in folgende drei fundamentalen Komponenten aufteilen:

- Systemmodul 1: Messsystem zur Messung von BRIRs
- Systemmodul 2: Postprocessing der BRIRs
- Systemmodul 3: Flexible Auralisationsumgebung

Die Anforderungen an die einzelnen Systemmodule sowie der Signalfluss der Letzteren sollen im Weiteren mit Hilfe von Blockschaltbildern dargestellt werden, wobei die nötigen mathematischen Konzepte bereits im Verlauf dieser Arbeit vorgestellt wurden. Konkrete Beschreibungen zur Implementierung oder gar Darstellungen des Quellcodes werden nur im Einzelfall direkt im Fließtext getätigt, wenn diese Form der Beschreibung sich als sinnvoll erweist, um nötige Besonderheiten der Implementierung aufzuzeigen. Ansonsten ist der Quellcode im Anhang sowie auf dem beigefügten Datenträger dieser Arbeit zu finden.

All die im Folgenden dargestellten Systemmodule werden unter der Annahme der theoretisch frei wählbaren Anzahl an Lautsprechern N für den Fall $N = 1$ beschrieben und für den Fall $N = 2$ implementiert. Dies hat keine funktionalen Auswirkungen auf das System, da es trotz dieser Beschreibungen weiterhin skalierbar bleibt; lediglich ist für einen Ausbau des Systems eine entsprechende Performanceabschätzung bei bekannter verfügbarer Rechenkapazität zu tätigen.

5.4 Systemmodul 1: Messsystem zur Messung von BRIRs

5.4.1 Anforderungsanalyse

Wie im Verlauf dieser Arbeit bereits beschrieben, sind aktuelle wissenschaftlich motivierte BRIR-Datensätze für die datenbasierte Simulation von Lautsprechern in einer horizontalen Raster-Auflösung von 2° gemessen worden (C. Pike & Romanov, 2017a)(Satongar et al., n. d. a)(Erbes et al., 2015)(Melchior et al., 2014). Auf dieser Grundlage soll auch dieses System zunächst entwickelt werden, um Möglichkeiten der Vergleichbarkeit zu Forschungszwecken und zur Weiterentwicklung zu bieten, beispielsweise hin zu einem System, welches mit geringer aufgelösten Datensätzen auskommt und trotzdem ausreichend gute perzeptive Qualität liefert. An dieser Stelle sind vor allem aktuelle kommerzielle Systeme wie die Produkte von

Smyth Research[2] zu erwähnen, welche mit Hilfe weniger Messungen eine äußerst authentische binaurale Auralisation ermöglichen.

Aus dieser funktionalen Anforderung heraus ein Messsystem zu konzipieren, mit welchem eine horizontale Raster-Auflösung von 2° gemessen werden kann, wurde ein System zur automatisierten Drehung des Kunstkopfmikrofons in der horizontalen Ebene entwickelt. Dies wurde bei der Anforderungsanalyse berücksichtigt. Ein Drehteller wurde konzipiert, welcher aus der Messumgebung angesteuert werden kann; dieses System wird in Absatz 5.4.2.2.1 beschrieben. Des Weiteren sollen diverse Kontrollstrukturen möglich sein, um die Messung mit den gewünschten Parametern durchzuführen. Eine Trennung der Anforderungen an das *Systemmodul 1 (Messsystem)* in funktionale Anforderungen und Benutzeranforderungen wird nicht durchgeführt, da alle Anforderungen für die einwandfreie Funktion des Systems wichtig sind und folglich keine benutzerspezifischen Anforderungen betrachtet werden.

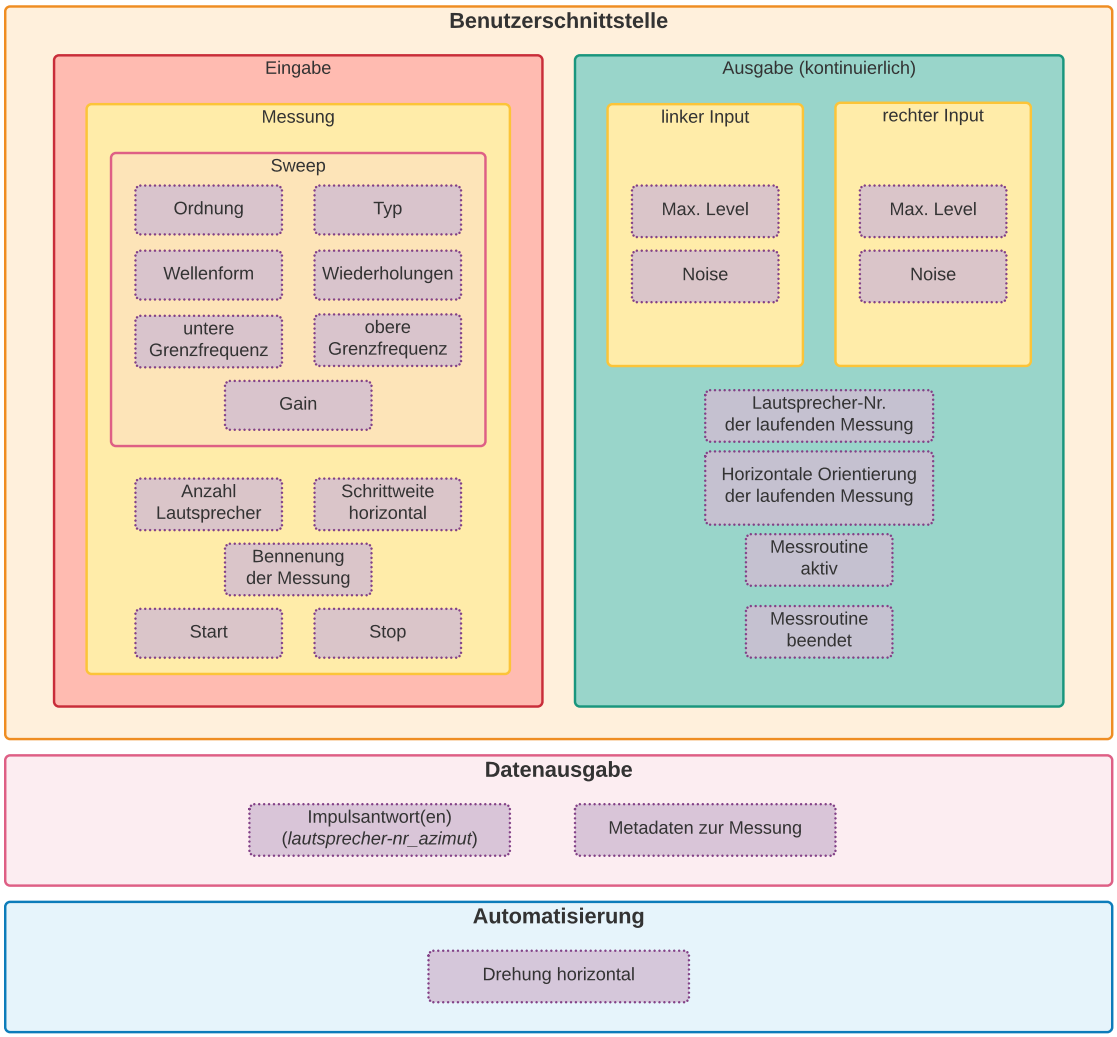

Abbildung 5.10: Anforderungen an das Messsystem zur Messung von binauralen Raumimpulsantworten (eigene Darstellung)

In Abbildung Abbildung 5.10 sind die Anforderungen an das System in den drei Kategorien der Benutzerschnittstelle, Datenausgabe sowie Automatisierung dargestellt. Die Benutzer-

[2] https://smyth-research.com/

schnittstelle soll sowohl eine Eingabe von Kenndaten zur Messung (bzw. einer Messreihe), die Sweep-basiert durchgeführt werden soll, und eine Möglichkeit des Startens und Stoppens der Letzteren bieten. Die Wahl einer Sweep-basierten Messung ist mit den Vorteilen des Verfahrens gegenüber anderen akustischen Messungen begründet, was in der Literatur ausführlich diskutiert worden ist (siehe beispielsweise Farina (2000)Müller et al. (2001)). Grundsätzlich ermöglichen sie ein hohes SNR, können Einflüsse harmonischer Verzerrungen trennen und sind weniger anfällig für Auswirkungen der Zeitvarianz; eine ausführliche Diskussion ist im Rahmen dieser Arbeit nicht vorgesehen. Des Weiteren soll die Anzahl der Lautsprecher und die gewünschte horizontale Raster-Auflösung der Messreihe (Schrittweite horizontal) eingegeben werden können. Eine Benennung der Messung bzw. Messreihe soll ermöglicht werden. Kontrollstrukturen zur Messung sollen ermöglicht werden, wobei die Letzteren kontinuierlich (d.h. für jede Einzelmessung) Werte ausgeben sollen, damit diese auch während einer Messreihe zur kontinuierlichen Kontrolle genutzt werden können. Hierbei soll eine Kontrolle des Pegels des linken und rechten Inputs, sowie eine Angabe des gemittelten Rauschpegels der BRIRs nach der Entfaltung möglich sein. Die Lautsprecher-Nummer sowie die horizontale Orientierung des Drehsystems der laufenden Messung sollen ausgegeben werden. Eine Ausgabe zur Aktivität bzw. Inaktivität der Messroutine ist des Weiteren zu ermöglichen. Die Datenausgabe der gemessenen BRIRs muss Lautsprecher- und Orientierungsinformationen (horizontale Orientierung unter Angabe des Azimutwinkels genügt) enthalten und einem festen Benennungsschema folgen, um die einwandfreie Weiterverarbeitung im nächsten Systemmodul bzw. das Interfacing in das Letztere zu garantieren. Des Weiteren sollen zur nachträglichen Kontrolle und einwandfreien Nachverfolgbarkeit der Parameter der Messung Metadaten ausgegeben werden, die wichtige Informationen zu den Messungen enthalten. Die Automatisierung richtet sich an die Nutzung eines Drehtellers, welcher sich entsprechend der Eingabe der Anzahl der Lautsprecher und der gewünschte horizontalen Raster-Auflösung bewegt und auf diese Weise die Messreihe in der gewünschten Genauigkeit ermöglicht.

5.4.2 Umsetzung

Diese Anforderungsanalyse und eine intensive Literaturrecherche führte zunächst zu einem wissenschaftlich frei verfügbaren Messumgebung in MATLAB mit dem Namen *ScanIR*, welche an der *New York University* im *Music and Audio Research Lab* entwickelt und betreut wird (Boren, 2011) (Vanasse et al., n. d.). Diese Messumgebung ist seit 2011 in stetiger Weiterentwicklung und bietet vielfältige Möglichkeiten der akustischen Messung und Datenanalyse mit ansprechender Benutzeroberfläche. Es ermöglicht die Erfassung akustischer Impulsantworten unter Verwendung einer Vielzahl von Messsignalen wie Sinus-Sweeps, Maximum-Length-Sequence (MLS) oder Golay-Code; dabei können einkanalige IRs, Stereo-HRIR/BRIRs sowie Mehrkanal-IRs gemessen werden. Nach der Messung und anschließender Entfaltung können direkt gängige akustische Kenndaten abgelesen, sowie graphische Darstellungen der IRs im Zeit- und Frequenzbereich sowie eine EDC genutzt werden. Des Weiteren soll auch die Nutzung eines Schrittmotors zur automatisierten Orientierungs-gebundenen Messung und eine anschließende Speicherung der Daten direkt als *SOFA*-File (zu diesem Dateinformat siehe Absatz 5.5.2.3.1) möglich sein.

Die Nutzung dieser Messumgebung stellte sich im Verlauf dieser Arbeit als nicht praktikabel heraus, da sie sich als stark fehleranfällig zeigte, was zu diversen Abstürzen und Abbrüchen von Messreihen führte. Auch die Speicherung der Messdaten als *SOFA*-File konnte nicht umgesetzt werden. Des Weiteren wurde bei der ausführlichen Testung festgestellt, dass die Ausgangskanäle auf eine Anzahl von $N = 2$ beschränkt sind, sowie die Nutzung eines Schrittmotors zur automatisierten Messung bei mehreren Ausgangskanälen nicht möglich ist. Diese Umstände führten letztlich dazu, eine eigene Messumgebung dieser Art, welche die Anforderungen aus der Anforderungsanalyse theoretisch erfüllen kann, in Max/MSP mithilfe der Spat-5-Library zu entwerfen.

5.4.2.1 Max/MSP und Spat-5 als Entwicklungsumgebung

Die Umsetzung bzw. Implementierung der im letzten Kapitel dargestellten und diskutierten Anforderungen erfolgt mit Hilfe der graphischen integrierten Entwicklungsumgebung *Max/MSP* von *Cycling '74* und der kostenfreien Library *Spat-5* des französischen *IRCAM* (franz. für *Institut de Recherche et Coordination Acoustique/Musique*); konkret Max/MSP in der Version 8.1.8 (64-bit für macOS) und Spat-5 in der Version 5.2.1. Die Motivation für die Implementierung in Max/MSP mit Spat-5 liegt in der Verfügbarkeit und Nutzung im Rahmen der Lehre und Forschung an der Hochschule der Medien Stuttgart. Da sowohl das Systemmodul 1 als auch das Systemmodul 3 dieser Arbeit mithilfe dieser Kombination von Entwicklungswerkzeugen umgesetzt wird, soll ihnen ein eigenes kleines Kapitel gewidmet werden.

Max/MSP ist eine modular aufgebaute, datenstrom- und objektbasierte Programmiersprache. Eine Anwendung bzw. ein sogenannter *Patch* ist aus *Objekten* aufgebaut, die bestimmte logische und/oder signalverarbeitende Aufgaben durchführen. Diese Objekte können triviale logische Verarbeitungsschritte durchführen oder aber mehrere Schritte ähnlich einer Funktion (oder Methode) verbinden, die entsprechende Parameter (welche als Argumente gesetzt werden) und auch Rückgabewerte hat. Objekte aus Libraries bzw. Packages von Drittanbietern werden auch *Max-Externals* genannt; diese sind zumeist in C oder C++ programmiert und bieten zum Teil eine komplexe Funktionalität, da sie eine Abfolge von Aufgaben eines spezifischen Prozesses zusammenführen. Man kann die Objekte bestimmter Max-Externals analog zur objektorientierten Programmierung auch als Instanzen einer Klasse bezeichnen und deren Attribute als Klassen- bzw. Instanzvariablen. Sie bieten eine Funktionalität auf Grundlage des im Objekt implementierten Algorithmus; dieser Algorithmus kann zwar durch die Nutzung von Attributen angepasst, jedoch nicht beliebig verändert werden, da der eigentliche Quellcode nicht zur Verfügung steht.

Die Library Spat (von franz. *Spatialisateur*) in der Version 5 bietet solche Max/MSP-Externals, welche im Bereich der digitalen Audiosignalverarbeitung - vor allem unter dem Aspekt der Echtzeit-Spatialisierung, der Nutzung von künstlichem algorithmischem Nachhall in diesem Zusammenhang sowie Berechnungen zur Schallausbreitung - viele nützliche Tools darstellen. Dabei bietet die Library ein modulares Framework, welches es ermöglicht diverse Use Cases abzudecken und das System anhand der verfügbaren Rechenkapazität zu skalieren.

Seit der Entstehung in den frühen 1990er-Jahren hat Spat viele algorithmische Entwicklungen, Verbesserungen und auch Refaktorisierung hin zu einem auf moderne Systeme mit einer hohen Kanalanzahl und viel Rechenkapazität optimieren Umgebung vorzuweisen. Die kritischsten Signalverarbeitungsblöcke sind somit hoch optimiert und vektorisiert und zeigen zur ursprünglichen rein Patch-basierten Implementierung eine deutlich bessere Performance. (Carpentier et al., 2015)

Max-Externals aus der Spat-5-Library werden nicht über die in Max/MSP üblichen Attribute gesteuert, sondern über eine Protokollschnittstelle und Syntax auf Basis von Open Sound Control (OSC). Die Gründe für die Nutzung dieser Schnittstelle liegen vor allem darin, dass die Objekte der Spat-5-Library aufgrund ihrer Anwendung zur Spatialisierung von komplexen Audioumgebungen (sei es rein kanalbasiert, szenenbasiert oder auch objektbasiert) nahezu immer algorithmische Verarbeitungen auf vielen Kanälen durchführen müssen, wobei jedem Kanal wiederum mehrere Attribute zugeordnet sind. Die gewählte Protokollschnittstelle bietet hier eine bessere Übersicht und Performance, da die Attribute nicht in Arrays, sondern in einer Baumstruktur gespeichert werden. Des Weiteren sind Spat-Objekte oft polymorph: Typ und Anzahl der exponierten Parameter können sich je nach Wahl des gewünschten vom Objekt durchgeführten Processings ändern. Die Anbindung an viele andere OSC-fähige Anwendungen, in denen die Spat-Library integriert ist und/oder mit denen sie interagieren kann, ist durch die Nutzung einer Steuerung auf Basis von OSC sichergestellt bzw. leicht zu realisieren und zu warten. OSC ist in der Audio-Gemeinschaft weit verbreitet und akzeptiert: Es erleichtert die Kommunikation zwischen Anwendungen, ermöglicht die Interoperabilität mit entfernten Geräten (z. B. über UDP/IP) und vieles mehr. Dazu kann es leicht implementiert werden und es sind mehrere Bibliotheken und Sprachbindungen verfügbar. Eine Kapselung von mehreren Nachrichten in sogenannten *Bundles*, die die Übertragung einer großen Menge synchroner Ereignisse vereinfachen, ist möglich. Betrachtet man die Anbindung der OSC-basierten Steuerung der Spat-5-Objekte mit dem Max/MSP-eigenen Nachrichten-Format, welches *Atom* genannt wird, ist eine solche Konvertierung trivial, da Atome und OSC-Argumente ähnliche Datentypen haben (int, float, Symbole, etc.). OSC-Bundles werden als *FullPacket* übertragen, die nur einen Zeiger auf eine Speicheradresse übermitteln; dies ermöglicht die sehr effiziente Übertragung großer Datenmengen. Der sogenannte *Max-Scheduler*, der die zeitliche Verarbeitung von Daten jeglicher Form in einer Max-Anwendung steuert und welcher im Unterunterabschnitt 5.6.2.1 ausführlicher beschrieben wird, wird dabei nur einmal pro Bundle ausgelöst und verarbeitet nicht jede im Bundle enthaltene Nachricht einzeln. Die am häufigsten verwendeten OSC-Adressmuster werden des Weiteren zur Kompilierzeit in einer Hash-Tabelle gespeichert, was CPU-intensive String-Operationen während der Laufzeit verhindert. (Carpentier, 2018a)

5.4.2.2 Messsystem

Um BRIRs in ausreichend hoher horizontaler Raster-Auflösung von maximal 2° (für weitere Ausführungen hierzu siehe Unterabschnitt 5.2.3) automatisiert messen zu können, wurde ein vollständiges Messsystem entwickelt, welches mithilfe eines Drehtellers und einer Computer-Anwendung nach Eingabe gewünschter Parameter eine Messroutine durchführt. Dabei wurde

darauf geachtet, dass die Messroutine eine Mindestqualität der gemessenen BRIRs garantiert und allen Anforderungen aus der Anforderungsanalyse genügt. Die gemessenen BRIRs können nach der Messung mithilfe der in Abschnitt 5.5 beschriebenen *Postprocessing-Blocks* in MATLAB bearbeitet und somit für die Auralisation in der *flexiblen Auralisationsumgebung* vorbereitet werden. Das Messsystem ist des Weiteren so konzipiert, dass es für weitere Entwicklungen im Rahmen der Messung von richtungsabhängigen Impulsantworten genutzt werden kann; dazu sind Erweiterungen an der Hardware und Software notwendig, die jedoch ohne kompletten Systemumstieg möglich sind.

Bevor die einzelnen Bestandteile des Systems genauer beschrieben und diskutiert werden, soll ein Blockschaltbild, welches in Abbildung 5.11 zu sehen ist, diese und deren Zusammenwirken darstellen. Obwohl das Messsystem funktional unabhängig von der Nutzung eines bestimmten Audiointerfaces ist, ist beispielhaft ein *RME MADIface Pro* als Audiointerface eingezeichnet, welches für die Messreihe im Rahmen der Erprobung des Gesamtsystems genutzt wurde; des Weiteren sind auch die genutzten aktiven *Genelec* Studiomonitore, sowie das genutzte Kunstkopfmikrofon *Neumann KU 100* namentlich erwähnt.

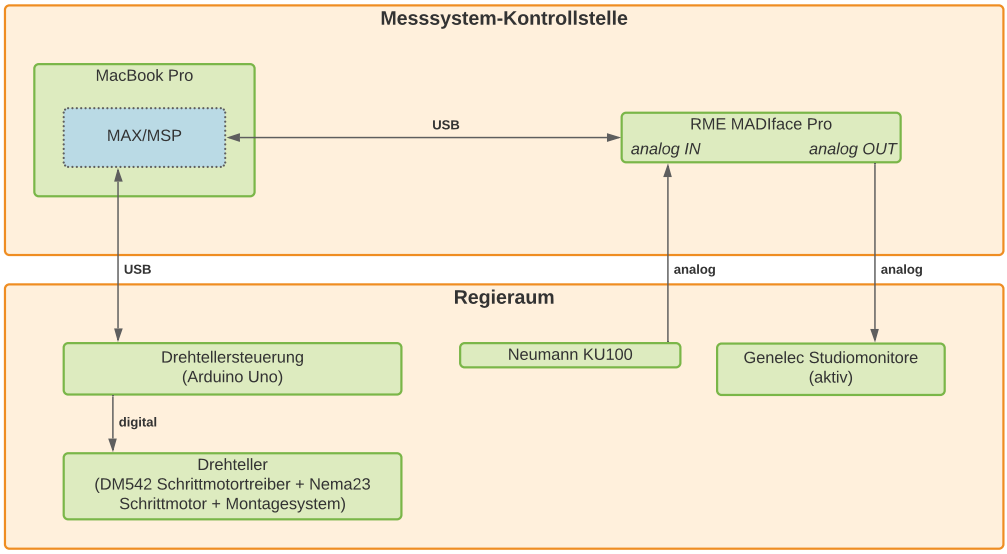

Abbildung 5.11: Blockschaltbild des Messsystems mit Nutzung eines RME MADIface Pro als Audiointerface (grün: Hardware-Geräte, blau: Software-Komponenten) (eigene Darstellung)

Es zeigt sich der zweiteilige Aufbau des Messsystems, welches sich technisch in eine *Messsystem-Kontrollstelle* und den zu vermessenen *Regieraum* bzw. das Mischkino teilt. Hierbei findet diese Trennung vorwiegend aufgrund akustischer Gründe statt: Die durchführende Person hat somit keinen Einfluss auf die Akustik des zu vermessenden Raumes und kann die *Messsystem-Kontrollstelle* außerhalb des Letzteren platzieren. So ist es auch möglich, automatisierte Messungen durchzuführen bzw. diese zu starten, ohne den zu vermessenden Raum nach Start der Messung noch verlassen oder bei Kontrolle der Messung wieder betreten zu müssen; dies ermöglicht eine flexible Kontrolle auch von langen Messreihen. Lediglich ein USB-Kabel nach USB-2.0-Standard, sowie analoge Mikrofonkabel, welche das Messsignal zu den Lautsprechern führen und die gemessenen Signale des Kunstkopfmikrofons zurück zum Audiointerface, müssen zwischen der *Messsystem-Kontrollstelle* und dem beispielhaft genannten *Regie-*

raum verlegt werden können. Der USB-2.0-Standard ist zwar nur bis zu einer maximalen Kabellänge vom 5 m spezifiziert, jedoch können auch längere Kabel unter Zuhilfenahme von Hubs oder Repeatern genutzt werden (STEMMER IMAGING AG, n. d.). Möglichkeiten der Optimierung des Systems hin zur digitalen Anbindung an eine zu vermessende Wiedergabeumgebung sind gegeben und können in weiteren Entwicklungsschritten durchgeführt werden.

Die Implementierung der Messsoftware wird mit Max/MSP und der Spat-5-Library von IRCAM umgesetzt. Die Anbindung des Drehtellers erfolgt über *Maxuino* und ist in Absatz 5.4.2.2.1 ausführlich dargestellt. Das *spat5.smk* Objekt (*smk = sweep measurement kit*) bietet für die Umsetzung der Messsoftware viele nützliche Funktionen, so bildet es den Kern der Implementierung. Diese hohe Funktionalität zeichnet sich durch die Sweep-Generierung auf Grundlage der angegebenen Kenndaten, das Aussenden der Sweeps, die anschließende Aufzeichnung der Sweeps inklusive Systemantwort, Entfaltung der Letzteren mit dem Anregungssignal und somit Gewinnung der Impulsantworten aus. In Echtzeit werden nach der Entfaltung Informationen zum SNR (Schätzung mithilfe Schröder-Rückwärtsintegration der Energie) der gemessenen Impulsantworten bereitgestellt, sowie Informationen zum maximal aufgezeichneten Pegel. Alle Berechnungen werden mit doppelt-genauer Gleitkommaarithmetik durchgeführt; das numerische Rauschen, das aus der spektralen Inversion des Sweeps resultiert, ist kleiner als -110 dB (Warusfel et al., 2018). Alle Daten (Rohaufnahmen, Sweep-Signal, IR-Daten usw.) werden zusammen mit verschiedenen textuellen Metadaten gespeichert; sowohl als 64-Bit-MATLAB-Dateien also auch *wav*-Files. Die *wav*-Files werden genutzt, um die BRIRs in das *Systemmodul 2: Postprocessing der BRIRs* zu überführen. Die automatisierte Steuerung des *spat5.smk* Objekts in Kombination mit dem Drehsystem wird durch Kontrollstrukturen umgesetzt, die an dieser Stelle nicht weiter betrachtet werden sollen. Die gemessenen BRIRs werden nach der Messung in einen *smk* genannten Ordner im gleichen Pfad der Anwendung gespeichert; der direkte Pfad zu den Daten lautet *smk/rir*. Dort können die Daten, welche nach dem Bennenungsschema *lautsprecher-nr_azimut* benannt und als *.wav* vorliegen in die in Abschnitt 5.5 erläuterte MATLAB-Umgebung eingeladen werden.

Die Abbildung 5.12 zeigt die Benutzeroberfläche der prototypisch implementierten Messsoftware für die automatisierte Messung von fein aufgelösten horizontalen BRIR-Datensätzen. Der Aufbau der Benutzeroberfläche ist hierbei in drei Hauptbereiche eingeteilt: Während im *Define Connection to Turntable* genannten Bereich der korrekte serielle Port zum in Absatz 5.4.2.2.1 beschriebenen Drehsystem ausgewählt und die Verbindung initialisiert werden muss, ist im *Measurement Settings* genannten Bereich die Eingabe der Anzahl der in der Messroutine gemessenen Lautsprecher, der gewünschten Schrittweite der horizontalen Drehung des Drehsystems, sowie aller Kenndaten zum genutzten Sweep der Messung zu tätigen. Des Weiteren kann den nach der Messung als Textfile gespeicherten allgemeinen Metadaten (Kenndaten der Messungen) noch weitere Informationen, wie beispielsweise eine Benennung der Messreihe oder Informationen zum Messaufbau bzw. dem gemessenen Lautsprechersetup, im Textfeld *Specific Measurement Metadata* hinzugefügt werden. Der letzte Bereich, welcher *Measurement Control* genannt wird, bietet sowohl die Möglichkeit die Messung/Messreihe zu

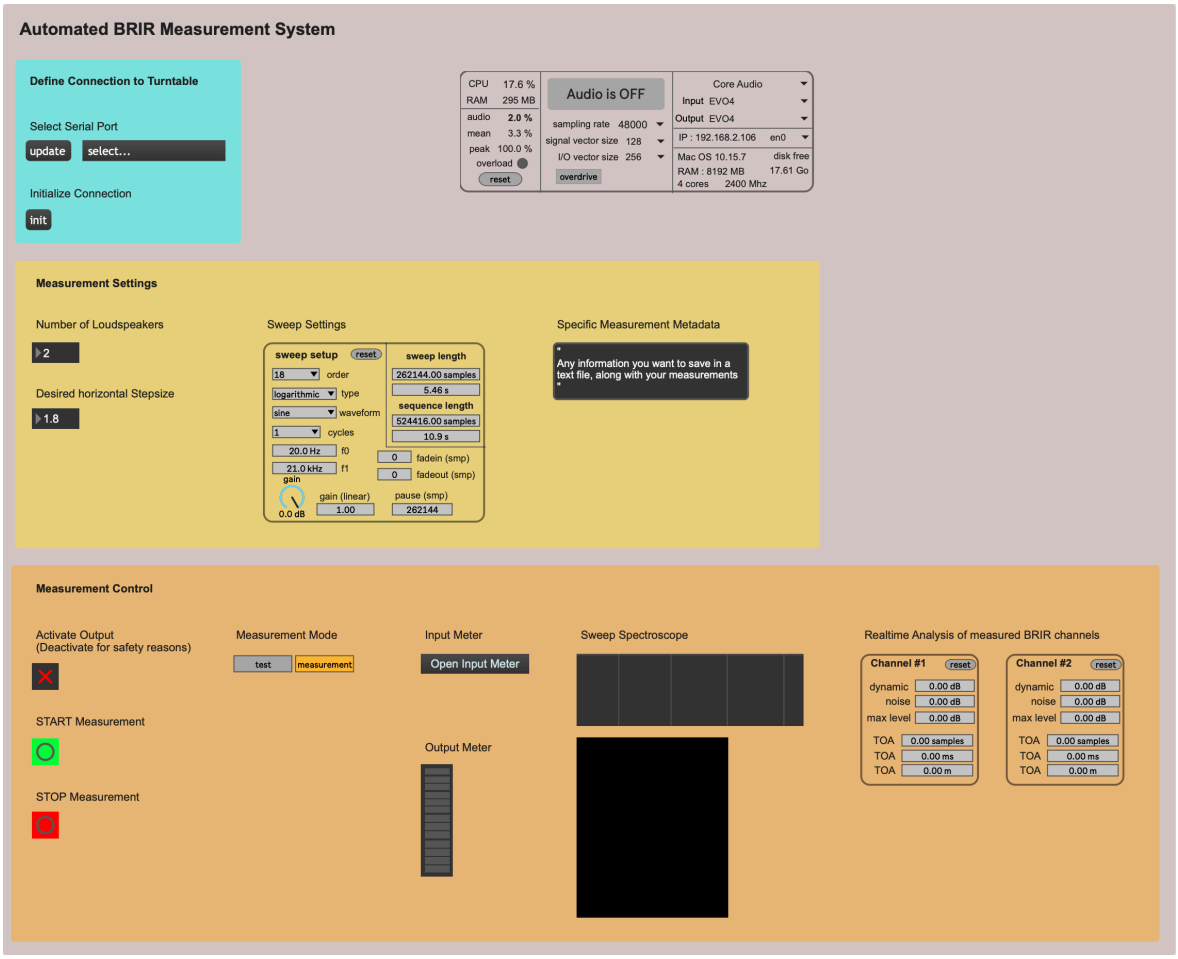

Abbildung 5.12: Benutzeroberfläche der Messsoftware zur automatisierten Messung von BRIR-Datensätzen (eigene Darstellung)

starten als auch zu stoppen, sowie eine Aktivierung des Audio-Outputs, welcher vor einer Messung zunächst noch freigegeben werden muss. Dies ist vor allem als Schutzmechanismus zu sehen, der als zweite Kontrollinstanz die nicht gewünschte Ausgabe von Messsignalen verhindert. Im *Measurement Mode* kann zwischen einem *Test*- und einem *Measurement*-Modus gewählt werden, wobei der *Test*-Modus keine Datenspeicherung umsetzt und auch das Drehsystem nicht bewegt; er ist also dazu gedacht, das Messsignal- und die Aufnahmeparameter - beispielsweise für das Erreichen eines bestimmten maximalen Rauschpegels - aufeinander abzustimmen, bevor die eigentliche Messung im *Measurement*-Modus beginnt. Das *Output Meter* zeigt den Spitzenpegel des ausgegebenen Messsignals, während das *Input Meter* den Spitzenpegel der beiden Eingangssignale darstellt; hierbei muss das *Input Meter* erst über den Button *Open Input Meter* in einem separaten kleinen Fenster geöffnet werden, bietet dann aber eine genauere Kontrolle des Spitzenpegels als das *Output Meter*. Das *Sweep Spectroscope* visualisiert das gewählte Messsignal bei jedem Abspielvorgang in Echtzeit. Die *Realtime Analysis of measured BRIR channels* liefert Informationen zu den gemessenen BRIRs nach der Entfaltung; hierbei sind vor allem die drei oberen Kenndaten interessant, die mit *dynamic* den (mithilfe Schröder-Rückwärtsintegration geschätzten) SNR ausgeben, sowie mit *noise* und *max level* den Rauschpegel sowie den Maximalpegel des aufgenommenen Messsignals (jeweils in dBFS) angeben. Des Weiteren befindet sich oben rechts im Anwendungsfenster eine Oberfläche, mit der sich allgemeine Einstellungen zur Audiosignalverarbeitung

tätigen lassen, wie die Wahl des genutzten Audiotreibers, des Audio-Eingabe- und Audio-Ausgabegeräts sowie der genutzten Abtastrate, Signalvektorgröße und der I/O-Vektorgröße (bzw. I/O-Audiobuffergröße).

Es ist zu erkennen, dass einige Anforderungen noch nicht vollständig erreicht sind: So ist die Benennung der Messung bzw. Messreihe momentan lediglich über das *Specific Measurement Metadata* genannte Textfeld möglich, welches diese Informationen zusammen mit den allgemeinen Metadaten der Messung in einem Textfile speichert. Sofern die BRIR-Daten, welche nach dem Schema *lautsprecher-nr_azimut* benannt werden, nach jeder Messreihe aus der Ordnerstruktur der Messsoftware entfernt werden, können Probleme bei der Zuweisung der Messdaten verhindert werden, welche momentan aufgrund der fehlenden direkten Benennung der Daten bzw. der dahinter liegenden Ordnerstruktur mit einem aussagekräftigen *Namen der Messung/Messreihe* möglich sind. Des Weiteren kommt es noch nicht zu einer Ausgabe der Informationen zum momentan gerade in Messung befindlichen Lautsprechers, sowie der aktuell vorliegenden (horizontalen) Orientierung des Drehsystems. Dies, sowie auch eine einfache Ausgabe des Status der Messroutine (*aktiv* oder *beendet*) ist noch zu implementieren.

Da die Messsoftware dazu genutzt werden soll, Messreihen routinemäßig zu steuern, im Folgenden beispielhaft der Ablauf einer Messroutine erläutert: Bei der Wahl von mehr als einem Lautsprecher werden zunächst alle Lautsprecher mit frontaler Kopforientierung (Azimutwinkel = 0°) gemessen, ehe entgegen des Uhrzeigersinns im mathematisch positiven Drehsinn in der gewählten Schrittweite horizontal weitergedreht wird, ehe für die nächste Orientierung wieder alle Lautsprecher gemessen werden. Somit muss das Drehsystem nur ein Mal eine komplette Umdrehung ausführen, auch wenn mehrere Lautsprecher gemessen werden. Dieses Vorgehen spart Zeit, da das Drehsystem wie in Absatz 5.4.2.2.1 beschrieben recht langsam bewegt wird, um die Bewegung kontrolliert und exakt auszuführen, sowie ein *Nachschwingen* des Systems zu vermeiden.

5.4.2.2.1 Drehsystem: Drehteller und Drehtellersteuerung Das entwickelte *Drehsystem* besteht aus einem *Drehteller*, der sich aus mechanischen und elektrischen Komponenten zusammensetzt, und einer Drehtellersteuerung, welche mithilfe eines *Arduino Uno*, ein Mikrocontrollerboard auf Grundlage eines ATmega328P Mikrocontrollers, und *Maxuino* zur Kommunikation mit Max/MSP umgesetzt wird. Das Drehsystem wurde im Rahmen dieser Arbeit für die Nutzung mit einem *Neumann KU 100* Kunstkopfmikrofon optimiert, ist jedoch so konzipiert, dass auch Erweiterungen für andere Positions- und Orientierungs-bezogene Messungen mit z.B. Mikrofonarrays oder eine Anbindung an einen größeren Drehteller für die Personen-bezogene Messung von BRIRs oder HRIRs möglich sind. Da die Umsetzung des Drehsystems aus Kostengründen zunächst ohne einen geschlossenen Regelkreis (also *open loop*) erfolgt, ist es nicht möglich, den Schrittmotor und damit das Drehsystem unter Eingabe eines gewünschten Wertes absolut zu positionieren. Des Weiteren ist die Schrittanzahl und die daraus resultierende Schrittweite ebenfalls nicht geregelt, sodass es theoretisch zu unerkannten Schrittausfällen kommen kann. Da der Schrittmotor und der genutzte Schrittmotortreiber jedoch so dimensioniert wurden, dass der Schrittmotor das benötige

Abbildung 5.13: *Neumann KU 100* Kunstkopfmikrofon auf dem Drehsystem (eigene Darstellung)

Drehmoment für die Drehung in seinem normalen Arbeitsbereich aufbringen kann und auch der Schrittmotortreiber nicht zu stark belastet wird, sind keine Schrittausfälle zu erwarten.

Mechanisch ist das Drehsystem so aufgebaut, dass es auch möglich ist, andere *Drehtelleraufbauten* direkt oder über ein Getriebe mit dem Schrittmotor zu verbinden und das System so zu erweitern. Die genutzte Bodenplatte ist quadratisch und hat eine Kantenlänge von 60 cm; somit kann die interaurale Achse des etwa 3,5 kg schweren *Neumann KU 100* Kunstkopfmikrofons auf eine Höhe von bis zu 1,6 m gebracht werden, ohne das aufgrund möglicher Hebelwirkungen der Aufbau umzukippen droht. Um des Weiteren ein Verrutschen der Bodenplatte zu verhindern, sind an ihrer Unterseite mehrere haftverstärkende Pads angebracht. Die interaurale Achse kann mit dem Drehsystem auf einer Höhe von 1,1 bis 1,6 m platziert werden, was für übliche menschliche Sitzpositionen und die in diversen Normen (z.B. ITU-R BS.775-3, DIN 15996) festgelegten Werte von 1,2 bis 1,3 m Höhe der akustischen Achse der Lautsprecher, welche sich im Lautsprecher-Layer direkt um den Kopf herum befinden, ausreichend ist. Die technische Möglichkeit für diese in der Höhe variable Platzierung des Kunstkopfmikrofons bietet das genutzte Mikrofonstativ, welches mit der Grundplatte des *Drehtelleraufbaus* bzw. der Drehplatte verschraubt ist. Der *Drehtelleraufbau* wird am Schrittmotor, welcher exakt mittig an der Bodenplatte befestigt ist, über einen Montage-Hub mithilfe zweier Schrauben befestigt und ist folglich leicht abnehmbar. Die Kraftwirkung auf diesen Montage-Hub bzw. den Schaft des Schrittmotors (Durchmesser: 8 mm) wird über eine mehrschrittige kegelförmige Abstufung der Drehplatte verringert; eine statisch motivierte Unterstützung der größten

kreisförmigen Plattform entlang der Außenkante mit Hilfe von beispielsweise Rändern ist des Weiteren möglich. Um eine Hebelwirkung auf den Aufbau prinzipiell zu vermeiden, sind alle Komponenten des Drehsystems so platziert, dass sich der resultierende Massenmittelpunkt über der Drehachse befindet. Der Schrittmotor selbst ist über zwei Schrittmotorhalterungen, sowie vier weitere Gewindestifte mehrfach mit der Bodenplatte verbunden. Da der untere Gewindeanschluss des *Neumann KU 100* Kunstkopfmikrofons sich nicht auf der interauralen Achse befindet, sondern 3 cm von ihr in Richtung Nase verschoben ist, muss der Anschluss des Kunstkopfmikrofon zur vertikalen Rotationsachse (z-Achse) versetzt werden, sodass sich die Rotationsachse exakt entlang der interauralen Achse in der Mitte des Kopfes befindet. senkrecht durch die interaurale Achse verläuft,Dies wird mithilfe einer 20 cm langen Stereoschiene erreicht; um die Hebelwirkung aufgrund der dadurch verursachten Verschiebung des Massenmittelpunktes des Kunstkopfmikrofons auszugleichen, werden entgegen des Versatzes des Kunstkopfmikrofons entlang der Stereoschiene Gewichte platziert. Da sich der 5-polige XLR-Anschluss des Kunstkopfmikrofons exakt unter der Drehachse befindet, muss es oberhalb der Stereoschiene mithilfe eines Verlängerungsrohrs erhöht werden, sodass der Anschluss weiterhin erreichbar bleibt; dies wird mit einem 13 cm langen Verlängerungsrohrs erreicht. Die Orientierung des Kunstkopfmikrofons kann in mehreren Schritten geprüft werden: Dem Drehsystem werden zwei Lote beigefügt, welche die Ausrichtung des Kunstkopfmikrofons zur Stereoschiene als auch der Stereoschiene zur Bodenplatte prüfen können; befindet sich die Letztere an/in einer geprüften bekannten Position/Orientierung, so lässt sich die davor gewonnene Information übertragen. Des Weiteren bietet die genutzte Stereoschiene im Blickfeld des Kunstkopfmikrofons Möglichkeiten der Anbringung eines Lasers, mit welchem (nach vorheriger Überprüfung der Orientierung des Kunstkopfmikrofons zur Stereoschiene) die Ausrichtung des Kunstkopfmikrofons vorgenommen werden kann.

Abbildung 5.14: Möglichkeiten zur laserbasierten Ausrichtung des Drehsystems (eigene Darstellung)

Da die mechanische, sowie auch elektrische Positionierungsgenauigkeit für die Qualität der BRIR-Datensätze und damit auch für die Nutzung des Systems nicht unerheblich ist, wurden etwaige Abweichungen, welche durch den mechanischen Aufbau begründet sind, geprüft. Hierbei können Abweichungen von etwa ± 0,5 cm im Bereich der üblichen Position eines

Kunstkopfmikrofons innerhalb der Horizontalebene festgestellt werden. Diese sind sehr wahrscheinlich durch Abweichungen bei der handwerklichen Umsetzung des Drehtellers zu begründen, wobei an dieser Stelle vor allem die Anbringung des schraubbaren Mikrofonstativs in der Drehachse hervorzuheben ist. Für die beispielhafte Prüfung des konzipierten Systems dieser Arbeit werden diese Abweichungen hingenommen, da sie bei händischer Drehung des Kunstkopfmikrofons ohne mechanisch-elektrisches Drehsystems in ähnlichem Ausmaß zu verzeichnen wären. In dem beispielhaft in dieser Arbeit simulierten 2.0-Stereoaufbau mit einem Abstand von 2,03 m zwischen Lautsprechern und Hörposition, kann diese Abweichung mithilfe von Trigonometrie (tan) in einen Winkel abgeschätzt werden, welcher mit $\pm$ 0,16° zu verzeichnen ist; ein Fehler in dieser Größenordnung scheint, wie auch Shotton et al. (2014) abschätzt, bei Betrachtung des Auflösungsvermögens unseres Gehörs nicht von Relevanz.

Abbildung 5.15: Möglichkeiten der Ausrichtung des Drehsystems mithilfe von Loten(eigene Darstellung)

Die elektrischen Komponenten des Drehsystems setzen sich aus einem 2-phasigen Hybridschrittmotor (nach NEMA 23 Baugrößen-Norm) mit einem minimalen Haltmoment von 2,8 Nm (bei 4,8 V Nennspannung und 3 A Nennstrom pro Phase) und einem Vollschrittwinkel von 1,8° sowie dem dazu passend dimensionierten Schrittmotortreiber DM542 zusammen. Der DM542 ist ein volldigitaler Schrittmotortreiber, der mit einem fortschrittlichen DSP-Regelalgorithmus ausgestattet ist und folglich die bestmögliche Nutzung des Motors bei der zur Verfügung gestellten Leistung garantiert. Mithilfe des ausreichend dimensionierten Schaltnetzteils kann der DM542 Schrittmotortreiber so den benötigten Strom geregelt aufbringen, um gewünschte Halte- bzw. Drehmomente zu garantieren. Die auf diese Weise innerhalb des Systems erreichten 2,8 Nm Haltemoment reichen für das zuvor auf Grundlage der Belastung des Motors durch die verteilte Masse entlang des Drehsystems bestimmte benötigte Drehmoment aus, sodass keine Schrittausfälle zu erwarten sind. Das maximale Drehmoment eines Schrittmotors bei geringen Geschwindigkeiten erreicht, sodass der Schrittmotor bei Drehung langsam angefahren und auch wieder abgebremst wird. Der benutzte Schrittmotortreiber DM542 ist des Weiteren in der Lage eine Schrittteilung (Mikroschrittbetrieb) durchzuführen, welche eine höhere Winkelauflösung bei prinzipiell höherer Laufruhe (gleichmäßigeres

Drehmoment, vibrations- und geräuschärmer) des Motors verursacht. Dabei ist jedoch darauf zu achten, dass das inkrementale Halte- bzw. Drehmoment pro Mikroschritt auf diese Weise abnimmt und es leichter zu Schrittausfällen bei Überlastung kommen kann. Mithilfe einer Übersetzung ist es möglich, dass verfügbare Halte- bzw. Drehmoment des gewählten Schrittmotors weiter zu erhöhen. Die Wahl auf einen Schrittmotor der beschriebenen Dimensionierung wurde aus einer Kosten-/Nutzenabschätzung getätigt: Er erreicht ohne Getriebeübersetzung das im System dieser Arbeit benötigte Drehmoment problemlos und bietet aufgrund seiner Größe einen ausreichend dimensionierten Schaft, um das Drehsystem direkt aufzunehmen; somit war es möglich ohne weitere Betrachtung hinsichtlich eines Getriebes oder eine Absetzung des Drehsystems vom eigentlichen Motor ein funktionsfähiges System aufzubauen. Der relative Fehler bei Drehung um einen Vollschritts von 1,8° liegt laut Datenblatt des Schrittmotors bei ± 5 %, was zu einem Fehler von ± 0,09° führt. Zieht man nun eine einfache Fehleraddition (welche Abhängigkeiten aufgrund der Unwissenheit über Letztere auslässt) des mechanischen Fehlers mit diesem Fehler in Betracht, so wird der Fehler des Gesamtsystems ± 0,25° groß; dies scheint in dieser Größenordnung weiterhin irrelevant. Mithilfe einer optischen Analyse des Messaufbaus sowie eine Analyse der akustischen Daten von fünf Messreihen konnte keine Abweichung der Start- und Endposition bei vollständiger 360°-Drehung festgestellt werden. Eine genauere elektrische Überprüfung des mithilfe der Systemkomponenten maximal zu erreichenden Drehmoments bzw. bestenfalls eine Messung einer Drehmomentkurve ist allerdings wünschenswert, um die Nutzbarkeit des Systems für weitere Aufbauten abzuschätzen.

Die Drehtellersteuerung basiert auf der Nutzung eines *Arduino Uno*, ein Mikrocontrollerboard auf Grundlage eines *ATmega328P* Mikrocontrollers, sowie *Firmata* und *Maxuino* zur seriellen Kommunikation mit der Computer bzw. der Anwendung, welche in Max/MSP implementiert ist. Dazu muss auf dem Arduino Mikrocontrollerboard die Firmata-Bibliothek *Firmata.h* eingebunden werden, welche das Firmata-Protokoll für die Kommunikation mit der Anwendung auf dem Computer implementiert. Dadurch kann benutzerdefinierte Firmware unter Zuhilfenahme dieses Protokolls geschrieben werden. Dies setzt Maxuino in Form eines Patches um, indem ein eingebundenes Javascript-Objekt ihm übergebene Befehle in Form von Max-Nachrichten in das Firmata-Protokoll überträgt und diese dann über die serielle Schnittstelle gesendet werden; umgekehrt wird auch übersetzt, wenn eingehende Nachrichten empfangen werden. Auf diese Weise ist es möglich mit gewöhnlichen Max-Nachrichten die Firmware und damit das Arduino Mikrokontrollerboard zu steuern. Die Nutzung der Steuerbefehle eines Schrittmotortreibers erfordert des Weiteren neben der Nutzung der Standard-Firmata-Bibliothek *Firmata.h* die Nutzung der Bibliothek *StepperFirmata.h*, welche innerhalb des sogenannten *ConfigurableFirmata*-Sketches vollständig funktional eingebunden wird; dieser Sketch muss folglich auf das genutzte Arduino Mikrocontrollerboard geladen werden, um Firmata mit Hilfe von Maxuino zur Steuerung von Schrittmotoren in vollem Funktionsumfang nutzen zu können. Dieser Sketch liegt auf dem Datenträger dieser Arbeit bei. Die Anbindung des Arduino Mikrocontrollerboards an den Computer erfolgt via USB.

Der Schrittmotortreiber DM542 wird mithilfe von drei digitalen Signalen gesteuert, deren High-Pegel zwischen 5 und 24 V liegen kann. Dies lässt die Steuerung des Schrittmotortrei-

bers mit einem Arduino Uno zu, welcher über 14 digitale I/O Pins verfügt und mit einer Spannung von 5 V als HIGH-Pegel arbeitet. Die drei vom Schrittmotortreiber DM542 genutzten Signale sind ENA, welches ein *Enablesignal* darstellt, das zu Beginn der Nutzung des Schrittmotortreibers auf HIGH gesetzt werden muss. Dieses Signal wird nicht zwangsläufig benötigt und ENA wird auch ohne Signalgeber aktiviert. Das DIR-Signal ist das *Directionssignal*, mit welchem über den HIGH- und LOW-Pegel die zwei Rotationsrichtungen des Motors gesteuert werden. Das PUL-Signal ist das *Pulsesignal*, das in der Maxuino-Umgebung auch als STEP-Signal bezeichnet wird und über den HIGH-Pegel jeweils einen Mikroschritt in der am Schrittmotortreiber sowie innerhalb der Maxuino-Umgebung bestimmten Mikroschrittauflösung (Schritte je Umdrehung) umsetzt. Für die Nutzung innerhalb dieses Systems wurde eine Mikroschrittauflösung von 12800 Mikroschritten pro Umdrehung gewählt, was zu einer Auflösung von etwa 0,028° pro Mikroschritt führt und 64 Mikroschritte für die gewünschte Schrittweite von 1,8° gewählt werden. Bevor Steuerdaten an den Schrittmotortreiber gesendet werden können, muss die Kommunikation innerhalb Maxuino/Firmata zunächst konfiguriert werden: Dies schließt eine Angabe der gewünschten Mikroschrittauflösung als auch die Angabe der genutzten digitalen Outputs des Arduino ein; die DIR-Steuerung des DM542 Schrittmotortreibers erfolgt über den digitalen Output des Pins 11, während die STEP- bzw. PUL-Steuerung über den Output des Pins 10 erfolgt. Eine ENA-Steuerung bleibt aus. Neben der eigentlichen Schritt- bzw. Mikroschrittauslösung ist auch eine Steuerung der Winkelgeschwindigkeit des Schrittmotors, sowie der Winkelbeschleunigung beim Anfahren und Stoppen möglich. Die in diesem Zusammenhang gesetzten Werte wurden empirisch in Versuchen mit dem Kunstkopfmikrofon *Neumann KU 100*, bei Betrachtung dessen Schwingungsverhaltens bei Drehung auf dem Drehtelleraufbau, sowie eines Tests hinsichtlich möglicher Schrittausfälle gefunden und auf eine Winkelgeschwindigkeit von $1\frac{rad}{s}$ (resp. $57,3\frac{°}{s}$), sowie Winkelbeschleunigungswerte von $0,1\frac{rad}{s^2}$ (resp. $5,7\frac{°}{s^2}$) gesetzt. Die recht hohe Mikroschrittauflösung in Kombination mit diesen niedrigen Geschwindigkeits- und Beschleunigungswerten führt zu einer hohen Laufruhe bei gleichzeitig nicht feststellbaren Schrittausfällen; die erreichte Laufruhe ist akustisch unauffällig. Die zwei typischen gesendeten Nachrichten zur Konfiguration und Schrittausführung sind von der Form `stepperConfig 0 1 12800 11 10` (0=Schrittmotor-Device-Nummer, 1=Schrittmotor-Typ (hier STEP+DIR), 12800=Schritte pro Umdrehung, 11=DIR-Pin, 10=STEP-Pin) sowie `stepperStep 0 0 64 100 10 10` (0=Schrittmotor-Device-Nummer, 0=DIR-Wert, 64=ausgeführte Schrittanzahl bei Befehl, 100=Winkelgescheindigkeit in $0,01 \cdot \frac{rad}{s}$, 10=Beschleunigung Anfahren in $0,01 \cdot \frac{rad}{s^2}$, 10=Beschleunigung Abbremsen in $0,01 \cdot \frac{rad}{s^2}$). Zu Beginn einer jeden Messroutine ist es nötig, *Maxuino* den seriellen Port des angeschlossenen Arduino Mikrokontrollerboards, welches das Drehsystem steuert, zu übergeben, und einen `init`-Befehl zur entsprechenden Initialisierung zu senden.

Die Kosten für das komplette Drehsystem belaufen sich auf etwa 250 Euro, wobei besonders hohe Kostenpunkte durch die leistungsstarken elektrischen Komponenten (Schrittmotor, Schrittmotortreiber, Schaltnetzteil), sowie das hochwertige Mikrofonstativ, welches an den Drehteller geschraubt wurde, zu verzeichnen sind.

Die Nutzung des Drehsystems mit den angegebenen Kennwerten der Motorsteuerung er-

folgt weitestgehend lautlos und ohne Vibrationen. Die geringen Vibrationen bei der Drehung sind insofern nützlich, da diese aufgrund der direkten Verbindung mit dem Drehtelleraufbau an den Letzteren weitergegeben werden und das Kunstkopfmikrofon in Schwingung versetzen können. Diese Schwingen müssten vor einer weiteren BRIR-Messung ausreichend gedämpft worden bzw. abgeklungen sein, was die komplette automatisierte Messung verzögert. Somit ist es für eine schnellen Abfolge von BRIR-Messungen nach einer Drehung wichtig, das die Drehung den kompletten bis zu 1,6 m langen Drehaufbau nicht in Schwingung versetzt.

Die automatisierte Messung kann nach Benutzer-Eingabe der für die Messung benötigten Kenndaten des Messsignals, sowie nach einer entsprechenden manuellen iterativen Einstellung des Ausgabepegels (Gain) auf Grundlage des zu erreichenden SNRs, sowie nach der Eingabe der gewünschten Schrittweite und der zu messenden Lautsprecher gestartet werden. Die Messroutine betrachtet dabei immer eine komplette Drehung um die Vertikalachse (z-Achse) im mathematischen Drehsinn, ehe es zu einem Stop der Letzteren kommt. Dabei beginnt die Messung in der Orientierung, welche im gewählten Koordinatensystem mit einem Azimutwinkel von 0° bezeichnet wird, in dem der Reihe nach (beginnend mit Lautsprecher 1, welcher - um Verwirrung zu vermeiden - derjenige Lautsprecher sein sollte, der mit dem ersten Output verbunden ist) die Lautsprecher zunächst für diese Orientierung gemessen, ehe der angegeben Schrittweite entsprechend eine Schrittweite entgegen des Uhrzeigersinns (mathematisch positiv) weiter gedreht wird. In dieser Orientierung wird die Messung für jeden Lautsprecher (zeitlich komplett voneinander abgesetzt) erneut durchgeführt, ehe wieder weiter gedreht wird. Dies wird so lange weitergeführt bis in allen Orientierungen alle Lautsprecher gemessen wurden. Dann kommt es zu einem Abbruch der Messung, welcher dem Benutzer des System ausgegeben wird.

Um eine ausreichende Trennung zwischen einer Drehung und der darauffolgenden Messung zu erreichen, ist in der Anwendung je eine Pause zwischen der Drehung, der Messung bzw. den Messungen und der darauffolgenden Drehung implementiert. Diese Pause berücksichtigt einen kurzen Abklingvorgang möglicher Schwingungen des Drehsystems nach Drehung sowie eine optimale akustische Trennung zu den Aufzeichnungen des Sinus-Sweeps nach Anregung des Raumes. Dabei wird der eigentliche Sweep sowie die darauffolgende Pause, welche noch in die Aufzeichnung und die darauf folgende Entfaltung der Signale eingeht, nicht durch eine Drehung des Schrittmotors gestört. Prinzipiell arbeitet der Schrittmotor mit den implementierten Werten zur Mikroschrittauflösung, Winkelgeschwindigkeit sowie zur Winkelbeschleunigung nahezu lautlos, jedoch nicht vollständig lautlos. Im Ruhezustand dagegen ist durch die elektrischen Komponenten keinerlei Störschall festzustellen. Die großzügig gesetzten Pausen führten in der beispielhaft durchgeführten Messreihe (exponentieller Sinus-Sweep: 20 Hz - 21 kHz, 2^{18} Samples bzw. 5,46 s Sweeplänge mit 5,46 s Pause nach Sweep somit 10,92 s Signallänge, 1,8° horizontale Raster-Auflösung) zu einer Messdauer von etwa einer Stunde für einen einzelnen Lautsprecher. Unter Betrachtung der Anzahl von 200 Einzelmessungen für diese Messreihe kann die benötigte Zeit für die Schrittmotorsteuerung, die Bewegung des Schrittmotors sowie die implementierten Pausen zusammenfassend mit etwa 23 Minuten angegeben werden; dies führt wiederum zu 7 Sekunden, die bei jeder Einzelmessung für den komplet-

ten Drehvorgang inklusive dessen Ein- und Ausleitung in der momentanen Implementierung genutzt werden. Da die Drehzeit beinahe an die eigentliche Signallänge der Messung heranreicht, ist sie unter Erachtung des Drehsystems als Hilfsmittel als (zu) lang zu bezeichnen; somit sollten an dieser Stelle Optimierungen angestrebt werden. Damit die Pausen sowie die Kenndaten der Schrittmotorbewegung (Winkelgeschwindigkeit, Winkelbeschleunigung) vom Anwender selbst definiert werden können, gibt es für sie eine Eingabemöglichkeit; die gewählten Parameter sollten jedoch nur mit Vorsicht unter Beachtung aller Einflussfaktoren abweichend von den standardmäßigen Parametern gewählt werden. Durch Anpassung kann die Messroutine jedoch deutlich beschleunigt werden, was gerade bei Messreihen mit vielen Lautsprechern und hoher Raster-Auflösung des Datensatzes nicht unerheblich den benötigten (Zeit-)Aufwand reduziert.

5.4.2.2.2 Beispielmessung Mit der Absicht, das Gesamtsystem zu erproben, wurde eine beispielhafte Messung eines standardisierten 2.0-Stereosetups mit einem Öffnungswinkel von 60° (DIN 15996) durchgeführt, welches sich im 3D-Audio-Produktionsraum (U48) der Hochschule der Medien Stuttgart befindet.

Abbildung 5.16: Vollständige Ansicht des Messsystems im beispielhaften Tonregieraum zur Erprobung des Gesamtsystems (eigene Darstellung)

Dieser Raum ist in Abbildung 5.16 zu sehen; er hat eine trapezartige Form mit nur zwei parallelen Wänden. Alle Lautsprecher, die für die Messungen genutzt wurden, sind aktive Genelec 8040B Studiomonitore. Diese Lautsprecher sind 2-Wege Lautsprecher, welche laut Herstellerangaben einen Übertragungsbereich (-3 dB) von 45 Hz bis 21 kHz haben. Die ebenfalls vom Hersteller bestimmte akustische Achse der Lautsprecher befindet sich im genannten Raum auf einer Höhe von 1,34 m und ist je 2,03 m vom Abhörort entfernt. Neben den BRIR-Messungen bei Anregung der Lautsprecher des 2.0-Stereosetups wurde mithilfe von fünf Lautsprechern eines 5.0-Surroundsetups (nach ITU-R BS.775) eine gemittelte Nachhall-

zeit des Raumes mit einem omnidirektionalen Messmikrofon bestimmt. Diese Nachhallzeitmessung entspricht nicht den Vorgaben der ISO 3382, welche das Verfahren zur Messung von Nachhallzeiten in Räumen festlegt und so gewährleistet, dass die gemessenen Nachhallzeiten verschiedener Räume objektiv miteinander verglichen werden können. Es stand für die Messreihe jedoch keine omnidirektionale Schallquelle (Dodekaeder) zur Verfügung, sodass die getätigten Messungen als zweckmäßig eingestuft werden. Es sind bei den mit jedem Lautsprecher ermittelten Nachhallzeiten auch keine substantiellen Abweichungen zu verzeichnen.

Um einen Einfluss der Messkette auf alle im Rahmen dieser Arbeit getätigten Messungen zu bewerten, sollen folgende Betrachtungen angestellt werden: Die Messungen der Nachhallzeit im Vorfeld der eigentlichen BRIR-Messungen wurden mit einem Earthworks M23 Messmikrofon[3] und einem RME MADIface Pro durchgeführt. Für das kalibrierte Messmikrofon liegt ein Messschrieb vor, der den Freifeld-Frequenzgang (auf Achse) des Mikrofons darstellt bzw. beschreibt; zwischen 10 Hz und 30 kHz sind Abweichungen von einem linearen Frequenzgang von maximal ± 0,3 dB angegeben. Des Weiteren gibt das Datenblatt einen Grenzschalldruckpegel von 140 dB SPL und einen Rauschpegel von 20 dB SPL-äquivalent (A-bewertet) an. Damit hält dieses Mikrofon den Anforderungen der IEC 61094 und ANSI Type 1 Normen für Messmikrofone Stand bzw. übertrifft sie. Das RME MADIface Pro besitzt laut Datenblatt für seine Ein- und Ausgänge einen Rauschabstand von minimal 113 dB RMS (unbewertet) und für den Eingangskanal bei 30 dB Gain ein THD+N von weniger als 0,001 %, des Weiteren erreicht es einen Frequenzgang der Ein- und Ausgänge, der innerhalb von 18 Hz bis 20,8 kHz maximal um ± 0,1 dB von einem linearen Frequenzgang abweicht. Diese Werte zeugen damit auch von einer ausreichenden Qualität des RME MADIface Pro, um den Ansprüchen akustischer Messungen zu genügen. Trotzdem wurde der Frequenzgang des RME MADIface Pro im Rahmen dieser Arbeit nochmals mithilfe einer *Kalibrationsmessung*, welche den Frequenzgang bei direkter Verbindung des Ausgangs mit dem Eingang (Loopback) misst, geprüft; Letzterer ist in Abbildung 5.17 dargestellt und zeigt, dass durch die Nutzung des RME MADIface Pro als Audiointerface der getätigten Messungen sowohl im Betragsfrequenz-, als auch Phasengang kein verfälschender Einfluss zu erwarten ist; der Betragsfrequenzgang im genutzten Übertragungsbereich von 20 Hz bis 21 kHz verläuft mit einer maximalen Abweichung von -0,1 dB linear und auch der Phasengang ist nahezu linear und ohne erwähnenswerte Phasensprünge.

Damit insbesondere die Lautsprecher in einem üblichen Arbeitsbereich bei den Messungen betrieben werden, wurde im Voraus der Messung der Übertragungsweg auf den Referenzabhörpegel eingestellt. Der Referenzabhörpegel der Wiedergabekanäle wird durch Messung des A-bewerteten Schalldruckpegels am Abhörpunkt eingestellt. Dazu wird die Empfindlichkeit jedes einzelnen der n Wiedergabekanäle so angepasst, dass für rosa Rauschen (um störende modale Eigenschaften des Raumes auszublenden, meist bandpassgefiltertes rosa Rauschen von 200 Hz bis 20 kHz) mit einem Digitalpegel von -18 dBFS RMS den nach folgender Gleichung bestimmten Schalldruckpegel erreicht:

$$L_{\mathrm{LISTref}} = 85 - 10 \cdot log(n) \qquad [dB(A)] \tag{5.5}$$

[3] http://earthworksaudio.com/measurement-microphones/m23/

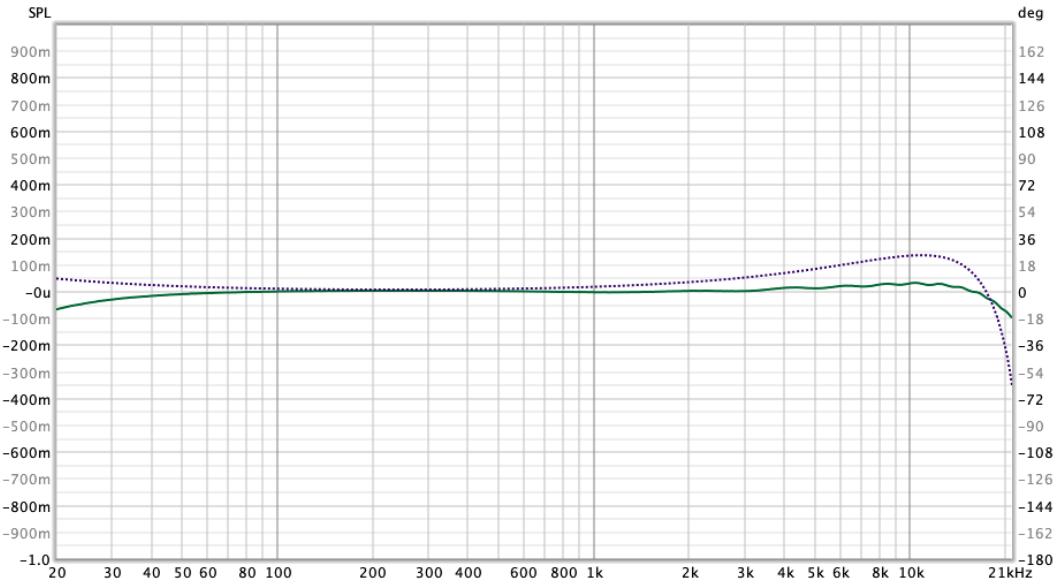

Abbildung 5.17: Kalibrationsmessung (Loopback) des RME MADIface Pro (1/48-Oktavglättung, Ausgang: -37 dBFS, Eingang: +27 dB Gain) (eigene Darstellung)

Für ein 2.0-Stereosetup und somit $n = 2$, welches beispielhaft simuliert werden soll, führt dies zu einem Schalldruckpegel von 82 dB(A), der am Abhörpunkt für jeden einzelnen Lautsprecher erreicht werden soll; dabei sollten die Abweichungen zwischen zwei Kanälen 0,5 dB nicht überschreiten. (Behr, n. d.) Die Messung des Schalldruckpegels am Abhörpunkt zeigten für die Lautsprecher des 2.0-Stereosetups eine Abweichung von 0,5 dB, wobei der linke Lautsprecher (L) 82,5 dB(A) bei gleichem Digitalpegel (-18 dBFS RMS) des bandbegrenzten rosa Rauschens lieferte; diese Abweichung ist nicht kritisch. Um den Prozess der Einstellung des Referenzabhörpegels nicht unnötig kompliziert zu gestalten, wurden auch die übrigen drei Lautsprecher (C, LS, RS), welche für die gemittelte Nachhallzeitmessung genutzt wurden, auf einen Schalldruckpegel von 82 dB(A) bei gleichbleibendem Digitalpegel hin überprüft; hier zeigte sich eine Abweichung des linken Surroundlautsprechers um 1,5 dB(A) nach oben, welche jedoch für die einfache Nachhallzeitmessung nicht weiter korrigiert wurde. Das Datenblatt der genutzten Genelec 8040B Studiomonitore gibt eine maximale Langzeit-RMS-Schallleistung von mehr als 99 dB SPL (unbewertet) im Abstand von 1 m an, sodass eine Nutzung der Studiomonitore in einem Abstand von 2 m am Abhörpunkt mit dem eingestellten Referenzabhörpegel innerhalb des Arbeitsbereichs des Lautsprechers stattfindet und folglich keine Begrenzungs- oder Verzerrungseffekte zu erwarten sind. Neben dieser Überprüfung des Referenzabhörpegels für die genutzten Lautsprecher wurde keine Entzerrung angewandt, um das Verhalten der Lautsprecher im Raum zu kompensieren bzw. zu korrigieren; dies steht nicht im Fokus dieser Arbeit und ist für eine authentische Auralisation virtueller Lautsprecher, die ein möglicherweise nicht perfektes Verhalten der Letzteren im Bezug auf die Abhörumgebung zeigen, auch nicht notwendig.

Für die gemittelte Messung der Nachhallzeit wurde ein logarithmischer Sinus-Sweep von 20 Hz bis 21 kHz, 2^{20} Samples bzw. 21,8 s Sweeplänge mit 21,8 s Pause nach dem Sweep und somit 43,7 s Signallänge verwendet. Die Pause nach dem Sweep ist dabei hinsichtlich der zu erwartenden Nachhallzeit des Raumes überdimensioniert lange gewählt. Es wird folglich sicher erreicht, dass jede einzelne Frequenz des Sweeps bis zum Ende des Erfassungszeitraums

in den Störsignalteppich hinein abgeklungen ist. Ein logarithmischer Sweep (der auch als exponentieller Sweep bezeichnet wird) bietet gegenüber einem linearen Sweep ein besseres SNR im tieffrequenten Bereich; er benötigt für jede Oktave die gleiche Zeit, was bedeutet, dass die Anregung der tieferen Frequenzen mehr Zeit beansprucht und somit mehr Energie in diesem Bereich aufbringt. Dies ist insbesondere bei der Messung von Raumimpulsantworten wichtig, da Störschall bzw. Störenergie zumeist im tieffrequenten Bereich durch Umgebungslärm (aufgrund unzureichender akustischer Trennung/Entkopplung der Räume) und/oder Einflüsse von elektrischen Geräten im Dauerbetrieb (z.B. Klimaanlage) auftreten. Durch eine Verlängerung des Sweeps und damit Vergrößerung der abgestrahlten Energie über alle Frequenzen hinweg kann der SNR weiter verbessert werden. Da wie erwähnt besonders im tieffrequenten Bereich unzureichende Anregungsenergien für eine fehlerfreie Messung auftreten, kann auch eine Färbung des Anregungssignals in Form eines Anhebung (*Pre-Emphasis*) der tiefen Frequenzen durchgeführt werden. Bei Verdopplung der Sweep-Länge erhöht sich der SNR in der Theorie um 3 dB. (Müller et al., 2001)

Der SNR der Messungen für die Bestimmung der Nachhallzeit liegt im Bereich von 50 dB (200 Hz) bis 75 dB (1 kHz und höher). Unter 200 Hz sinkt das SNR weiter stetig und erreicht bei 100 Hz lediglich noch einen Wert von etwa 40 dB. In diesem kontinuierlichen Anstieg des SNR hin zu den tiefen Frequenzen zeigt sich die beschränkte abgestrahlte Energie der genutzten Lautsprecher im tieffrequenten Bereich. Dies wird des Weiteren verstärkt, da den Lautsprechern eine starke Bassabsenkung über die Anpassungsmöglichkeiten, welche die Lautsprecher integriert bieten, *aufgeprägt* wird. Inwiefern diese Bassabsenkung raumakustisch ausreichend motiviert ist, soll im Rahmen dieser Arbeit nicht geklärt werden. Jedoch fällt während der durchgeführten Messungen auf, dass der Raum ein deutliches modales Verhalten und langes Ausschwingverhalten im tieffrequenten Bereich aufweist. Neben diesen raumakustischen Auffälligkeiten ist auch eine deutliche Körperschallanregung der im Raum genutzten Lautsprecheraufhängung bei Anregung im tieffrequenten Bereich wahrnehmbar. Vor allem der genannte Bereich um 100 Hz ist in den Messdaten auffällig (Lautsprecher L, R und C) und zeigt ein Abklingverhalten, das bis zu 0,8 s reicht. Abschließend kann jedoch gesagt werden, dass mit den durchgeführten Messdaten und deren erreichten SNR eine ausreichende Abschätzung der Nachhallzeit gegeben werden kann.

Für die in dieser Arbeit betrachteten Räume wie Tonstudioregien oder Mischkinos sind arithmetische Mittelwerte der Nachhallzeiten in den Terzbändern von 200 Hz bis 4 kHz gemäß EBU Tech. 3276 und SSF-01.1 bzw. nach Dolby- oder THX-Vorgabe von 0,2 bis 0,4 Sekunden zu erwarten (Weinzierl, 2008); so soll dies auch für den beispielhaft betrachteten Raum gelten. Eine ausreichend genaue Auswertung der fünf gemessenen Impulsantworten hinsichtlich der Nachhallzeit des Raumes stellt heraus, dass diese Vorgabe - unter Beachtung möglicher Abweichungen aufgrund der nicht normgerechten Messungen - prinzipiell erfüllt wird: Die gemittelte Nachhallzeit (nach *T30*-Estimation) über alle genannten Terzbänder führt zu 0,2 s. Frequenzunspezifisch wird - ebenfalls nach *T30*-Estimation - ein gemittelter Wert von 0,25 s festgestellt.

Für die Messung der BRIR-Datensätze wurde ebenfalls ein logarithmischer Sinus-Sweep von 20 Hz bis 21 kHz verwendet, der jedoch aus praktischen Gründen der Reduzierung des Zeitauf-

wands für die komplette Messreihe (1,8 ° horizontale Raster-Auflösung) lediglich eine Dauer von 2^{18} Samples bzw. 5,46 s Sweeplänge mit 5,46 s Pause nach dem Sweep und somit 10,92 s Signallänge hatte. Dies entspricht einer Sweeplänge, die für BRIR-Datensatzmessungen zur Darstellung virtueller Lautsprecher in ähnlich dimensionierten Räume in wissenschaftlicher Literatur gefunden wird (C. Pike & Romanov, 2017b)(Satongar et al., n. d. b)(Melchior et al., n. d.). Betrachtet man die Theorie zur Messung mit Sweeps, so erhöht sich bei jeder Verdopplung der Messdauer der SNR um 3 dB (Farina, 2000). Somit wären im Vergleich zur Messung der Raumimpulsantworten mit dem Messmikrofon SNRs von 44 dB (200 Hz) bis 69 dB (1 kHz und höher) zu erwarten; dabei sind jedoch negative Einflüsse des Kunstkopfmikrofons gegenüber dem kalibrierten Messmikrofon mit optimalen Eigenrauschwerten nicht beachtet. Jedoch können in den gemessenen BRIRs SNRs von 45 dB (200 Hz) bis 64 dB (1,6 kHz) gemessen werden; über 1,6 kHz sinkt der SNR jedoch wieder leicht und erreicht Werte von beispielsweise 58 dB bei 11 kHz. Bei 100 Hz und 150 Hz werden jedoch nur knapp 30 dB erreicht. Die theoretische Abschätzung des erreichbaren SNR mithilfe der erstgetätigten Messreihe stimmt weitestgehend mit den ermittelten Werten aus den Messdaten überein. Diese Werte erreichen trotz der Wahl der gleichen Messmethode mit logarithmischen Sinus-Sweeps derselben Länge nicht die SNRs, welche in ähnlichen Systemen aus der Literatur erreicht werden, die Werte von etwa 60 bis 85 dB im Frequenzbereich von 200 Hz bis 18 kHz (**ïż£Pike2017**), etwa 65 bis 80 dB (**ïż£Erbesa**) oder gar von einem Peak-to-Tail-Rauschabstand von ca. 90 dB (Melchior et al., n. d.) nennen. Erbes et al. (2015) nennt des Weiteren hierbei die Nutzung von linearen Sweeps mit Bassbetonung (*Pre-Emphasis*) von 2^{18} und 2^{17} Samples Länge, die anderen Quellen einen logarithmischer Sinus-Sweep der Länge 2^{18}.

Es zeigt sich bei Analyse der Messdaten jedoch, dass ein SNR von etwa 60 dB über dem weiten (und wichtigen) Frequenzbereich ab 800 Hz und höher erreicht wird. Somit wird eine Kürzung der BRIRs nach dem Abklingen des Raumes bis zur Nachhallzeit *RT60* befürwortet, sodass der Übergang in den recht hohen Rauschteppich nicht negativ zum Höreindruck beiträgt. Die damit erreichte Dynamik von 60 dB zeit sich in Hahne et al. (2019) als kritische Schwelle und sollte somit (gerade noch) einen natürlichen Höreindruck des Ausklingverhaltens des Raumes ermöglichen. Hierbei ist jedoch zu beachten, dass das lange Ausschwingverhalten des Beispielraumes im Tieftonbereich, welches ebenfalls charakteristisch für den Raum ist, so nicht (vollständig) abgebildet wird. Die damit verbundene kurze Form der BRIRs ist jedoch nützlich unter Beachtung der im weiteren Verlauf (Abschnitt 5.6) diskutierten Signalverarbeitungseffizienz des Systems. Trotzdem sollte das Messsystem und der Raum im Bezug auf die erreichten Ergebnisse dieser Beispielmessung nochmals geprüft werden, um eine Abschätzung der störenden (vor allem tieffrequenten) Einflüsse des Raumes auf die Leistungsfähigkeit der Messroutine zu ermöglichen; hierbei wird auch eine Durchführung der Messroutine in einem anderen Raum zu Vergleichszwecken empfohlen.

5.5 Systemmodul 2: Postprocessing der BRIRs

In diesem Kapitel wird das Systemmodul 2 entworfen und umgesetzt, welches die Postprocessing-Schritte umsetzt, die eine einwandfreie Verbindung der Messumgebung (siehe Abschnitt 5.4)

mit der flexiblen Auralisationsumgebung (siehe Abschnitt 5.6) ermöglichen. Da diese Verarbeitungsschritte niemals vollständig getrennt von den Systemmodulen davor und danach betrachtet werden können, ist es nötig einige Schlüsselelemente der flexiblen Auralisationsumgebung in diesem Kapitel vorwegzunehmen. Für eine nähere Betrachtung dieser Elemente sei auf Abschnitt 5.6 verwiesen.

5.5.1 Anforderungsanalyse

Die Anforderungsanalyse des *Systemmodul 2: Postprocessing der BRIRs* liefert wenige, jedoch harte und funktionale Anforderungen: Dabei ist vor allem die einwandfreie Weitergabe der Daten aus der Messumgebung (Systemmodul 1) an die Renderingumgebung bzw. flexible Auralisationsumgebung (Systemmodul 3) zu erwähnen; diese Anbindung sollte den Anforderungen der flexiblen Auralisationsumgebung genügen. Aufgrund der Nutzung dieses Systemmoduls als eine Art *Verbindungsmodul* sind keine direkten Benutzeranforderungen zu definieren, vielmehr ergeben sich diese Anforderungen an das Systemmodul 2 aus den Anforderungen der Systemmodule 1 und 3.

Folgende Schritte sind notwendig, um alle Anforderungen einer einwandfreien Verbindung zu erfüllen:

- Interfacing der Messdaten aus Systemmodul 1 und Bildung von BRIR-Datensätzen für jeden Lautsprecher
- Normalisieren aller Messdaten unter Bestimmung eines Normalisierungsfaktors
- Verkürzung aller Messdaten unter Zuhilfenahme einer *Onset-Protection* und raumakustischer Parameter wie *RT60* (oder *EDT*)
- Zweiteilige Speicherung (dynamisch/statisch) der BRIR-Datensätze unter Nutzung der *SimpleFreeFieldHRIR-SOFA-Convention* und *wav*-Files

Der erste Punkt der vorangegangenen Aufzählung ergibt sich aus der Tatsache, dass die unbearbeiteten Messdaten aus der Messumgebung noch nicht als BRIR-Datensatz vorliegen, sondern in einzelnen zwei-kanaligen Impulsantworten für jede einzelne Messung, die mit Hilfe des Benennungsschemas *lautsprecher-nr_azimut* benannt sind. Diese Impulsantworten müssen matriziert und damit zu BRIR-Datensätzen zusammengefasst werden. Die Messdaten einer Messreihe sind nicht normalisiert und erreichen damit aufgrund der in der Regel konservativen Aussteuerung innerhalb der Messumgebung, in der auf eine Aussteuerungsreserve für unvorhersehbare raumakustisch oder aufgrund des Einflusses der kopfbezogenen Übertragung bedingte Überhöhungen geachtet wird, noch nicht den erreichbaren Maximalpegel. Um dies für die weitere Verarbeitung der Daten in den Faltungsalgorithmen sicherzustellen, soll ein Normalisierungsfaktor bestimmt werden. Dieser Normalisierungsfaktor soll dabei aus allen Messungen einer Messreihe bestimmt werden und gleichförmig auf alle Messdaten angewendet werden, sodass eine gleichförmige Verstärkung der Messreihe erreicht wird. Dies ist wichtig, um keine Abweichungen im Pegel zwischen einzelnen Lautsprechern bei der anschließenden Auralisation zu erhalten. Des Weiteren sind die Impulsantworten noch nicht optimal

gekürzt, was jedoch für die spätere Verarbeitung in der flexiblen Auralisationsumgebung negative Auswirkungen hat. Zum einen kommt es aufgrund der Samples in der Impulsantwort vor dem eigentlichen Einschwingen des Lautsprechers zu einer Latenz in der digitalen Audiosignalverarbeitung bei der Faltung. Dies soll verhindert werden, indem mithilfe einer auf Grundlage der Maximalwertfindung basierenden - nennen wir es - *Onset-Protection* auditiv nicht benötige Samples vor der eigentlichen Impulsantwort entfernt werden; dabei werden sowohl Samples entfernt, die aufgrund nicht kompensierter Latenz in der Messumgebung als auch aufgrund von Laufzeiten durch die direkte Wegstrecke zwischen Lautsprecher und Mikrofon in den Impulsantworten zu finden sind. An dieser Stelle soll nicht weiter auf eine Trennung der Ursachen für diese vorhandenen führenden Samples eingegangen werden, sondern unter Aspekten der Latenzminimierung der späteren Audiosignalverarbeitung nicht benötigte führende Samples entfernt werden.

Die rechtsseitige Kürzung der Impulsantworten soll divers umgesetzt werden können: So soll es möglich sein, die gewünschte Zeitspanne vom Beginn der vorliegenden Daten bis zur Kürzung der Impulsantwort manuell einzugeben und somit Parameter, die aus vorangegangenen Untersuchungen bestimmt worden sind, zu nutzen. Des Weiteren soll es jedoch auch ermöglicht werden, mithilfe einer integrieren Funktionsabfrage eine EDC zu generieren und daraus gewonnene raumakustische Parameter für die weitere Verarbeitung der Impulsantworten zu nutzen. Diese Analyse ist abhängig von der Qualität der importierten Impulsantwort und sollte folglich immer unter Betrachtung der Letzteren analysiert und eingeschätzt werden. So kann an dieser Stelle (analog zu Absatz 5.4.2.2.2) eine mit einem omnidirektionalen Messmikrofon gewonnene Raumimpulsantwort genutzt werden, welche bestenfalls nach dem ISO-Standard raumakustischer Messungen mithilfe einer omnidirektionalen Schallquelle (Dodekaeder) gemessen wurde und diesem ISO-Standard vollständig genügt oder aber ein Signal des Kunstkopfmikrofons kann nach diesen Parametern analysiert werden; die Nutzung einer Raumipulsantwort nach (quasi) ISO-Standard ist als sinnvoller zu erachten und sollte der Nutzung der Mikrofonsignale des Kunstkopfes, die eine ausgeprägte Richtwirkung haben, vorgezogen werden. Da die in Abschnitt 5.6 beschriebene flexible Auralisationsumgebung eine dynamische sowie statische Verarbeitung der BRIRs durch Trennung der Anteile in Direktschall/frühe Reflexionen sowie späte Reflexionen/diffuser Nachall auf Grundlage einer Mixing Time vorsieht, ist eine Trennung und Speicherung in unterschiedlichen Datenformaten (*SimpleFreeFieldHRIR*-SOFA-Convention und *wav*-File) nötig; die Nutzung der *SimpleFreeFieldHRIR*-SOFA-Convention sowie *wav*-Files begründet sich hierbei wiederum vor allem durch die nötige Anbindung an das nachfolgende Systemmodul 3, die flexible Auralisationsumgebung.

5.5.2 Umsetzung

Für die Umsetzung des Systemmoduls 2 wird *MATLAB* genutzt, eine Software des US-amerikanischen Unternehmens *MathWorks* zur Lösung mathematischer Probleme sowie zur grafischen Darstellung der Ergebnisse. Dabei zeichnet sich *MATLAB* vorrangig durch die Möglichkeiten der numerischen Berechnung mithilfe von Matrizen aus, so wird es zur numerische Simulation sowie Datenerfassung, Datenanalyse und -auswertung eingesetzt.

Somit bietet sich MATLAB für die Matrizierung der gemessenen BRIRs, sowie für die alle Verarbeitungsschritte des Postprocessings, welche über eine große Datenmenge in den entstandenen Matrizen bzw. Arrays ausgeführt werden müssen, an. Des Weiteren bietet die sogenannten *SOFA API* (siehe Absatz 5.5.2.3.1), die aufgrund des hinter *SOFA*-Files liegenden Containers nativ in MATLAB unterstützt wird, alle benötigten Werkzeuge, um die Anbindung der BRIRs an das Systemmodul 3 der flexiblen Auralisationsumgebung zu gewährleisten.

In MATLAB wird in einer proprietären Programmiersprache programmiert; kleinere Programme können als sogenannte Skripte oder Funktionen zu geschlossenen Einheiten verpackt werden. Die im Folgenden beschriebenen Verarbeitungsschritte sind somit auch in Form von MATLAB-Skripten umgesetzt, welche im Anhang dieser Arbeit vollständig aufgeführt werden. Da die Programmierung der Verarbeitungsschritte auf den mehrdimensionalen Matrizen spezifische und zugleich grundlegende Schritte in Form von Matrixoperationen bzw. Indizierungen auf diesen Matrizen darstellen, soll im Folgenden nicht auf die Implementierung eingegangen werden, sondern vielmehr die dahinter liegenden Ziele dieser Verarbeitungsschritte beschrieben werden; die konkrete Implementierung ist des Weiteren im Anhang einzusehen.

Da das Systemmodul 2 eine Art *Zwischenmodul* darstellt, das - wie bereits erwähnt - keine konkreten Benutzeranforderungen stellt, besitzt es keine graphische Benutzeroberfläche und ist folglich nicht sonderlich *anwenderfreundlich*. Somit sollte es im Gegensatz zum Systemmodul 1 und vor allem Systemmodul 3 nur mit dem nötigen Fachwissen angewendet werden. Optionen der Verarbeitung ergeben sich nur durch Anpassung von Variablen in den Skripten, was direktes Programmieren in den Letzteren voraussetzt.

5.5.2.1 Interface aus Messumgebung, Normalisierungsfaktor und Position des Samples mit absolutem maximalem Wert

In der automatisierten Messumgebung, welche mit Hilfe von Max/MSP und der Spat-5-Library umgesetzt wurde, werden die BRIRs mit Hilfe des Benennungsschemas *lautsprechernr_azimut* für jeden gemessenen Lautsprecher der Messreihe und für jede gemessene horizontale Orientierung des Kunstkopfes, die durch den Azimutwinkel entlang des rechtsdrehenden Kugelkoordinatensystems (oder auch Polarkoordinatensystems) charakterisiert wird, gemessen. Diese BRIRs liegen als zweikanalige *wav*-Files vor. Das MATLAB-Skript mit dem Namen *twospeakers_load_normalisationfactor_maxposition.m* ist das erste Skript, welches im Zuge des Postprocessings ausgeführt werden muss:

Bevor die BRIRs übergeben bzw. eingeladen werden, sollen in MATLAB zunächst Metadaten initialisiert werden, welche grundlegend für die Messung und auch spätere Auralisation sind: Zum einen wird mit `complete_horizontal_grid` der genutzte Kreis bzw. Kreisbogen in Grad angegeben, welcher den Bereich der Messung - startend von der Orientierung mit Azimutwinkel 0° - aufspannt. Hierbei ist wichtig, dass Messungen mit dem Messsystem immer beim Azimutwinkel 0° starten und dann schrittweise höhere Azimutwinkel bis zu (im Falle einer Schrittweite von 1,8°) maximal 358,2° erreichen. Dies ist zweckdienlich, da so eine

gleichmäßige Auflösung der BRIR-Daten erreicht wird - BRIR-Datensätze aus aktueller wissenschaftlicher Literatur bieten ebenfalls eine volle horizontale Kreisauflösung. Somit ist das Messsystem an dieser Stelle eingeschränkt, sodass spezielle Messungen von Kreisausschnitten oder andere Startpositionen als die die frontalen Kopforientierung nicht berücksichtigt werden. Dies muss auch bei der Bearbeitung der BRIR-Daten in MATLAB beachtet werden, da die Verarbeitung hier auf einen vollen kreisförmigen Datensatz angepasst ist, der jedoch in seiner Raster-Auflösung bzw. Schrittweite prinzipiell flexibel ist. Diese Schrittweite (`stepsize`) ist ein weiterer Parameter, welcher an dieser Stelle initialisiert wird; aus den beiden Parametern des `complete_horizontal_grid` und der `stepsize` lässt sich dann die Anzahl der BRIRs `number_of_BRIRs` durch eine einfache Division berechnen. Ebenfalls wird die Abtastrate der Messdaten `FS` initialisiert. Auch wenn das Skript an dieser Stelle keine Benutzerabfragen bietet, ist es für die einwandfreie Funktion der weiteren Postprocessing-Schritte notwendig, dass diese Angaben mit den Metadaten der genutzten Messdaten übereinstimmen.

Das Einladen der BRIRs, die als zweikanalige *wav*-Files vorliegen, findet automatisiert statt, indem alle *wav*-Files im Pfad, in dem das ausgeführte Skript liegt, mit Hilfe der `audioread()`-Funktion sortiert eingeladen werden. Dabei wird die Sortierung mithilfe des `natsortfiles()`-Funktion[4] von Stephen Cobeldick durchgeführt; diese führt keine naive Sortierung in natürlicher Reihenfolge durch, sondern sortiert die Dateinamen und Dateierweiterungen getrennt, um eine Wörterbuchsortierung zu gewährleisten, bei der kürzere Dateinamen immer vor längeren sortiert werden. Ebenso werden Dateipfade an jedem Dateitrennzeichen aufgeteilt, und jede Ebene der Dateihierarchie wird separat sortiert. Somit kann bei Nutzung des gewählten Benennungsschemas der BRIRs (*lautsprecher-nr_azimut*) die richtige rechtsdrehende Sortierung der Daten anhand des Azimutwinkels gewährleistet werden. Die auf diese Weise in der richtigen Reihenfolge geladenen Daten werden in eine dreidimensionale Matrix mit der Dimension *BRIR-Länge x Anzahl Messpunkte x 2* geschrieben, welche `all_irs_measured_X` genannt wird, wobei das letztgenannte `X` in diesem Fall für die Nummer des Lautsprechers steht. Dieses Unterscheidungsschema für unterschiedliche Lautsprecher wird innerhalb der Skripte für alle Matrizen und Variablen verfolgt; ein deutlich besserer Programmierstil als die Nutzung einer laufende Nummerierung im Namen wäre eine entsprechende Verteilung der Daten in einem *Struct* oder *Cell* und anschließendem Zugriff auf die Letzteren via Indizierung. Dies ist weiterhin umsetzbar und sollte - besonders bei Erweiterung des Systems hin zu einer deutlich größeren Anzahl an Lautsprechern - genutzt werden. An dieser Stelle soll auch noch eine Anmerkung zur Nutzung des Skriptes mit Messreihen für mehrere Lautsprecher gegeben werden: Das Einladen der BRIRs ist in der momentanen Implementierung problematisch, da alle *wav*-Files, welche im Pfad des Skriptes liegen eingeladen werden; somit findet keine automatische Trennung von Files mehrerer Lautsprecher statt. Es ist somit unbedingt darauf zu achten, die beiden Sektionen des Einladens abschnittsweise für jeden Lautsprecher auszuführen und zwischen diesen beiden Ausführungen die Messdaten im Pfadordner anzupassen. Dies ist absolut nicht anwenderfreundlich und bietet großes Fehlerpotential, konnte jedoch bis Ende dieser Arbeit nicht gelöst werden. Eine nähere Betrachtung der einlesenden Pfadstruktur und ein Anlegen von einem eindeutig benannten Ordner für die Messdaten eines

[4] http://de.mathworks.com/matlabcentral/fileexchange/47434-natural-order-filename-sort

jeden Lautsprechers sollten die Problematik jedoch beheben. Zwei Fehlerabfragen überprüfen im Verlauf des Importprozesses die Anzahl der eingeladenen BRIRs und deren Übereinstimmung mit der Angabe in den Metadaten (`number_of_BRIRs`); dieser Abgleich finden auch hinsichtlich der Abtastrate `FS` statt.

Nach dem vollständigen Einladen der BRIR-Daten in die Matrizen wird ein Normalisierungsfaktor aus den Letzteren bestimmt; dies wird umgesetzt, indem der betragsmäßig maximale Wert (`max_X`) aller vorhandenen Samples gesucht wird. Dabei muss der Absolutbetrag genutzt werden, da die BRIRs sowohl im ersten als auch vierten Quadranten Werte annehmen. Ist dieser Wert gefunden, wird des Weiteren auch das Sample innerhalb der Matrix aller BRIRs eines Lautsprechers bestimmt, welches diesen betragsmäßig maximalen Wert annimmt (`maxsample_X`). Mithilfe dieser Information wird die `max_sampleposition_X` bestimmt, welche die Position des gefundenen betragsmäßig maximalen Samples innerhalb der einzelnen BRIR beschreibt, beginnend vom Beginn der BRIR; dies wird unter Zuhilfenahme des Wissen über die Länge einer einzelnen BRIR umgesetzt. Dieser Wert wird im darauffolgenden Skript (*twospeakers_normalize_truncate.m*) genutzt, um die linksseitige Kürzung der BRIRs umzusetzen. Der Normalisierungsfaktor `normalisationfactor_X` ist im Weiteren bei einer Normalisierung auf 0 dBFS der Quotient aus 1 und dem gefundenen betragsmäßig maximalen Wert eines Samples `max_1` und wird so im Skript bestimmt.

Neben einer abschließenden Fehlerabschätzung der vorangegangenen Prozesse, die einen Fehlermeldung ausgeben, wenn die genannten Werte für das beispielhaft für zwei Lautsprecher durchgeführte Skript sich in abgeschätzten Grenzen zueinander zu stark unterscheiden, findet ein Vergleich des `normalisationfactor` für beide Lautsprecher (`normalisationfactor_1` und `normalisationfactor_2`) statt, der den kleineren Normalisierungsfaktor für die im nächsten Skript durchgeführte Normalisierung wählt. Dies ist insofern umzusetzen, damit es bei der gleichmäßigen Normalisierung aller Messdaten nicht zu einer Erhöhung des maximalen Samplewertes auf über 1 und somit zu *Clipping* kommt. Eine getrennte Normalisierung der BRIR-Datensätze ist nicht zielführend, wenn das Lautsprechersetup in der Auralisation die gleichen akustischen Eigenschaften wie in den Messungen der Messreihe abbilden soll. Des Weiteren werden zum Schluss noch die berechneten `max_sampleposition_1` und `max_sampleposition_2` miteinander verglichen und es wird der kleinere der beiden Werte als `max_sampleposition` gesetzt. Dies garantiert, dass bei der folgenden linksseitigen Kürzung der BRIRs mit Hilfe diesen Wertes nicht zu viel gekürzt und damit möglicherweise das Einschwingverhalten verändert wird.

5.5.2.2 Normalisierung und links- sowie rechtsseitige Kürzung

Im darauffolgend auszuführenden MATLAB-Skript *twospeakers_normalize_truncate.m* wird die Normalisierung anhand des bestimmten Normalisierungfaktors `normalisationfactor` durchgeführt. Des Weiteren findet die links- und rechtsseitige Kürzung der BRIRs statt. Für die linksseitige Kürzung wird eine *Onset-Protection-Zeit* (`onsetprotectionseconds`) initialisiert, welche einen Zeitbereich vor (also linksseitig) dem über alle BRIRs gefundenen

betragsmäßigen Maximum aufspannt; dieser Zeitbereich soll das Einschwingverhalten bewahren. Wird dieser Bereich linksseitig der BRIRs überschritten, so werden die BRIRs an der gewählten Position abgeschnitten. Dieses Vorgehen ermöglicht eine kontrollierte linksseitige Kürzung der BRIRs ohne Sichtkontrolle über alle BRIRs hinweg. Der Wert der *Onset-Protection-Zeit* (`onsetprotectionseconds`) muss aus Erfahrung gewählt werden sollte einen Wert von 10 ms nicht unterschreiten. Die `headcutsamples` bestimmen sich aus der Differenz der `max_sampleposition` und der gewählten `onsetprotectionsamples` und entsprechen der Anzahl an Samples, welche am Anfang der BRIRs gekürzt werden. Sollten vor der Position des betragsmäßig maximalen Werts linksseitig weniger Samples als die gewünschten `onsetprotectionsamples` zur Verfügung stehen, so findet keine linksseitige Kürzung statt. Die linksseitige Kürzung wird direkt auf den Matrizen `all_irs_measured_X` umgesetzt. Die auf diese Weise *simpel durchgeführte Onset-Protection* geht von der Annahme aus, dass in den genutzten BRIR-Daten der Direktschallanteil zu Beginn der Impulsantwort den betragsmäßig maximalen Wert `max_X` in der kompletten Impulsantwort annimmt; dies ist für die Räume (Tonstudioregien, Mischkinos), die mit diesem System simuliert werden sollen, und Messpositionen, welche sich im Sweet-Spot bzw. der Sweet-Area vielkanaliger Lautsprechersetups befinden, eine plausible Annahme, da keine frühen Reflexionen zu erwarten sind, die stärker als der Direktschall aus der Impulsantwort heraustreten.

Die Normalisierung wird durchgeführt, indem die Matrizen aller Lautsprecher, die alle BRIRs des jeweiligen Lautsprechers beinhalten und `all_irs_measured_X` genannt werden, mit dem selben `normalisationfactor` multipliziert werden. Neben der linksseitigen Kürzung der BRIRs wird auch eine rechtsseitige Kürzung durchgeführt. Diese begründet sich auf der Angabe bzw. Bestimmung der *RT60* oder *EDT*; die Letztere wird durch mehrere dem Skript beigefügte Funktionen ermöglicht, wobei zunächst die Erstellung einer *EDC* mithilfe der Funktion `getSchroeder` nötig ist, bevor über die Funktion `calcRTX` unter Angabe der entsprechenden Regressionsbereiche die Kriterien *EDT*, *T20*, *T30* bzw. auch direkt *T60* (Regressionsbereich von 0 bis -60 dB) bestimmt werden können. Dabei werden Funktionen der bereits beschriebenen *ScanIR*-Mess- und Analyseumgebung genutzt (Vanasse et al., n. d.), die lizenzrechtlich für diese nicht kommerzielle Nutzung zur Verfügung stehen. Diese Funktionen sind im dem Anhang beigefügten Skript auskommentiert, da beispielhaft die direkte Eingabe der *RT60* betrachtet wird. Des Weiteren wird im Verlauf des Skriptes nur die *RT60* für die rechtsseitige Kürzung in Betracht gezogen und alle Teile des Codes, die sich auf die *EDT* beziehen auskommentiert. Auch wenn diese raumakustischen Parameter direkt eingegeben werden können bzw. deren Werte aus der Skript-internen Bestimmung überprüft und jederzeit angepasst werden können, wird eine Art *Sicherheitsparameter* eingeführt, der `endtruncationsafetyseconds` bzw. `endtruncationsafetysamples` genannt wird. Dieser führt unabhängig vom Kriterium und dessen Wert, welches für die rechtsseitige Verkürzung genutzt wird, zu einer Verschiebung der Letzteren, sodass die BRIRs länger werden und tatsächlich erst *nach* Erreichen von z.B. *RT60* gekürzt werden. Dies soll die Tatsache beachten, dass in kleinen, stark gedämpften Räumlichkeiten Bestimmungen der *RT60* aufgrund unzureichender Diffusität der späten Reflexionen bzw. starker erster Reflexionen sehr fehleranfällig sind. Die Parameter `IRendtruncationFromRT60` bzw. `IRendtruncationFromEDT` spiegeln die Werte für die im weiteren Verlauf der Bearbeitung tatsächlich durchgeführte

Verkürzung wieder. Eine Betrachtung oder gar automatische Bestimmung des SNR der BRIRs und eine daraus resultierende rechtsseitige Verkürzung ist nicht implementiert. Der SNR muss also unbedingt direkt während der Messung oder, was eine genauere Analyse ermöglicht, nach der Messung in einer anderen Umgebung als dieser geprüft werden; dies sollte vor allem auch deswegen getan werden, um eine Nutzung der raumakustischen Kriterien zu bewerten. An dieser Stelle bestehen also Möglichkeiten für Weiterentwicklungen des Skripts, hin zu einer SNR-Bestimmung und/oder -Betrachtung in der automatisierten Postprocessing-Umgebung.

Die bereits linksseitig gekürzten BRIR-Matrizen werden in einem letzten Verarbeitungsschritt des Skriptes rechtsseitig gekürzt, indem die Matrizen `all_irs_truncatedFromRT60_normalized_X` bzw. `all_irs_truncatedFromEDT_normalized_X` erstellt werden. Wie bereits vor der Bestimmung bzw. Betrachtung der raumakustischen Parameter wird des Weiteren erneut eine Trennung in Matrizen des linken und rechten Ohrs durchgeführt, diese werden `irs_leftear_truncatedFr` bzw. `irs_rightear_truncatedFromRT60_normalized_X` usw. genannt.

5.5.2.3 Speicherung und Benennung der BRIR-Datensätze

Ehe im weiteren Verlauf dieses Kapitels die konkrete Benennung der BRIR-Datensätze besprochen wird, was vor allem für die Anbindung an die zweigliedrige flexible Auralisierungsumgebung wichtig ist, soll zunächst das *SOFA (Spatially Oriented Format for Acoustics)* vorgestellt und ein Überblick über die Nutzung des Letzteren sowie den strukturellen Ablauf der Generierung und Speicherung eines *SOFA*-Files gegeben werden. Hierbei steht des Weiteren die Motivation zur Nutzung der *SimpleFreeFieldHRIR*-Convention mit BRIRs im Fokus; dieses Kapitel liefert somit wichtige Informationen über die Nutzung der BRIR-Daten bei der dynamischen Binauralsynthese, die mit dem *spat5.binaural* Objekt umgesetzt wird.

Die in diesem Kapitel erläuterten Vorgehensweisen werden in den Skripten *twospeakers_saveFullDynamic.* und *twospeakers_splittingBRIRs_saveDynamic+Static.m* umgesetzt und somit sind die Erläuterungen auf beide der genannten Skripte anwendbar. Das Skript *twospeakers_splittingBRIRs_saveDy* führt des Weiteren vor der Speicherung der BRIR-Daten noch eine Trennung der BRIRs zur Nutzung einer getrennt dynamischen und statischen Faltung wie sie Abschnitt 5.6 beschrieben ist durch; das Vorgehen dieser Trennung wird in Absatz 5.5.2.3.3 beschrieben.

5.5.2.3.1 SOFA (Spatially Oriented Format for Acoustics) SOFA (Spatially Oriented Format for Acoustics) ist ein Dateiformat, welches *räumlich orientierte akustische Daten* speichern kann und von der AES (Audio Engineering Society) in der AES69-2015 (SOFA 1.0) sowie AES69-2020 (SOFA 2.0) standardisiert wird (Audio Engineering Society, 2015)(Audio Engineering Society, 2020). Ziel dieser Standardisierung ist eine bestmögliche Austauschbarkeit dieser Daten zu erreichen, welche Inkompatibilität zwischen Systemen der Erfassung, Berechnung und des Renderings verhindert und des Weiteren eine problemlose Erweiterbarkeit des Dateiformats ermöglicht. Da die räumliche Darstellung von Audio weiterhin Gegenstand aktueller Forschung und Entwicklung ist, steht dieser Standard unter Revision und es ist am

6. Dezember 2020 der Standard AES69-2015 durch den neuen Standard AES69-2020 ersetzt worden; da dieser neue Standard jedoch zum Zeitpunkt des Beginns dieser Masterarbeit noch nicht zugänglich war, beziehen sich alle im Folgenden getätigten Aussagen auf den älteren Standard AES69-2015. Viele in der AES69-2020 nun standardisierten Dateiformate bestimmter gängiger Messumgebungen bzw. Datenstrukturen (sogenannte *SOFA-Conventions*) waren vor der Standardisierung zwar verfügbar, jedoch noch in einer Art Entwicklungs- bzw. Betaphase und wurden von Software, welche die Nutzung von SOFA prinzipiell fördert, noch nicht einheitlich unterstützt. Die im SOFA-File gespeicherten räumlich orientierten akustischen Daten können kopfbezogene Übertragungsfunktionen im Zeitbereich oder Frequenzbereich (HRIRs oder HRTFs) sein, aber auch binaurale Raumimpulsantworten (BRIRs) oder *räumliche* Raumimpulsantworten (SRIRs), welche z.B. mit einem Mikrofonarray aufgenommen wurden. Des Weiteren bietet der Standard auch Dateiformate an, die nicht direkt Inhalte zu räumlichem Audio tragen, so z.B. einen einfachen Datentyp wie eine endliche Impulsantwort (FIR) oder eine Übertragungsfunktion (TF) (*GeneralFIR* oder *GeneralTF*). Dies ist im Rahmen binauraler Wiedergabe vor allem deswegen interessant, da auch für Kopfhörer-Entzerrungsfilter ein eigenes Dateiformat, die *SimpleHeadphoneIR*-Convention existiert. Das einzige in der AES69-2015 *tatsächlich fertig standardisierte* Dateiformat für kopfbezogene Übertragung von räumlichem Audio ist die *SimpleFreeFieldHRIR*-Convention.

Folgende Anforderungen sollen mit dem AES69-Standard für die Speicherung und Darstellung der Daten erreicht werden (Audio Engineering Society, 2015):

- Beschreibung des Messaufbaus mit beliebiger Geometrie, d. h. nicht beschränkt auf Sonderfälle wie ein regelmäßiges Gitter oder einen konstanten Abstand.
- Selbstbeschreibende Daten mit einer konsistenten Definition, d. h. alle erforderlichen Informationen über den Messaufbau müssen als Metadaten in der Datei vorhanden sein.
- Flexibilität zur Beschreibung von Daten multipler Bedingungen (Zuhörer, Abstände usw.) in einer einzigen Datei.
- Verfügbarkeit als Binärdatei mit Datenkompression für effiziente Speicherung und Übertragung.
- Vordefinierte Beschreibungen für die gängigsten Messaufbauten, die als *Conventions* bezeichnet werden; Empfehlungen zur Benennung von Attributen, Variablen und Dimensionen werden in diesen *Conventions* gegeben.

Folgende generelle Spezifikationen sind bei der Nutzung des AES69-Standards zu beachten (Audio Engineering Society, 2015):

- Unterstützte Koordinatensysteme: Kartesisches oder Kugelkoordinatensystem (Rechtssysteme, mathematisch positiver Drehsinn)
- Objekte: Receiver (beliebiger akustischer Sensor, vorhanden in beliebiger positiver ganzzahliger Anzahl), Listener (soll eine logische Einheit aller Receiver darstellen, wobei sich die relativen Positionen der Receiver zum Listener nicht verändern; eine AES69-Datei darf nicht mehr als ein Listener-Objekt enthalten), Emitter (beliebige Schallquelle, vorhanden in beliebiger positiver ganzzahliger Anzahl), Source (soll eine logische Einheit

aller Emitter darstellen, wobei sich die relativen Positionen der Emitter zur Source nicht verändern; eine AES69-Datei darf nicht mehr als ein Source-Objekt enthalten), Room (soll das Volumen beziffern, das den Messaufbau umschließt; Freifeld-Messaufbau ist als Sonderfall zu betrachten)

- Positionen der Objekte und Relationen zwischen ihnen: Globales Koordinatensystem (muss die Position der Source- und Listenerobjekte innerhalb des Raums definieren; Source- und Listenerobjekte müssen sich das gleiche globale Koordinatensystem teilen, wobei der Ursprung des globalen Koordinatensystems beliebig gewählt werden kann), lokales Koordinatensystem (Source- und Listenerobjekte, die sich im globalen Koordinatensystem befinden, definieren das Source-bezogene bzw. Listener-bezogene lokale Koordinatensystem; Achsen dieser Koordinatensysteme können frei gewählt werden und müssen nicht parallel zum globalen Koordinatensystem liegen; die lokalen Koordinatensysteme müssen die Positionen von Emittern innerhalb des Sourceobjekts bzw. die Positionen von Receivern innerhalb des Listenerobjekts definieren, wobei sich alle Emitter resp. Receiver das gleiche lokale Source- resp. Listener-Koordinatensystem teilen; soll die Richtwirkung von Emittern und Receivern berücksichtigt werden, dann ist das Richtdiagramm eines Emitters in einem Emitter-bezogenen Koordinatensystem zu definieren bzw. das Richtdiagramm eines Receivers in einem Receiver-bezogenen Koordinatensystem, die Emitter- und Receiverobjekte, die sich in den Source-bezogenen und Listener-bezogenen Koordinatensystemen befinden, müssen wiederum diese Emitter- bzw. Receiver-bezogenen lokalen Koordinatensysteme definieren.)
- Orientierung/Lage bzw. *Ausrichtung* der Objekte: Die Orientierung/Lage von Source/Listener sowie Emitter/Receiver kann berücksichtigt werden, wobei die Angabe der Orientierung/Lage mit Hilfe der Drehung des jeweiligen lokalen Koordinatensystems im globalen Koordinatensystem erfolgt. Dazu werden zwei orthogonale Vektoren mit den Namen `View` und `Up` genutzt, welche die Ausrichtung des lokalen Koordinatensystems definieren. Der `View`-Vektor definiert dabei die Richtung der positiven x-Achse des jeweiligen lokalen Koordinatensystems, also den Einheitsvektor $u_x = (1 \quad 0 \quad 0)$ in kartesischen Koordinaten bzw. den Einheiheitsvektor $u_x = (0° \quad 0° \quad 1)$ in Kugelkoordinaten. Der `Up`-Vektor definiert wiederum die Richtung der positiven z-Achse des jeweiligen lokalen Koordinatensystems, also den Einheitsvektor $u_x = (0 \quad 0 \quad 1)$ in kartesischen Koordinaten bzw. den Einheiheitsvektor $u_x = (0° \quad 90° \quad 1)$ in Kugelkoordinaten. Dabei werden diese Vektoren im jeweils übergeordneten Koordinatensystem definiert, also im globalen Koordinatensystem für die lokalen Koordinatensysteme der Listener- und Source-Objekte, sowie in den Listener- bzw. Source-bezogenen Koordinatensystemen für die jeweiligen lokalen Receiver- und Emitter-bezogenen Koordinatensysteme. Soweit nicht anders definiert, stimmen die Orientierungen der lokalen Koordinatensysteme immer mit denen der übergeordneten Koordinatensysteme überein. An dieser Stelle sei des Weiteren zu erwähnen, dass in Kugelkoordinaten der `View`-Vektor den *Azimut- bzw. Elevationswinkel (Gier- bzw. Nickwinkel)* der Source- bzw. Listener-Orientierung beschreibt, sowie der `Up`-Vektor den Roll-Winkel.

Die SOFA-Conventions bauen auf dem numerischen Container netCDF auf, welcher eine

Reihe von Softwarebibliotheken und Datenformaten für die Erstellung, den Zugriff und die gemeinsame Nutzung wissenschaftlicher Daten unterstützt. Für SOFA bietet netCDF eine strukturierte Darstellung von multidimensionalen Daten und Metadaten. Die frei zugänglichen und nutzbaren Spezifikationen enthalten eine vollständige Definition, Beispiele für verschiedene Implementierungen, sowie Schnittstellen zur Anwendungsprogrammierung sind als vorkompilierte Bibliotheken für Programmiersprachen wie C++, Octave oder JAVA verfügbar; des Weiteren wird netCDF nativ in MATLAB unterstützt. Dieser numerische Container definiert schlussendlich den Binärstrom der Daten bei der Serialisierung bzw. Speicherung in einem einzigen File, dem SOFA-File mit der Dateiendung *.sofa*.

In Anlehnung an die netCDF-Terminologie definiert SOFA *Dimensionen* und speichert Daten in *Variablen* und *Attributen*. Dabei war es für die Entwicklung des SOFA essentiell, dass Daten aus diversen bisher verfügbaren bzw. aus der wissenschaftlichen Arbeit bekannten Datensätzen eineindeutig in die Conventions überführt werden konnten, so werden Empfehlungen zur Benennung von *Attributen*, *Variablen* und *Dimensionen* innerhalb einer netCDF-Datei als ebendiese Conventions bezeichnet. Eine *Variable* repräsentiert hierbei ein Array von Werten des selben netCDF-Datentyps (double oder character/string), wobei diese Variable entweder nulldimensional (Skalar), eindimensional oder mehrdimensional sein kann; dabei sollte sie einer der im nächsten Abschnitt erläuterten *AES-69-Dimensionen* entsprechen. Eine Variable kann zugehörige *Attribute* haben, die vom Datentyp string sind.

Dimension	Wert	Beschreibung
M	unbegrenzt	Anzahl der Messungen; muss eine ganze Zahl größer als Null sein.
R	unbegrenzt	Anzahl der Receiver; muss eine ganze Zahl größer als Null sein.
E	unbegrenzt	Anzahl der Emitter; muss eine ganze Zahl größer als Null sein.
N	unbegrenzt	Anzahl der Daten-Samples, die eine Messung beschreiben; muss eine ganze Zahl größer als Null sein.
S	unbegrenzt	Anzahl der Character in einem String; muss eine ganze Zahl größer als Null sein.
I	1	Einelementige Dimension, definiert einen skalaren Wert.
C	3	Koordinatentripel, immer dreiwertig; der gewählte Koordinatentyp definiert die Bedeutung der einzelnen Dimensionen.

Tabelle 5.1: Dimensionen nach AES69-2015 (Audio Engineering Society, 2015)

Die *AES-69-Dimensionen* sind in ihrer vordefinierten Form in Tabelle 5.1 dargestellt. Die Dimension von Variablen wird je nach Convention entsprechend dieser Dimensionen definiert. Dabei sind Angaben von Dimensionen, welche bei der Definition von Variablen klein geschrieben werden, hierbei bestimmend für alle weiteren Variablen, die die selbe Dimension benutzen. Die Interpretation der Dimension N hängt vom Datentyp ab und ist durch die Convention im entsprechenden Attribut zu definieren; so stellt sie für eine Impulsantwort die Anzahl der FIR-Filtertaps (Samples der IR) bzw. für eine Übertragungsfunktion die Anzahl der Frequenzbins dar.

Bei der Nutzung des SOFA unter Zuhilfenahme von Conventions, die für Daten aus Freifeldbedingungen definiert wurden (wie die *SimpleFreeFieldHRIR*-Convention), sollen vor den weiteren Betrachtungen ein paar Ausführungen zur praktischen Nutzung dieser Conventions gegeben werden: Der Listener sollte im Ursprung des globalen Koordinatensystems positioniert werden und dabei seine *Standardausrichtung* bzw. default-mäßige (ursprüngliche) Orientierung in Richtung der positiven x-Achse, d.h. in der horizontalen Ebene in Frontalrichtung haben. Unter der Annahme, dass sich ein Emitter in der Mitte der zugehörigen Source

befindet, spiegelt die Position/Orientierung der Source dann direkt die Beziehung zwischen ihr und dem Listener bzw. den jeweiligen Receivern wider, was die Richtungsinformationen beim Schalleinfall eindeutig kodiert.

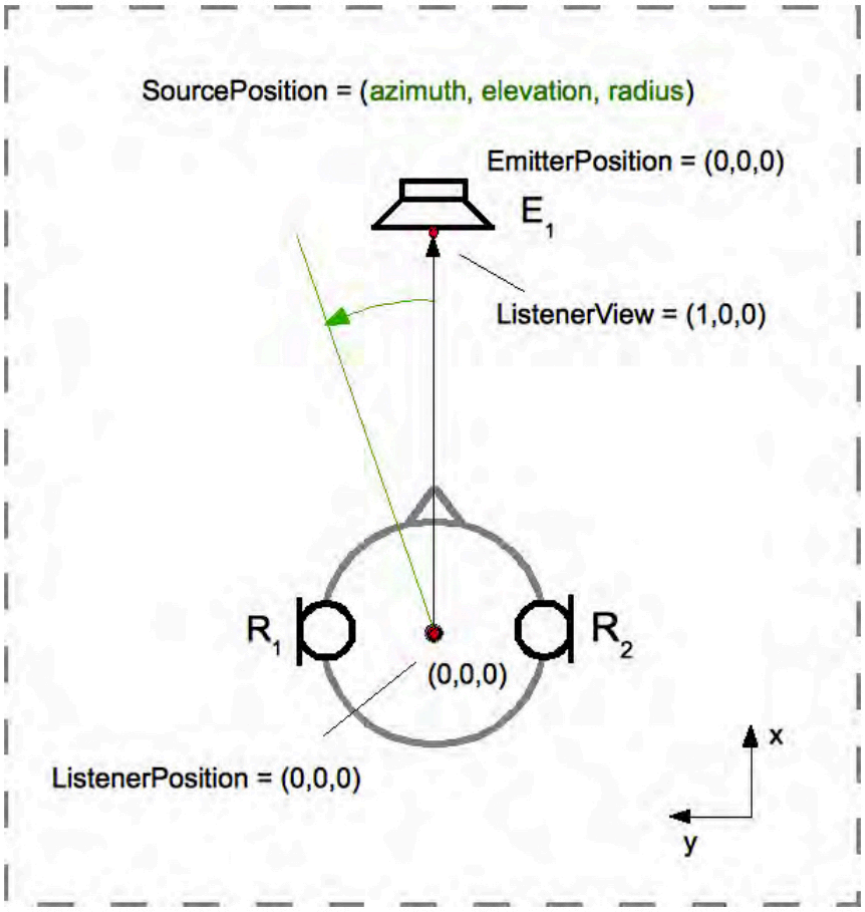

Abbildung 5.18: Beispiel eines HRIR-Messaufbaus im Freifeld mit *Emitter* E_1 und dem *Listener* mit seinen beiden *Receivern* R_1 *und* R_2 (Audio Engineering Society, 2015)

In Abbildung 5.18 ist schematisch ein Messaufbau für eine Messung eines HRIR-Datensatzes, welcher der *SimpleFreeFieldHRIR*-Convention genügt, dargestellt. Die relevanten positions- und lagebezogenen Daten und die zugehörigen Vektoren sind daraus ersichtlich, der Listener befindet sich im globalen Koordinatenursprung und somit gilt `ListenerPosition` = (0 0 0), des Weiteren ist seine ursprüngliche Orientierung entlang der positiven x-Achse gewählt, sodass gilt `ListenerView` = (1 0 0) - damit ist das Listener-bezogene lokale Koordinatensystem definiert; die beiden Receiver R_1 und R_2 befinden sich jeweils in einem festen Abstand zur `ListenerPosition`, welche in diesem Fall jeweils den Abstand des Trommelfells zum Kopfmittelpunkt beschreiben, der default-mäßig mit 9 cm gesetzt ist. Die in Abbildung 5.18 eingezeichnete `EmitterPosition` = (0 0 0) ist nicht im globalen Koordinatensystem angegeben, sondern im lokalen Koordinatensystem der Source, welche in dieser Abbildung nicht direkt eingezeichnet ist - jedoch ist in dieser Convention, wie im letzten Absatz beschrieben, davon auszugehen, dass sich der Emitter in der Mitte der Source befindet. Die `SourcePosition` lässt sich im globalen Koordinatensystem - welches in diesem Fall auch dem Listener-bezogenen lokalen Koordinatensystem entspricht - eindeutig mit Hilfe der

Kugelkoordinaten (Azimut, Elevation, Radius) beschreiben, sodass die räumlichen Beziehung zwischen Listener und der Source damit dargestellt ist.

Geht man von der Messung eines HRIR-Datensatzes im reflexionsarmen Raum aus, so wird dieser - je nach bestmöglicher Praktikabilität - zum Beispiel bei fester Positionierung des Listeners, welcher sich aus den zwei Receivern R_1 und R_2 in Form der Ohren bzw. Trommelfelle zusammensetzt und den auf einem gewählten Gitter positionierten idealerweise exakt punktförmigen Lautsprechern, den Emittern bzw. Sources, von den Positionen aus gemessen, die der Datensatz der kopfbezogenen Übertragung beinhalten soll; der Lautsprecheraufbau wird dabei aus praktischen Gründen zumeist gedreht, da ansonsten zu viele Lautsprecher nötig wären. Es ist jedoch auch möglich, einen solchen HRIR-Datensatz mithilfe eines fest positionierten Lautsprecheraufbaus, der die Messpunkte partiell abdeckt, und einem sich innerhalb dieses Aufbaus drehenden Listeners zu erreichen. Die Voraussetzungen für diese Äquivalenz sind die reflexionsarmen Bedingungen des Messraums. Unabhängig von der Messmethode besitzen diese beiden HRIR-Datensätze für jede HRIR-Messung eine relative Position und Orientierung des Listeners (inklusive seiner Receiver) zum Lautsprecher, dem Emitter/der Source, die üblicherweise in einem kopfbezogenen Kugelkoordinatensystem mit den bekannten Dimensionen *Azimutwinkel* ($\in [0, 360)$: horizontaler Winkel in Grad, gemessen gegen den Uhrzeigersinn von der positiven x-Achse), *Elevationwinkel* ($\in [-90, 90]$: vertikaler Winkel in Grad, gemessen von der x-y-Ebene, positive Winkel bei Position oberhalb der x-y-Ebene im positiven Halbraum) und *Distance* ($\in [0, \infty)$: Abstand in Metern vom Ursprung) angegeben werden. Die im vorangegangenen Abschnitt definierte Äquivalenz zwischen *Emitter* und *Source* ist weiterhin gültig. Eine beliebige Schallquellposition, die in einem binauralen Renderer verarbeitet wird, lässt sich so also eineindeutig mithilfe der genannten Positions- und Lageparameter zwischen Listener und Source in beide Richtungen beschreiben.

Die *SimpleFreeFieldHRIR*-Convention wurde aufgrund der verfügbaren Schnittstelle in den im Rahmen dieser Arbeit konzipierten und in Abschnitt 5.6 näher beschriebenem binauralen Renderer genutzt; dies stellt - wie der Name des Dateiformats, welches für die Nutzung von HRIR-Daten, die im reflexionsarmen Umfeld gemessen oder numerisch simuliert wurden, bereits aussagt - einen *Workaround* für die Nutzung von BRIRs dar, der kurz motiviert und dargestellt werden soll. Alle im AES-69-Standard in Form von Variablen und Attributen definierten Daten und Metadaten können im weiteren Verlauf nicht vollständig beschrieben werden, da deren Analyse den Rahmen dieser Arbeit übersteigen würde. Aus diesem Grund werden diejenigen Variablen und Attribute vorgestellt, welche in den im Zuge dieser Arbeit erstellten SOFA-Files genutzt werden und folglich relevante Daten bzw. Metadaten tragen.

In der folgenden Tabelle werden die Variablen und Attribute der *SimpleFreeFieldHRIR*-Convention dargestellt (Audio Engineering Society, 2015); dafür wird die englische Sprache genutzt, wie in der Convention vorgegeben (die Übersetzung ins Deutsche sollte selbsterklärend sein). Für die SOFA-Files, die im Rahmen dieser Arbeit generiert wurden, wurden nicht alle dieser Variablen und Attribute neu gesetzt, sondern es wurden auch Default-Werte übernommen. Um diesbezüglich einen kurzen Überblick zu erhalten, werden die gesetzte Variablen/Attribute in fetter schwarzer Schrift und jene, die im Default übernommen wurden,

in normaler Schrift dargestellt. Variablen/Attribute, die gar nicht genutzt wurden, sind kursiv.

Name	Default-Wert	Flags (r = nur lesen, m = verpflichtend)	Dimensionen	Datentyp
GLOBAL:Conventions	SOFA	rm		attribute
GLOBAL:Version	2.0	rm		attribute
GLOBAL:SOFAConventions	SimpleFreeFieldHRIR	rm		attribute
GLOBAL:SOFAConventionsVersion	1.0	rm		attribute
GLOBAL:APIName		**rm**		**attribute**
GLOBAL:APIVersion		**rm**		**attribute**
GLOBAL:ApplicationName				*attribute*
GLOBAL:ApplicationVersion				*attribute*
GLOBAL:AuthorContact		**m**		**attribute**
GLOBAL:Comment				**attribute**
GLOBAL:DataType	FIR	rm		attribute
GLOBAL:History				**attribute**
GLOBAL:License	No license provided, ask the author for permission.	m		attribute
GLOBAL:Organization		**m**		**attribute**
GLOBAL:References				*attribute*
GLOBAL:RoomType	**free field**	**m**		**attribute**
GLOBAL:Origin				*attribute*
GLOBAL:DateCreated		m		attribute
GLOBAL:DateModified		m		attribute
GLOBAL:Title		m		attribute
ListenerPosition	[0 0 0]	m	[IC, MC]	double
ListenerPosition:Type	cartesian	m		attribute
ListenerPosition:Units	metre	m		attribute
ReceiverPosition	[0 0.09 0; 0 -0.09 0]	m	[rCI, rCM]	double
ReceiverPosition:Type	cartesian	m		attribute
ReceiverPosition:Units	metre	m		attribute
SourcePosition	**[0 0 1]**	**m**	**[IC, MC]**	**double**
SourcePosition:Type	spherical	m		attribute
SourcePosition:Units	degree, degree, metre	m		attribute
EmitterPosition	[0 0 0]	m	[eCI, eCM]	double
EmitterPosition:Type	cartesian	m		attribute
EmitterPosition:Units	metre	m		attribute
GLOBAL:DatabaseName		**m**		**attribute**
GLOBAL:ListenerShortName		**m**		**attribute**
ListenerUp	[0 0 1]	m	[IC, MC]	double
ListenerView	[1 0 0]	m	[IC, MC]	double
ListenerView:Type	cartesian	m		attribute
ListenerView:Units	metre	m		attribute
Data.IR	**[0 0]**	**m**	**[mRn]**	**double**
Data.SamplingRate	**48000**	**m**	**[I]**	**double**
Data.SamplingRate:Units	hertz	m		attribute
Data.Delay	**[0 0]**	**m**	**[IR, MR]**	**double**
SourceUp	[0 0 1]		[IC, MC]	double
SourceView	[1 0 0]		[IC, MC]	double
SourceView:Type	cartesian			attribute
SourceView:Units	metre			attribute

Tabelle 5.2: Variablen und Attribute der *SimpleFreeFieldHRIR*-Convention nach AES69-2015 (Audio Engineering Society, 2015)

Die praktische Implementierung der Daten und Metadaten zeigt beispielhaft Listing 1. Dabei ist zunächst zu erwähnen, dass diese Implementierung mithilfe der *SOFA-API für MATLAB* erfolgte, welche nötige Funktionen und Datenstrukturen zur Verfügung stellt, um ein SOFA-File zu schreiben; diese *SOFA-API für MATLAB* ist frei verfügbar[5] und muss im genutzten MATLAB-Pfad liegen und vor Benutzung mit `SOFAstart` geladen werden.

In Listing 1 ist zu erkennen, dass die Anzahl der Messungen `M` sich nach der Länge eines zuvor initialisierten Vektors `azi` richtet; dabei gibt dieser Vektor `azi` die Azimutwinkel wider, für die je eine BRIR gemessen wurde und der Postprocessing-Routine vorliegt. Hierbei wird - wie bereits an anderer Stelle in Unterunterabschnitt 5.5.2.1 beschrieben - von vorliegenden BRIRs einer gleichmäßig abgestuften Messreihe der vollständigen horizontalen Drehung ausgegangen, sodass die Einträge in `azi` sich durch die Angabe eines vollen Kreisbogens von 360° bzw. [0, 360)° und der gewählten `stepsize` der Messung vollständig ergeben. Die Dimension des Vektors richtet sich also nach der Schrittweite der Messung

[5] https://sourceforge.net/projects/sofacoustics/

bzw. der gleichbedeutenden Variablen `stepsize`. Der angegebene Azimutwinkel folgt hierbei einem Rechtssystem im mathematisch positiven Drehsinn gegen den Uhrzeigersinn wie im AES69-Standard festgelegt und bei der Konzeption des Messsystems, sowie dem Interfacing der Messdaten in die Postprocessing-Umgebung befolgt. Die Anzahl der Daten-Samples `N`, welche im Fall der Nutzung der *SimpleFreeFieldHRIR*-Convention den FIR-Filtertaps bzw. der Anzahl der Samples der Impulsantwort entsprechen, ergibt sich aus der Länge der ersten Dimension der Matrix `all_irs_truncatedFromRT60_1`, welche alle BRIRs des - in diesem beispielhaften Fall - ersten Lautsprechers beinhaltet. Die Anzahl der Receiver `R` wird auf 2 gesetzt, was den beiden menschlichen Ohren entspricht; die Anzahl der Emitter `E` entspricht 1 wie es default-mäßig in der Convention genutzt wird und folglich nicht mehr neu gesetzt wird. Auf Grundlage dieser nun gesetzten AES-69-Dimensionen wird dem zuvor in MATLAB über die Funktion `SOFAgetConventions` geladenen SOFA-Objekt der *SimpleFreeFieldHRIR*-Struktur eine leere (MxRxN)-dimensionale Matrix übergeben, welche in der darauffolgenden for-Schleife mit den BRIRs befüllt wird. Dabei werden im gleichen Schritt auch die zugehörigen `SourcePositions` in das SOFA-Objekt geschrieben, sodass die korrekte Verbindung aus der Position bzw. Lage mit der jeweiligen BRIR erreicht wird. Hier deutet die Nutzung der Variablen `SourcePosition` zunächst auf einen Widerspruch hin, da sich bei der Messung der BRIRs lediglich die Orientierung des Listeners, nicht jedoch die Position des emittierenden Lautsprechers im Raum geändert hat; dies soll im nächsten Abschnitt ausführlicher beschrieben werden. Da es aufgrund der Nutzung von BRIRs nicht zur Extraktion des linearphasigen Anteils in Form eines echten breitbandigen Delays kommt, muss das `Data.Delay` für alle im SOFA-File befindlichen Messungen `M` und für beide Receiver `R` null betragen. Die `Data.SamplingRate` wird aus den zuvor übergebenen Metadaten übernommen und ebenfalls an das SOFA-Objekt übergeben. Des Weiteren werden die für die Datenstruktur wichtigsten charakterisierenden Attribute an das SOFA-File übergeben.

Listing 5.1: Definition und Befüllung des SOFA-Files

```
%% Starte SOFA-API & Setze netCDF Daten-Kompression

SOFAstart;
compression=0; % netCDF Daten-Kompression (0=keine Kompression,
    9=hoechste Kompression)


%% Lade und generiere leere SOFA-Conventions-Datenstruktur

Obj = SOFAgetConventions('SimpleFreeFieldHRIR');


%% Definiere Azimuth- und Elevationswinkel fuer SOFA-
    Datenstruktur

azi=0:stepsize:359; % stepsize ist Schrittweite der horizontalen
    BRIR-Messung, befindet sich in Workspace aus vorangegangenem
    Skript
ele=0;
```

```
%% Definiere SOFA-Dimensionen & befuelle SOFA-Object mit BRIR-
   Daten

M=length(azi);
N=size(all_irs_truncatedFromRT60_1,1); %
   all_irs_truncatedFromRT60_1 ist beispielhafte Matrix mit allen
    BRIRs einer Messung aus vorangegangenem Skript (gekuerzt nach
    RT60)
R=2;
% E=1 ist default in der SimpleFreeFieldHRIR-Convention

Obj.Data.IR = NaN(M,R,N); % Data.IR hat Dimension [M R N]

for ii=1:M % ii geht ueber alle Messungen M

     Obj.Data.IR(ii,1,:)=
        irs_leftear_truncatedFromRT60_normalized_1(:,ii); %
        irs_leftear_truncatedFromRT60_normalized_1 ist
        beispielhafte Matrix mit BRIRs des linken Ohrs einer
        Messung (normalisiert und gekuerzt nach RT60)
     Obj.Data.IR(ii,2,:)=
        irs_rightear_truncatedFromRT60_normalized_1(:,ii); %
        irs_leftear_truncatedFromRT60_normalized_1 ist
        beispielhafte Matrix mit BRIRs des rechten Ohrs einer
        Messung (normalisiert und gekuerzt nach RT60)

    Obj.SourcePosition(ii,:)=[azi(ii) ele 1];
end

Obj.Data.Delay(:,:) = [0 0]; % kein breitbandiges Delay im SOFA-
   File, da keine Extraktion des linearphasigen Anteils der BRIRs
    moeglich
Obj.Data.SamplingRate = Fs; % Sampling Rate aus Messung
   uebergeben


%% Befuelle SOFA-Variablen mit Attributen

Obj.GLOBAL_APIName = 'SOFA API for MATLAB';
Obj.GLOBAL_APIVersion = SOFAgetVersion('API');
Obj.GLOBAL_AuthorContact = 'me099@hdm-stuttgart.de';
Obj.GLOBAL_Comment = 'BRIRs of single speakers';
Obj.GLOBAL_History = 'created with a script';
Obj.GLOBAL_Organization = 'Hochschule der Medien Stuttgart';
Obj.GLOBAL_RoomType = 'reverberant';
Obj.GLOBAL_DatabaseName = 'U48_L_Hochschule der Medien Stuttgart'
   ;
```

```
Obj.GLOBAL_ListenerShortName = 'Neumann KU 100';

%% Update SOFA-Objekt vor Speicherung

Obj=SOFAupdateDimensions(Obj);


%% Speichere SOFA-File 'U48_1-
   L_dynamic_truncatedFromRT60_normalized.sofa' im aktuellen
   MATLAB-Pfad im Ordner 'Dynamic SOFA Files'

currentFolder = pwd;

SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files','U48_1-
   L_dynamic_truncatedFromRT60_normalized.sofa');

disp(['Saving:  ' SOFAfn]);
Obj=SOFAsave(SOFAfn, Obj, compression)
```

Neben den grundlegenden Daten und Metadaten, welche die *SimpleFreeFieldHRIR*-Convention beinhaltet und welche für die Nutzung innerhalb des konzipierten Systems mit BRIRs keinen systemischen Einfluss auf die Funktionalität des in Abschnitt 5.6 beschrieben binauralen Renderers haben, ist ein Aspekt jedoch wie bereits erwähnt näher zu betrachten: Wie in Listing 1 dargestellt, werden die BRIRs in das SOFA-File unter Zuhilfenahme der Variable `SourcePosition`, welche mithilfe der Azimutwinkel - in Falle dieses Systems ohne Elevationswinkel, da die Messung bzw. binaurale Darstellung nur in der horizontalen Ebene erfolgt - initialisiert wird, geschrieben. Nun stellt ein Lautsprecher im Falle einer BRIR-Messreihe, die aus der Definition heraus unter reverberanten bzw. halligen Bedingungen generiert wurde, jedoch nur eine einzige Source (bzw. in dem betrachteten Fall gleichbedeutend einen einzigen Emitter unter Annahme einer idealen Punktschallquelle) dar, jedoch liegen für verschiedene Kopforientierungen, die in der AES-69-Nomenklatur als `ListenerView` bezeichnet werden, BRIRs dieses Lautsprechers vor. Hier findet sich ein scheinbarer Widerspruch. Der `ListenerView` in der *SimpleFreeFieldHRIR*-Convention ist per Definition defaultmäßig entlang der positiven x-Achse ausgerichtet, trotzdem lässt sich mit dem in Abschnitt 5.6 ausführlich beschriebenen *spat5.binaural* Objekt Head-Tracking realisieren. Dies geschieht, in dem bei Veränderung der Kopforientierung das komplette Listener-bezogene Koordinatensystem gedreht wird, was gleichbedeutend mit einer Drehung des `ListenerView`-Vektors um den Koordinatenursprung ist. Auf diese Weise verändert sich die relative Orientierung des Listeners inkl. seiner Receiver zur Source und es werden andere kopfbezogene Filter, die durch die `SourcePositions` im SOFA-File definiert wurden, für das binaurale Rendering ausgewählt; diese `SourcePositions` werden entsprechend der Drehung des Listener-bezogenen Koordinatensystems verändert, da sie auch im Listener-bezogenen Koordinatensystem ausgelesen werden. Im Falle eines HRIR-Datensatzes geschieht dies für ein komplettes Gitter um den Listener herum, da Sources als Quellen der direkten binauralen Synthese frei platziert werden können und es keinen Raumeinfluss gibt. Bei einem einzelnen BRIR-Datensatz

zur Darstellung virtueller Lautsprecher ist die Source/der Emitter jedoch der in diesem Datensatz dargestellte (virtuelle) Lautsprecher in einem (virtuellen) Raum; somit entspricht dieser einer raumfesten Schallquelle. Die Kopfdrehung und die daraus resultierende relative Änderung der Orientierungen zueinander kann jedoch äquivalent zur Drehung bei einem HRIR-Datensatz erfolgen. Es können bei Nutzung eines BRIR-Datensatz in der *SimpleFreeFieldHRIR*-Convention jedoch keinen virtuellen Schallquellen frei platziert werden wie dies im *spat5.binaural* Objekt theoretisch möglich ist und bei der direkten Binauralsynthese auch so umgesetzt wird. Die Daten zur Position und Orientierung des Lautsprechers, welches als omnidirektionale Source behandelt wird, befinden sich fest im Datensatz verankert; somit dürfen dem *spat5.binaural* Objekt auch keine weiteren Positionsdaten zu einer gewünschten Schallquelle übergeben werden, sondern es erfolgt eine Auswahl der kopfbezogenen Filter alleinig auf der Grundlage der Änderung der Kopforientierung und des Listener-bezogenen Koordinatensystems. Praktisch muss dem *spat5.binaural* Objekt somit eine neutrale Source (0°, 0°, 1) übergeben werden.

Nach dieser Befüllung des SOFA-Objekts mit den gewünschten Daten und Metadaten erfolgt schlussendlich die Speicherung in einem *.sofa*-File. Dies geschieht mit der entsprechenden Funktion der SOFA-API, die `SOFAsave` heißt.

Bei Betrachtung der Daten und vor allem Metadaten, welche im *.sofa*-File gespeichert werden bzw. generell gespeichert werden können, fällt auf, dass Informationen zu den `headcutsamples`, zum `normalisationfactor`, zur Latenz der Messung (kompensiert oder unkompensiert, sowie I/O-Round-Trip-Latenz der Messung) und zum Typ und der Dauer einer möglicherweise umgesetzten Fensterung der BRIRs nicht gespeichert werden können. Dies ist insofern bedauerlich, da so diesbezüglich keine Rückschlüsse mehr auf die unverarbeiteten Daten gezogen werden können. Das *.miro*-Dateiformat der *TH Köln* (Bernshütz, 2013) ermöglicht die Speicherung dieser Informationen, kann jedoch aufgrund der nötigen Systemanbindung an das vorgegebene Interface des Systemmodul 3 nicht genutzt werden.

5.5.2.3.2 Speicherung der *FullDynamic*-BRIRs Die BRIRs werden im Skript *twospeakers_saveFullDynamic.m* in ihrer vollständigen Länge nach der links- und rechtsseitigen Kürzung, also in Form der Matrizen `all_irs_truncatedFromRT60_X` bzw. `all_irs_truncatedFromEDT_X` je nach Wahl Kriteriums der rechtsseitigen Kürzung in einem SOFA-File gespeichert. Dabei muss das Benennungschema in der Form *0-U48_1-L_dynamic_truncatedFromRT60_normalized.sofa* vorliegen; das File muss mit einer führenden *0* beginnen, damit es in der flexiblen Auralisationsumgebung später auch bei Wahl des voll dynamischen Processings richtig ausgewählt wird. Die weiteren Angaben wie *U48* können sich auf die Bezeichnung eines spezifischen Raumes beziehen, während *1-L* die Bezeichnung des Lautsprechers im gewählten Lautsprechersetup ist. Der Zusatz *dynamic* soll zur Verdeutlichung dienen, dass dieser Datensatz dynamisch bzw. zeitlich variierend verarbeitet wird. *truncatedFromRT60* sowie *normalized* beziehen sich lediglich auf die auf diesen Daten durchgeführten Verarbeitungsschritte.

5.5.2.3.3 Zweistufige Speicherung Das Skript *twospeakers_splittingBRIRs_saveDynamic+Static.m* führt des Weiteren vor der Speicherung der BRIR-Daten noch eine Trennung der BRIRs auf Grundlage der angegebenen Mixing Times `split_times_seconds` durch. Nach dieser Trennung wird jeweils eine Kombination aus *SOFA*- und *wav*-Files gespeichert, welche so benannt sind, dass sie in der standardmäßig implementierten Auralisationsumgebung, die Wahlmöglichkeiten der Mixing Times zwischen *40 ms* und *130 ms* bietet, richtig angewählt werden.

5.6 Systemmodul 3: Flexible Auralisationsumgebung

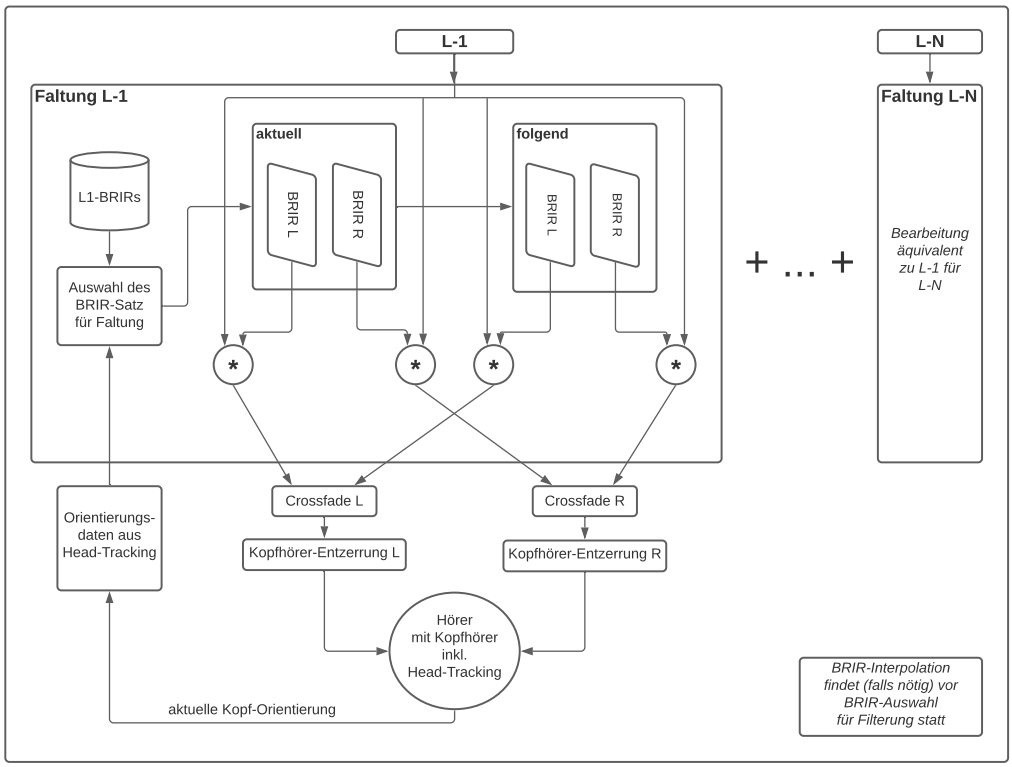

Abbildung 5.19: Struktur der dynamischen Binauralsynthese (eigene Darstellung)

Bevor im weiteren Verlauf dieses Kapitels die konkreten Anforderungen an die flexible Auralisationsumgebung untersucht und die konkrete Umsetzung dargestellt wird, zeigt Abbildung 5.19 nochmals die grundlegende Struktur der dynamischen Binauralsynthese für die Simulation von virtuellen Lautsprechern, welche in diesem System umgesetzt wird. Dabei wird deutlich, dass durch Änderungen der Kopforientierung und daraus resultierenden Änderungen der BRIRs für Lautsprecher, die sich an einer bestimmten Position im Raum befinden, die entsprechend variierenden Filter (aktuell -> folgend) durch Überlagerung von zwei parallelen Filtern, welche mithilfe eines Crossfades überblendet werden, umgesetzt sind. Dies geschieht nun für jeden Lautsprecher (L-1 bis L-N) eines mehrkanaligen Lautsprechersetups in dieser Form; danach kommt es zu einer einfachen Superposition der Ohrsignale, welche für jeden Lautsprecher generiert wurden, ehe als letzter Schritt der Verarbeitungskette eine Kopfhörer-Entzerrung stattfindet.

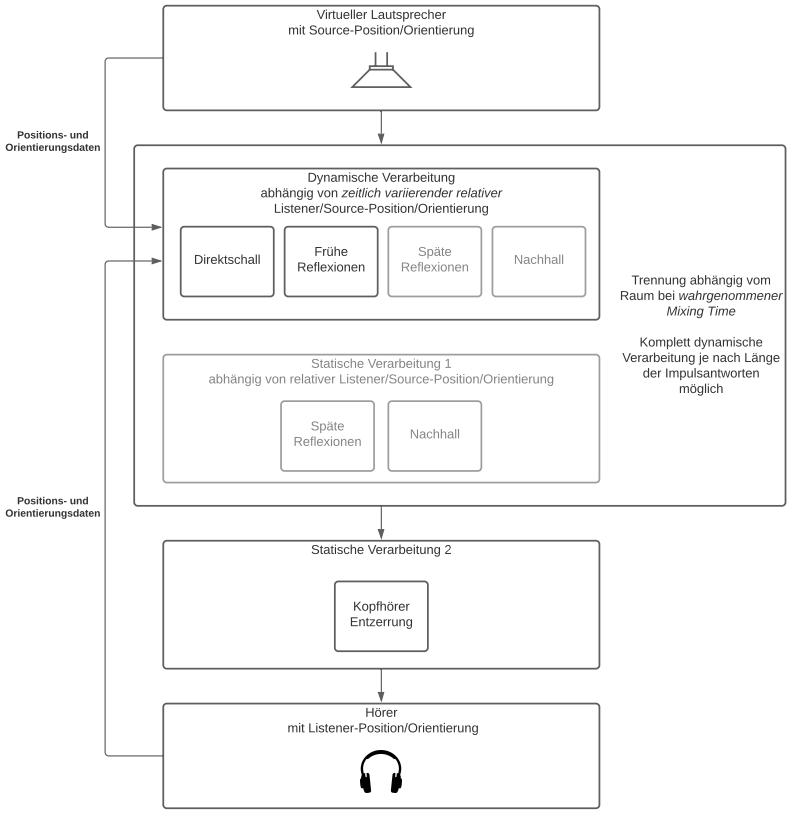

Abbildung 5.20: Signalfluss der flexiblen Auralisationsumgebung (eigene Darstellung)

Diese beschriebene Struktur ist vollständig dynamisch und wird durch Orientierungsdaten des Kopfes, welche aus einem Head-Tracking-System gewonnen werden, jederzeit bzw. durch die vom Gesamtsystem vorgegebene effektive Update-Rate angepasst. Nun ist es aufgrund der Diffusität des Nachhalls, welcher gemittelt keine Richtungsinformationen mehr enthält, möglich, ab einem bestimmten Zeitpunkt in der binauralen Raumimpulsantwort auf die dynamische Anpassung der kopfbezogenen Filter komplett zu verzichten und eine rein statische Verarbeitung durchzuführen. Dieser Zeitpunkt in der Raumimpulsantwort, zu welchem diese Diffusität erreicht ist bzw. so stark ausgeprägt ist, dass eine Änderung der dynamischen Verarbeitung hin zu statischer Verarbeitung nicht mehr wahrgenommen wird, wird *wahrgenommene Mixing Time* genannt. Auf dieser Grundlage wird der Signalfluss der flexiblen Auralisationsumgebung umgesetzt, der in Abbildung 5.20 exemplarisch für einen virtuellen Lautsprecher dargestellt ist.

5.6.1 Anforderungsanalyse

In Abbildung 5.21 sind alle Anforderungen an die flexible Auralisationsumgebung dargestellt, welche im Rahmen dieser Arbeit konzipiert und prototypisch implementiert wird. Dabei wird zwischen funktionalen Anforderungen und Benutzeranforderungen unterschieden. Die funktionalen Anforderungen sind hierbei Anforderungen an das System, um die im Verlauf dieser Arbeit herausgearbeiteten Systemparameter nach dem aktuellen Stand der Technik zu erreichen. Benutzeranforderungen sind spezifisch für die Nutzbarkeit des Systems an der Hochschule der Medien zu sehen; sie bilden also keine funktional systemrelevanten Bausteine.

Harte Anforderungen, die für die einwandfreie Funktion der Auralisationsumgebung unabdingbar erfüllt werden müssen, sind mit schwarzer Schrift dargestellt, während weiche Anforderungen, deren Nicht-Umsetzung keine - nennen wir es - direkte Fehlfunktion verursacht, in grauer Schrift hinterlegt sind. Dabei sind auch drei weiche Anforderungen unter den funktionalen Anforderungen geführt; nämlich die Filter-Interpolation und ITD-Individualisierung sowie Delay-Interpolation. An dieser Stelle ist zu erwähnen, das diese Interpolationen je nach Schrittweite bzw. Raster-Auflösung der BRIR-Messungen sehr wohl harte Anforderungen zur Erfüllung einer gewünschten Qualität der Wiedergabe sind. Diese Interpolationen finden - falls überhaupt nötig - (zumindest in dem in dieser Arbeit konzipierten System) jedoch im bereits im Abschnitt 5.5 beschriebenen Systemmodul 2 statt und sind folglich im Systemmodul 3 der flexiblen Auralisationsumgebung zwar theoretisch möglich, jedoch nicht mehr unbedingt nötig und somit weich. Des Weiteren ist eine getrennte ITD-Verarbeitung (und damit ITD-Individualisierung und Delay-Interpolation) beim Rendering mit BRIRs aufgrund der nicht umsetzbaren Trennung des linearphasigen Anteils in ein echtes breitbandiges Delay nicht möglich; mögliche Methoden eine Verarbeitung dieser Art zu erreichen hat Lindau (2014) untersucht.

Im Folgenden sollen nun schrittweise die Anforderungen an die flexible Auralisationsumgebung im Detail betrachtet werden. In diesem Zusammenhang werden auch direkt mögliche Probleme und deren Zusammenhänge diskutiert und Lösungsmöglichkeiten unterbreitet, die im drauffolgenden Kapitel, welches die Umsetzung konkret beschreibt, wieder aufgegriffen und diskutiert werden.

Der erste Block der funktionalen Anforderungen befasst sich mit dem Head-Tracking. Wie in den vorangegangenen Kapiteln zu lesen, ist Head-Tracking für eine plausible binaurale Darstellung zwingend notwendig; der Begriff der *dynamischen* Binauralsynthese definiert sich daraus. Wichtige Anforderungen an das Head-Tracking sind eine minimale Genauigkeit, die der Sensor des Head-Trackers bei der Lagebestimmung liefert, sowie eine verschwindend geringe Latenz der Lagebestimmung des Kopfes. Die Genauigkeit des Sensors als auch die wichtigste zur Latenz beitragende Komponente wird hauptsächlich durch die Update-Rate des Sensors bestimmt; eine weitere Latenz-Komponente, welche meist dem Head-Tracking als Komplettsystem zugeschrieben wird, ist die der seriellen Übertragung. Neben der eigentlichen *direkten* Latenz dieser Übertragung kommt es häufig zu Phänomenen der Asynchronität bei der seriellen Datenübertragung zwischen Head-Tracker und dem System, welches die gelieferten Head-Tracker-Daten verarbeitet, die die Latenz vergrößern bzw. die effektive Update-Rate der Lagebestimmung. Der Anforderung einer im Gesamtsystem irrelevanten Latenz an die serielle Kommunikation zwischen Head-Tracker und der digitalen Signalverarbeitung ist der zweite große Anforderungsblock der funktionalen Anforderungen gewidmet.

Die meisten funktionalen Anforderungen finden sich jedoch im großen Block der digitalen Signalverarbeitung: Hier spielt wie auch in den vorangegangenen Bausteinen erneut die Latenz eine Rolle, welche vom Signalverarbeitungsweg verursacht wird. Die Kopfhörer-Entzerrung garantiert die *Schnittstellenanpassung* zwischen genutztem Kunstkopfmikrofon (oder der genutzten individuellen Aufnahmeumgebung) und Kopfhörer; diese funktionale Anforderung

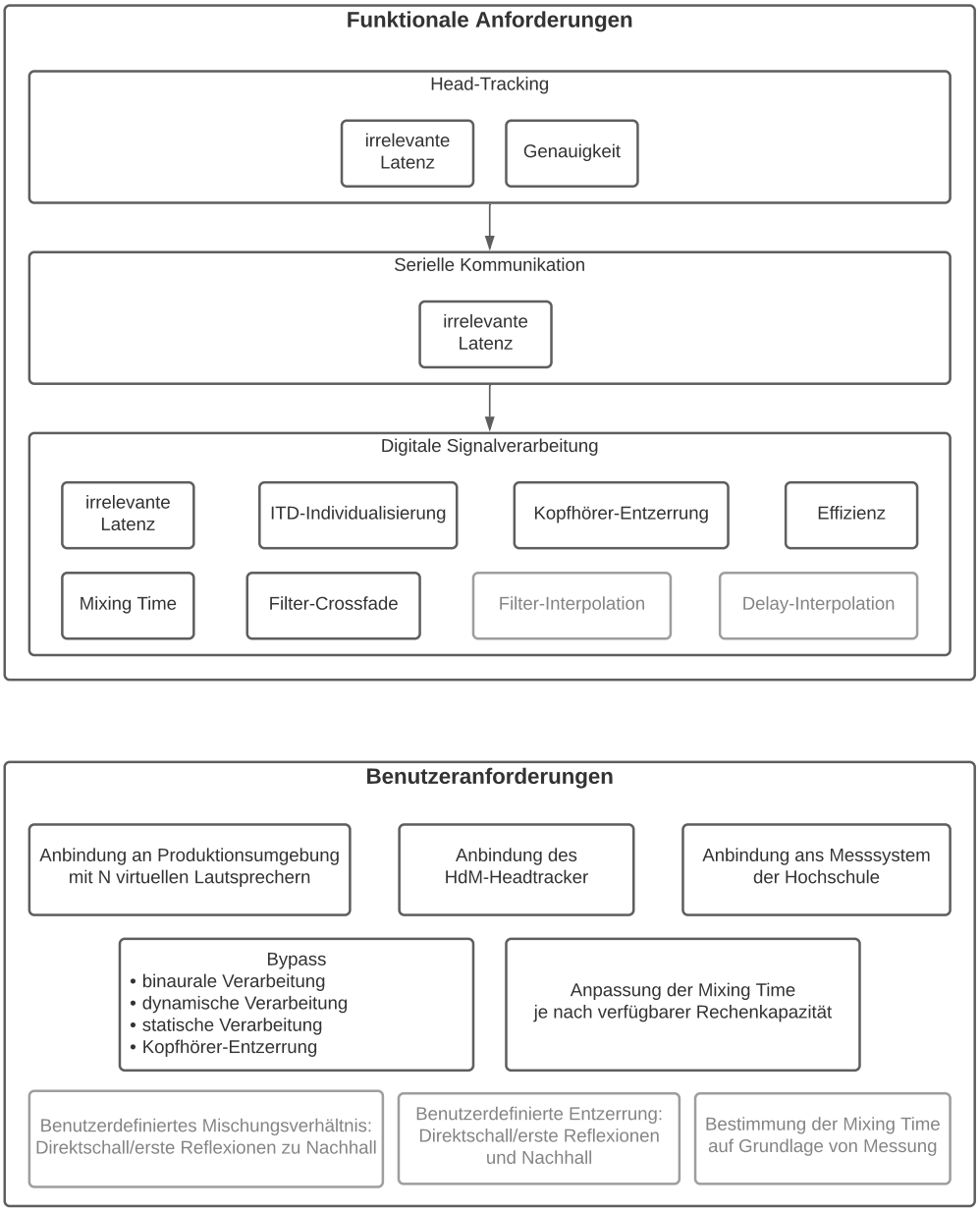

Abbildung 5.21: Funktionale Anforderungen sowie Benutzeranforderungen an die Flexible Auralisationsumgebung zur Darstellung von Regieräumen/Mischkinos an der Hochschule der Medien Stuttgart (schwarz: harte Anforderungen, grau: weiche Anforderungen

ist womöglich nicht die Wichtigste für eine plausible Auralisation, jedoch ist sie unbedingt zu berücksichtigen bei der Reduktion des verfärbenden Einflusses eines Kopfhörers für eine authentische binaurale Wiedergabe über Kopfhörer. Die Anforderungen der Effizienz und (wahrnehmbaren) Mixing Time gehen Hand in Hand: Die Effizienz der digitalen Signalverarbeitung ist im Falle der Faltung bzw. Filterung von BRIRs, deren Länge im Bereich der Nachhallzeit *RT60* des zu simulierenden Raumes liegt (für die in dieser Arbeit betrachteten Räume wie Tonstudioregien oder Mischkinos sind arithmetische Mittelwerte der Nachhallzeiten in den Terzbändern von 200 Hz bis 4 kHz gemäß EBU Tech. 3276 und SSF-01.1 bzw. nach Dolby- oder THX-Vorgabe von 0,2 bis 0,4 Sekunden zu erwarten (Weinzierl, 2008)), und der Trennung der dynamischen und statischen Faltung der BRIRs anhand der Mixing Time zu analysieren. Hierbei kommt es neben dieser Trennung vor allem auf die Wahl des

Faltungsalgorithmus an; die direkte Faltung zeigt lediglich bei kurzen Impulsantworten eine gute Performance, während die schnelle FFT-basierte Faltung bei längeren Impulsantworten bessere rechnerische Effizienz zeigt. Mit Hilfe der (wahrnehmbaren) Mixing Time ist einer Trennung der Faltungsoperationen in einen dynamisch verarbeiteten Teil und einen statischen Teil ohne Verluste der wahrgenommenen Plausibilität bzw. Authetizität möglich; diese Mixing Time soll für jeden Raum individuell und nach individuellem Hörempfinden einstellbar sein. Eine weitere wichtige funktionale Anforderung sind Crossfades zwischen den beiden parallelen Filtern bzw. Faltungen, welche für jeden Lautsprecher bei Kopfdrehung ineinander übergeblendet werden müssen; dabei ist die wirksame Filterung die zu einem Zeitpunkt für die jeweilige Kopforientierung gültige, ehe die Kopfdrehung eine Überblendung in den nächsten wirksamen Filter verursacht. Die Crossfades sind hierbei nötig, damit es nicht zu Sprüngen in der Wahrnehmung kommt und um sogenannte *Switching-Artefakte* bei der Umschaltung zwischen den genutzten FIR-Filterkernen zu verhindern; bei einer geringen Raster-Auflösung kann die Form und Dauer der Crossfades im Zusammenhang mit dem genutzten Crossfade-Scheduling besonders bedeutsam sein. Die wie bereits beschrieben als weich dargestellten funktionalen Anforderungen der Filter-Interpolation (und gegebenenfalls Delay-Interpolation, siehe (Lindau, 2014)) beschreiben Anforderungen an den Datensatz, welcher eine ausreichende Raster-Auflösung besitzen sollte; sofern diese Rasterauflösung nicht durch die Messdaten direkt erreicht wird, sind interpolierte Daten zumeist nötig, um weiche, nicht wahrnehmbare Übergänge zwischen benachbarten Filtern bei der Nutzung von *Zwischenpositionen* zu erreichen (Mackensen, Felderhof et al., 1999). Dies steht in direktem Zusammenhang mit der genutzten Überblendung mithilfe *passend* geformter Crossfades und muss für eine plausible bzw. authentische Wahrnehmung untersucht und aufeinander abgestimmt werden. Diese Interpolationen finden in Echtzeit statt oder werden offline berechnet und in den Datensätzen hinterlegt, was zu einer höheren Raster-Auflösung der Letzteren führt; somit ist diese funktionale Anforderung an die flexible Auralisationsumgebung bei ausreichend aufgelösten Messdaten oder Interpolation in Systemmodul 2 weich bis nahezu nicht vorhanden. Da die Interpolationsberechnungen des Weiteren aufgrund höherer Komplexität recht rechenintensiv sein können, scheint eine Auslagerung der Berechnung in einen Offline-Prozess oder gar einen separaten Prozessor wie im BRS-System umgesetzt (Horbach et al., 1999) zumeist sinnvoll.

Die Benutzeranforderungen sollen eine optimale Anbindung an die Nutzung im Umfeld der Hochschule gewährleisten: Dabei ist eine flexible Anbindung an Lautsprechersetups mit mehreren Lautsprechern N - wie sie in den Räumen der Hochschule zu finden sind - zu gewährleisten. Somit muss das System in der Lage sein, Lautsprechersignale in ausreichender Anzahl aus der gewählten Produktionsumgebung zu empfangen und diese im gewählten Lautsprecher-Setup in Form von virtuellen Lautsprechern zu binauralisieren. Des Weiteren ist die Anbindung des *HdM-Headtrackers* zu gewährleisten, welcher mit Hilfe eines Arduino Pro Micro auf Basis eines ATmega32U4 Mikrocontrollers serielle Daten einer Adafruit BNO055 IMU, die die aktuelle Lage des Kopfes beschreiben, via USB an den Rendering-Computer weitergibt. Eine Schnittstelle zwischen der Messumgebung bzw. vorausgegangener Postprocessing-Blocks und der Auralisationsumgebung muss konzipiert werden. Das binaurale Rendering soll so aufgebaut sein, dass es flexibel angepasst werden kann wie die Namensgebung der *flexiblen*

Auralisationsumgebung bereits beschreibt: Harte Anforderungen sind hierbei die Möglichkeit des Bypass der kompletten binauralen Verarbeitung, als auch jeweils der dynamischen Verarbeitung sowie der statischen Verarbeitung der BRIRs und der Kopfhörer-Entzerrung. Die Mixing Time soll entsprechend der im *Systemmodul 2: Postprocessing der BRIRs* festgelegten Zeiten ausgewählt werden können; dabei kann eine Auswahl unter verschiedenen Aspekten vorgenommen werden, sei es durch die verfügbare Rechenleitung oder psychoakustisch motiviert. An dieser Stelle ist auch eine Möglichkeit der Bestimmung der Mixing Time aus der Analyse einer Impulsantwort wünschenswert, welche eine für den zu auralisierenden Raum passende *wahrnehmbare Mixing Time* berechnet und dem Nutzer vorschlägt; dies ist als weiche Anforderung formuliert. Die weiteren weichen Benutzeranforderungen liegen lediglich in der vom Benutzer gewünschten klangästhetischen Anpassung der Verarbeitung, welche fernab der möglichst authentischen Simulation eines Tonregieraums oder eines Mischkinos liegt; hierzu wird die Möglichkeit des benutzerdefinierten Mischungsverhältnisses zwischen dem dynamischen Anteil des Direktschalls bzw. der ersten Reflexionen zum statischen Nachhall gefordert, sowie die Möglichkeit der Entzerrung der Anteile mit einem einfachen Kuhschwanzfilter. Eine weitere Parametrisierung, welche weitere Anpassungsmöglichkeiten bietet, ist aufgrund der datenbasierten Verarbeitung als Faltungshall jedoch nicht so leicht umsetzbar und wird deswegen in der Anforderungsanalyse nicht weiter betrachtet.

5.6.2 Umsetzung

Wie bereits in Unterunterabschnitt 5.4.2.1 beschrieben, wird auch das *Systemmodul 3: Flexible Auralisationsumgebung* in der Entwicklungsumgebung Max/MSP zusammen mit der Spat-5-Library von IRCAM umgesetzt. Eine allgemeine Motivation zur Wahl dieser Entwicklungswerkzeuge wurde schon gegeben. Da die Interaktion der Echtzeitsignalverarbeitung von Audio mit den Orientierungsdaten aus dem Head-Tracking-System für die Funktion der flexiblen Auralisationsumgebung von starker Relevanz ist, sei im folgenden Kapitel kurz erläutert, wie Max/MSP zeitkritische Events im Allgemeinen umsetzt und wie Spat-5 im Speziellen arbeitet.

5.6.2.1 Max-Scheduler und eventbasierte Verarbeitung von Audio in Spat-5

In Max/MSP werden Events in folgenden zwei Kategorien eingeteilt: Events hoher Priorität, sogenannte Scheduler-Events oder Events niedriger Priorität, sogenannte Queue-Events. Zeitkritische Events, die Timing-Informationen tragen und somit eine zeitgebundene Übertragung bzw. Ausführung benötigen, sind Scheduler-Events. Queue-Events tragen keine Zeitinformation, d. h. sie werden so schnell wie möglich verarbeitet, aber zu keinem geplanten Zeitpunkt und nicht mit hoher Priorität. In der standardmäßigen Nutzung von Max/MSP kann es jedoch vorkommen, dass es zu Threading-Problemen im Zusammenhang mit diesen beiden Prioritätsstufen kommt. Das kann dazu führen, dass Scheduler-Events nicht mehr zur geplanten Zeit ausgeführt werden, da beispielsweise gerade ein Queue-Event mit langerer Verarbeitungszeit ausgeführt wird - das Scheduler-Event muss warten. Nur wenn die Max/MSP-Option *Overdrive* aktiviert ist, haben Scheduler-Events hoher Priorität tatsächlich Vorrang

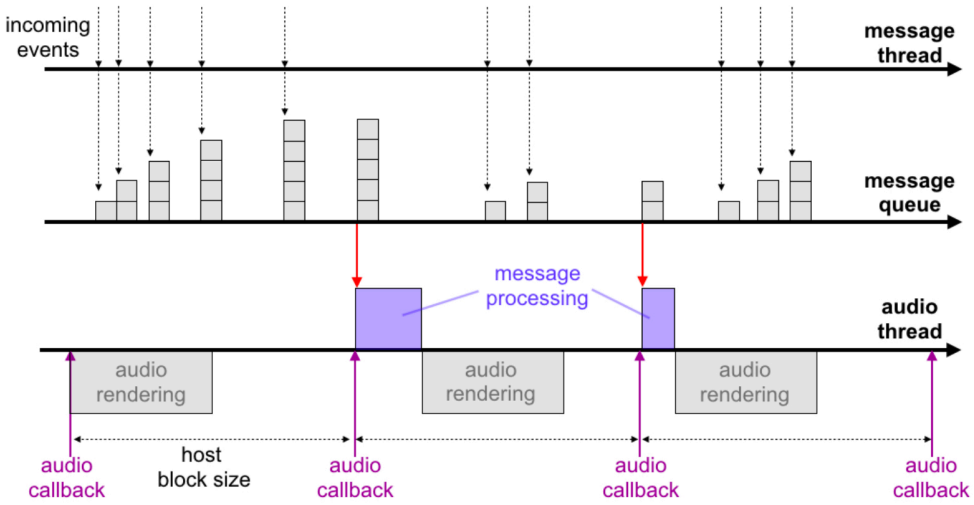

Abbildung 5.22: Terminierung von Events durch den Max-Scheduler im Audio-Interrupt-Verfahren (Carpentier, 2018a)

vor Queue-Events niedriger Priorität. Wenn *Overdrive* aktiviert ist, verwendet Max/MSP zwei Threads für die Ausführung von Events, sodass ein Scheduler-Event die Ausführung eines Queue-Events unterbrechen kann bevor das Letztere fertig ausgeführt ist. Andernfalls verarbeitet Max/MSP sowohl Events hoher Priorität als auch Events niedriger Priorität im selben Thread, wobei kein Event ein anderes unterbricht. Neben der Einhaltung der exakten Timings von Scheduler-Events hat die Nutzung von zwei parallelen Threads auch den Vorteil der Nutzung von zwei getrennten Prozessoren auf modernen Computern mit Mehrkernprozessoren. In Max/MSP können diverse Einstellungen vorgenommen werden, die das Verhalten des Schedulers und der Queue betreffen: Sowohl das Intervall, in dem der Scheduler und die Queue abgefragt werden, als auch die Anzahl der Events, die pro Abfrage ausgeführt werden, können eingestellt werden.

Neben der Nutzung des *Overdrive*-Modus zur Einhaltung des Timings von Scheduler-Events, können die Scheduler-Events auch im Audio-Thread verarbeitet werden; dies geschieht im Modus *Sheduler in Audio Interrupt*, welcher auch kurz *Audio Interrupt* genannt wird. Bei Nutzung dieses Modus ist das Verarbeiten der Scheduler-Events an die Signalvektorgröße gekoppelt und die Abfrage der Scheduler-Events erfolgt ein Mal pro Signalvektor bzw. Audioblock. Hierbei ist zu beachten - wie auch in Abbildung 5.22 grafisch dargestellt -, dass die Eventverarbeitung in diesem Modus direkt im Audio-Thread stattfindet und es somit, je nach Menge der vorhandenen Events im *Message Queue* pro *Audio Callback*, zu zeitlich unterschiedlich langem *Messsage Processing*, also der Verarbeitung der Events vor dem eigentlichen *Audio Rendering* kommen kann. Dies kann auch dazu führen, dass es zu *Audiodropouts* kommt, wenn das *Messsage Processing* so viel Zeit in Anspruch nimmt, dass das *Audio Rendering* nicht mehr vollständig innerhalb einer Audioblockgröße und vor dem nächsten *Audio Callback* ausgeführt werden kann; der Audio-Thread kann dann mit seinen Echtzeitanforderungen nicht mehr mithalten, so kommt es zu Knacksern und Störungen in der Audioausgabe. Dieses Problem kann mithilfe zweier Vorgehensweisen minimiert werden: Zum einen kann der sogenannte *Poll Throttle* Parameter des Max-Scheduler angepasst werden. Dieser begrenzt

die Anzahl der Events, welche innerhalb einer Scheduler-Abfrage - im Falle der Nutzung des *Sheduler in Audio Interrupt* Modus also innerhalb einer Audioblockgröße - bedient werden. Des Weiteren kann die Audioblockgröße erhöht werden, was sowohl dem *Messsage Processing* als auch eigentlichen *Audio Rendering* mehr Zeit zur Verfügung stellt; nachteilig ist hierbei jedoch, dass somit die Latenz bis zur Verarbeitung der Orientierungsdaten des Head-Trackers vergrößert wird. Der *Poll Throttle* Parameter wurde für die Entwicklung des im weiteren Verlauf dieser Arbeit beschriebenen Max-Patches für die flexible Auralisationsumgebung in der default-mäßigen Einstellung von *40* belassen. Bei der im Zusammenhang der Entwicklungsparameter dieses Systems beschriebenen Wahl des Max-Signalvektors und damit Audioblocks der dynamischen Verarbeitung mit 128 Samples, führt dies zu maximal verarbeiteten 40 Scheduler-Events pro 2,6 ms. Dies erscheint aufgrund des im Weiteren ebenso beschriebenen Vorliegens von Head-Tracker-Events in einem Zeitintervall von minimal 10 ms, mehr als ausreichend. Dieser Parameter wurde im Rahmen der Entwicklung des Systems nicht weiter untersucht, jedoch ist eine Vergrößerung der verfügbaren Rechenleistung bei Verringerung des Parameters nicht zu erwarten, da die maximale Anzahl von 40 Scheduler-Events pro 2,6 ms bei Nutzung des *HdM-Headtrackers* nie erreicht wird.

Dieses beschriebene Verhalten wird standardmäßig von allen Audio-Objekten der Spat-5-Library genutzt - unabhängig von den *Overdrive*- oder *Audio Interrupt*-Einstellungen der Host-Anwendung - und ist somit für die Betrachtungen der erreichten Latenz der dynamischen Signalverarbeitung ausschlaggebend. Diese Struktur garantiert eine gleichbleibende zeitliche Verarbeitung der Orientierungsdaten mit der Audioblockgröße/dem Signalvektor, die unabhängig von der Taktung und *Thread Safety* der Event-Abfragen in den anderen Threads ist; dies ist prinzipiell wünschenswert. Im Vergleich zur Nutzung von normalen Max-Nachrichten vereinfacht die Spat-5-spezifische Kapselung von Ereignissen in OSC-Nachrichten die Programmierung von thread-sicheren Warteschlangen für die Synchronisation von Daten - auch jenen, die in den verschiedenen Max-Threads (Audio-Thread, Queue-Thread, Scheduler-Thread usw.) gemeinsam genutzt werden (Carpentier, 2018b). Trotzdem wird in Audio-Objekten die FIFO-Warteschlange der OSC-Nachrichten nach dem Prinzip des *Sheduler in Audio Interrupt*-Modus umgesetzt (siehe Abbildung 5.22. Als Nachteil ist hier zu nennen, dass es auf diese Weise durch das *Messsage Processing* im Audio-Thread zu einer weiteren Beanspruchung des Letzteren kommt, welche Ressourcen für die eigentlichen Audiosignalverarbeitung verschwendet; dies ist gerade bei den zeitlich variierenden Faltungsoperationen - welche noch dazu wie in Unterunterabschnitt 5.6.2.5 beschrieben im genutzten binauralen Processing-Core als direkte Faltung implementiert sind - kritisch unter Nutzung von BRIRs, die eine nicht zu vernachlässigende Länge besitzen. Betrachtet man des Weiteren noch die Tatsache, das Max/MSP ohne Erweiterungen des Programmcodes unter Nutzung von *poly* -Objekten, alle Audiosignalverarbeitungsschritte im gleichen Audio-Thread auf einem Prozessorkern ausführt und somit kein Multi-Threading nutzt, so ist diese Verarbeitung bei Betrachtung von mehreren Lautsprechern N - wie es im System prinzipiell skalierbar bleiben soll - als noch kritischer zu betrachten.

5.6.2.2 HdM-Headtracker

Der *HdM-Headtracker* ist ein Do-It-Yourself Headtracker, welcher an der Hochschule der Medien Stuttgart im Rahmen eines Forschungsprojekts entwickelt wurde und stetig weiterentwickelt wird. Dieser Headtracker wird im Rahmen der Konzeption und Entwicklung des Simulationssystems dieser Arbeit verwendet und ist folglich als integraler Bestandteil zu verstehen; die Studenten der Hochschule der Medien arbeiten in weiteren Projekten mit diesem Headtracking-System und folglich soll die Nutzung dieses Systems weiter gefördert werden.

Der *HdM-Headtracker* setzt sich aus dem *Adafruit BNO055 9-DOF-Sensor IMU-Breakout* und einem *Arduino Pro Micro* (5 V / 16 MHz mit einem ATmega32U4 Mikrocontroller in den Abmessungen der Version von *SparkFun Electronics*) zusammen. Dabei nutzt das *Adafruit BNO055 IMU-Breakout* einen *Bosch BNO055 IMU-Sensor*, welcher Sensordaten algorithmisch aus einem 14-Bit-Beschleunigungssensor, einem triaxialen geomagnetischen Sensor und einem triaxialen 16-Bit-Gyroskop mit geschlossenem Regelkreis über einen 32-Bit-ARM Cortex-M0 Mikrocontroller fusioniert bzw. auswertet und diese Daten in Echtzeit als absolute Orientierungsdaten in Quaternionen und Kardan- bzw. Eulerwinkel ausgibt(**BoschSensortec2020**). Die Nutzung von drei Sensoren erhöht die Genauigkeit und verringert das Driften der Orientierungsdaten; eine genauere Analyse der durchgeführten Informationsfusion soll an dieser Stelle nicht durchgeführt werden. Die Anbindung an den *Arduino Pro Micro* erfolgt über den I^2C Datenbus. Folglich sind neben der Spannungsversorgung von 3,3 bis 5 V nur zwei weitere Verbindungen in Form der Takt-(*SCL* = Serial Clock) und Datenleitung (*SDA* = Serial Data) nötig; diese können mit einer 3 oder 5 V Logik arbeiten, Pull-Up-Wiederstände mit 10 $k\Omega$ sind bereits im Breakout-Board integriert. Die Ausgabe der Quaternionen und Kardan- bzw. Eulerwinkel erfolgt mit einer Update-Rate von bis zu 100 Hz. Diese 100 Hz Update-Rate erreichen nicht die 120 Hz Update-Rate gängiger Headtracking-Systeme wie in Unterabschnitt 5.2.6 beschrieben. Der *Arduino Pro Micro* lässt sich via USB mit dem Rendering-Computer verbinden. Um die beiden für die Ausgabe der Orientierungsdaten benötigten elektrischen Komponenten des Headtrackers an einem Kopfhörer zu befestigen, wurde ein kleines leichtes Gehäuse entwickelt, welches mittig auf dem Bügel eines Kopfhörers befestigt werden kann.

Wie in Abschnitt 3.4 mathematisch erläutert, ist die Berechnung von Drehungen mithilfe von Quaternionen genauer als mit Hilfe der Euler- bzw. Kardanwinkel. Aus diesem Grund nutzt dieses System auch die Ausgabe der Quaternionen des *Adafruit BNO055*; des Weiteren bietet der im Folgenden beschrieben binaurale Processingcore *spat5.binaural* auch eine direkte Nutzung der Orientierungsdaten in der mathematischen Darstellung der Quaternionen. Die Anbindung des *HdM-Headtrackers* an die Anwendung (den Patch) der flexiblen Auralisationsumgebung erfolgt mithilfe des Subpatches `headtrackerReceiver`, welcher ebenfalls im Rahmen der Entwicklung dieses Head-Tracking-Systems an der Hochschule der Medien Stuttgart entwickelt wurde. Dieser Subpatch führt die im Weiteren näher beschriebene Abtastung des seriellen Datenstroms durch, welche die Extraktion der sich zeitlich verändernden Orientierungsdaten unter Nutzung der Quaternionen liefert und diese in die eigentliche Auralisationsanwendung weitergibt. Des Weiteren setzt dieser Subpatch auch den *Reset* der

Orientierung um, welcher in der Anwendung der flexiblen Auralisationsumgebung durchgeführt werden kann. Der `headtrackerReceiver` Subpatch muss im gleichen Ordner wie der Hauptpatch liegen.

Der *Arduino Pro Micro* muss mit entsprechendem Code bespielt werden, um die via I^2C übertragenen Sensordaten des *Adafruit BNO055* richtig auszulesen und via USB an den Rendering-Computer zu übertragen. Dazu werden mehrere Bibliotheken benötigt, nämlich die standardmäßig in der Arduino IDE vorhandene Wire Library (`<Wire.h>`) zur I^2C-Kommunikation sowie die Adafruit Sensor Library (`<Adafruit_Sensor.h>`) und die Adafruit BNO055 Library (`<Adafruit_BNO055.h>`) für sensorspezifische Funktionen. Der Arduino Sketch, welcher für den *HdM-Headtracker* im Rahmen dieser Arbeit genutzt wurde, ist auf dem dieser Arbeit beigefügten Datenträger zu finden und liegt den Anlagen bei (*headtracker_Micro_BNO055.ino*). Die Implementierung dieses Sketches liefert die vollständige Beschreibung aller drei Rotationsfreiheitsgrade mit Hilfe von Quaternionen, welche dem `headtrackerReceiver` Subpatch und damit der flexiblen Auralisationsumgebung so auch vollständig übergeben wird. Aufgrund der lediglich in einem horizontalen Ring aufgelösten BRIR-Daten dieses System hat letztlich nur die Rotation um die z-Achse (Gieren) eine Auswirkung auf das Rendering. Der Sketch und `headtrackerReceiver` Subpatch werden trotzdem so genutzt, dass vom Head-Tracking-System erkannte Rotationen um alle Rotationsachsen ausgegeben werden; dies stellt prinzipiell zunächst keine Einschränkungen für die tatsächlich benötigte Information dar und soll somit für etwaige Weiterentwicklungen von beispielsweise BRIR-Datensätzen hin zu anderen Raster-Auflösungen zunächst so beibehalten werden.

Die seriellen Headtracker-Daten werden in Max/MSP mithilfe einer 100 Hz Abtastung des seriellen Ports empfangen. Die Baudrate entspricht 115200 Baud und ist damit ausreichend groß dimensioniert. Dabei kommt es aufgrund der möglichen Asynchronität der Headtracker-internen Abtastung zur Abtastung des seriellen Ports zu einer maximalen Latenz von 20 ms bis eine Kopforientierung an die binaurale Rendering-Engine weitergegeben wird; dies entspricht einer minimalen effektiven Update-Rate der Abtastung der Kopforientierung bis zum Eingang in die Rendering-Engine von 50 Hz, die bis auf 100 Hz bei vollständiger Synchronität der Updates ansteigen kann. Bei der getätigten Betrachtung der Datenverarbeitung im Headtracking-System des *HdM-Headtrackers* müssen allerdings Vereinfachungen getroffen werden, da nicht alle Abhängigkeiten der Datenabfrage/-weitergabe und deren Interaktion zueinander im Rahmen dieser Arbeit aufgelöst werden können; beispielsweise wurde die Kommunikation der IMU mit dem Arduino über den I^2C-Datenbus, sowie die Schnittstelle zwischen Arduino und der seriellen (USB-)Übertragung zum Rendering-Computer nicht in die Betrachtung mit einbezogen. Folgende Beschreibungen sind also als Abschätzung zu sehen, die jedoch in dieser Form für das System als ausreichend erachtet werden können und auch bei der Analyse anderer Systeme dieser Art so umgesetzt werden. Eine exakte Latenzbestimmung ist jedoch nur mithilfe einer Messreihe möglich.

Die für den *HdM-Headtracker* angegebenen abgeschätzte maximale Latenz von 20 ms steht in Übereinstimmung mit in der Literatur gefundenen Daten für ähnliche binaurale Rendering-Systeme; beispielsweise der Latenz eines Polhemus FasTrak (120 Hz Update-Rate) bei der Nutzung im BRS-System, welche mit 20 ms unter Einschluss aller Asynchronitäten zwischen

Headtracker, Microcontroller und DSP angegeben wird(Rathbone, 2000). Prinzipiell erscheint dieser Wert zur Erreichung einer ausreichenden TSL nicht als kritisch.

Neben des Reaktionsvermögens des Headtracker-Systems, welche durch die Interaktion aller Bestandteile der Signalkette eine mTSL annimmt, ist auch dessen Genauigkeit der absoluten Orientierungsangabe ein Faktor, welcher die Qualität des Systems bestimmt. Dabei ist vorrangig der Sensor bzw. die IMU zu betrachten, welche neben Aspekten des Ausflösungsvermögens (ausgedrückt als Anzahl von bits) auch hinsichtlich der Genauigkeit (Unsicherheit einer Orientierungsbestimmung) untersucht werden muss. Das vergleichbare System *Mr. Headtracker*, welches am *Institute of Electronic Music and Acoustics (IEM) Graz* entwickelt wurde, nutzt ebenfalls das *Adafruit BNO055* mit einem *Arduino Mini* unter Nutzung von Quaternionen und eine Datenübertragung via MIDI in 14-bit, die über einen *Arduino USB2Serial* realisiert wird. Untersuchungen zeigten hier eine gemittelten Abweichung der Orientierungsdaten von 0,5° bis 2,5°, wobei bei Drehungen um die z-Achse (*yaw*), welche für das System dieser Arbeit von Relevanz ist, sogar nur Abweichungen von 0,5° festgestellt werden. Diese Genauigkeit ist unter Beachtung der genutzten BRIR-Raster-Auflösung von 1,8° und der menschlichen Differentielle Wahrnehmbarkeitsschwelle (Just Noticeable Difference) (JND) für das Richtungshören in der Horizontalebene von 1° (Perrott & Saberi, 1990) mehr als ausreichend.

5.6.2.3 Allgemeine Einstellmöglichkeiten

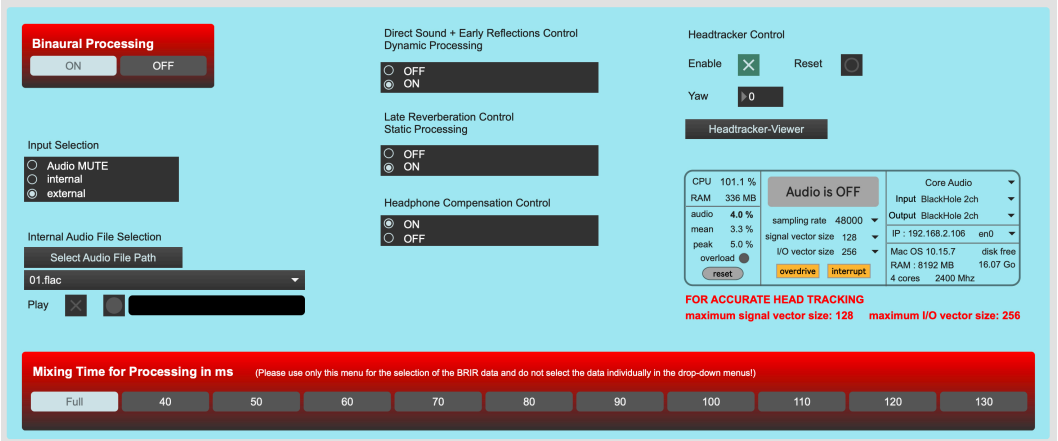

Abbildung 5.23: Allgemeine Einstellmöglichkeiten und Interfacing in die Flexible Auralisationsumgebung (eigene Darstellung)

Der erste Teil der Benutzeroberfläche der flexiblen Auralisationsumgebung bieten dem Anwender verschiedene allgemeine Einstellmöglichkeiten: Neben der Möglichkeit, die Simulation der virtuellen Lautsprecher komplett zu umgehen (*Binaural Processing: ON/OFF*), kann die dynamische sowie statische Verarbeitung der BRIRs jeweils separat deaktiviert werden (*Direct Sound + Early Reflections Control, Dynamic Processing: ON/OFF* sowie *Late Reverberation Control, Static Processing: ON/OFF*) und auch die Kopfhörer-Entzerrung aktiviert und deaktiviert werden (*Headphone Compensation Control: ON/OFF*). Dies bietet dem Anwender Möglichkeiten, den Einfluss der dynamischen als auch statischen Verarbeitung getrennt voneinander zu hören und zu evaluieren; genauso kann auch der Einfluss der

Kopfhörer-Entzerrung beurteilt und je nach Anwendungsfall ein- oder ausgeschaltet werden. Im default-mäßigen Zustand nach Öffnen der Anwendung sind sowohl die dynamische als auch die statische Verarbeitung und die Kopfhörer-Entzerrung aktiviert. Der *Headtracker Control* genannte Bereich bietet einen Schalter *Enable*, welcher das Ein- und Ausschalten des Headtrackings ermöglicht, sowie einen Button *Reset*, der die aktuelle Kopforientierung im übertragenen Sinne der weiteren binauralen Verarbeitung als `ListenerView` = (1 0 0) definiert und folglich das Listener-bezogene Koordinatensystem verschiebt; somit kann die echte Kopforientierung der Kopforientierung im simulierten Lautsprechersetup beliebig angepasst werden. Vielmehr dient dieser Button allerdings dem Zurücksetzen von Sensor-bedingten Drifts, die das Listener-Koordinatensystem ungewollt verschieben. Eine Anzeige des Gier-Winkels (engl. *yaw*) zeigt die momentane Kopforientierung entlang der z-Achse des Listener-bezogenen Koordinatensystems; weitere Drehwinkel in Form von Kardan-Winkeln wie der Nick- (engl. *pitch*) und der Roll-Winkel (engl. *roll*) können vom benutzten *HdM-Headtracker* zwar auch aufgelöst und angezeigt werden, sind jedoch für das konzipierte System aufgrund der Nutzung eines BRIR-Datensatzes, der lediglich BRIRs für 1,8°-aufgelöste Kopforientierungen in der Horizontalebene beinhaltet und diese anhand der durch das Headtracking-System dem Rendering-System übergebenen Kopforientierungsdaten faltet, nicht von Relevanz. Über den Button *Headtracker-Viewer* ist eine Darstellung der aktuellen Kopforientierung in einem Pop-up-Fenster möglich. Über den sogenannten *spat5.monitor* lassen sich allgemeine Einstellungen zur Audiosignalverarbeitung tätigen, wie die Wahl des genutzten Audiotreibers, des Audio-Eingabe- und Audio-Ausgabegeräts sowie der genutzten Abtastrate, Signalvektorgröße und der I/O-Vektorgröße (bzw. I/O-Audiobuffergröße); auch der *Overdrive* sowie *Audio Interrupt*-Modus ist an- und abwählbar. Wichtige Einstellungen wie die Abtastrate, Signalvektorgröße, I/O-Vektorgröße als auch die Aktivierung des *Overdrive* sowie *Audio Interrupt*-Modus werden default-mäßig beim Öffnen der Anwendung richtig eingestellt, da sie für die Funktion der Anwendung kritisch sind; folglich sollten sie auch nur unter Beachtung der damit verbundenen Abhängigkeiten und auf eigene Gefahr verändert werden.

5.6.2.4 Interfacing der BRIR-Daten und der Lautsprechersignale

Wie in Unterunterabschnitt 5.5.2.3 beschrieben, werden die BRIR-Datensätze aus der Postprocessing-Umgebung in Form des Datenformats *SOFA* gespeichert, sodass sie von der flexiblen Auralisationsumgebung eingelesen werden können. Die gewählte Convention ist hierbei die *SimpleFreefieldHRIR-Convention*, da sie vom *spat5.binaural* Objekt, welches als Processingcore für die dynamische Verarbeitung gewählt wurde, eingelesen werden können. Der späte Anteil der BRIR bei frontaler Kopfausrichtung, welcher entsprechend der in der Postprocessing-Umgebung gewählten Mixing Times vom frühen Anteil getrennt wurde, liegt nach der Trennung als statischer Anteil in *.wav*-Files vor. Diese können vom *spat5.conv* Objekt geladen werden, welches die statischen Faltungsoperationen in der Implementierung umsetzt.

Die BRIR-Daten müssen vor der ersten Nutzung sowohl für die dynamische Verarbeitung, als auch für die statische Verarbeitung eingeladen werden; dies geschieht praktisch durch jeweils einen Button, der konkret auf die jeweilige Verarbeitung hinweist (*1-L _ Select Dynamic SOFA-File Path*, *2-R _ Select Dynamic SOFA-File Path*, *1-L _ Select Static WAV-File Path*,

2-R _ Select Static WAV-File Path). *1-L* und *2-L* sind in diesem Fall die Bezeichnungen für die beiden in dieser Beispielanwendung simulierten Lautsprecher *L (links)* und *R (rechts)* nach DIN 15996 bzw. ITU-R BS.775. Nach Betätigung eines solchen Buttons kann ein Pfad zu den gewünschten Files über ein Dialogfenster angegeben werden. Bei Befolgung des in Unterunterabschnitt 5.5.2.3 vorgestellten Benennungsschemas werden die richtigen BRIR-Datenfiles bei Auswahl der gewünschten Mixing Time im *Mixing Time for Processing in ms* genannten Menü dann automatisch eingeladen. Wichtig ist, dass die Auswahl der BRIR-Daten ausschließlich über dieses Menü erfolgt! Die Auswahl über die vier Dropdown-Menüs, welche nach Angabe der Pfade zu den Daten befüllt werden, ist zwar möglich, jedoch aufgrund der auf diese Weise ungleichen Befüllung hinsichtlich der Mixing Time nicht gewünscht, da dies bei unsachgemäßer Auswahl zu Fehlern im Rendering führt.

Der Pfad zu den Kopfhörer-Entzerrungsfiltern muss unter Anwendung eines analogen Vorgehens ebenfalls ausgewählt werden, der Name des zu betätigenden Buttons ist *Select Headphone Compensation WAV-File Path*. Die Files der Köpfhörer-Entzerrung, welche als *.wav*-Files vorliegen müssen, müssen keinem Benennungsschema folgen und können folglich beliebig benannt sein.

Um Lautsprechersignale in der flexiblen Auralisationsumgebung zu empfangen, muss im Menü *Input Selection* die Option *external* ausgewählt sein. Die Option *internal* wählt einen in der Anwendung implementierten Audioplayer aus, dem via Betätigung des Buttons *Select Audio File Path* ein Pfad zu den gewünschten Audiofiles über ein Dialogfenster gegeben werden kann; dies geschieht im Bereich der *Audio File Selection for internal Playback*. Im Fall der beispielhaften Anwendung für $N = 2$ Lautsprecher sind Stereo-Files zu nutzen; Mono-Files werden auch abgespielt, jedoch dann nur auf dem ersten Lautsprecher nach Society of Motion Picture and Television Engineers (SMPTE)-Reihenfolge, also dem linken Lautsprecher (L) der Simulation. Das gewünschte Audiofile für die Wiedergabe kann mit Hilfe eines Dropdown-Menüs ausgewählt werden, ein *Play*-Button steht für den Wiedergabestart zur Verfügung. Direkt neben dem *Play*-Button befindet sich des Weiteren ein Bereich für graphische Warnmeldungen, wenn das Audiofile nicht gefunden werden konnte. Bei Wahl der Option *external* im Menü *Input Selection* erfolgt das Audio-Interfacing aus einer externen Wiedergabeanwendung oder Produktionsumgebung mit Hilfe eines virtuellen Audiogeräts, das als virtuelles Audio-Ausgabegerät in der Wiedergabe- bzw. Produktionsumgebung und als virtuelles Audio-Eingabegerät in der Anwendung des binauralen Renderers fungiert. Hierbei gibt es mehrere frei verfügbare virtuelle Audiogeräte: Unter macOS ist *Soundflower*[6] eine beliebte Wahl, *BlackHole*[7] von *existential audio* ermöglicht die gleiche Funktionalität und kann ab macOS Catalina genutzt werden; *Soundflower* wird offiziell seit einiger Zeit nicht mehr unterstützt und führte bei Betriebssystemen ab macOS Catalina zunächst zu Problemen. Mit diesen beiden virtuellen Audiogeräten sind bis zu $L = 16$ Lautsprechersignale aus der gewählten Wiedergabe-/Produktionsumgebung für die binaurale Simulation möglich. Unter Windows sind ähnliche virtuelle Audiogeräte vorhanden, so kann zum Beispiel das *JACK Audio Connection Kit*[8] sowohl unter Windows als auch macOS genutzt werden. Das genutzte

[6] https://github.com/mattingalls/Soundflower
[7] http://existential.audio/blackhole/
[8] https://jackaudio.org/

virtuelle Audio-Eingabegerät muss in der flexiblen Auralisationsumgebung als *Input Device* ausgewählt werden; die Reihenfolge der Lautsprecherkanäle muss dabei SMPTE-Vorgaben folgen.

5.6.2.5 Dynamische Verarbeitung des Direktschalls und der frühen Reflexionen

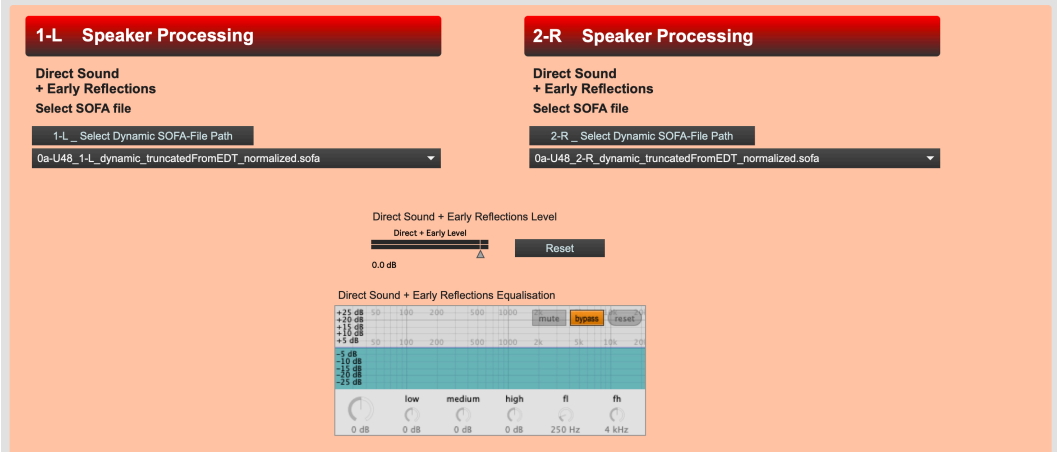

Abbildung 5.24: Benutzeroberfläche zur dynamischen BRIR-Verarbeitung der Flexible Auralisationsumgebung (eigene Darstellung)

Die Spat-5-Library stellt diverse Bausteine in Form von hochfunktionalen Objekten für die Umsetzung der auf Grundlage der Anforderungsanalyse konzipierten flexiblen Auralisationsumgebung bereit; zu nennen sind hierbei das *spat5.binaural* -Objekt, welches für die zeitvarierenden Faltungen auf Grundlage der Kopforientierung genutzt wird, sowie das *spat5.conv* -Objekt, welches für die statischen BRIR-Faltungen auf Grundlage der Wahl der Mixing Time sowie für die Kopfhörer-Entzerrung Anwendung findet. Das *spat5.conv* -Objekt kann hierbei je nach Wahl der Block- bzw. Partionierungsgrößen auf der Grundlage des genutzten Faltungsalgorithmus nach Gardner (n. d.) latenzfrei arbeiten. Es ist zu erwähnen, dass zu Beginn dieser Arbeit aufgrund der lückenhaften Dokumentation zum *spat5.binaural*-Objekt dessen Implementierung als direkte Faltung im Zeitbereich nicht bekannt war; die vom Entwickler *IRCAM* bereitgestellten Benchmark-Daten[9] des Objekts stellten zunächst seine prinzipielle Leistungsfähigkeit heraus, jedoch ohne Angabe genutzter FIR-Filterlängen. Das Objekt wurde für die Umsetzung des Systems genutzt, wobei sich im Verlauf dieser Arbeit aufgrund der hohen CPU-Last die Implementierung der Faltung als direkte Faltung im Zeitbereich herausstellte, welche auch von Thibaut Carpentier[10](Hauptentwickler und Leiter des Spat-Projekts am *IRCAM*) via persönlichem E-Mail-Kontakt bestätigt wurde. Prinzipiell ist die Nutzung einer direkten Faltung im Zeitbereich für die Audio-Funktionalität des konzipierten und implementierten Systems nicht kritisch; jedoch ist die Effizienz der Signalverarbeitung in der Anfordungsanalyse als harte Anforderung gekennzeichnet, welche so - auch im Hinblick auf die Erweiterung des Systems für viele Lautsprecher N - nicht erreicht werden kann. Dies ist als größter Kritikpunkt des Gesamtsystems zu betrachten.

[9] *Spat5-Benchmark.pdf* zu finden im Spat-5-Package nach Download unter http://forum.ircam.fr/projects/detail/spat/
[10] http://www.ircam.fr/person/thibaut-carpentier/

Die dynamische Verarbeitung des Direktschalls und früher Reflexionen je nach Wahl der Mixing Time (falls gewünscht und mit der verfügbaren Rechenkapazität möglich, können auch spätere Reflexionen und damit Teile des bzw. der vollständige Nachhall in die dynamische Verarbeitung einbezogen werden) erfolgt mithilfe je einer Instanz des Objekts *spat5.binaural* und der hinter diesem Objekt liegenden Audio-Engine. Diese Audio-Engine liefert einen Algorithmus zur dynamischen Binauralsynthese nach Abbildung 5.19. Dabei können diverse Attribute gesetzt werden, die die Verarbeitung im Detail anpassen und entsprechend der Anforderungsanalyse für dieses System interessant sind:

- *Crossfades* zwischen den parallelen Filtern, welche mit der Kopforientierung variierend genutzt werden, sind implementiert; dabei kann zwischen einem linearen, cosinusförmigen oder quadriert-cosinusförmigen Crossfade-Typ gewählt werden. Des Weiteren kann die Crossfade-Länge in Millisekunden eingestellt werden.
- *ITD-Individualisierung* kann über einen Skalierungsfaktor in Prozent (50 bis 200 Prozent) vorgenommen werden, wobei hierbei auf breitbandige Delay-Daten, welche im eingeladenen *SOFA*-File vorliegen, oder auf gängige Kopfmodelle zugegriffen werden kann. Es ist zu beachten, dass diese Funktion nur genutzt werden kann, wenn die ITD-Informationen aus den kopfbezogenen Übertragungsfunktionen als linearphasiger Anteil in Form eines echten breitbandigen Delays extrahiert worden sind und die Filter folglich in einer minimalphasigen Form vorliegen. Die variierende interaurale Verzögerung wird durch eine variable fraktionierte Verzögerungsleitung umgesetzt. Dabei sind verschiedene Interpolationsverfahren für die Umsetzung dieser Fraktionierung nutzbar, sowie die Interpolationsdauer einstellbar. Des Weiteren kann diese Funktion auch komplett deaktiviert werden, was im Zuge der Nutzung in diesem System unbedingt umgesetzt werden muss. Eine ITD-Extraktion aus BRIRs ist nicht umsetzbar; Versuche der Nutzbarmachung eines interauralen Delays können in Lindau (2014) nachgelesen werden.

Die dynamische Verarbeitung in der entwickelten flexiblen Auralisationsumgebung findet blockweise statt; dies ist wie in Unterabschnitt 5.2.5 beschrieben ein gängiges Vorgehen bei der Implementierung von zeitlich variierenden kopfbezogenen Filtern. Dabei fällt die Wahl der Audioblockgröße auf 128 Samples, welche so auch in den Systemen des BRS(Horbach et al., 1999) und von Lindau (2014) unter kritischen Abhörbedingungen genutzt wird. Unter Beachtung der Event-basierten Audiosignalverarbeitung im *Audio Interrupt*-Modus, wie sie vom genutzten *spat5.binaural* Objekt umgesetzt wird, gilt, dass hochpriorisierte Scheduler-Events wie die mithilfe eines *metro*-Objekts (Metronom, welches im Abstand eines definierten Zeitintervalls einen Impuls (Bang) sendet) alle 10 ms abgefragten seriellen Headtracker-Daten spätestens bei Erreichen des nächsten *Audio Callbacks* abgefragt, innerhalb der Zeitspanne einer Blockgröße gerendert und dann beim darauffolgenden *Audio Callback* ausgegeben werden. Dies bedeutet, dass das durch die Head-Tracker-Events bestimmte Audio zwei Audioblockgrößen nach Auftreten des Events zu hören ist. Diese genannten zwei Audioblockgrößen entsprechen 256 Samples und damit bei der genutzten Abtastrate des Systems von 48 kHz einer Länge von $5{,}\overline{3}$ ms. Dies und die Tatsache, dass die Crossfade-Länge aufgrund der hohen (und perzeptiv ausreichenden) 1,8°-genauen Raster-Auflösung des genutzten BRIR-Datensatzes und des sofortigen Event-basierten Crossfade-Schedulings auf die Hälfte der genutzten Blockgröße und

somit 1,$\bar{3}$ ms (linearer Crossfade) gesetzt wurde (vorrangig um *Switching Artefakte* bei der Filterauswahl zu verhindern), zeigt, dass das System damit *gängige* Zeiten bis zum Rendering eines neuen Filters erreicht. Wie in Unterabschnitt 5.2.5 beschrieben ist bei Lindau (2014) je die kleine erste Partionierungsgröße des Systems (128 bzw. 256 Samples) ausschlaggebend für das Filter-Update und somit steht das Letztere bei Nutzung eines linearen Crossfades über die Dauer der kleinen Partionierung maximal 5,8 ms bzw. 11,6 ms nach Änderung der Kopforientierung zur Verfügung. Es ist zu erkennen, dass das entwickelte System rein signalverarbeitungstechnisch sogar ein wenig reaktiver als das System von Lindau (2014) ist, eine Erhöhung der Audioblockgröße auf 256 Samples scheint möglich. In wissenschaftlich nicht auswertbaren Hörversuchen des Systems dieser Arbeit hat sich jedoch bei schnellen ruckartigen Kopfbewegungen eine leichte, wenn auch wahrnehmbare Verschlechterung der Lokalisierung von virtuellen Lautsprechern eingestellt, wenn die Audioblockgröße auf 256 Samples erhöht wurde; dies führte bei schnellen Kopfbewegung zu einem kurzen *Stehenbleiben* der virtuellen Lautsprecher. Es sei jedoch kritisch angemerkt, dass solche schnellen ruckartigen Bewegungen im normalen Anwendungsfall des Systems nicht von großer Relevanz sind. Die Audioblockgröße der dynamischen Verarbeitung entspricht der gewählten Signalvektorgröße der Max/MSP-Entwicklungsumgebung; dieser beschreibt, wie viele Samples von Max/MSP auf einmal blockweise verarbeitet werden.

Die Trennung zwischen der dynamischen Verarbeitung, welche mit Hilfe der Headtracker-Daten zeitlich variierend umgesetzt wird, und der statischen Verarbeitung der späten Anteile der binauralen Raumimpulsantworten auf Grundlage des Prinzips des *wahrnehmbaren Mixing Time* findet durch Auswahl der Letzteren in einem Menü statt. In diesem *Mixing Time for Processing in ms* genannten Menü ist die erste Auswahlmöglichkeit in Form eines Buttons *Full* genannt, was einer vollständig dynamischen Verarbeitung der in den *SOFA*-Files vorliegenden BRIRs entspricht. Diese Verarbeitung ist aufgrund der direkten Faltung im Zeitbereich, wie sie das *spat5.binaural* Objekt umsetzt, bei der gewählten Länge der BRIRs auf Grundlage der Nachhallzeit *RT60* von Tonregieräumen/Mischkinos im Bereich von 0,2 bis 0,4 s auch für einen leistungsstarken Computer für zwei virtuelle Lautsprecher $N = 2$ wie in der Beispielanwendung nicht umzusetzen; jedoch soll diese Möglichkeit prinzipiell zur Verfügung gestellt werden. Abstufungen der wählbaren *Mixing Time* sind in den darauffolgenden Wahlmöglichkeiten des Menüs von *40 ms* beginnend bis zu *130 ms* möglich. Diese Auswahl begründet sich auf Forschungsergebnisse wie sie in Unterabschnitt 5.2.7 zu lesen sind, welche eine Abschätzung zulassen, dass in den für die Simulation genutzten Räumen eine Mixing Time von minimal *40 ms* einen sinnvollen Wert darstellt, der erreicht werden kann. Höhere *wahrnehmbaren Mixing Times* bis *130 ms* entsprechen wissenschaftlich gefundenen Werten für größere Räume(Lindau, 2014), sodass auch diese über das Menü ausgewählt werden können - sei es, um die *wahrnehmbaren Mixing Time* an den tatsächlich simulierten Raum anzupassen, für psychoakustische Experimente oder um verfügbare Rechenleistung zugunsten einer *möglicherweise geringfügig* authentischeren Auralisation auszureizen. Eine Erweiterung dieser Wahlmöglichkeiten hin zu kleineren und/oder größeren Mixing Times bei Nutzung der gleichen Menüstruktur ist möglich, soll jedoch an dieser Stelle nicht weiter erläutert werden. Sollte die Option *Full* im Menü ausgewählt worden sein, werden die DSP-Ressourcen der statischen Verarbeitung der BRIRs freigegeben, da diese aufgrund der dann

vollständig dynamischen Verarbeitung über die komplette Länge der BRIRs nicht umgesetzt wird.

Das System dieser Arbeit verzichtet aus rein praktischen Gründen zunächst auf eine Überblendung zwischen der dynamischen und statischen Verarbeitung, welche weiteren MATLAB-Implementierungsaufwand in der Postprocessing-Umgebung des Systemmoduls 2 bedeutet. Typische Überblendungen in vergleichbaren Systemen sind mit kurzen linearen (kleinste Blockgröße der dynamischen Verarbeitung, 5,8 ms) oder cosinus-förmigen (1,3 ms sowie auch 20 ms) Crossfades umgesetzt (siehe Unterabschnitt 5.2.7), die eine nicht hörbare Trennung garantieren sollen. Der momentane Entwicklungsstand ohne Überblendung ist jedoch ebenfalls frei von hörbaren Artefakten und kann aufgrund der in Unterunterabschnitt 5.6.2.6 beschriebenen sample-genauen Delayline zwischen dynamischer und statischer Verarbeitung ein exaktes Aneinanderfügen der beiden Teile garantieren, sodass diese Überblendung als Ausblick für die weitere Arbeiten am System gegeben werden soll.

5.6.2.6 Statische Verarbeitung der späten Reflexionen und des Nachhalls

Abbildung 5.25: Benutzeroberfläche zur statischen BRIR-Verarbeitung der Flexible Auralisationsumgebung (eigene Darstellung)

Neben dieser dynamischen, anhand der Orientierung des Kopfes des Hörers bestimmten zeitlich variierenden Filterung der Lautsprechersignale mit den Anteilen der BRIRs, welche den Direktschall und die frühen Reflexionen beinhalten, werden die *gleichen Lautsprechersignale* mit den Anteilen der späten Reflexionen und somit des diffusen Nachhalls gefiltert. Diese Filterung wird auf Grundlage des Prinzips des *wahrnehmbaren Mixing Time* jedoch statisch und somit nicht zeitlich variierend umgesetzt. Dafür sind im *Systemmodul 2: Postprocessing der BRIRs* die BRIRs anhand frei wählbarer Mixing Times getrennt worden, wobei in der beispielhaften Auralisationsumgebung Mixing Times von *40 ms* bis *130 ms* in Schrittweiten von *10 ms* genutzt werden. Während für die dynamische Verarbeitung die BRIRs im Datenformat *SOFA* vorliegen, werden für die statische Verarbeitung *wav*-Files genutzt. Diese *wav*-Files sind wie auch die *SOFA*-Files nach dem in Unterunterabschnitt 5.5.2.3 beschriebenen Benennungschema benannt und werden nun entsprechend der Wahl der Mixing Time im Menü *Mixing Time for Processing in ms*gleichförmig zu den *SOFA*-Files in das Faltungsobjekt *spat5.conv* eingeladen. Es sei an dieser Stelle nochmals erwähnt, dass immer ein *SOFA*- und ein *wav*-File je Lautsprecher für eine bestimmte Mixing Time (sowie auch bei Nutzung

der vollständig dynamischen Verarbeitung - dann entsprechen die *wav*-Files für die statische Verarbeitung leeren Files, die keine Information enthalten und folglich auch zu keiner Signalausgabe des Faltungsobjekts führen) zusammengehören und so im Dateinamen auch kodiert sind. Die statische Filterung erfolgt je Lautsprecher mit der BRIR der frontalen *ungedrehten* Kopfausrichtung, die einem Azimut-Winkel von 0° entspricht. Es wird die statische Filterung also für jeden Lautsprecher durchgeführt und nicht nur eine BRIR für alle Lautsprechersignale nach Erreichen der Mixing Time benutzt; dies ist aus der von Lindau (2014) beschrieben Hörbarkeit der Veränderung von Emitter- und Listener-Positionen auch nach der *wahrnehmbaren Mixing Time*, die auf Grundlage unterschiedlichen Kopforientierungen beruht, begründet. Somit werden modale Eigenschaften des Raumes, sowie Effekte, die durch Wandnähe auftreten, für jeden Lautsprecher berücksichtigt.

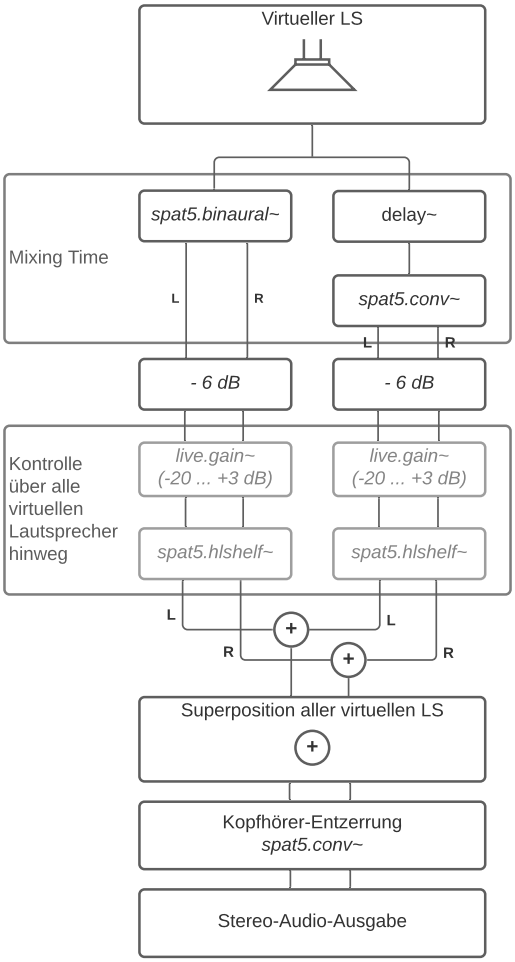

Abbildung 5.26: Vereinfachter Signalfluss eines virtuellen Lautsprechersignals in Max/MSP (eigene Abbildung)

Während die Audioblockgröße der dynamischen Verarbeitung nicht beliebig gewählt werden kann, um ein ausreichendes Reaktionsvermögen zu gewährleisten, ist die statische Verarbeitung prinzipiell unabhängig von der Kopfbewegung. Trotzdem kann die Audioblockgröße auch in dieser Faltungseinheit nicht beliebig gewählt werden; es muss zu einer Anpassung der dynamischen und statischen Verarbeitung insofern kommen, dass die Zusammenführung bzw. Addition der beiden Signale so vollzogen wird, dass keines der beiden Signale zum an-

deren versetzt ist, sondern es zu einer sample-genauen Verbindung der beiden kommt. In Abbildung 5.26 ist der Signalweg der Verarbeitung eines Lautsprechersignals exemplarisch und vereinfacht dargestellt. Das Lautsprechersignal fließt in zwei parallele Pfade, wobei ein Pfad zum *spat5.binaural* Objekt und der andere Pfad zum *spat5.conv* Objekt führt. Da die Audioblockgröße des *spat5-binaural* Objekts der Signalvektorgröße der Max/MSP Entwicklungsumgebung entspricht, wird durch das Objekt keine weitere Latenz verursacht, welche nicht bereits durch die genutzte Signalvektorgröße der Entwicklungsumgebung bestimmt ist. Somit wird im Pfad der dynamischen Verarbeitung bzw. im *spat5-binaural* Objekt keine weitere Latenz verursacht. Der zweite Pfad ist der der statischen Verarbeitung des diffusen Nachhalls: Da die genutzte BRIR in diesem Pfad der akustischen Antwort des Raumes nach der gewählten Mixing Time entspricht, muss diese Faltung zum dynamischen Teil bzw. Pfad verzögert werden. Dies geschieht mit einer samplegenauen Delayline, die mithilfe des *delay* Objekts umgesetzt wird. Diese Delayline ist jedoch nicht direkt auf die Anzahl der Samples, welche der Mixing Time unter Beachtung der gewählten Abtastrate von 48 kHz entsprechen, eingestellt, sondern nimmt die Latenz, welche durch die Blockgröße der gewählten statischen Faltung entsteht, mit in Betracht. Diese Faltung entspricht einer FFT-basierten partionierten schnellen Faltung (Overlap-Save) im Frequenzbereich; das Objekt kann dabei eine gleichmäßig oder ungleichmäßig partionierte Faltung umsetzen. Die Faltung ist aufgrund der ohnehin nötigen Verzögerung zum dynamischen Teil nicht zeitkritisch, sodass prinzipiell eine beliebige Partionierung gewählt werden kann. Die gewählte Partionierung in der Implementierung ist ungleichmäßig, wobei die erste Partionierungsgröße 2048 Samples entspricht (die im Objekt auch als *Blockgröße* der Faltung bezeichnet wird). Die Blockgröße wird im Vergleich zu den 128 Samples der dynamischen Verarbeitung groß gewählt, um die Rechenlast an dieser Stelle nicht unnötig zu erhöhen. Die daraus resultierende Latenz bestimmt sich aus der gewählten Blockgröße und dem Max/MSP-Signalvektor folgendermaßen: *latenz = blockgröße - signalvektorgröße*. Mit Hilfe dieser Information und der signalverarbeitungstechnisch benötigten sample-genauen Verzögerung, die sich aus der gewählten Mixing Time bestimmt, kann nun der Wert einer Delayline bestimmt werden, welcher sich aus dem Verhältnis zwischen signalverarbeitungstechnisch benötigter Verzögerung und der Latenz der gewählten Faltung aufgrund der gewählten Audioblockgröße ergibt. Folgende zwei Beispiele sollen zum besseren Verständnis gegeben werden: Bei einer Mixing Time von *40 ms* muss das Audiosignal eines virtuellen Lautsprechers im Pfad der statischen Faltung um 40 ms resp. 1920 Samples (bei 48 kHz Abtastrate) verzögert werden. Nun verursacht das *spat5.conv* mit der Wahl der Blockgröße (und damit ersten kleinen Partionierungsgröße) von 2048 Samples bei einer Signalvektorgröße von 128 Samples eine Latenz von 1920 Samples, also genau 40 ms. Damit muss das Lautsprechersignal im Pfad der statischen Faltung überhaupt nicht direkt verzögert werden, da die Verzögerung durch die Latenz der Faltungsoperation verursacht und dem Signal damit aufgeprägt wird. Wählt man nun jedoch eine Mixing Time von 80 ms, so reichen die 40 ms Latenz der Faltungsoperation nicht mehr aus, um das Signal ausreichend zu verzögern. Somit wird dem Signal zusätzlich vor der Faltungsoperation noch eine Verzögerung von exakt 40 ms resp. 1920 Samples (bei 48 kHz Abtastrate) mit Hilfe des *delay* Objekts aufgeprägt; so erreicht das statisch mit einer einzelnen BRIR gefaltete Lautsprechersignal vor der Addition mit dem dynamischen Teil die nötige Verzögerung, um eine sample-genaue

Zusammenführung der beiden Signale zu garantieren. An dieser Stelle muss erwähnt werden, dass das *delay* Objekt unabhängig von der Signalvektorgröße sample-genau arbeitet und die Verzögerungen somit sample-genau ausführen kann. Die Verzögerung wird durch Statusabfrage des Max/MSP-DSP hinsichtlich der Abtastrate der Signalverarbeitung und der gewählten Signalvektorgröße sample-genau angepasst. Somit sind prinzipiell auch andere Abtastraten als 48 kHz sowie andere Signalvektorgrößen unter Beibehaltung der Funktionalität möglich, jedoch muss beachtet werden, dass die hervorgerufene Latenz des *spat5.conv* Objekts bei einer Signalvektorgröße von 64 Samples $41{,}\overline{3}$ ms (48 kHz Abtastrate) entspricht und folglich die Mixing Time von *40 ms* nicht mehr umgesetzt werden kann; größere Signalvektoren führen zu geringeren Latenzen und sind folglich unkritisch.

Neben diesen genannten Signalverarbeitungsschritten, welche die Hauptfunktion der Anwendung als binaurale Faltungs-Engine garantieren, zeigt Abbildung 5.26 weitere Verarbeitungsschritte: Nach der dynamischen als auch statischen Faltung wird eine Pegelabsenkung von 6 dB durchgeführt, welche bei der anschließenden Superposition des Signals des dynamischen Pfades mit dem Signal des statischen Pfades sowie bei der beispielhaften Superposition in der Beispielanwendung von zwei Lautsprechern genug *Headroom* garantieren soll. Hierbei ist auch die anschließende Möglichkeit der Erhöhung des Pegels um 3 dB mithilfe eines *live.gain* Objekts in beiden Pfaden beachtet. Starke Pegelanhebungen in bestimmten Frequenzbereichen, die durch den Kuhschwanzfilter (umgesetzt über ein *spat5.hlshelf* Objekt) theoretisch möglich sind, werden hierbei jedoch nicht mit einbezogen und können folglich zu *Clipping* führen. Die beiden letztgenannten Funktionalitäten entstehen aus weichen Benutzeranforderungen: So ist es möglich, mithilfe zweier Fader für je den dynamischen und den statischen Signalanteil aller Lautsprecher ein benutzerdefiniertes Mischungsverhältnis zwischen dem Direktschall und den frühen Reflexionen zu den späten Reflexionen bzw. dem diffusen Nachhall einzustellen, indem in beiden Bereichen die Signalpegel erhöht und erniedrigt werden können (siehe *Direct Sound + Early Reflections Level* und *Late Reverberation Level*, sowie jeweils einem *Reset*-Button zur Zurücksetzung auf 0 dB); dies geschieht mithilfe von *live.gain* Objekten. Des Weiteren ist es möglich für alle virtuellen Lautsprecher über einen Kuhschwanzfilter eine benutzerdefinierte Entzerrung vorzunehmen (siehe *Direct Sound + Early Reflections Equalisation* und *Late Reverberation Equalisation*); diese Kuhschwanzfilter befinden sich default-mäßig im Bypass.

5.6.2.7 Kopfhörer-Entzerrung

Die Kopfhörer-Entzerrung wird in der prototypisch implementierten Anwendung ebenfalls mit einem *spat5.conv* Objekt umgesetzt, indem die Impulsantworten der Filter als *.wav*-File eingeladen werden. Dazu muss zunächst der Pfad zu den Files ausgewählt werden, was über den Button *Select Headphone Compensation WAV-File Path* möglich ist. Nach Angabe des Pfades sind über ein Dropdown-Menü alle im Pfad gefundenen Filter anwählbar. Der Betragsfrequenzgang sowie Phasengang des genutzten Filters wird dargestellt; dies ermöglicht eine Einschätzung des Filters und dessen Einfluss auf das Klangbild. Die Latenz der Faltungsoperation wird an dieser Stelle minimiert, indem die Blockgröße und damit erste Partionierungsgröße des *spat5.conv* Objekts der Signalvektorgröße entspricht; somit arbeitet das

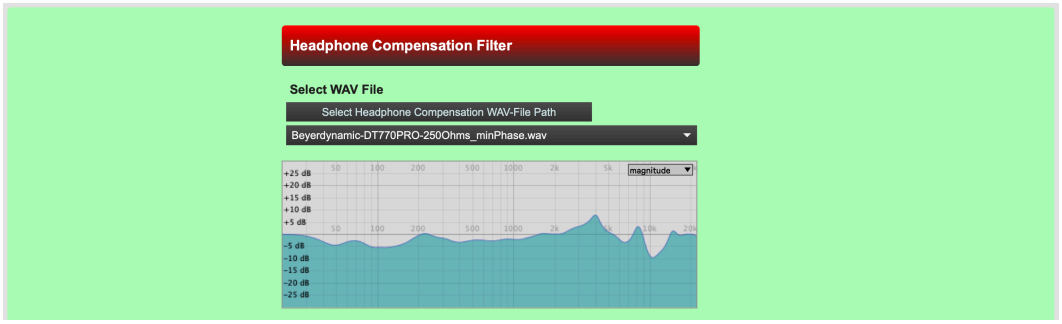

Abbildung 5.27: Benutzeroberfläche zur Kopfhörer-Entzerrung in der Flexible Auralisationsumgebung (eigene Darstellung)

Objekt latenzfrei. Der Wert der Signalvektorgröße wird abgefragt und dem Objekt übergeben. Neben diesen Parametern der Implementierung wird des Weiteren eine minimalphasige Filterform empfohlen, welche sich in Hörversuchen Filterformen mit unbeschränktem gemischten Phasenverhalten als perzeptiv gleichwertig herausgestellt haben (Lindau, 2014); hierbei ist jedoch anzumerken, dass in diesen Hörversuchen eine andere Inversionsmethode (Hochpass-regulierte Least-Mean-Square (LMS)-Inversion) genutzt wurde. Damit wird die Latenz weiter minimiert. Die Kopfhörer-Entzerrungsfilter, welche in diesem System genutzt werden und dem System beiliegen, wurden im Gegensatz zu den BRIRs nicht selbst gemessen. Sie stammen aus einem frei verfügbaren Datensatz von Kopfhörer-Entzerrungsfiltern gängiger Kopfhörer[11], die von Bernschütz (2013) auf einem *Neumann KU 100* Kunstkopfmikrofon gemessen wurden; damit ist die *Schnittstellenkorrektur* zum genutzten Kunstkopfmikrofon der BRIR-Messungen sichergestellt. Diese Filter wurden in einer Messreihe bestimmt, in der jeder Kopfhörer 12 Mal auf- und wieder abgesetzt wurde, um Abhängigkeiten des Kopfhörersitzes zu mitteln; ein halbautomatischer Log-Spline-Inversionsalgorithmus wurde zur Filtererstellung genutzt, die Filter liegen in minimal- und linearphasiger Form vor. Etwaige Einflüsse eines anderen Produktionszeitraums des Kunstkopfmikrofons oder des von Bernschütz (2013) genutzten Messequipments können im Bezug auf eine einwandfreie *Schnittstellenkorrektur* nicht ausgeschlossen werden, werden jedoch als unwahrscheinlich erachtet.

5.6.2.8 Latenz-Betrachtung

Wie die Ausführungen in dieser Arbeit zeigen, ist für eine plausible und authentische Wahrnehmung bei binauraler Simulation neben einem bestmöglichen Abgleich der interauralen und monoauralen Cues der Simulation mit denen des Hörenden unter Beachtung der *Schnittstelle* bei Wiedergabe über einen bestimmten Kopfhörer die Latenz des Gesamtsystems eine wichtige Komponente, um die dynamischen Fähigkeiten unseres Hörsinns nicht aus der Hörwahrnehmung auszuschließen bzw. sie ausreichend genau zu ermöglichen. Hierbei spielen diverse Systemkomponenten eine Rolle, welche nur bei Betrachtung des komplexen Zusammenwirkens zu einer (minimalen) Gesamtlatenz, der mTSL, des Systems führen. Diese mTSL kann nur durch eine Messreihe und eine Mittelung der gemessenen Werte hinreichend genau bestimmt werden (Wenzel, 1997). Auch wenn eine solche Messreihe im Rahmen dieser Arbeit nicht durchgeführt wurde, soll nun abschließend eine Abschätzung bei Betrachtung aller

[11] http://audiogroup.web.th-koeln.de/wdr_irc.html

Systemkomponenten (beginnend beim Head-Tracking-System bis zur Audioausgabe über das Audiointerface) hinsichtlich der Systemlatenz gegeben werden.

Der *HdM-Headtracker* bietet eine Update-Rate von 100 Hz. Diese Head-Tracker-Daten werden im `headtrackerReceiver` Subpatch alle 10 ms und somit mit einer Update-Rate von 100 Hz abgetastet. Eine vollständige Asynchronität dieser Update-Raten führt zu einer effektiven Update-Rate von 50 Hz. Somit ist - unter vereinfachter Betrachtung des kompletten Head-Tracking-Systems mithilfe dieser beiden Update-Raten - spätestens 20 ms nach Auftreten einer (tatsächlich) abgetasteten Kopforientierung diese Information auch in der Max-Umgebung als *Message* vorhanden, die im *Message Queue* (siehe Abbildung 5.22) gelistet werden. Es ist - wie bereits dargelegt - in diesem Abschnitt des Gesamtsystems jedoch noch mit weiteren Latenzen aufgrund der seriellen Datenübertragung zu rechnen. Unter der Annahme, dass die Bildung dieser *Message Queue* keine weitere Latenz verursacht, kann als nächstes die Event-Terminierung im *spat5.binaural* Renderingcore, der die dynamische Anpassung in Form von zeitlich variierenden Filter umsetzt, betrachtet werden. Dieser Renderingcore arbeitet mit einer Audioblockgröße von 128 Samples, welche auch dem Signalvektor des Max-Entwicklungsumfelds entspricht. Somit wird durch diesen fernab der dynamischen Verarbeitung keine weitere Latenz verursacht. Jedoch sollte die Laufzeit in den BRIRs bis zum eigentlichen Impuls beachtet werden, welche in der Faltungsoperation ebenfalls eine Latenz verursacht. Die beispielhaft gemessenen BRIR-Datensätze wurden mit einer `onsetprotectionseconds` von 10 ms (Dauer bis zum Erreichen des betragsmäßig enthaltenen Maximums in den BRIRs) gespeichert, was zu einer zusätzlichen Latenz von etwa 10 ms an dieser Stelle der Verarbeitung führt. Wie in Unterunterabschnitt 5.6.2.5 beschrieben werden die Filterwechsel 5,3 ms nach dem Auftreten der durch das Head-Tracker-Event verursachten *Message*, die sich in der FIFO *Message Queue* befinden, ausgegeben. Somit kann vom Auftreten einer abgetasteten Kopforientierung bzw. deren Änderung bis hin zum abgeschlossenen daraus folgenden dynamischen Audiorendering eine minimale Latenz des Systems von etwa 35,3 ms geschätzt werden. Latenzen, die durch den minimalphasigen Kopfhörer-Entzerrungsfilter entstehen, sind ausreichend klein und werden als irrelevant betrachtet. Mit dem vom Autor genutzten Audiointerface *Audient EVO 4* konnte für die Audioausgabe eine minimale I/O-Buffergröße (I/O-Vektorgröße) von 256 Samples ohne wahrnehmbare Artefakte genutzt werden. Dies führt rein aus der Betrachtung des Output-Buffers auf eine weitere Latenz von 5,3 ms; die gesamte Output-Latenz ist jedoch aufgrund weiterer Latenzen, die durch den Audiotreiber und etwaige Einflüsse durch elektronische Komponenten hervorgerufen werden, als größer einzuschätzen. Eine Messung der Roundtrip-Latenz des genutzten Audiointerfaces zur Abschätzung der Latenz, die durch die Ausgabe hervorgerufen wird, wird vom Autor für jeden Anwender empfohlen, konnte jedoch aus Zeitgründen selbst nicht mehr umgesetzt werden. Betrachtet man erreichte mTSL wie 43 ms (Lindau, 2014), 50 ms (Horbach et al., 1999) oder 41,2 ms (C. W. Pike, 2019), sollte ein ähnlicher Wert jedoch auch bei hoher Output-Latenz des genutzten Interfaces mit dem Gesamtsystem und den gewählten Parametern der Implementierung erreicht werden können. Nimmt man mögliche nicht beschriebene Einflüsse in der Latenz des Head-Tracking-Systems außer Acht, kann die komplette Output-Latenz auf einen Wert von bis zu 15 ms ansteigen und eine ähnliche Performance wäre erreicht. Unter Betrachtungen von Werten der gerade wahrnehmbaren Latenz TSL von 85

ms (Darstellung von virtuellen Lautsprechern) (Horbach et al., 1999) oder 114 ms (Virtuelle auditorische Umgebung (VAE)) (Lindau, 2014) scheinen die erreichten Werte noch weniger problematisch.

6 Fazit

Diese Arbeit hatte zur Aufgabe ein funktionsfähiges System zur binauralen Simulation von Regieräumen (und Mischkinos) zu entwerfen und prototypisch zu implementieren. Dieses System sollte dabei als Gesamtsystem fungieren, welches alle nötigen Schritte zur Umsetzung in entsprechenden Räumen an der Hochschule der Medien Stuttgart betrachtet.

Unter diesem Vorhaben wurden zunächst die grundlegenden Prinzipien des räumlichen Hörens, sowie die Verfahren der Darstellung bzw. Simulation von Letzterem in knapper Form recherchiert und diskutiert. Aus dieser Diskussion stellte sich das datenbasierte Rendering von virtuellen Lautsprechern als für den Anwendungszweck passend heraus; das System des *Binaural Room Scanning* stand hierbei als erstes System dieser Art unter besonderer Aufmerksamkeit der Betrachtungen. Um den Stand der Technik aller Systemkomponenten eines solchen Systems zu prüfen, wurden in einer ausführlichen methodischen Analyse verschiedene Parameter solcher Systeme (als auch vergleichbarer Systeme der Darstellung von VAE) untersucht und die Letzteren in ein flexibles Gesamtsystem überführt, welches sich aus den Systemmodulen 1 bis 3 zusammensetzt. Für jedes der drei Systemmodule wurden funktionale Anforderungen als auch Benutzeranforderungen definiert, welche eine bestmögliche Nutzbarkeit an der Hochschule der Medien Stuttgart im Rahmen der Forschung und Lehre garantieren. Die Umsetzung der Systemmodule 1 und 3 erfolgte in der Entwicklungsumgebung Max/MSP unter Nutzung der Spat-5-Library von IRCAM, während Systemmodul 2 in MATLAB realisiert wurde. Während der Umsetzung aller Module wurden iterativ mehrere Schritte gegangen, die schlussendlich zu den nun vorliegenden Systemmodulen geführt haben, welche die wichtigen funktionalen Anforderungen prinzipiell erfüllen können.

Systemmodul 1: Messsystem zur Messung von BRIRs bietet die Möglichkeit der automatisierten Messung von orientierungsabhängigen Impulsantworten, wie sie auch in diesem System zur Darstellung virtueller Lautsprecher (ein BRIR-Datensatz für jeden Lautsprecher) benötigt werden. Die für dieses Messsystem konzipierte und prototypisch implementierte Messsoftware bietet neben gängigen Einstellungen der akustischen Sweep-basierten Messung auch Möglichkeiten zur direkten Kontrolle der Messung und deren Messdaten. Es können Impulsantwort-Datensätze in flexiblen horizontalen Raster-Auflösungen für mehrere Lautsprecher automatisiert gemessen werden, wobei das entwickelte Drehsystem (in der momentanen Implementierung) mithilfe von Mikroschritten des zugrundeliegenden Schrittmotors eine theoretische Schrittweite des Rasters von bis zu 0,03° ermöglicht. Bezieht man jedoch den ermittelten Abbildungsfehler des Drehsystems von 0,25° unter Bezugnahme des beispielhaft gemessenen 2.0-Stereosetups ein, so ist die Wahl einer solchen Auflösung wenig sinnvoll. Des Weiteren ist diese unter Aspekten der benötigten horizontalen Raster-Auflösung für eine plausible binaurale Simulation nicht nötig, sodass BRIR-Datensätze mit einer Raster-Auflösung

von 1,8° für zwei Lautsprecher eines 2.0-Stereosetups gemessen wurden. Eine Abschätzung der erreichten Qualität der BRIR-Datensätze wurde unter Betrachtungen des SNR getätigt, wobei hier Abweichungen zu in der Literatur beschriebenen Messungen bei prinzipiell gleichen Messparametern festgestellt werden mussten; und dies, obwohl der Störschall im Raum prinzipiell als nicht relevant eingestuft werden konnte. Der erreichte SNR reicht für eine vom Autor als ausreichend empfundene Auralisation aus, jedoch konnten im zeitlichen Rahmen dieser Arbeit und aufgrund von Einschränkungen während der Corona-Pandemie keine Hörtests durchgeführt werden, welche diese Aussage prüfen. Der genutzte Raum zeigt deutliche akustische Auffälligkeiten, deren Einfluss auf die Messung geprüft werden muss. Eine Fehleranalyse des kompletten Messsystems unter Zuhilfenahme einer anderen Räumlichkeit scheint also sinnvoll.

Systemmodul 2: Postprocessing der BRIRs ist als Schnittstelle zwischen Systemmodul 1 und 3 zu verstehen, welches wenige qualitätsbestimmende Parameter des Gesamtsystems inne trägt. Trotzdem ist es für die Verarbeitung der gemessenen BRIRs hin zu echten BRIR-Datensätzen im *SOFA*-Format, welche gleichförmig links- und rechtsseitig gekürzt sowie normalisiert sind, von Relevanz. Das Systemmodul ist skriptbasiert in MATLAB umgesetzt und es werden drei Skripte benötigt, um alle Verarbeitungsschritte auszuführen. Dabei werden die BRIRs eines Lautsprechers automatisiert in die Umgebung eingeladen, sortiert und schrittweise in Form einer großen Matrix normalisiert und mithilfe von Parametern für die links- und rechtsseitige Kürzung gekürzt. Dieser Parameter für die linksseitige Kürzung wird `onsetprotectionseconds` genannt und bestimmt die Zeit, welche nach dem betragsmäßigen Maximum aller eingeladenen BRIRs (welches dem Direktschall zugehörig angenommen wird) linksseitig dieses Maximums nach der Kürzung noch vorhanden bleibt. Auf diese Weise kann die Latenz aufgrund führender leerer Samples in den BRIRs verkürzt und kontrolliert werden. Die rechtsseitige Kürzung bestimmt sich momentan lediglich aus raumakustischen Kenndaten der *RT60* bzw. *EDT*, welche entweder direkt angegeben werden können oder aber durch Funktionen innerhalb der Skripte bestimmt werden können. Eine Betrachtung unter Aspekten des Erreichens des Rauschteppichs wird nicht durchgeführt. Mithilfe der `endtruncationsafetyseconds` kann die rechtsseitige Kürzung des Weiteren noch hinsichtlich der raumakustischen Parameter verlängert werden, um einen tatsächlichen *Decay* auch tatsächlich innerhalb der gekürzten BRIRs zu erreichen. Es ist wichtig zu erwähnen, dass all diese Verarbeitungsschritte mit Parametern (`normalisationfactor`, `onsetprotectionseconds`, `endtruncationsafetyseconds`) über alle BRIRs eines gemessenen Lautsprechersetups bestimmt und ausgeführt werden müssen, damit es nicht zu Verschiebungen der BRIRs untereinander oder eine ungleichmäßigen Verstärkung im Zuge der Normalisierung kommt. Der letzte Schritt des Systemmoduls ermöglicht neben der Speicherung der BRIR-Daten in kompletter Länge in einem *SOFA*-File auch die Trennung der BRIRs und Speicherung in einem *SOFA*-File und einem *wav*-File nach Angabe von gewünschten Mixing Times, welche im nachfolgenden Systemmodul 3 verarbeitet werden sollen. Momentan sind diese ganzen Prozesse nur für zwei Lautsprecher implementiert, da die Erprobung des Systems durch ein 2.0-Stereosetup erfolgte; eine Erweiterung hin zu mehreren Lautsprechern sollte ohne größeren Aufwand möglich sein. Hierbei ist auch eine Verbesserung des Programmierstils unter Nutzung der Datenstrukturen eines *Structs* oder *Cells* für alle Lautsprecher des Systems und

anschließendem Zugriff auf die Letzteren via Indizierung wünschenswert. Etwaige Fensterungen der BRIRs oder ein Ein- und Ausblenden in Form von Fades bei der Trennung der BRIRs sind des Weiteren noch nicht implementiert und sollten vorrangig zur Vermeidung von Artefakten in der Renderingumgebung in weiteren Schritten umgesetzt werden. Systemmodul 2 ist aufgrund der Implementierung ohne Benutzerschnittstelle des Weiteren nicht sonderlich *anwenderfreundlich* und setzt somit ein gewisses Maß an Fachwissen voraus, um die Skripte richtig auszuführen. Hier können weitere Entwicklungsschritte umgesetzt werden, um die Anwenderfreundlichkeit deutlich erhöhen.

Systemmodul 3: Flexible Auralisationsumgebung bildet als binaurale Renderingumgebung den Kern des Gesamtsystems. *Flexibel* ist diese Auralisationsumgebung deshalb, da sie aus drei Grundbausteinen besteht, welche flexibel angepasst und im Zuge weiterer Entwicklungen auch getauscht werden können. Die flexiblen Anpassungmöglichkeiten beziehen sich hierbei auch auf die separate Hörbarmachung und im Pegel und durch einen Kuhschwanzfilter durchgeführte Anpassungen der einzelnen Grundbausteine. Sie wurde dabei zunächst für den Fall einer Simulation von zwei virtuellen Lautsprechern umgesetzt, um einen Systemeinstieg zu ermöglichen. Den ersten und größten Grundbaustein der Signalverabeitung der Auralisationsumgebung bildet der der *Dynamischen Verarbeitung*. Dieser liefert die zeitlich variierende Filterung bei Änderung der Kopforientierung (welche durch einen an der Hochschule entwickelten sensorbasierten Head-Tracker verfolgt wird) unter Zuhilfenahme des Direktschallanteils und der ersten Reflexionen in den BRIRs. Der zweite Grundbaustein wird *Statische Verarbeitung 1* genannt und bildet die Filterung mit den Anteilen der BRIRs, welche auf Grundlage des Parameters der *wahrnehmbaren Mixing Time* als diffus genug empfunden werden, um auf eine dynamische Anpassung der Filterung bei Änderung der Kopforientierung komplett zu verzichten. Da ein vollständig diffuses Schallfeldes in einem kleinen stark gedämpften Raum wie einem Regieraum zumeist nicht erreicht wird und aufgrund unterschiedlicher Anregungspositionen des Raumes durch jeden einzelnen Lautsprecher hörbare Unterschiede in den BRIRs zueinander auch fernab der Mixing Time bestehen, ist diese *Statische Verarbeitung 1* ebenso wie die *Dynamische Verarbeitung* für jeden Lautsprecher des Systems implementiert. Die *wahrnehmbaren Mixing Time* kann standardmäßig in einem Bereich von *40 ms* bis *130 ms* ausgewählt werden; dies entspricht Werten, welche sich in Hörversuchen verschieden großer Räumlichkeiten mit zusätzlich verschiedenen gemittelten Absorptionskoeffizienten als ausreichend herausgestellt haben. In Räumen wie Regieräumen und Mischkinos, die nach Norm eine gemittelte Nachhallzeit von 0,2 bis 0,4 s erreichen, ist eine *wahrnehmbare Mixing Time* von *40 ms* zumeist eine treffende Wahl. Da für die *Dynamischen Verarbeitung* deutlich mehr Rechenkapazitäten beansprucht werden, ist neben der Trennung anhand psychoakustischer Parameter auch eine rein pragmatische Trennung möglich, die eine Nutzung der flexiblen Auralisationsumgebung auf dem gewünschten Computer noch möglich macht. Die in der Wissenschaft genannten Parameter für das dynamische Update der kopfbezogenen Filterung können mit dem System ausreichend erreicht werden, sodass eine minimale Gesamtsystemlatenz von etwa 45 ms (BRIR-`onsetprotectionseconds`: 10 ms, Max/MSP-Signalvektorgröße: 128 Samples, I/O-Buffergröße: 256 Samples) möglich scheint. Diese muss jedoch in einer Messreihe bestimmt werden, um eine abschließenden Aussage diesbezüglich treffen zu können. Der Grundstein der *Statischen Verarbeitung 2* bildet eine latenzfreie Implementierung

von Kopfhörer-Entzerrungsfiltern, welche bestenfalls in minimalphasiger Filterform vorliegen sollten, um weitere Latenzen zu vermeiden. Die flexible Auralisationsumgebung ist auf dem Computer, der für die Entwicklung des Systems zur Verfügung stand (MacBook Pro Retina 13Ënde 2013, 2,4 GHz Dual-Core Intel Core i5, 8 GB 1600 MHz DDR3, macOS Catalina 10.15.7) bei keiner bis wenig CPU-Belastung durch andere Prozesse mit einer Mixing Time von *40 ms* artefaktfrei nutzbar. Eine Erhöhung auf eine Mixing Time von *50 ms* ist jedoch bereits nicht mehr durchführbar und führt zu deutlichen Audiodropouts in Form von Knacksern. Diese Performance ist ernüchternd, besonders auch im Vergleich mit Systemen wie in Lindau et al. (2007) (Intel CoreDuo 2 GHz Thinkpad, 2 GB RAM, 6 Quellen in 1°/5° (hor./ver.) Auflösung mit BRIRs der Länge 3 s, Blockgröße: 256 Samples), jedoch vor dem Hintergrund der direkten Faltungsoperationen im binaualen Renderingcore des *spat5.binaural* -Objekts erklärbar. Auch auf einem Mac Pro mit deutlich mehr Rechenleistung (2 x 2,66 GHz 6-Core Intel Xeon, 48 GB 1333 MHz DDR3, macOS Sierra 10.12.6) sind ebenfalls lediglich Mixing Times von *100 ms* erreichbar. Die direkte Faltung ist nur für sehr kurze Impulsantworten mit $M \leq 32$ rechnerisch effizient; diese Längen werden selbst bei einer Mixing Time von *40 ms*, welche bei einer Abtastrate von 48 kHz zu einer Länge des dynamisch zu verarbeitenden Anteils der BRIR von 1920 Samples führt, um ein Vielfaches überschritten. Somit kann die Weiterentwicklung dieses Systems hin zu einer FFT-basierten schnellen Faltung in allen Bereichen der Verarbeitung die Performance hinsichtlich der Effizienz der digitalen Audiosignalverarbeitung in sehr hohem Maße steigern. Dies muss der größte Kritikpunkt bei der prototypischen Implementierung dieses Systems sein. Des Weiteren entspricht auch die Anbindung der BRIR-Daten an die Auralisationumgebung nicht dem Standard neuerer Systeme, welche z.B. mithilfe der *MultiSpeakerBRIR*-Convention direkt eine Speicherung und Weitergabe der BRIR-Daten von Lautsprechersetups mit mehreren Lautsprechern inklusiver aller benötigten Metadaten in Form von Positions- und Orientierungsdaten in einem einzigen File ermöglichen (siehe z.B. (C. Pike & Romanov, 2017a)).

Ausführliche Hörtests, welche erneut die Plausibilität bzw. Authentizität des Systems auf subjektiver Basis evaluieren, liegen außerhalb der Möglichkeiten dieser Masterarbeit und sind aufgrund der erreichten systemtechnischen Parameter, welche sich in einschlägiger Literatur in Systemen gleicher oder ähnlicher Art als perzeptiv ausreichend erwiesen haben, auch nicht erforderlich, um die Qualität des Systems zu bewerten. Da es in dieser Arbeit primär darum ging, ein funktionsfähiges System zur Simulation und Auralisation zu entwerfen, wurden Hardfacts zu psychoakustisch motivierten Systemparametern übernommen und umgesetzt. Eine Anpassung oder Veränderung dieser Parameter auf Grundlage von perzeptiv motivierten Untersuchungen ist weiterhin möglich, war jedoch nicht Inhalt dieser Arbeit.

Literatur

33408-1, D. (1987). *Körperumrißschablonen für Sitzplätze* (Techn. Ber.). Beuth (German national standard). Berlin.

Ahrens, A., Joshi, S. N. & Epp, B. (2020). Perceptual Weighting of Binaural Lateralization Cues across Frequency Bands. *JARO - J. Assoc. Res. Otolaryngol.*, *21*(6), 485–496. https://doi.org/10.1007/s10162-020-00770-3

Algazi, V. R., Duda, R. O., Thompson, D. M. & Avendano, C. (2001). THE CIPIC HRTF DATABASE. *Proc. 2001 IEEE Work. Appl. Signal Process. to Audio Acoust.*, (October), 99–102.

Arcoverde Neto, E. N., Duarte, R. M., Barreto, R. M., Magalhães, J. P., Bastos, C. C., Ren, T. I. & Cavalcanti, G. D. (2014). Enhanced real-time head pose estimation system for mobile device. *Integr. Comput. Aided. Eng.*, *21*(3), 281–293. https://doi.org/10.3233/ICA-140462

Armelloni, E., Giottoli, C. & Farina, A. (2003). Implementation of real-time partitioned convolution on a DSP board. *IEEE Work. Appl. Signal Process. to Audio Acoust.*, *2003-Janua*(November), 71–74. https://doi.org/10.1109/ASPAA.2003.1285822

Ashby, T., Mason, R. & Brookes, T. (2013). Head movements in three dimensional localisation. *134th Audio Eng. Soc. Conv. 2013*, 361–370.

Audio Engineering Society. (2015). *AES69-2015: AES standard for file exchange - Spatial acoustic data file format* (Techn. Ber.).

Audio Engineering Society. (2020). *AES69-2020: AES standard for file exchange - Spatial acoustic data file format* (Techn. Ber.).

Băcilă, B. & Lee, H. (2019). Binaural Room Impulse Response (BRIR) Database for 6DOF spatial perception research, In *AES 146th Conv.*

Begault, D. R., Wenzel, E. M., Lee, A. S. & Anderson, M. R. (2000). Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source, In *108th AES Conv.*

Behr, S. (n. d.). *Optimierung der Akustik und der Abhörsituation im Tonstudio der TU Graz* (Techn. Ber.).

Bernschütz, B. (2013). A Spherical Far Field HRIR/HRTF Compilation of the Neumann KU 100. *Fortschritte der Akust. – AIA-DAGA 2013*, 592–595.

Bernshütz, B. (2013). MIRO - measured impulse response object: data type description.

Blauert, J. (1974). *Räumliches Hören* (1. [Blauert Jens, Hrsg.; Bd. 1). Stuttgart, Hirzel.

Boren, B. (2011). Multichannel Impulse Response Measurement in Matlab, In *AES 131st Conv.* New York.

Brandtsegg, Ø., Saue, S. & Lazzarini, V. (2018). Live convolution with time-varying filters. *Appl. Sci.*, *8*(1), 1–29.

Brigham, E. O. (1997). *FFT-Anwendungen*. Oldenbourg Verlag München.

Brinkmann, F., Aspöck, L., Ackermann, D., Lepa, S., Vorländer, M. & Weinzierl, S. (2019). A round robin on room acoustical simulation and auralization. *J. Acoust. Soc. Am.*, *145*(4), 2746–2760.

Brinkschulte, U. & Ungerer, T. (2010). *Mikrocontroller und Mikroprozessoren* (3.). Springer-Verlag Berlin Heidelberg.

Carpentier, T. (2018a). *A new implementation of Spat in Max* (Techn. Ber.). www.spatrevolution.com

Carpentier, T. (2018b). *A new implementation of Spat in Max* (Techn. Ber.). www.spatrevolution.com

Carpentier, T., Noisternig, M., Warusfel, O., Carpentier, T., Noisternig, M., Warusfel, O., Years, T., Looking, S., Carpentier, T., Noisternig, M. & Warusfel, O. (2015). Twenty Years of Ircam Spat : Looking Back , Looking Forward. *Proc. 41st Int. Comput. Music Conf.*, 270–277.

Carty, B. & Lazzarini, V. (2009). Frequency-domain interpolation of empirical HRTF data. *126th Audio Eng. Soc. Conv. 2009*, *2*, 735–745.

Carty, B. & Lazzarini, V. (2010). Hrtfearly & hrtfreverb: Flexible binaural reverberation processing. *Int. Comput. Music Conf. ICMC 2010*, 450–453.

Chevalier, N. P., Majdak, P., Wilk, E. & Görne, T. (2018). *Convention e-Brief 455 Rapid HRTF measurement in a loudspeaker dome* (Techn. Ber.).

Christensen, F., Møller, H., Minnaar, P., Plogsties, J. & Olesen, S. K. (1999). Interpolating Between Head-Related Transfer Functions Measured with Low Directional Resolution. *Audio Eng. Soc. Conv.*, *5047*.

Daniel, P., Fastl, H., Fedtke, T., Genuit, K., Grabsch, H.-P., Niederdränk, T., Schmitz, A., Vorländer, M. & Zollner, M. (2007). Kunstkopftechnik - Eine Bestandsaufnahme. Eine Mitteilung aus dem Normenausschusses Psychoakustische Messtechnik. *ACUSTICA/Acta Acustica/Nuntius Acusticus*, *93*(1), 1–58.

Eberhard, P. & Ziegler, P. (2021). *Euler- und Kardan-Winkel: Merkblatt Maschinendynamik M8.1* (Techn. Ber.). Institut für Technische und Numerische Mechanik, Universität Stuttgart. http://info.itm.uni-stuttgart.de/courses/madyn/merkblaetter/M08.pdf

Erbes, V., Geier, M., Weinzierl, S. & Spors, S. (2015). Database of single-channel and binaural room impulse responses of a 64-channel loudspeaker array, In *138th AES Conv.* Warschau, Polen.

Estrella, J. (2010). *Zur Extraktion von interauralen Laufzeitdifferenzen in binauralen Raumimpulsantworten* (Techn. Ber.).

et al. Morgan, C. T. (1963). *Human Engineering Guide to Equipment Design*. New York, McGraw-Hill.

Farina, A. (2000). *Simultaneous Measurement of Impulse Response and Distortion With a Swept-Sine Technique* (Techn. Ber.).

Fruhmann, M., Mackensen, P. & Theile, G. (2002). Reduction of dynamic cues in auralized binaural signals. *Acta Acust. united with Acust.*, *88*(3), 443–445.

Gardner, W. G. (n. d.). Efficient Convolution without Input-Output Delay. *J. Audio Eng. Soc.*, *43*(3), 127–136.

Goldstein, H., Poole, C. P. & Safko, J. L. (2006). *Klassische Mechanik*. Wiley.

Hahne, W., Erbes, V. & Spors, S. (2019). On the Perceptually Acceptable Noise Level in Binaural Room Impulse Responses. *Fortschritte der Akust. – DAGA 2019*, 1–4.

Hak, C. C., Wenmaekers, R. H. & Van Luxemburg, L. C. (2012). Measuring room impulse responses: Impact of the decay range on derived room acoustic parameters. *Acta Acust. united with Acust.*, *98*(6), 907–915. https://doi.org/10.3813/AAA.918574

Hammershoi, D. (2002). Methods for binaural recording and reproduction. *Acta Acust. united with Acust.*, *88*(3), 303–311.

Hammershoi, D. & Møller, H. (2002). Methods for binaural recording and reproduction. *Acta Acust. united with Acust.*, *88*(3), 303–311.

Hidaka, T., Yamada, Y. & Nakagawa, T. (2005). A new definition of boundary point between early reflections and late reverberation in room impulse response. *Forum Acusticum Budapest 2005 4th Eur. Congr. Acustics*, (1), 1–4. https://doi.org/10.1121/1.2743161

Hoene, C., Patiño Mejía, I. C. & Cacerovschi, A. (n. d.). *MySofa: Design Your Personal HRTF* (Techn. Ber.). http://www.aes.org/e-lib.

Hofman, P. M., Van Riswick, J. G. & Van Opstal, A. J. (1998). Relearning sound localization with new ears. *Nat. Neurosci.*, *1*(5), 417–421. https://doi.org/10.1038/1633

Hofmann, P. (2009). *Freie Rotation im Raum - Quaternionen und Matrizen* (Techn. Ber.). https://www.uninformativ.de/bin/SpaceSim-2401fee.pdf

Horbach, U., Karamustafaoglu, A., Pellegrini, R. S. & Mackensen, P. (1999). Design and Applications of a Data-based Auralization System for Surround Sound. *106th AES Conv.*, Convention Paper 4976. http://www.aes.org/e-lib/browse.cfm?elib=8204

Horbach, U. & Pellegrini, R. S. (1998). Design of Positional Filters for 3D Audio Rendering. *105th AES Conv.*, Convention Paper 4798.

Howard, D. M. & Angus, J. A. S. (2017). *Acoustics and Psychoacoustics*. 5th edition. | New York; London : Routledge, 2016., Routledge. https://doi.org/10.4324/9781315716879

Jacoby, R. H., Adelstein, B. D. & Ellis, S. R. (1996). Improved temporal response in virtual environments through system hardware and software reorganization. *Stereosc. Displays Virtual Real. Syst. III*, *2653*, 271–284. https://doi.org/10.1117/12.237447

Kearney, G. & Doyle, T. (2015). A HRTF database for virtual loudspeaker rendering. *139th Audio Eng. Soc. Int. Conv. AES 2015*, 1–10.

Keyrouz, F. (2008). *Efficient Binaural Sound Localization for Humanoid Robots and Telepresence Applications* (Diss.). Technische Universität München.

Kirkeby, O. & Nelson, P. A. (1999). Digital filter design for inversion problems in sound reproduction. *AES J. Audio Eng. Soc.*, *47*(7), 583–595.

Kudo, A., Higuchi, H., Hokari, H. & Shimada, S. (2005). An improved method for accurate sound localization. *Audio Eng. Soc. - 118th Conv. Spring Prepr. 2005*, *4*, 1551–1565.

Kulkarni, A., Isabelle, S. K. & Colburn, H. S. (1999). Sensitivity of human subjects to head-related transfer-function phase spectra. *J. Acoust. Soc. Am.*, *105*(5), 2821–2840. https://doi.org/10.1121/1.426898

Kuttruff, H. (2000). *Room Acoustics* (4.). London, Spoon Press
Das Scan-IR-Paper sagt, dass in diesem Buch alle wichtigen akustischen Analysemethoden erklärt werden.

Ledoux, T. (2011). *Quanternionen* (Techn. Ber.). http://www.mathematik.uni-dortmund.de/%7B~%7Dlschwach/SS11/Seminar%7B%5C_%7DII/Quaternionen.pdf

Li, S. & Peissig, J. (2020). Measurement of head-related transfer functions: A review. MDPI AG. https://doi.org/10.3390/app10145014

Lindau, A. (2014). *Binaural Resynthesis of Acoustical Environments. Technology and Perceptual Evaluation.* (Diss.). Technische Universität Berlin.

Lindau, A., Hohn, T. & Weinzierl, S. (2007). Binaural resynthesis for comparative studies of acoustical environments. *Audio Eng. Soc. - 122nd Audio Eng. Soc. Conv. 2007*, *3*(May), 1394–1403.

Lindau, A. & Weinzierl, S. (2012). Assessing the plausibility of virtual acoustic environments. *Acta Acust. united with Acust.*, *98*(5), 804–810. https://doi.org/10.3813/AAA.918562

Lust, M. (2001). *Quaternionen - mathematischer Hintergrund und ihre Interpretation als Rotationen* (Techn. Ber.). https://www.uni-koblenz.de/%7B~%7Dcg/veranst/ws0001/sem/Lust%7B%5C_%7Dquaternion.pdf

Mackensen, P. (2004). *Auditive Localization. Head movements, an additional cue in Localization* (Diss.). Technische Universität Berlin.

Mackensen, P., Felderhof, U., Theile, G., Horbach, U. & Pellegrini, R. (1999). Binaural room scanning—A new tool for acoustic and psychoacoustic research. *J. Acoust. Soc. Am.*, *105*(2), 1343–1344. https://doi.org/10.1121/1.426373

Mackensen, P., Theile, G., Fruhmann, M., Spikofski, G., Horbach, U. & Karamustafaoglu, A. (1999). Der virtuelle Surround Sound Abhörraum – Theorie und Praxis. *ITG-Fachbericht 158*, 15–20.

Meesawat, K. & Hammershøi, D. (2003). The time when the reverberation tail in a binaural room impulse response begins. *AES 1105th Conv.*, 1–9.

Melchior, F., Marston, D., Pike, C., Satongar, D. & Lam, Y. W. (n. d.). *A Library of Binaural Room Impulse Responses and Sound Scenes for Evaluation of Spatial Audio Systems* (Techn. Ber.). http://usir.salford.ac.u

Melchior, F., Marston, D., Pike, C., Satongar, D. & Lam, Y. W. (2014). A Library of Binaural Room Impulse Responses and Sound Scenes for Evaluation of Spatial Audio Systems, In *Proc. 40th Ger. Annu. Conf. Acoust.* Oldenburg. http://usir.salford.ac.u

Menzer, F. (2011). Binaural reverberation using two parallel feedback delay networks. *Proc. AES Int. Conf.*, 1–10.

Miller, J. D., Anderson, M. R., Wenzel, E. M. & Mcclain, B. U. (2003). Latency Measurement of a Real-Time Virtual Acoustic Environment Rendering System. *Proc. 2003 Int. Conf. Audit. Disp.*, *2000*(July), 1–21.

Mills, A. W. (1958). On the Minimum Audible Angle. *J. Acoust. Soc. Am.*, *30*(4), 237–246.

Minnaar, P., Olesen, S. K., Christensen, F. & Møller, H. (2001). Localization with binaural recordings from artificial and human heads. *AES J. Audio Eng. Soc.*, *49*(5), 323–336.

Minnaar, P., Christensen, F., Møller, H., Olesen, S. K. & Plogsties, J. (1999). Audibility of All-Pass components in binaural synthesis. *Proc. 106th Conv. Audio Eng. Soc.*, *4911*(L 5), 1–16.

Møller, H. (1992). Fundamentals of binaural technology. *Appl. Acoust.*, *36*(3-4), 171–218. https://doi.org/10.1016/0003-682X(92)90046-U

Møller, H., Jensen, C. B., Hammershøi, D. & Sørensen, M. F. (1995). Design criteria for headphones. *J. Audio Eng. Soc.*, *43*(4), 218–232.

Müller, S., Member, A. & Massarani, P. (2001). *Transfer-Function Measurement with Sweeps* (Techn. Ber.).

Muller-Tomfelde, C. (2001). Time-varying filter in non-uniform block convolution. *Computer (Long. Beach. Calif).*, 1–5.

Parks, A., Braasch, J. & Clapp, S. (2013). Auralization of measured room impulse responses considering head movements. *135th Audio Eng. Soc. Conv. 2013*, 799–802.

Perrott, D. R. & Saberi, K. (1990). Minimum audible angle thresholds for sources varying in both elevation and azimuth. *J. Acoust. Soc. Am.*, *87*(4), 1728–1731. https://doi.org/10.1121/1.399421

Pike, C., Melchior, F. & Tew, A. (2016). *Descriptive analysis of binaural rendering with virtual loudspeakers using a rate-all-that-apply approach* (Techn. Ber.). http://www.aes.org/e-lib

Pike, C. & Romanov, M. (2017a). *An impulse response dataset for dynamic data-based auralisation of advanced sound systems* (Techn. Ber.). http://epubs.surrey.ac.uk/811764/

Pike, C. & Romanov, M. (2017b). *An impulse response dataset for dynamic data-based auralisation of advanced sound systems* (Techn. Ber.). http://epubs.surrey.ac.uk/811764/

Pike, C. W. (2019). *Evaluating the Perceived Quality of Binaural Technology* (Techn. Ber.).

Pöntynen, H., Santala, O. & Pulkki, V. (2016). Conflicting dynamic and spectral directional cues form separate auditory images. *140th Audio Eng. Soc. Int. Conv. 2016, AES 2016*.

Preis, D. (1982). Phase Distortion and Phase Equalization in Audio Signal Processing - a Tutorial Review. *AES J. Audio Eng. Soc.*, *30*(11), 774–794.

Rathbone, B. (2000). *Evaluierung von Kunstköpfen als Messvorrichtung für datenbasierte binaurale Raumsynthese* (Diss.).

Rayleigh, L. (1907). On our perception of sound direction. *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, *13*(74), 214–232.

Romblom, D. & Bahu, H. (2018). *Optimization and prediction of the spherical and ellipsoidal ITD model parameters using offset ears* (Techn. Ber.). http://www.aes.org/e-lib

Rossing, T. D. (2007). *Springer Handbook of Acoustics* (1.). Springer-Verlag Berlin Heidelberg.

Satongar, D., Lam, Y. W. & Pike, C. (n. d. a). *MEASUREMENT AND ANALYSIS OF A SPATIALLY SAMPLED BINAURAL ROOM IMPULSE RESPONSE DATASET* (Techn. Ber.).

Satongar, D., Lam, Y. W. & Pike, C. (n. d. b). *MEASUREMENT AND ANALYSIS OF A SPATIALLY SAMPLED BINAURAL ROOM IMPULSE RESPONSE DATASET* (Techn. Ber.).

Satongar, D., Lam, Y. W. & Pike, C. (2014). MEASUREMENT AND ANALYSIS OF A SPATIALLY SAMPLED BINAURAL ROOM IMPULSE RESPONSE DATASET, In *21st Int. Congr. Sound Vib.*

Schanze, R. (2016). Mooresches Gesetz: Defintion und Ende von Moore's Law – Einfach erklärt. Verfügbar 20. Januar 2021 unter https://www.giga.de/ratgeber/specials/mooresches-gesetz-defintion-und-ende-von-moore-s-law-einfach-erklaert/

Schärer, Z. & Lindau, A. (2009). Evaluation of equalization methods for binaural signals. *126th Audio Eng. Soc. Conv. 2009*, *1*, 15–31.

Schroeder, M. R. (1965). New Method of Measuring Reverberation Time. *J. Acoust. Soc. Am.*, *37*(6), 1187–1188. https://doi.org/10.1121/1.1939454

Sengpiel, E. (1995). Der Ohrabstand - welcher ?

Shotton, M., Pike, C. & Melchior, F. (2014). *A Motorised Telescope Mount as a Computer-Controlled Rotational Platform for Dummy Head Measurements* (Techn. Ber.).

Slate, S. (2020). Steven Slate Audio VSX | Perfect Mixes Just Got Easier. Verfügbar 20. Januar 2021 unter https://stevenslateaudio.com/vsx

Smith, S. W. (1999). *The Scientist and Engineer's Guide to Digital Signal Processing*. https://doi.org/10.1016/S0065-2458(08)60403-9

Stade, P. & Arend, J. M. (2016). A Perception-Based Parametric Model for Synthetic Late Binaural Reverberation. *Proc. 42nd DAGA*, 63–66.

Stade, P., Bernschütz, B. & Rühl, M. (2012). A Spatial Audio Impulse Response Compilation Captured at the WDR Broadcast Studios. *27th Tonmeistertagung - VDT Int. Conv.*

STEMMER IMAGING AG. (n. d.). Kabel für USB 2.0. Verfügbar 15. April 2021 unter https://www.stemmer-imaging.com/de-de/produkte/serie/usb-kabel/%7B%5C#%7D:%7B~%7D:text=Die%20maximale%20L%7B%5C%22%7Ba%7D%7Dnge%20f%7B%5C%22%7Bu%7D%7Dr%20USB,oder%20aktive%20Repeater%20eingesetzt%20werden.

Thurlow, W. R., Mangels, J. W. & Runge, P. S. (1967). Head movements during sound localization. *J. Acoust. Soc. Am.*, *42*(2), 489–493.

Vanasse, J., Genovese, A. & Roginska, A. (n. d.). *Multichannel Impulse Response Measurements in MATLAB: An Update on ScanIR* (Techn. Ber.).

Völk, F. (2011). System theory of binaural synthesis. *131st Audio Eng. Soc. Conv. 2011*, *1*, 501–517.

Warusfel, O., Carpentier, T., Bahu, H. & Noisternig, M. (2018). *Measurement of a head-related transfer function database with high spatial resolution* (Techn. Ber.). https://www.researchgate.net/publication/315812259

Wefers, F. (2014). *Partitioned convolution algorithms for real-time auralization* (M. Vorländer, Hrsg.). Berlin, Logos Verlag. http://publications.rwth-aachen.de/record/466561

Wefers, F. & Vorländer, M. (2011). Optimal filter partitions for real-time FIR filtering using uniformly-partitioned FFT-based convolution in the frequency-domain. *Proc. 14th Int. Conf. Digit. Audio Eff. DAFx 2011*, (January), 155–162.

Weinzierl, S. (2008). *Handbuch der Audiotechnik* (1. Auflage). Springer-Verlag Berlin Heidelberg.

Wenzel, E. M. (1997). Analysis of the role of update rate and system latency in interactive virtual acoustic environments. *Audio Eng. Soc. 103rd Conv.*, Convention Paper 4633.

Werner, S., Götz, G. & Klein, F. (2017). Influence of head tracking on the externalization of auditory events at divergence between synthesized and listening room using a binaural headphone system. *142nd Audio Eng. Soc. Int. Conv.*, 1–8.

Werner, S., Klein, F., Mayenfels, T. & Brandenburg, K. (2016). A summary on acoustic room divergence and its effect on externalization of auditory events. *8th Int. Conf. Qual. Multimed. Exp. QoMEX 2016*, (June 2016). https://doi.org/10.1109/QoMEX.2016.7498973

Zhong, X.-l. & Xie, B.-s. (2014). Head-Related Transfer Functions and Virtual Auditory Display. In *Soundscape Semiot. - Localis. Categ.* InTech. https://doi.org/10.5772/56907

Anhang

A. MATLAB-Skript: twospeakers_load_normalisationfactor_maxposition.m

twospeakers_load_normalisationfactor_maxposition.m

Contents

Metadata of Max/MSP measurements

```
complete_horizontal_grid = 360;
stepsize = 1.8;
number_of_BRIRs = complete_horizontal_grid/stepsize;
FS = 48000;
```

1-L Speaker

```
%  BRIR import and matrixing

%  !!! Please make sure that ONLY the correct BRIR data can be found in
%  this directory, because all wav files of the directory will be read OR
%  adjust the directories accordingly !!!
struct_1 = dir(fullfile('*.wav'));
[~,ndx] = natsortfiles({struct_1.name});
struct_1 = struct_1(ndx);

if number_of_BRIRs ~= length(struct_1)
    error('Number of found BRIRs (1-L Speaker) does not match given measurement metadata.')
end

cell_1 = cell(length(struct_1),1);
fs_cell_1 = cell_1;
for ii = 1:length(struct_1)
    [cell_1{ii},fs_cell_1{ii}] = audioread(struct_1(ii).name);
end

if fs_cell_1{1,1} ~= FS
   error('Sampling Rate of imported BRIRs (1-L Speaker) does not match given measurement metadata.')
end

all_irs_measured_1 = permute(cat(3,cell_1{:}), [1,3,2]);


%  Find absolute maximum value in all given BRIR-data (max_1)
%  Find sample with maximum absolute value (maxsample_1)
%  Find position of the the maxsample_1 from the beginning of a single BRIR (max_sampleposition_1)

max_1 = max(abs(all_irs_measured_1(:)));
maxsample_1 = find(all_irs_measured_1 == max_1);

BRIR_length_1 = size(all_irs_measured_1,1);
i = maxsample_1;
a = 0;
while i >= BRIR_length_1
    i=i-BRIR_length_1;
    a=a+1;
end

max_sampleposition_1 = maxsample_1 - a * BRIR_length_1;


% Set normalization factor (normalisationfactor_1)

normalisationfactor_1 = db2mag(0) / max_1;
```

```
Undefined function 'natsortfiles' for input arguments of type 'cell'.

Error in twospeakers_load_normalisationfactor_maxposition (line 21)
[~,ndx] = natsortfiles({struct_1.name});
```

2-R Speaker

```
%  BRIR import and matrixing

%  !!! Please make sure that ONLY the correct BRIR data can be found in
%  this directory, because all wav files of the directory will be read OR
%  adjust the directories accordingly !!!
struct_2 = dir(fullfile('*.wav'));
[~,ndx] = natsortfiles({struct_2.name});
```

```
struct_2 = struct_2(ndx);

if number_of_BRIRs ~= length(struct_2)
    error('Number of found BRIRs (2-R Speaker) does not match given measurement metadata.')
end

cell_2 = cell(length(struct_2),1);
fs_cell_2 = cell_2;
for ii = 1:length(struct_2)
    [cell_2{ii},fs_cell_2{ii}] = audioread(struct_2(ii).name);
end

if fs_cell_2{1,1} ~= FS
    error('Sampling Rate of imported BRIRs (2-R Speaker) does not match given measurement metadata.')
end

all_irs_measured_2 = permute(cat(3,cell_2{:}), [1,3,2]);


%  Find absolute maximum value in all given BRIR-data (max_2)
%  Find sample with maximum absolute value (maxsample_2)
%  Find position of the maxsample_2 from the beginning of a single BRIR (max_sampleposition_2)

max_2 = max(abs(all_irs_measured_2(:)));
maxsample_2 = find(all_irs_measured_2 == max_2);

BRIR_length_2 = size(all_irs_measured_2,1);
i = maxsample_2;
a = 0;
while i >= BRIR_length_2
    i=i-BRIR_length_2;
    a=a+1;
end

max_sampleposition_2 = maxsample_2 - a * BRIR_length_2;


% Set normalization factor (normalisationfactor_2)

normalisationfactor_2 = db2mag(0) / max_2;
```

Throw ERROR if dataset seems incorrect

```
The values of the first three error queries are initially set quite arbitrarily
under rough estimates and consequently can also work incorrectly.
```

```
if (max_1 > max_2+0.01) || (max_2 > max_1+0.01)
    error('The measurement data seems erroneous.')
end

if (max_sampleposition_1 > max_sampleposition_2+FS*0.003/20) || (max_sampleposition_2 > max_sampleposition_1+FS*0.003/20)
    error('The measurement data seems erroneous.')
end

if (normalisationfactor_1 > normalisationfactor_2+1) || (normalisationfactor_2 > normalisationfactor_1+1)
    error('The measurement data seems erroneous.')
end

if size(all_irs_measured_1) ~= size(all_irs_measured_2)
    error('Impulse responses must be of the same length, have the same number of channels, and be recorded with the same step size.')
end
```

Set the normalizationfactor and the position of the maxsample (max_sampleposition) for further processing

```
This guarantees that the datasets of the two speakers are normalized and
left-truncated in the same way (executed in script 'twoloudspeakers_normalize_truncate.m')
```

```
if normalisationfactor_1 < normalisationfactor_2
    normalisationfactor = normalisationfactor_1;
else normalisationfactor = normalisationfactor_2;
end

if max_sampleposition_1 < max_sampleposition_2
    max_sampleposition = max_sampleposition_1;
else max_sampleposition = max_sampleposition_2;
end
```

Published with MATLAB® R2018a

B. MATLAB-Skript: twospeakers_normalize_truncate.m

twospeakers_normalize_truncate.m

Contents

Left truncation of BRIRs with value headcutsamples

```
Truncate the BRIRs to the left of max_sampleposition specifying an onsetprotection which leads to the value headcutsamples
If the onsetprotection and thus the headcutsamples have a larger value than the samples to the left of the max_sampleposition, no truncation takes place
```

```
onsetprotectionseconds = 0.01;
onsetprotectionsamples = onsetprotectionseconds * FS;
headcutsamples = max_sampleposition - onsetprotectionsamples;

if max_sampleposition > onsetprotectionsamples
    all_irs_measured_1(1:headcutsamples, :, :) = [];
    all_irs_measured_2(1:headcutsamples, :, :) = [];
end
```

```
Undefined function or variable 'FS'.

Error in twospeakers_normalize_truncate (line 9)
onsetprotectionsamples = onsetprotectionseconds * FS;
```

Normalisation of BRIR data and Splitting into seperate arrays for left and right ear

```
Normalize BRIR data uniformly to 0 dBFS with the determined normalization factor
Split the data into arrays for the left and right ear
```

```
all_irs_normalized_1 = all_irs_measured_1 * normalisationfactor;

irs_leftear_normalized_1 = all_irs_normalized_1(:,:,1);
irs_rightear_normalized_1 = all_irs_normalized_1(:,:,2);


all_irs_normalized_2 = all_irs_measured_2 * normalisationfactor;

irs_leftear_normalized_2 = all_irs_normalized_2(:,:,1);
irs_rightear_normalized_2 = all_irs_normalized_2(:,:,2);
```

Integrated determination of the room acoustic parameters to determine a possible right-side truncation of the BRIRs

```
Besides the direct input of the parameters, they can also be determined by means of attached functions via an EDC (Schroeder backward integration)
For this purpose, an omnidirectional room impulse response can be imported directly or a single BRIR signal can be used
```

```
% Possibility of importing an omnidirectional room impulse response for the determination of room acoustic parameters
% ir_room_mono = audioread(...);

% Integrated determination of the room acoustic parameters,
% exemplarily performed with a signal of the left ear at frontal head orientation and excitation by the left loudspeaker

% EDC via getSchroeder function
% EDC_ir_leftear_normalized = getSchroeder(irs_leftear_normalized_1(:,1,1));

% Calculation of decay times EDT and RT60 with its different regression
% ranges
% EDT = calcRTX(EDC_ir_leftear_normalized,0,-10,FS,true);
% T60 = calcRTX(EDC_ir_leftear_normalized,0,-60,FS,false);
% T30 = calcRTX(EDC_ir_leftear_normalized,-5,-35,FS,true);
% T20 = calcRTX(EDC_ir_leftear_normalized,-5,-25,FS,true);
```

Specification of room acoustic parameters via direct input of the them from another determination

In this case, the reverberation time was determined using averaged omnidirectional room impulse responses when the sample auralization room was excited successively with one loudspeaker each from five loudspeakers (according to ITU-R BS.775).

```
RT60 = 0.25;

RT60insamples = RT60 * FS;
%EDTinsamples = EDT * FS;

endtruncationsafetyseconds = 0.1;
endtruncationsafetysamples = endtruncationsafetyseconds * FS;

IRendtruncationFromRT60 = RT60insamples + endtruncationsafetysamples;
%IRendtruncationFromEDT = EDTinsamples + endtruncationsafetysamples;
```

Right-side truncation of BRIRs based on RT60

```
all_irs_truncatedFromRT60_normalized_1 = all_irs_normalized_1(1:IRendtruncationFromRT60,:,:);

irs_leftear_truncatedFromRT60_normalized_1 = all_irs_truncatedFromRT60_normalized_1(:,:,1);
irs_rightear_truncatedFromRT60_normalized_1 = all_irs_truncatedFromRT60_normalized_1(:,:,2);
```

```
all_irs_truncatedFromRT60_normalized_2 = all_irs_normalized_2(1:IRendtruncationFromRT60,:,:);

irs_leftear_truncatedFromRT60_normalized_2 = all_irs_truncatedFromRT60_normalized_2(:,:,1);
irs_rightear_truncatedFromRT60_normalized_2 = all_irs_truncatedFromRT60_normalized_2(:,:,2);
```

Right-side truncation of BRIRs based on EDT

```
%all_irs_truncatedFromEDT_normalized_1 = all_irs_normalized_1(1:IRendtruncationFromEDT,:,:);

%irs_leftear_truncatedFromEDT_normalized_1 = all_irs_truncatedFromEDT_normalized_1(:,:,1);
%irs_rightear_truncatedFromEDT_normalized_1 = all_irs_truncatedFromEDT_normalized_1(:,:,2);


%all_irs_truncatedFromEDT_normalized_2 = all_irs_normalized_2(1:IRendtruncationFromEDT,:,:);

%irs_leftear_truncatedFromEDT_normalized_2 = all_irs_truncatedFromEDT_normalized_2(:,:,1);
%irs_rightear_truncatedFromEDT_normalized_2 = all_irs_truncatedFromEDT_normalized_2(:,:,2);
```

Published with MATLAB® R2018a

C. MATLAB-Skript: twospeakers_saveFullDynamic.m

twospeakers_saveFullDynamic.m

Contents

Start SOFA API // Set netCDF compression parameter // Get an empty 'SimpleFreeFieldHRIR' convention structure

```
SOFAstart;
compression = 0;
Obj = SOFAgetConventions('SimpleFreeFieldHRIR');
```

```
Undefined function or variable 'SOFAstart'.

Error in twospeakers_saveFULLDynamic_15_newVariablenAttribute (line 6)
SOFAstart;
```

Define the HRIR/BRIR grid

```
In this case a full horizontal circle around the listener with zero elevation
is used, with grid resolution according to the stepsize of the measurement
```

```
azi = 0:stepsize:359;     % azimuth angles
ele = 0;                  % elevation angles
```

Check if the BRIR datasets of the used loudspeakers have the same dimensions

```
if size(all_irs_truncatedFromRT60_normalized_1) ~= size(all_irs_truncatedFromRT60_normalized_2)
    error('BRIR datasets of the used loudspeakers must have the same dimensions.')
end
```

Definiere SOFA-Dimensionen and befülle SOFA-Object mit BRIR-Daten

```
1-L Loudspeaker
```

```
M=length(azi);
N=size(all_irs_truncatedFromRT60_1,1);
%N=size(all_irs_truncatedFromEDT_1,1);
R=2;
% E=1 ist default in der SimpleFreeFieldHRIR-Convention

Obj.Data.IR = NaN(M,R,N); % Data.IR muss von der Dimension [M R N] sein

for ii=1:M % ii geht über alle Messungen M

 Obj.Data.IR(ii,1,:)=irs_leftear_truncatedFromRT60_normalized_1(:,ii);
 Obj.Data.IR(ii,2,:)=irs_rightear_truncatedFromRT60_normalized_1(:,ii);

    %Obj.Data.IR(ii,1,:)=irs_leftear_truncatedFromEDT_normalized_1(:,ii);
 %Obj.Data.IR(ii,2,:)=irs_rightear_truncatedFromEDT_normalized_1(:,ii);

    Obj.SourcePosition(ii,:)=[azi(ii) ele 1];
end

Obj.Data.Delay(:,:) = [0 0];
Obj.Data.SamplingRate = Fs;


% Befülle SOFA-Variablen mit Attributen

Obj.GLOBAL_APIName = 'SOFA API for MATLAB';
Obj.GLOBAL_APIVersion = SOFAgetVersion('API');
Obj.GLOBAL_AuthorContact = 'me099@hdm-stuttgart.de';
Obj.GLOBAL_Comment = 'BRIRs of single speakers';
Obj.GLOBAL_History = 'created with a script';
Obj.GLOBAL_Organization = 'Hochschule der Medien Stuttgart';
Obj.GLOBAL_RoomType = 'reverberant';
```

```
Obj.GLOBAL_DatabaseName = 'U48_L_Hochschule der Medien Stuttgart';
Obj.GLOBAL_ListenerShortName = 'Neumann KU 100';


% Update der SOFA-Dimensionen

Obj=SOFAupdateDimensions(Obj);


% Speichere das SOFA-File im CurrentFolder

currentFolder = pwd;

SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files','0-U48_1-L_dynamic_truncatedFromRT60_normalized.sofa');
%SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files','0-U48_1-L_dynamic_truncatedFromEDT_normalized.sofa');

disp(['Saving:  ' SOFAfn]);
Obj=SOFAsave(SOFAfn, Obj, compression)
```

Definiere SOFA-Dimensionen and befülle SOFA-Object mit BRIR-Daten

2-R Loudspeaker

```
M=length(azi);
N=size(all_irs_truncatedFromRT60_2,1);
%N=size(all_irs_truncatedFromEDT_2,1);
R=2;
% E=1 ist default in der SimpleFreeFieldHRIR-Convention

Obj.Data.IR = NaN(M,R,N); % Data.IR muss von der Dimension [M R N] sein

for ii=1:M % ii geht über alle Messungen M

 Obj.Data.IR(ii,1,:)=irs_leftear_truncatedFromRT60_normalized_2(:,ii);
 Obj.Data.IR(ii,2,:)=irs_rightear_truncatedFromRT60_normalized_2(:,ii);

    %Obj.Data.IR(ii,1,:)=irs_leftear_truncatedFromEDT_normalized_2(:,ii);
 %Obj.Data.IR(ii,2,:)=irs_rightear_truncatedFromEDT_normalized_2(:,ii);

    Obj.SourcePosition(ii,:)=[azi(ii) ele 1];
end

Obj.Data.Delay(:,:) = [0 0];
Obj.Data.SamplingRate = Fs;


% Befülle SOFA-Variablen mit Attributen

Obj.GLOBAL_APIName = 'SOFA API for MATLAB';
Obj.GLOBAL_APIVersion = SOFAgetVersion('API');
Obj.GLOBAL_AuthorContact = 'me099@hdm-stuttgart.de';
Obj.GLOBAL_Comment = 'BRIRs of single speakers';
Obj.GLOBAL_History = 'created with a script';
Obj.GLOBAL_Organization = 'Hochschule der Medien Stuttgart';
Obj.GLOBAL_RoomType = 'reverberant';
Obj.GLOBAL_DatabaseName = 'U48_L_Hochschule der Medien Stuttgart';
Obj.GLOBAL_ListenerShortName = 'Neumann KU 100';


% Update der SOFA-Dimensionen

Obj=SOFAupdateDimensions(Obj);


% Speichere das SOFA-File im CurrentFolder

currentFolder = pwd;

SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files','0-U48_2-R_dynamic_truncatedFromRT60_normalized.sofa');
%SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files','0-U48_1-L_dynamic_truncatedFromEDT_normalized.sofa');

disp(['Saving:  ' SOFAfn]);
Obj=SOFAsave(SOFAfn, Obj, compression)
```

Published with MATLAB® R2018a

D. MATLAB-Skript: twospeakers_splittingBRIRs_saveDynamic+Static

?twospeakers_splittingBRIRs_saveDynamic+Static.m

```
Error using evalin
Undefined function or variable 'twospeakers_splittingBRIRs_saveDynamic'.
```

Contents

Angabe der gewünschten Mixing Times

```
split_times_seconds = [0.04,0.05,0.06,0.07,0.08,0.09,0.1,0.11,0.12,0.13];

split_times_samples = split_times_seconds .* FS;
```

Teile die BRIRs (für die zwei Lautsprecher 1-L und 2-R) in dynamisch und statisch zu verarbeitenden Teil auf Grundlage der gewählten Mixing Times

```
Dynamic_1 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(split_times_samples, 2))
   Dynamic_1{i} = all_irs_truncatedFromEDT_normalized_1(1:round(split_times_samples(i)),:,:);
end

Static_1 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(split_times_samples, 2))
   Static_1{i} = all_irs_truncatedFromEDT_normalized_1((round(split_times_samples(i))+1):end,:,:);
end


Dynamic_2 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(split_times_samples, 2))
   Dynamic_2{i} = all_irs_truncatedFromEDT_normalized_2(1:round(split_times_samples(i)),:,:);
end

Static_2 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(split_times_samples, 2))
   Static_2{i} = all_irs_truncatedFromEDT_normalized_2((round(split_times_samples(i))+1):end,:,:);
end
```

Speichere den dynamisch zu verarbeitenden Teil der BRIRs in je einem SOFA-File

```
% Start SOFA API // Set netCDF compression parameter // Get an empty 'SimpleFreeFieldHRIR' convention structure
SOFAstart;
compression=0;
Obj = SOFAgetConventions('SimpleFreeFieldHRIR');

% Define the HRIR/BRIR grid
% In this case a full horizontal circle around the listener with zero elevation
% is used, with grid resolution according to the stepsize of the measurement
azi1 = 0:stepsize:359;     % azimuth angles
ele1 = 0;                  % elevation angles


% For-Loop for SOFA-File-Creation

for i = 1:round(size(Dynamic_1,1))
    N = size(Dynamic_1{i},1);

    M=length(azi1);
    Obj.Data.IR = NaN(M,2,N);

    ii=1;
    for aa=1:length(azi1)
        Obj.Data.IR(ii,1,:)=Dynamic_1{i,1}(:,ii,1);
        Obj.Data.IR(ii,2,:)=Dynamic_1{i,1}(:,ii,2);
        deg = [azi1(ii),ele1];
        degCell = num2cell(deg);
  [azim,elev]=degCell{:};
        Obj.SourcePosition(ii,:)=[azim elev 1];
  Obj.SourcePosition(ii,:)=[azim elev 1];
  ii=ii+1;
    end

Obj.Data.Delay(:,:) = [0 0];
Obj.Data.SamplingRate = Fs;

Obj=SOFAupdateDimensions(Obj);

Obj.GLOBAL_APIName = 'SOFA API for MATLAB';
Obj.GLOBAL_APIVersion = SOFAgetVersion('API');
Obj.GLOBAL_AuthorContact = 'me099@hdm-stuttgart.de';
Obj.GLOBAL_Comment = 'BRIRs of single speakers';
Obj.GLOBAL_History = 'created with a script';
Obj.GLOBAL_Organization = 'Hochschule der Medien Stuttgart';
Obj.GLOBAL_RoomType = 'reverberant';
Obj.GLOBAL_DatabaseName = 'U48_L_Hochschule der Medien Stuttgart';
Obj.GLOBAL_ListenerShortName = 'Neumann KU 100';

currentFolder = pwd;
```

```
SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files',sprintf('%i-U48_1-L_dynamic_truncatedFromEDT_normalized_%g sec.sofa',i,split_times_seconds(i)));

disp(['Saving:  ' SOFAfn]);
Obj=SOFAsave(SOFAfn, Obj, compression)

end


for i = 1:round(size(Dynamic_2,1))
    N = size(Dynamic_2{i},1);

    M=length(azi1);
    Obj.Data.IR = NaN(M,2,N);

    ii=1;
    for aa=1:length(azi1)
        Obj.Data.IR(ii,1,:)=Dynamic_2{i,1}(:,ii,1);
        Obj.Data.IR(ii,2,:)=Dynamic_2{i,1}(:,ii,2);
        deg = [azi1(ii),ele1];
        degCell = num2cell(deg);
  [azim,elev]=degCell{:};
        Obj.SourcePosition(ii,:)=[azim elev 1];
  Obj.SourcePosition(ii,:)=[azim elev 1];
  ii=ii+1;
    end

Obj.Data.Delay(:,:) = [0 0];
Obj.Data.SamplingRate = Fs;

Obj=SOFAupdateDimensions(Obj);

Obj.GLOBAL_APIName = 'SOFA API for MATLAB';
Obj.GLOBAL_APIVersion = SOFAgetVersion('API');
Obj.GLOBAL_AuthorContact = 'me099@hdm-stuttgart.de';
Obj.GLOBAL_Comment = 'BRIRs of single speakers';
Obj.GLOBAL_History = 'created with a script';
Obj.GLOBAL_Organization = 'Hochschule der Medien Stuttgart';
Obj.GLOBAL_RoomType = 'reverberant';
Obj.GLOBAL_DatabaseName = 'U48_L_Hochschule der Medien Stuttgart';
Obj.GLOBAL_ListenerShortName = 'Neumann KU 100';

currentFolder = pwd;

SOFAfn=fullfile(currentFolder,'Dynamic SOFA Files',sprintf('%i-U48_2-R_dynamic_truncatedFromEDT_normalized_%g sec.sofa',i,split_times_seconds(i)));

disp(['Saving:  ' SOFAfn]);
Obj=SOFAsave(SOFAfn, Obj, compression)

end
```

Speichere den statisch zu verarbeitenden Teil der BRIRs je einem in wav-File

```
Static_0deg_1 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(Static_1,1))
    Static_0deg_1{i,1} = squeeze(Static_1{i,1}(:,1,:));

    audiowrite(sprintf('%i-U48_1-L_static_truncatedFromEDT_normalized_%g sec.wav',i,split_times_seconds(i)), Static_0deg_1{i,1},  FS, 'BitsPerSample', 64);
end


Static_0deg_2 = cell(round(size(split_times_samples,2)),1);

for i = 1:round(size(Static_2,1))
    Static_0deg_2{i,1} = squeeze(Static_2{i,1}(:,1,:));

    audiowrite(sprintf('%i-U48_2-R_static_truncatedFromEDT_normalized_%g sec.wav',i,split_times_seconds(i)), Static_0deg_2{i,1},  FS, 'BitsPerSample', 64);
end
```

Published with MATLAB® R2018a

G. Danksagung

Im Rahmen dieser Abschlussarbeit möchte ich mich ganz herzlich bei allen beteiligten Personen bedanken.

Im Besonderen bei:

Prof. Dr. Frank Melchior und Prof. Oliver Curdt für die Betreuung.

Meiner Freundin Melissa, meinen Eltern, meiner Band und engen Freunden für die tatkräftige Unterstützung bei der Fertigstellung der Arbeit, besonders in deren Endphase.

H. Digitaler Datenträger

Auf dem dieser Arbeit beigelegten Datenträger befinden sich:

- Vorliegende Arbeit als PDF-Version
- Messsoftware des Systemmoduls 1 als Max/MSP-Patch (inkl. aller Subpatches)
- Postprocessing-Routinen des Systemmoduls 2 als MATLAB-Skripte (inkl. aller benötigten Funktionen)
- Flexible Auralisationsumgebung des Systemmoduls 3 als Max/MSP-Patch (inkl. aller Subpatches)
- BRIRs in Form von SOFA- und wav-Files
- Arduino-Sketch für die *Drehtellersteuereung* sowie den *HdM-Headtracker*