



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Mihai Bodolea

Entwicklung einer Virtual Reality Umgebung für einen realitätsnahen Höreindruck

*Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau*

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

Mihai Bodolea

**Entwicklung einer Virtual Reality
Umgebung für einen realitätsnahen
Höreindruck**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Flugzeugbau
am Department Fahrzeugtechnik und Flugzeugbau
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr.-Ing. habil. Thomas Kletschkowski
Zweitprüfer: Prof. Dr.-Ing Sten Böhme

Abgabedatum: 06.09.2022

Kurzreferat

Mihai Bodolea

Thema der Masterthesis

Entwicklung einer Virtual Reality Umgebung für einen realitätsnahen Höreindruck

Stichworte

Virtuelle Realität, Unity, Akustik, Raumakustik, Schallfeldmessung, Frequenzantwort, Finite-Elemente-Methode, COMSOL Multiphysics, MATLAB

Kurzreferat

Mit der rapiden Entwicklung der Technologien im Bereich der virtuellen Realität gewann auch die Simulation der Akustik in den virtuellen Umgebungen immer mehr an Bedeutung. Während in der Wissenschaft innovative Methoden zur physikalisch basierten Audiowiedergabe in einer VR-Umgebung entwickelt wurden, erzielten die kommerziellen VR-Systemen keine großen Entwicklungen in diesem Bereich. In dieser Arbeit wird ein Konzept für die realitätsnahe Wiedergabe eines Schallfelds im tieffrequenten Bereich in der virtuellen Umgebung der Spiel-Engine von Unity entwickelt. Das Konzept basiert auf der Verbindung der Ergebnisse einer Finite-Elemente-Methode Simulation mit der virtuellen Umgebung aus Unity. Anhand von durchgeführten Kartierungsmessungen wird gezeigt, dass die mit dem entwickelten Konzept erzeugte Schallverteilung mit einer realen Schallverteilung übereinstimmt.

Title of the paper

Development of a virtual reality environment for a realistic listening impression

Keywords

Virtual Reality, Unity, Acoustics, Room Acoustics, Sound field measurement, Frequency Response, Finite Element Method, COMSOL Multiphysics, MATLAB

Abstract

With the rapid development of virtual reality technologies, the simulation of acoustics in virtual environments also became increasingly important. While innovative methods for physically based audio reproduction in a VR environment have been developed in the academic field, no major developments have been made in this area in commercial VR systems. In this thesis, a concept is developed for the realistic reproduction of a sound field in the low-frequency range in the virtual environment of the commercial Unity game engine. The concept is based on combining the results of a finite element method simulation with the virtual environment from Unity. Based on performed mapping measurements, it is shown that the sound distribution generated with the developed concept matches a real sound distribution.

Inhaltsverzeichnis	
Symbole und Abkürzungen	V
Abbildungsverzeichnis	VIII
Tabellenverzeichnis	XI
1. Einleitung	1
2. Theoretische Grundlagen	5
2.1. Physik der Schallfelder	5
2.2. Raumakustik	11
2.3. Digitale Signalverarbeitung	15
3. Akustik in der virtuellen Realität	21
3.1. Steam Audio Plug-in	29
3.2. Resonance Audio Plug-in.....	32
4. Definition und Auswahl der Messumgebungen	34
4.1. Mock-up eines Crew Rest Compartments	35
4.2. Finite-Elemente Modell eines Crew Rest Compartments.....	39
4.3. Virtual Reality Modell eines Crew Rest Compartments.....	45
5. Schallfeldmessungen in den unterschiedlichen Messumgebungen	49
5.1. Messvorbereitungen in der realen Messumgebung	50
5.2. Messung der Frequenzantwort der unterschiedlichen Umgebungen	54
5.3. Kartierung der Schallfelder in den unterschiedlichen Messumgebungen.....	61
6. Vergleich der Messergebnisse der Schallfeldmessungen	68
6.1. Vergleich der realen und der virtuellen Umgebung.....	69
6.2. Vergleich der realen Umgebung mit der FEM-Berechnung	72
6.3. Erkenntnisse	75
7. Verbesserung der Akustik in der virtuellen Umgebung	76
7.1. Beschreibung des Verbesserungskonzeptes.....	76
7.2. Implementierung des Verbesserungskonzeptes	79
7.3. Vergleich des Verbesserungskonzeptes mit der realen- und FEM-Umgebung	81
8. Zusammenfassung und Ausblick	86
Literaturverzeichnis	88
Anhang	92

Symbole und Abkürzungen

Griechische Buchstaben

Symbol	Einheit	Bedeutung
α	-	Absorptionsgrad
θ	rad	Einfallwinkel der Schallwelle auf die Wandoberfläche
κ	-	Isentropenexponent
λ	m	Wellenlänge
ξ	m	Teilchenauslenkung
ρ_0	kg/m ³	Ruhedichte
ρ_{\sim}	kg/m ³	Wechseldichte
φ	rad	Phasenwinkel
$\Delta\varphi$	rad	Phasenunterschied
ψ_A	-	Eigenvektoren des Systems <i>A</i>
ψ_E	-	Eigenvektoren des Systems <i>E</i>
ω	1/s	Kreisfrequenz

Lateinische Buchstaben

Symbol	Einheit	Bedeutung
<i>A</i>	m ²	gesamte Wandfläche des Raumes
ΔA	V	Amplitudenintervall / Auflösung
<i>A</i> _{max}	V	maximale Amplitude
<i>a</i>	m/s ²	Beschleunigung
<i>b</i>	bit	Bit-Anzahl
<i>c</i>	m/s	Schallgeschwindigkeit
<i>c</i> _p	J/kg·K	Spezifische Wärmekapazität bei konstantem Druck
<i>c</i> _v	J/kg·K	Spezifische Wärmekapazität bei konstantem Volumen
<i>E</i> _M	mV/Pa	Mikrofonempfindlichkeit
<i>f</i>	Hz	Frequenz
<i>f</i> _E	Hz	Eigenfrequenz
<i>f</i> _{Gr}	Hz	Grenzfrequenz
<i>f</i> _m	Hz	Messfrequenz
<i>f</i> _{max}	Hz	maximale Frequenz eines Signales
<i>f</i> _s	Hz	Abtastfrequenz
Δf_m	Hz	Messfrequenzschritt
<i>G</i> (<i>f</i>)	-	fouriertransformiertes Ausgangssignal <i>g</i> (<i>t</i>)
<i>g</i> (<i>t</i>)	-	Ausgangssignal im Zeitbereich
<i>H</i> (<i>f</i>)	-	fouriertransformierte Übertragungsfunktion <i>h</i> (<i>t</i>)
<i>h</i> (<i>t</i>)	-	Übertragungsfunktion im Zeitbereich
<i>h</i> _{max}	m	maximale Netzelementgröße
<i>h</i> _{min}	m	minimale Netzelementgröße
<i>k</i>	1/m	Wellenzahl
<i>L</i>	m	Summe aller Kantenlängen des Raumes
<i>L</i> _A	dB	Amplitudenpegel
<i>L</i> _p	dB	Schalldruckpegel

$L_{p,ref}$	dB(Z)	Referenzschalldruckpegel
l_x	m	Kantenlänge des Raumes in x-Richtung
l_y	m	Kantenlänge des Raumes in y-Richtung
l_z	m	Kantenlänge des Raumes in z-Richtung
$N_{f_{Gr}}$	-	Anzahl der Eigenmoden bis zur Grenzfrequenz f_{Gr}
n	-	Anzahl der Knotenpunkte
n_x	-	Ordnung der Raummode in x-Richtung
n_y	-	Ordnung der Raummode in y-Richtung
n_z	-	Ordnung der Raummode in z-Richtung
p_0	Pa	Ruhedruck
p_{\sim}	Pa	Wechseldruck / Schalldruck
Q_m	1/s ²	Monopol
q_d	N/m ³	Dipol
r	-	Reflexionsfaktor
r_{max}	-	maximale Wachstumsrate
$S(f)$	-	fouriertransformiertes Eingangssignal $s(t)$
$s(t)$	-	Eingangssignal im Zeitbereich
T	°C	Temperatur
t	s	Zeit
t_m	s	Messdauer
U	V	Spannung
U_A	V	Ausgangsspannung
U_E	V	Eingangsspannung
V	m ³	Volumen
V_m	dB	Verstärkungsfaktor Mikrofonsignal
v_0	m/s	Ruheschnelle
v_{\sim}	m/s	Schallschnelle
x	m	eindimensionale Laufkoordinate
Z	kg/m ² ·s	akustische Impedanz
Z_0	kg/m ² ·s	Schallkennimpedanz
Z_w	kg/m ² ·s	Wandkennimpedanz

Mathematische Operatoren

∂	Partielle Ableitung
Δ	Laplace-Operator
∇	Gradient
$Im\{.\}$	Imaginärteil
$Re\{.\}$	Realteil
$(.)^*$	komplex konjugiert
$(.)^T$	Transponiert

Schreibweisen

\vec{n}	Vektor n
\underline{n}	komplexe Größe n
\hat{n}	Amplitude der Größe n
\tilde{n}	Effektivwert der Größe n

Indizes

B	Berechnet mit FEM
FEM	Bezogen auf die FEM-Berechnung
i	das i -te Element
K	Kartierung
M	gemessen
$Real$	Bezogen auf die reale Messung
S	Simuliert mit VR
$Unity\ verbessert$	Bezogen auf die verbesserte VR Umgebung
\ddot{u}	überschlägig
0	Ruhegröße
\sim	Wechselgröße

Abkürzungen

ADU	Analog-Digital-Umsetzer
BEM	Boundary Element Method
CRC	Crew Rest Compartment
DAU	Digital-Analog-Umsetzer
DFT	Diskrete Fourier Transformation
FEM	Finite-Elemente-Methode
FFT	Fast Fourier Transformation
HAW	Hochschule für Angewandte Wissenschaften Hamburg
HRIR	Head Related Impulse Response
HRTF	Head Related Transfer Function
LLfM	LiveLink for MATLAB
LTI	Linear Time Invariant
LZI	Linear Zeitinvariant
MAC	Modal Assurance Criterion
Mik.	Mikrofon
RIR	Raum Impulsantwort
RTF	Raum Transferfunktion
SQ	Spiegelschallquelle
TCP/IP	Transmission Control Protocol / Internet Protocol
vgl.	vergleiche
VR	Virtuelle Realität (engl. Virtual Reality)

Abbildungsverzeichnis

Abbildung 2-1: Unterteilung der Raumakustik [nach (Lerch, et al., 2009)]	11
Abbildung 2-2: Analoge und digitale Darstellung eines Signals [nach (Technische Universität Chemnitz, 2022)]	16
Abbildung 2-3: Hauptkomponenten eines Analog-Digital-Umsetzers [aus (Hesselmann, 1987)].....	16
Abbildung 2-4: Beziehungen zwischen Eingangssignal und Ausgangssignal und Systemverhalten eines LZI-Systems im Zeit- und Frequenzbereich [aus (Vorländer, 2020)].....	19
Abbildung 2-5: Links : FFT-Analyse von harmonischen Signalen mehrerer Frequenzen, die schwarzen Kurven weisen keinen Leakage-Effekt auf; Rechts : FFT-Analyse mit Hanning Fenster [nach (DEWESoft, 2022)]	20
Abbildung 3-1: Modellierung der virtuellen Akustik [nach, (Serafun, et al., 2018), (Vorländer, et al., 2014)].....	21
Abbildung 3-2: Berechnungsverfahren der Impulsantwort [nach (Vorländer, 2020, p. 146)].....	23
Abbildung 3-3: Simulation der geometrischen Raumakustik mit Raytracing (links) [nach (Svensson, 2002)] und Spiegelschallquellen (rechts) [nach (Vorländer, 2020)]	25
Abbildung 3-4: aixCAVE System der RWTH Aachen [aus (Schröder, et al., 2010)].....	28
Abbildung 3-5: Einstellungsmöglichkeiten für die Schallquelle in dem Steam Audio Plug-in.....	29
Abbildung 3-6: Globale Einstellungen der Steam Audio Engine.....	31
Abbildung 3-7: Materialzuweisung zur Verfeinerung der Impulsantwortberechnung; Jede Farbe repräsentiert ein unterschiedliches Material [nach (Resonance Audio (b), 2022)].....	33
Abbildung 4-1: Verteilung der Raummoden bis $f=500$ Hz in der Schallabsorptionseinrichtung (Oben), in der Klimakammer (Mitte) und im Mock-up des CRCs (Unten) [nach (Melcher, 2022)].....	36
Abbildung 4-2: Crew Rest Compartment und das eingesetzte Mock-up [eigene Darstellung].....	37
Abbildung 4-3: Komponenten und Aufbau der Messtechnik	38
Abbildung 4-4: Modell des CRC Mock-ups in COMSOL Multiphysics	40
Abbildung 4-5: Randbedingungen des Modells; Links : Raumabgrenzungen mit Absorptionsgrad; Rechts : Lautsprechermembran mit Beschleunigung und schallhartes Mikrofonverstärkergehäuse	41
Abbildung 4-6: Schalldruckdifferenzen entlang einer Schnittlinie für Netze mit verschiedenen Feinheitsgraden.....	43
Abbildung 4-7: Vernetztes CRC Mock-up	44
Abbildung 4-8: CRC Mock-up in der virtuellen Umgebung.....	45
Abbildung 4-9: Einstellungen der Schallquelle und des Mikrofons für das Resonance Audio Plug-in.....	46
Abbildung 4-10: Lautsprechereinstellungen für das Steam Audio Plug-in.....	47
Abbildung 5-1: Position der Messmikrofone, des Lautsprechers und der Mikrofonverstärker für die Messung der Frequenzantwort	49

Abbildung 5-2: Messpunkte für die Kartierungsmessung; Position des Lautsprechers und der Mikrofonverstärker bleibt unverändert	50
Abbildung 5-3: Messaufbau für die Messung der Übertragungsfunktion des Lautsprechers	51
Abbildung 5-4: Gemessene Membranbeschleunigung des Lautsprechers	51
Abbildung 5-5: Messaufbau für die Kalibrierung der Messmikrofone	52
Abbildung 5-6: Gemessene Empfindlichkeiten der Messmikrofone	53
Abbildung 5-7: Gemessene Frequenzantwort des CRC Mock-ups	54
Abbildung 5-8: Berechnete Frequenzantwort des CRC Mock-ups	55
Abbildung 5-9: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial Holzplatte ; Resonance Audio Plug-in	57
Abbildung 5-10: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial Holzdecke ; Resonance Audio Plug-in	57
Abbildung 5-11: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial Sperrholz ; Resonance Audio Plug-in	58
Abbildung 5-12: Simulierte Frequenzantwort der CRC Mock-ups; Steam Audio Plug-in	59
Abbildung 5-13: Messaufbau für die Kartierung des Schallfeldes in der Ebene E2	61
Abbildung 5-15: Gemessene Schalldruckpegelverteilung in der Ebene E2; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	62
Abbildung 5-14: Gemessene Schalldruckpegelverläufe in der Ebene E1; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	62
Abbildung 5-16: Berechnete Schalldruckpegelverläufe in der Ebene E1; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz;	63
Abbildung 5-17: Berechnete Schalldruckpegelverläufe in der Ebene E2; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	64
Abbildung 5-18: Simulierte Amplitudenpegelverläufe in der Ebene E1 mit dem Plug-in Resonance Audio; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz;	65
Abbildung 5-19: Simulierte Amplitudenpegelverläufe in der Ebene E2 mit dem Plug-in Resonance Audio; linker Verlauf: fm1, K = 180 Hz ; rechter Verlauf: fm2, K = 170 Hz	65
Abbildung 5-21: Simulierte Amplitudenpegelverläufe in der Ebene E2 mit dem Plug-in Steam Audio; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	66
Abbildung 5-20: Simulierte Amplitudenpegelverläufe in der Ebene E1 mit dem Plug-in Steam Audio; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	66
Abbildung 6-2: Graphische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung-Steam Audio Plug-in	71
Abbildung 6-1: Grafische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung-Resonance Audio	71
Abbildung 6-3: Graphische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung - FEM-Modell	74
Abbildung 7-1: Konzept zur Verbesserung der Akustik in VR	77
Abbildung 7-2: Simulierte Amplitudenpegelverläufe mit dem Verbesserungskonzept in der Ebene E1; linker Verlauf: fm1, K = 180 Hz; rechter Verlauf: fm2, K = 170 Hz	80

Abbildung 7-3: Simulierte Amplitudenpegelverläufe mit dem Verbesserungskonzept in der Ebene E2; linker Verlauf: $f_{m1}, K = 180 \text{ Hz}$; rechter Verlauf: $f_{m2}, K = 170 \text{ Hz}$	81
Abbildung 7-4: Amplitudenpegeldifferenzen zwischen COMSOL und Unity verbessert in E1, $f_{m2}, K = 170 \text{ Hz}$	82
Abbildung 7-5: Amplitudenpegeldifferenzen zwischen der realen Umgebung und Unity verbessert in E2, $f_{m2}, K = 170 \text{ Hz}$	83
Abbildung 7-7: Grafische Darstellung der MAC-Faktoren für den Vergleich reale Umgebung-Unity verbessert.....	85
Abbildung 7-6: Grafische Darstellung der MAC-Faktoren für den Vergleich FEM-Unity verbessert	85

Tabellenverzeichnis

Tabelle 2.1: Schallpegelwerte mit den entsprechenden Schalldrücken [nach (Möser, 2015)].....	10
Tabelle 2.2: Auflösung ΔA und Quantisierungsfehler bei unterschiedlichen Bit-Anzahlen	18
Tabelle 4.1: Eigenschaften der untersuchten Messumgebungen.....	35
Tabelle 4.2: Technische Spezifikationen der Messtechnikkomponenten	39
Tabelle 4.3: Eingestellte Umgebungsbedingungen des Ausbreitungsmediums	40
Tabelle 5.1: Einstellungen der myDAQ Messkarten.....	51
Tabelle 6.1: MAC-Analyse zwischen der realen Umgebung und dem Resonance Audio Plug-in.....	71
Tabelle 6.2: MAC-Analyse zwischen der realen Umgebung und dem Steam Audio Plug-in.....	71
Tabelle 6.3: MAC-Analyse zwischen der realen Umgebung und dem FEM-Modell.....	74
Tabelle 7.1: MAC-Faktoren FEM-Unity verbessert	84
Tabelle 7.2: MAC-Faktoren reale Umgebung-Unity verbessert.....	84

1. Einleitung

Die virtuelle Realität (VR) ist eine schnell wachsende Technologie, die einem die Möglichkeit bietet, in eine virtuelle, dreidimensionale Umgebung einzutauchen und mit dieser in Echtzeit zu interagieren. Wie der Name andeutet, soll diese Umgebung eine nach den Wünschen des Anwenders personalisierte Realität darstellen. Die technologischen Entwicklungen in den letzten Jahren treiben nicht nur die Verfügbarkeit der VR-Systeme und VR-Komponenten, sondern auch die Popularität der virtuellen Realität an. Der Vorteil der Personalisierung ist für viele Branchen attraktiv. So können zum Beispiel in der Unterhaltungsindustrie neue Erlebnisse angeboten werden, um neue Kunden zu gewinnen. Es können auch andere Bereiche identifiziert werden, die von der virtuellen Realität profitieren können. In der Architektur können fotorealistic Räume, Gebäude oder sogar ganze Städte dreidimensional modelliert werden, sodass die Notwendigkeit der realen Entwurfsmodelle entfällt. Entwicklungsprozesse technischer Produkte können mit der virtuellen Realität schneller durchgeführt werden (Vorländer, 2021). Auch weitere Bereiche, wie zum Beispiel die Wissenschaft, die Medizin oder die Lehre, können von der VR profitieren (Schröder, et al., 2010).

Bei der jahrelangen Weiterentwicklung der virtuellen Realität wurden die visuellen Aspekte in den Vordergrund gestellt. Die nur durch Bilder erzeugte Immersion ist für die meisten Applikationen ausreichend. Da aber der Mensch seine Umgebung mit allen Sinnesorganen aufnimmt, kann der Immersionsgrad deutlich erhöht werden, wenn auch zusätzliche Stimuli in die Simulation integriert werden. Im Gegensatz zu der visuellen Wahrnehmung unserer Umgebung, welche durch das Schließen der Augen unterbrochen werden kann, ist die auditive Wahrnehmung nicht unterbrechbar. Beim Eintauchen in eine virtuelle Umgebung beeinflusst somit auch die Akustik den Immersionsgrad. Diese zu simulieren wurde im Laufe der Zeit immer wichtiger und gewann an Aufmerksamkeit seitens der Wissenschaft. So entwickelten Forscherteams neue Methoden und Algorithmen, um die Akustik in einer virtuellen, dreidimensionalen Umgebung so realitätsnah wie möglich zu gestalten und in VR-Systemen zu integrieren.

Viele der heute verfügbaren Methoden für eine realitätsnahe Audiosimulation finden eine Anwendung verstärkt im wissenschaftlichen Umfeld. Das liegt daran, dass man dafür komplexe Rechnerarchitekturen braucht, die für den gewöhnlichen VR-Benutzer nicht

zur Verfügung stehen. Außerdem haben einige davon die Entwicklungsphase noch nicht verlassen, sodass sie auf dem Markt noch nicht erschienen sind. Die Audiowiedergabe virtueller Applikationen, welche auf dem kommerziellen Markt zu finden sind, wird in den meisten Fällen von der Spiel-Engine übernommen. Meistens erzeugen sie eine zufriedenstellende Akustik, die aber nicht der Raumakustik der Szene gleichzusetzen ist.

Einige Unternehmen haben sich mit diesem Problem befasst, indem sie verschiedene Plug-ins für kommerzielle Spiel-Engines entwickelt haben. So entstanden Softwarepakete wie Steam Audio von der Firma Valve Corporation®, Resonance Audio von Google oder SoundToolKit des Unternehmens Techmo - Voice Technologies. Über die Qualität der Simulationen mit diesen Plug-ins lässt sich leider nicht viel sagen, weil sie kaum untersucht wurde. Preuß führte im Rahmen seiner Abschlussarbeit eine Untersuchung der Qualität der akustischen Plug-ins durch. Anhand eines Vergleichs der Akustiksimulationen mit den Plug-ins Steam Audio und SoundToolKit und der Simulation mit der Software EASE konnte festgestellt werden, dass die Ergebnisse große Unterschiede aufweisen und die Plug-ins die etablierte Simulationssoftware EASE nicht ersetzen können (Preuß, 2021). Obwohl das Problem der fehlenden raumakustischen Simulationen für den kommerziellen Markt erkannt wurde, sind die heutigen Lösungen noch nicht reif genug, um mit den Ergebnissen der Simulationen aus der Wissenschaft und Forschung verglichen zu werden.

Daraus resultiert, dass die Notwendigkeit besteht, eine Methode zu entwickeln mit der eine realitätsnahe raumakustische Simulation in einer kommerziellen Umgebung realisiert werden kann, damit in der Zukunft Anwendungen sowohl in der Wissenschaft und Forschung als auch in der Industrie und Produktentwicklung möglich sind.

Ziel der Arbeit

Somit liegt das Ziel der vorliegenden Arbeit in der Entwicklung einer Methode, um eine realitätsnahe Raumakustik im tieffrequenten Bereich für die virtuelle Umgebung der kommerziellen Spiel-Engine Unity zu erzeugen. Die Raumakustik soll mittels einer Berechnung mit der Finite-Elemente-Methode (FEM) simuliert werden und in Echtzeit mit der virtuellen Umgebung von Unity verkoppelt, sodass ein VR Erlebnis mit realitätsnaher Akustik ermöglicht wird.

Methodik

Für die Erfüllung des Zieles wird das akustische Verhalten einer ausgewählten Umgebung anhand einer FEM-Berechnung ermittelt. Bevor die Ergebnisse der FEM-Berechnung für die Entwicklung der Methode eingesetzt werden, müssen sie mit der realen Umgebung verglichen werden, um sicherzustellen, dass sie die Realität wiedergeben. Zusätzlich wird auch das Verhalten der Audio Plug-ins Steam Audio und Resonance Audio im tieffrequenten Bereich mit der realen Umgebung verglichen. Die entwickelte Methode beruht auf einer Verbindung zwischen MATLAB und COMSOL Multiphysics, sowie zwischen MATLAB und Unity. Somit wird die Position des VR Benutzers aus Unity exportiert und anhand dieser eine realitätsnahe Audiowiedergabe mit den Berechnungsdaten aus COMSOL Multiphysics erzeugt.

Struktur der Arbeit

Angelehnt an das beschriebene Verfahren lässt sich die Arbeit wie folgt strukturieren:

Das zweite Kapitel befasst sich mit dem theoretischen Basiswissen, welches für das Verständnis und die Bearbeitung der angestrebten Methode notwendig ist. Es werden die Grundlagen der Schallwellen und deren Verbreitung in einem Schallfeld präsentiert, sowie das Grundwissen der wellentheoretischen Raumakustik vermittelt. Man geht im zweiten Kapitel auch auf die Grundlagen der digitalen Signalverarbeitung ein.

Im dritten Kapitel wird auf die Erzeugung der virtuellen akustischen Simulation näher eingegangen. Hier werden die Mechanismen hinter den einzelnen Komponenten einer Auralisation beschrieben, und einige Anwendungen präsentiert. Weiterhin wird die Funktionsweise der untersuchten Audio Plug-ins erläutert.

Das vierte Kapitel definiert die eingesetzte Messumgebung. Es werden die benötigten Messvorbereitungen für die reale Messung, die Erstellung des FEM-Modells und die Erzeugung der virtuellen Umgebung präsentiert.

Anschließend werden im fünften Kapitel die Durchführung der Messungen vorgestellt und die Messergebnisse dargestellt und bewertet.

Im sechsten Kapitel werden die Messungen miteinander verglichen, sodass eine Aussage über die Anwendbarkeit der Audio Plug-ins aber auch der FEM-Berechnung zur Erzeugung einer realitätsnahen akustischen Nachbildung getroffen werden kann.

Mit den Ergebnissen aus dem fünften und sechsten Kapitel wird im siebten Kapitel eine Methode präsentiert, mit der sich das Ziel der Arbeit realisieren lässt. Diese wird in Unity angewandt und mit den Messergebnissen aus der realen und FEM-Umgebung verglichen.

Im letzten Teil wird eine Aussage über die Erfüllung des vorgenommenen Ziels der Arbeit getroffen und die weiteren Schritte für die Verbesserung und Entwicklung der Methode vorgeschlagen.

2. Theoretische Grundlagen

In diesem Kapitel werden die Grundlagen der Physik der Schallfelder, der Raumakustik, welche ein Teilgebiet der Akustik darstellt, sowie der digitalen Signalverarbeitung präsentiert. Die vorgestellten mathematischen und physikalischen Zusammenhänge und die Begrifflichkeiten haben für die Bearbeitung und das Verständnis der Arbeit eine hohe Relevanz.

2.1. Physik der Schallfelder

Die Akustik ist die Wissenschaft vom Schall und befasst sich mit dessen Entstehung, Eigenschaften und Ausstrahlung (Kuttruff, 2006), (Lerch, et al., 2009).

Wie die Definition besagt, ist der Schall ein zentrales Phänomen in der Akustik. Als Schall lassen sich die Schwingungen eines Mediums beschreiben, die durch eine externe Kraft aus ihrer Gleichgewichtslage gebracht werden und sich als freie Wellen durch das Medium, auch Schallfeld genannt, ausbreiten (Weinzierl, 2008). Die Schallfelder können gasförmiger, flüssiger oder fester Natur sein. In gasförmigen und ausgedehnten flüssigen Medien treten praktisch keine Schubspannungen auf, sodass man nur Longitudinalwellen, die parallel zur Ausbreitungsrichtung schwingen, beobachten kann (Meyer & Neumann, 1979).

Der Schall kann auf molekularer Ebene als eine Schwingung der Luftmoleküle¹ um ihre Ruhelage betrachtet werden. Diese verursachen zeit- und ortsabhängige Kompressionen und Expansionen des Mediums, die durch lokale Dichte-, Druck- und Geschwindigkeitsänderung beschrieben werden können. Die Überlagerungen der dynamischen Anteile und der ruhenden Anteile bilden die Gesamtfeldgrößen. Mathematisch lassen sich diese folgendermaßen ausdrücken (Meyer & Neumann, 1979):

$$\rho_{ges} = \rho_0 + \rho_{\sim} \quad (2-1)$$

$$p_{ges} = p_0 + p_{\sim} \quad (2-2)$$

$$v_{ges} = v_0 + v_{\sim} \quad (2-3)$$

¹ In dieser Arbeit wird ausschließlich Luft als Ausbreitungsmedium betrachtet.

wobei der Index „0“ die Ruhegrößen und der Index „~“ die Wechselgrößen bezeichnet. In der Betrachtung des Schalls spielen nur die Wechselgrößen eine bedeutende Rolle. Aus diesem Grund wird im weiteren Verlauf der Arbeit auf die Indizierung der Wechselgrößen, soweit es möglich ist, verzichtet.

Die Geschwindigkeit v bezeichnet die Geschwindigkeit, mit der die Luftmoleküle um ihre Ruhelage schwingen und wird Schallschnelle oder nur Schnelle bezeichnet. Sie entspricht nicht der Schallgeschwindigkeit c , mit der sich die Longitudinalwellen im Medium ausbreiten. Die Schnelle ist eine vektorielle Feldgröße und man kann sie als die partielle Ableitung der Auslenkung $\vec{\xi}$ nach der Zeit eines Moleküls mathematisch beschreiben (Kuttruff, 2006):

$$\vec{v}(\vec{r}, t) = \frac{\partial \vec{\xi}(\vec{r}, t)}{\partial t} \quad (2-4)$$

Bei den schnellen Druck- und Dichteänderungen wird angenommen, dass keine Zeit für einen Wärmetransfer mit den benachbarten Teilchen bleibt. Dies bedeutet, dass aus thermodynamischer Sicht ein adiabatischer Prozess stattfindet. Wird noch berücksichtigt, dass die Schwankungen des Druckes p und der Dichte ρ in Bezug auf die Ruhegrößen sehr klein sind, kann die allgemeine Isentropenbeziehung linearisiert werden. Daraus resultiert folgende Beziehung zwischen dem Druck und der Dichte (Meyer & Neumann, 1979):

$$p = c^2 \cdot \rho \quad \text{mit} \quad c = \sqrt{\kappa \frac{p_0}{\rho_0}} \quad (2-5)$$

Der Isentropenexponent $\kappa = c_p/c_v$ drückt das Verhältnis der spezifischen Wärmekapazitäten bei konstantem Druck und bei konstanter Dichte aus. Für Luft beträgt er den Wert $\kappa = 1,4$ (Möser, 2015).

Betrachtet man nun ein sehr kleines Luftvolumen, in dem die Ruhegrößen Druck, Dichte und Schnelle konstant angenommen werden, so können folgende Gleichungen für den eindimensionalen Fall hergeleitet werden:

$$\frac{\rho}{\rho_0} = -\frac{\partial \xi}{\partial x} \quad (2-6)$$

$$\rho_0 \frac{\partial \xi}{\partial t} = - \frac{\partial p}{\partial x} \quad (2-7)$$

Die Teilchenauslenkung ξ wird in der Akustik selten verwendet. Sie kann eliminiert werden, indem man Gleichungen (2-4) und (2-5) in den Gleichungen (2-6) und (2-7) einsetzt.

$$\frac{\partial v}{\partial x} = - \frac{\partial p}{\rho_0 c^2 \partial t} \quad (2-8)$$

$$\rho_0 \frac{\partial v}{\partial t} = - \frac{\partial p}{\partial x} \quad (2-9)$$

Daraus kann man erkennen, dass nur die Angabe der Schnelle und des Druckes für die Beschreibung des Schalles erforderlich ist. Man kann in einem weiteren Schritt Gleichung (2-8) nach der Zeit und Gleichung (2-9) nach dem Ort ableiten und sie zusammenfügen. Der daraus resultierte Ausdruck ist nur noch von dem Schalldruck abhängig (Möser, 2015):

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (2-10)$$

Gleichung (2-10) repräsentiert die Wellengleichung für den eindimensionalen Fall. Da aber die Schallverteilung normalerweise ein dreidimensionales Phänomen ist, wird die Wellengleichungen für die drei Raumrichtungen erweitert. Mit der vektoriellen Schnelle, welche als $div v$ für alle Raumrichtungen ausgedrückt werden kann und dem skalaren Druck, der durch die Notation $grad p$ in dem dreidimensionalen Raum repräsentiert wird, lassen sich die Gleichungen (2-8) und (2-9) neu aufstellen. Erneut kann man sie ineinander setzen und die Schnelle eliminieren, sodass sich die allgemeine Wellengleichung für den dreidimensionalen Raum² ergibt (Möser, 2015):

$$\Delta p(\vec{r}, t) = \frac{1}{c^2} \frac{\partial^2 p(\vec{r}, t)}{\partial t^2} \quad (2-11)^3$$

² Für eine ausführliche Herleitung der Wellengleichung wird (Möser, 2015) empfohlen.

³ Der Vektor \vec{r} enthält die Achsen des kartesischen Koordinatensystems.

Zur Beschreibung des Schallfeldes ist demnach die Angabe des Schalldrucks ausreichend. Man kann die Wellengleichung auch für die Schnelle oder für die Dichte formulieren. Sie kann für jedes beliebige Koordinatensystem aufgestellt werden, somit beschränkt sich ihre Gültigkeit nicht nur auf das kartesische Koordinatensystem (Kuttruff, 2006).

Die im Schallfeld auftretenden Wechselgrößen lassen sich sehr einfach durch harmonische Schwingungen repräsentieren. Diese werden am einfachsten mithilfe der trigonometrischen Funktionen mathematisch beschrieben.

Somit kann der dynamische Druck p einer fortschreitenden Schallwelle als eine Kosinusfunktion ausgedrückt werden (Meyer & Neumann, 1979):

$$p(x, t) = \hat{p} \cdot \cos \left(\omega \left(t \pm \frac{x}{c} \right) + \varphi \right) \quad (2-12)$$

mit

$$\omega = 2\pi f$$

Die Größe \hat{p} repräsentiert die Amplitude der Schwingung. Mit φ kann ein Phasenwinkel berücksichtigt werden. Die Kreisfrequenz ω ist von der Frequenz f der Schwingung abhängig. Führt man die Wellenlänge λ und Wellenzahl k ein,

$$\lambda = \frac{c}{f} \quad (2-13)$$

$$k = \frac{2\pi}{\lambda} = \frac{\omega}{c} \quad (2-14)$$

so lässt sich Gleichung (2-12) zu

$$p(\vec{r}, t) = \hat{p} \cdot \cos (\omega t \pm k\vec{r} + \varphi) \quad (2-15)$$

für den dreidimensionalen Raum mit dem Vektor $\vec{r} = (x, y, z)^T$ umschreiben. Das Vorzeichen des ortsabhängigen Terms spiegelt die Ausbreitungsrichtung der Wellenkomponente wider. Bei einem negativen Vorzeichen verläuft diese in positiver Achsenrichtung, bei einem positiven Vorzeichen in negativer Achsenrichtung.

Zur Vereinfachung kann Gleichung (2-15) in ihrer komplexen Form dargestellt werden (Kuttruff, 2017):

$$p(\vec{r}, t) = \text{Re}\{\hat{p}(\vec{r}) \cdot e^{i(\omega t \pm k\vec{r} + \varphi)}\} \quad (2-16)$$

und mit der komplexen Amplitude des Schalldruckes $\underline{p}(\vec{x})$ umgeformt:

$$p(\vec{r}, t) = \text{Re}\{\underline{p}(\vec{r}) \cdot e^{i(\omega t \pm k\vec{r})}\} \quad (2-17)$$

$$\underline{p}(\vec{r}) = \hat{p}(\vec{r}) \cdot e^{i\varphi} \quad (2-18)$$

Die komplexe Schreibweise kann man auch auf andere Feldgrößen anwenden. Sie hat den Vorteil, dass Differential- und Integraloperationen sehr einfach durchgeführt werden können.

Setzt man den komplexen Schalldruck in die Wellengleichung ein, so kann die Helmholtz-Gleichung formuliert werden. Der Vorteil dieser Schreibform ist die Eliminierung der Zeitabhängigkeit. Es entsteht somit eine Wellengleichung im Frequenzbereich (Möser, 2015):

$$\Delta \underline{p} + k^2 \underline{p} = 0 \quad (2-19)$$

Eine wichtige Größe des Schallfeldes ist die akustische Impedanz Z , welche sich als das Verhältnis von Amplitude des Schalldruckes zur Amplitude der Schnelle im eindimensionalen Fall ausdrücken lässt. Für den Fall einer fortschreitenden Welle werden die komplexen, ortsabhängigen Amplituden der Feldgrößen eingesetzt. Dieses Verhältnis bildet die Schallkennimpedanz \underline{Z}_0 des Schallfeldes und ist ein Maß für den Widerstand des Schallfeldes gegen eine fortlaufende Schallwelle (Meyer & Neumann, 1979):

$$\underline{Z}_0 = \frac{\underline{p}(\vec{r})}{\underline{v}(\vec{r})} = \rho_0 c_0 \quad (2-20)$$

Der für die Akustik interessante Druckbereich befindet sich zwischen der Hörschwelle ($2 \cdot 10^{-5} \text{ Pa}$) und der Schmerzgrenze des Ohres (200 Pa) für eine Frequenz von $f = 1000 \text{ Hz}$. Da sich dieser Bereich über mehrere Zehnerpotenzen erstreckt, wird er in einer logarithmischen Skala als Pegel angegeben. Der Schalldruckpegel L_p wird somit als logarithmisches Verhältnis, bezogen auf die Hörschwelle, gebildet (Möser, 2015):

$$L_p = 20 \cdot \log_{10} \left(\frac{\tilde{p}}{p_{ref}} \right) \quad (2-21)$$

Um einen Eindruck für die Größenordnung des eingeführten Schalldruckpegels zu bekommen, präsentiert Tabelle 2.1 die Schallemissionen alltäglicher Vorkommnisse als Schalldrücke und Schallpegelwerte.

Tabelle 2.1: Schallpegelwerte mit den entsprechenden Schalldrücken [nach (Möser, 2015)]

Situation	Schalldruck in [Pa]	Schalldruckpegel in [dB]
Hörschwelle	$2 \cdot 10^{-5}$	0
Wald bei wenig Wind	$2 \cdot 10^{-4}$	20
Bibliothek	$2 \cdot 10^{-3}$	40
Büro	$2 \cdot 10^{-2}$	60
befahrene Straße	$2 \cdot 10^{-1}$	80
Presslufthammer, Sirene	$2 \cdot 10^0$	100
Start von Düsenflugzeugen	$2 \cdot 10^1$	120
Schmerzgrenze	$2 \cdot 10^2$	140

2.2. Raumakustik

Die Betrachtungen aus Abschnitt 2.1 werden nun auf die Ausbreitung des Schalls in einer geschlossenen Umgebung angewendet. In einem Raum treten Effekte, wie zum Beispiel die Reflexion und die Schallabsorption an den Raumbegrenzungen, auf. Dadurch entsteht eine besondere Zusammensetzung des Schallfelds, welche von der Raumakustik charakterisiert wird (Kuttruff, 2017). Die Raumakustik besitzt mehrere Methoden für die Beschreibung und Behandlung der Schallausbreitung. Man unterscheidet zwischen der wellentheoretischen, geometrischen und statistischen Raumakustik. Die Differenzierung beruht auf dem Verhältnis der Wellenlänge λ zu den Raumabmessungen. Laut Gleichung (2-13) hängt die Wellenlänge von der Frequenz f des Schalls ab. Ist also die Frequenz des Schalls und die Schallgeschwindigkeit c der Umgebung bekannt, so kann die Behandlungsmethode festgelegt werden (Meyer & Neumann, 1979). Abbildung 2-1 präsentiert die unterschiedlichen Bereiche der Raumakustik und deren Eigenschaften.

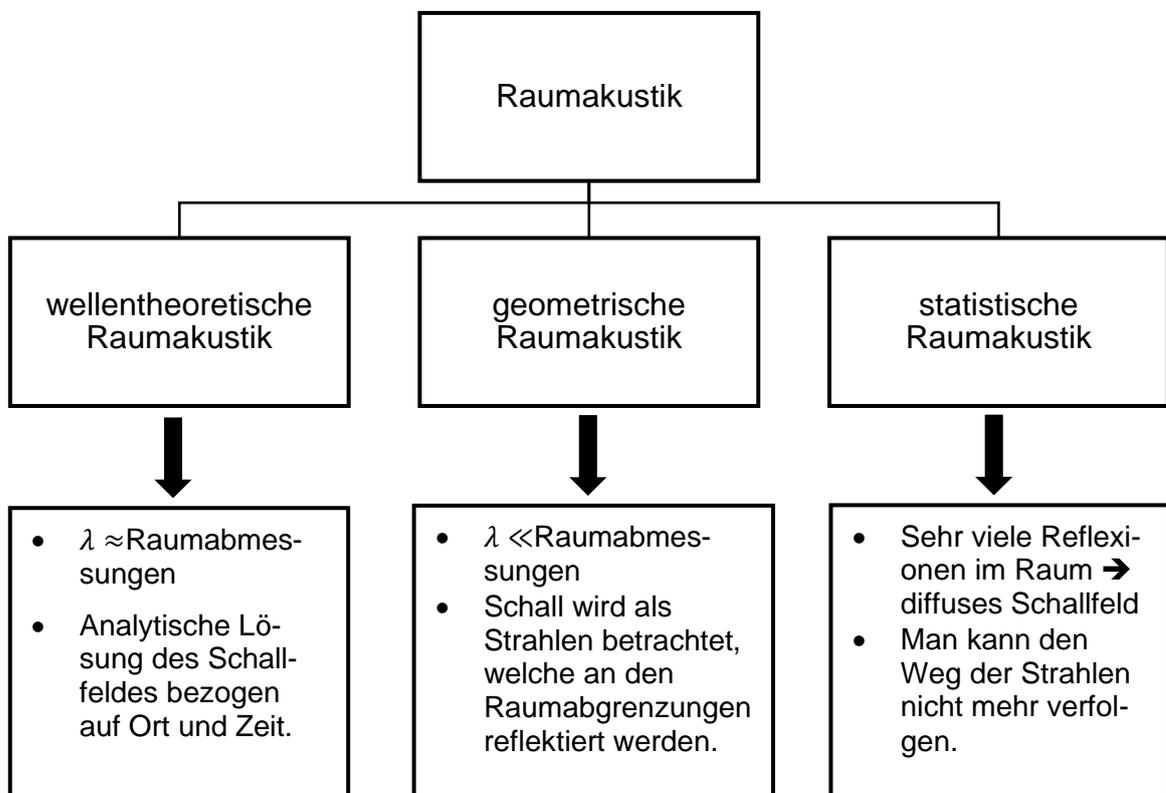


Abbildung 2-1: Unterteilung der Raumakustik [nach (Lerch, et al., 2009)]

Wird ein Schallfeld in einem geschlossenen Raum betrachtet, so kann man die Reflexionen der fortschreitenden Schallwellen an den Wänden beobachten. Die Reflexionspunkte wirken ähnlich wie eine Schallquelle und erzeugen eine zurücklaufende Welle.

Aufgrund der Absorption der Begrenzungsflächen hat die reflektierte Schallwelle eine von dem komplexen Reflexionsfaktor \underline{r} abhängige Phasenverschiebung und geringere Amplitude. Der Absolutwert des Reflexionsfaktors $|\underline{r}|$ liegt im Bereich zwischen 0 und 1, wobei $|\underline{r}| = 0$ eine vollständig absorbierende Fläche und $|\underline{r}| = 1$ eine schallharte Fläche repräsentiert. Der Reflexionsfaktor hängt von der Impedanz \underline{Z}_W der reflektierenden Wand ab und wird durch folgende Gleichung ausgedrückt:

$$\underline{r} = \frac{\underline{Z}_W \cdot \cos \theta - \underline{Z}_0}{\underline{Z}_W \cdot \cos \theta + \underline{Z}_0} \quad (2-22)$$

θ = Einfallswinkel der Schallwelle

Die Wandimpedanz \underline{Z}_W ist ähnlich, wie die spezifische Impedanz des Mediums definiert:

$$\underline{Z}_W = \frac{\underline{p}_W}{\underline{v}_W} \quad (2-23)$$

Für die unmittelbar vor der Wandoberfläche komplexen Amplituden des Schalldruckes \underline{p}_W und der Schnelle \underline{v}_W werden nur die Anteile, welche normal zur Wand sind, betrachtet.

Mithilfe des Faktors \underline{r} kann der Absorptionsgrad α des Wandmaterials eingeführt. Er spiegelt den absorbierten Anteil des gesamten einfallenden Schalls wider (Kuttruff, 2017).

$$\alpha = 1 - |\underline{r}|^2 \quad (2-24)$$

Treffen sich die fortlaufenden und die reflektierten Wellen, so entstehen aufgrund der konstruktiven und destruktiven Interferenz stehende Wellen. Weil die Raumbegrenzungen in der Realität die Schallwelle weder vollständig reflektieren noch vollständig

absorbieren können und das Ausbreitungsmedium eine dämpfende Wirkung auf den Schall hat, besteht das Schallfeld aus einer Mischung aus fortlaufenden und stehenden Wellen. Die fortlaufenden Wellen bilden das aktive Feld, die stehenden Wellen das reaktive Feld. In einer stehenden Welle beträgt der Phasenunterschied $\Delta\varphi$ zwischen der Schallschnelle und dem Schalldruck den Wert $\Delta\varphi = 90^\circ$ (Möser, 2015).

Die eingeschlossene Luftmasse eines Raumes ist ein schwingungsfähiges Medium. Somit besitzt sie Eigenmoden und entsprechende Eigenfrequenzen. Betrachtet man einen idealisierten quadratischen Raum mit den Kantenlängen l_x , l_y und l_z in x -, y - und z -Richtung, so kann man die Eigenfrequenzen f_E des Raumes folgendermaßen berechnen (Möser, 2015):

$$f_E = \frac{c}{2} \cdot \sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2} \quad (2-25)$$

Für die Variablen n_x , n_y und n_z werden ganze Zahlen, die größer oder gleich null sind, eingesetzt. Diese repräsentieren die Ordnung der Raummoden entlang der entsprechenden Achsen.

Jede Raummode ist durch eine dreidimensionale, stehende Welle abgebildet, die sich aus den einzelnen Wellenmoden in den drei Raumrichtungen zusammensetzt. Es können drei Raummodentypen entstehen, die nach ihrer Ausbreitungsrichtung eingeteilt werden (Meyer & Neumann, 1979).

- Axiale Moden: Die Schallwellen verlaufen parallel zu einer Hauptkoordinatenachse. Zwei der Ordnungszahlen n_x , n_y , n_z sind Null.
- Tangentiale Moden: Die Ausbreitung der Schallwellen erfolgt senkrecht zu einer Raumkante und tangential zu einem Wandpaar. Eine der Ordnungszahlen n_x , n_y , n_z hat den Wert Null.
- Schräge oder oblique Moden: Die Wellen fallen schräg auf alle Wandpaare. Keine der Ordnungszahlen n_x , n_y , n_z hat den Wert Null.

Aufgrund der Energiedissipation durch Reibung an den Raumwänden, der Wandabsorption und der Dämpfung im Ausbreitungsmedium, ist es erforderlich, das Schallfeld mit einer Schallquelle kontinuierlich anzuregen, um einen bestimmten Schallzustand

aufrechtzuerhalten. Bei der Anregung des Schallfeldes in einer Eigenfrequenz ist das reaktive Feld dominant und die mit der Anregungsfrequenz f_E verbundene Mode prägt sich im Raum aus. Es entstehen somit Bereiche mit ausgeprägten Druckmaxima, Druckminima und Knotenpunkten. Mit der Erhöhung der Anregungsfrequenz werden immer mehr Raummoden erzeugt. Durch die Entstehung von vielen stehenden Wellen, die sich überlagern, können die einzelnen Moden im Schallfeld nicht mehr untereinander differenziert werden. Es bildet sich ein diffuses Feld, in dem kaum Druckvarianzen zu erkennen sind (Kuttruff, 2006).

Anhand der folgenden Formel wird die Anzahl $N_{f_{Gr}}$ der angeregten Eigenmoden in einem Raum mit dem Volumen V bis zu der Grenzfrequenz f_{Gr} berechnet (Meyer & Neumann, 1979):

$$N_{f_{Gr}} = \frac{4\pi V}{3} \left(\frac{f_{Gr}}{c}\right)^3 + \frac{\pi A}{4} \left(\frac{f_{Gr}}{c}\right)^2 + \frac{f_{Gr}L}{8c}$$

mit

(2-26)

$$A = 2(l_x l_y + l_x l_z + l_y l_z)$$

$$L = 4(l_x + l_y + l_z)$$

Für den Vergleich zweier unterschiedlichen Raummoden kann das Modal Assurance Criterion (MAC) eingesetzt werden. Das MAC ist ein statistischer Indikator, mit dem der Grad an Ähnlichkeit zweier Moden ermittelt werden kann. Mit diesem Verfahren lassen sich analytisch errechnete Raummoden mit experimentell gemessenen Raummoden vergleichen.

Der Eigenvektoren ψ_A der Raummode m wird mit dem Eigenvektor ψ_E der Raummode n verglichen. Die quantitative Methode drückt das normierte Skalarprodukt der Eigenvektoren aus.

$$MAC(m, n) = \frac{|\{\psi_A\}_m^T \{\psi_E\}_n^*|^2}{(\{\psi_A\}_m^T \{\psi_A\}_m^*) (\{\psi_E\}_n^T \{\psi_E\}_n^*)} \quad (2-27)$$

Gleichung (2-27) stellt die Berechnungsformel des MAC-Faktors in allgemeiner Form dar und berücksichtigt auch komplexe Eigenvektoren. Der MAC-Wert ist allerdings eine

reelle Zahl zwischen null und eins. Bei einem Wert über 0,9 kann man annehmen, dass bei den zu vergleichenden Raummoden eine hohe Ähnlichkeit existiert (Pastor, et al., 2012).

2.3. Digitale Signalverarbeitung

Die genaue Erfassung physikalisch-technischer Vorgängen aus unserer Umgebung ist essenziell für die Verarbeitung, Analyse und daraus folgende Verwendungen.

Analoge Signale werden mithilfe der rechnergestützten Erfassung digitalisiert und für die weitere Verarbeitung aufbereitet.

Analoge Signale zeichnen sich dadurch aus, dass ihr Amplituden- und Zeitverlauf von kontinuierlicher Natur ist.

Man unterteilt analoge Signale in deterministische und nicht deterministische Signale. Der Unterschied besteht in der Möglichkeit der Vorausbestimmung des zeitlichen Verlaufes des Signals. So kann man bei deterministischen analogen Signalen bei einem vorgegebenen Zeitpunkt den Amplitudenwert bestimmen. Nicht deterministische Signale kommen in der Natur zum Beispiel als Rauschen vor. Periodische Schwingungen sind ein klassisches Beispiel für deterministische Signale (Hesselmann, 1987).

Für die Digitalisierung eines analogen Signals tastet ein Analog-Digital-Umsetzer (ADU) das Signal in bestimmten Zeitintervallen ab und kodiert sie in digitaler Form als Binärzahlen. Abbildung 2-2 präsentiert das gleiche Signal in analoger und digitaler Form. Ist der Prozess fehlerfrei und unter Beachtung bestimmter Regeln durchgeführt, so bietet die Digitalisierung von analogen Signalen eine Mehrzahl an Vorteilen an.

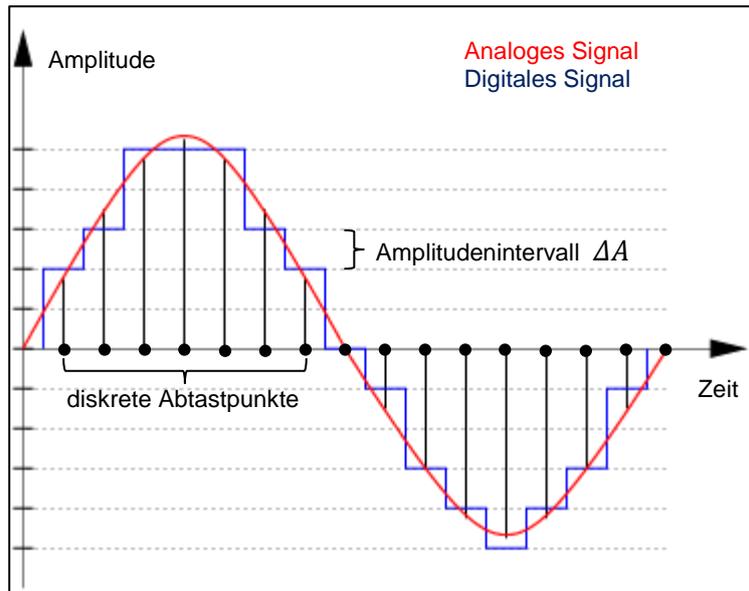


Abbildung 2-2: Analoge und digitale Darstellung eines Signals
[nach (Technische Universität Chemnitz, 2022)]

Mittels des ADUs werden zeit- und wertekontinuierlichen Signale in zeit- und wertedis-
kreten Signalen kodiert. Die Hauptkomponenten eines ADUs sind in Abbildung 2-3 dar-
gestellt.

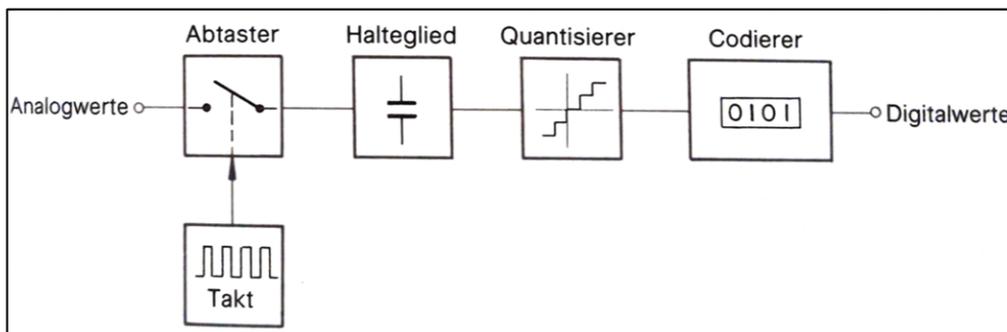


Abbildung 2-3: Hauptkomponenten eines Analog-Digital-Umsetzers [aus (Hesselmann, 1987)]

Die durch einen Sensor erfasste Messgröße wird zuerst zu bestimmten äquidistanten
Zeitpunkten von dem Abtaster erfasst. Um ein fehlerfreies Ergebnis zu erzeugen, muss
die Abtastfrequenz f_s nach dem Nyquist-Theorem gewählt werden (Hesselmann, 1987).

$$f_s > 2f_{max} \quad (2-28)$$

f_{max} ist die maximale Frequenz, die das abgetastete Signal hat. Besteht das Signal aus mehreren Frequenzanteilen, so wird der maximale Frequenzanteil berücksichtigt. Durch die Verletzung des Nyquist-Theorems wird durch den Abtaster ein verfälschtes Signal aufgenommen, welches fiktive Frequenzanteile aufweist. Dieser Fehler wird als Aliasing-Effekt bezeichnet. Um diesen Fehler bei Signalen mit unbekanntem Frequenzen zu vermeiden, kann das Signal vor dem Abtaster mit einem Antialiasing-Filter gefiltert werden. Somit können die Signalanteile mit einer Frequenz $f > 0,5f_s$ eliminiert werden. Der Antialiasing-Filter verhält sich also wie einen Tiefpassfilter.

Zusätzlich zu der Abtastung des zeitlichen Verlaufs des analogen Signals muss auch der kontinuierliche Amplitudengang diskretisiert werden, also in binäre Zahlen umkodiert werden. Ein ADU ist aus technischen Gründen auf eine bestimmte Bit-Anzahl limitiert, wodurch die Auflösung des diskretisierten Signals beeinflusst wird. Die Auflösung ΔA kann für einen Wertebereich A_{max} und eine Bit-Anzahl b folgendermaßen berechnet werden (Hesselmann, 1987):

$$\Delta A = \frac{A_{max}}{2^{b-1}} \quad (2-29)$$

Bei Messgrößen mit negativen und positiven Amplitudengrößen wird die verfügbare Bit-Anzahl um ein Bit reduziert, weil die Kodierung des Vorzeichens ein Bit benötigt. Der Prozess der Wertediskretisierung heißt Quantisierung.

Wie man aus Abbildung 2-2 erkennen kann, können die Werte der analogen Amplitude innerhalb eines Amplitudenintervalls als zwei unterschiedliche binäre Werte ausgedrückt werden. Dadurch können Abweichungen von dem wahren Wert entstehen, die einen halben Quantisierungsschritt betragen. Dieser Effekt der Quantisierung wird als Quantisierungsfehler bezeichnet. Aus Gleichung (2-29) folgt, dass das Amplitudenintervall ΔA und die Bit-Anzahl indirekt proportional sind. Somit hat der Quantisierungsfehler bei mehreren verfügbaren Bits einen kleineren Wert. Aus Tabelle 2.2 sind die Auflösungen ΔA und die Quantisierungsfehler für ein ADU mit der maximalen Eingangsamplitude $A_{max} = \pm 5 V$ bei unterschiedlichen verfügbaren Bit-Anzahlen b beispielhaft gezeigt.

Tabelle 2.2: Auflösung ΔA und Quantisierungsfehler bei unterschiedlichen Bit-Anzahlen

Verfügbare Bit-Anzahl n	Auflösung ΔA in [V]	Quantisierungsfehler in [V]
8	0,078	0,039
12	0,0049	$2,45 \cdot 10^{-3}$
24	$1,192 \cdot 10^{-6}$	$5,96 \cdot 10^{-7}$
32	$4,657 \cdot 10^{-9}$	$2,329 \cdot 10^{-9}$

Aus der Sicht der Signalverarbeitung können akustische Phänomene als eine Signalfaltung des Eingangssignals, also der Anregung eines Schallfeldes, mit der Impulsantwort der Umgebung betrachtet werden. Fast alle Übertragungssysteme aus der Akustik lassen sich als lineare zeitinvariante (LZI oder engl. LTI) Systeme beschreiben. Ist die Anregung $s(t)$ und das Verhalten des Systems $h(t)$ bekannt, so kann das Ausgangssignal $g(t)$ wie folgt berechnet werden (Vorländer, 2020):

$$g(t) = s(t) * h(t) \quad (2-30)$$

Mit Hilfe der Fourier Transformation lassen sich Funktionen zwischen dem Zeitbereich und dem Frequenzbereich transformieren. Es können somit Spektralanalysen durchgeführt werden, bei denen ein Signal in seine einzelnen Frequenzkomponenten mit den zugehörigen Amplituden zerlegt wird. Die Fourier Transformation ist ein wichtiges Werkzeug der Signalverarbeitung, weil eine Faltung im Zeitbereich mit einer Multiplikation im Frequenzbereich gleich ist. Ein weiterer Vorteil ist die Möglichkeit, beliebige Signale zu analysieren. Besteht das Signal aus einzelnen harmonischen Komponenten, so liefert die Fourier-Analyse jeden Frequenzanteil mit seiner entsprechenden Amplitude (Kuttruff, 2006).

Ein LZI-System kann ebenfalls im Frequenzbereich betrachtet werden, wenn das Eingangs- und Ausgangssignal und die Impulsantwort in den Frequenzbereich transformiert wird (vgl. Abbildung 2-4)

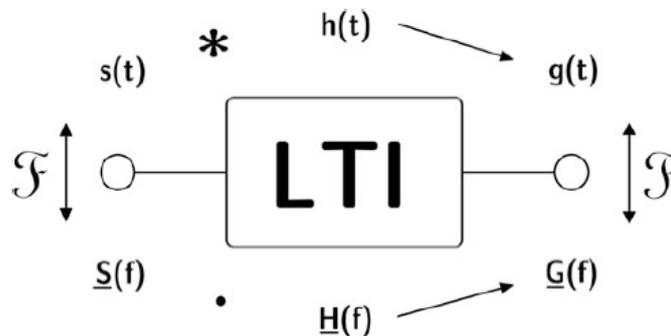


Abbildung 2-4: Beziehungen zwischen Eingangs- und Ausgangssignal und Systemverhalten eines LTI-Systems im Zeit- und Frequenzbereich [aus (Vorländer, 2020)]

Für die Transformation zeitdiskreter Signale wird die schnelle Fourier-Transformation (FFT) eingesetzt. Sie ist eine Form der diskreten Fourier-Transformation (DFT), hat aber den Vorteil, dass für die Analyse eine kürzere Rechendauer benötigt wird. Die DFT und die FFT können nur auf Datensätze mit finiten Abtastwerten, auch Samples genannt, angewendet werden.

Somit hat auch das Frequenzspektrum diskrete Stellen, an denen sogenannte Frequenzlinien gebildet werden. Die maximale Anzahl an Frequenzlinien stimmt mit der Anzahl der erzeugten diskreten Datenpunkte überein. Daraus resultiert, dass die Auflösung des Spektrums von der Abtastfrequenz, mit der das Eingangssignal erfasst wurde, abhängt (Müller & Möser, 2004), (DEWESoft, 2022).

Die FFT ist für die Analyse von Datensätzen mit einer Samplelänge gleich einer Potenz von zwei ausgelegt. Sind ganze Perioden des Eingangssignals in dem analysierten Datenblock enthalten, so fällt die ermittelte Frequenz mit einer Spektrallinie zusammen. Die Höhe dieser Linie entspricht der Amplitude des Eingangssignales. Bei einem Datenblock mit unvollendeten Schwingungen ist das Eingangssignal unstetig und das Ergebnis der FFT eine Verteilung von Spektrallinien über das ganze Spektrum. Die Signalfrequenz und Amplitude werden durch Spektrallinien benachbarter Frequenzen repräsentiert, die mit steigender Differenz von der tatsächlichen Frequenz an Höhe verlieren. Solche Fehler sind als Leakage-Effekt bekannt. Abbildung 2-5 stellt ein Beispiel für das Auftreten des Leakage-Effektes dar.

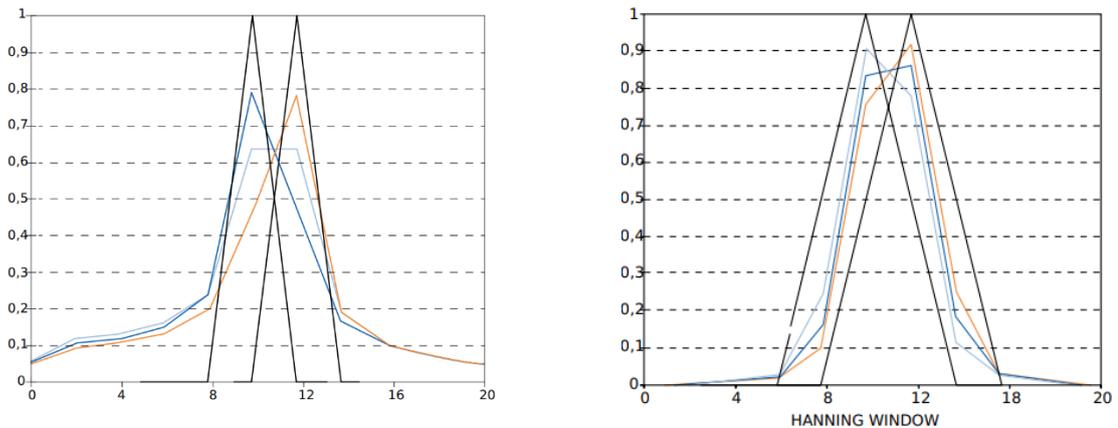


Abbildung 2-5: **Links:** FFT-Analyse von harmonischen Signalen mehrerer Frequenzen, die schwarzen Kurven weisen keinen Leakage-Effekt auf; **Rechts:** FFT-Analyse mit Hanning Fenster [nach (DEWESoft, 2022)]

Man kann diesen Fehler durch zwei Methoden vermeiden oder reduzieren:

- Auswahl einer Abtastfrequenz, sodass der Datensatz ganze Perioden des Signals enthält. Der Leakage-Effekt tritt somit nicht auf.
- Faltung des Eingangssignals mit einer Fensterfunktion, sodass der Einfluss des Anfangs- und Endbereich des Signals unterdrückt wird. Eine komplette Eliminierung des Leakage-Effekts ist nicht möglich, wird aber je nach Fensterfunktion unterschiedlich stark unterdrückt.

3. Akustik in der virtuellen Realität

Im Rahmen dieses Kapitels wird der aktuelle Stand der akustischen Entwicklungen und Technologien im Bereich der Virtual Reality präsentiert. Weiterhin werden die im späteren Verlauf der Arbeit eingesetzten Audio-Plug-ins *Steam Audio Plugin* der Firma Valve Corporation und Googles *Resonance Audio* aus der Spiel-Engine Unity vorgestellt.

Die Erzeugung akustischer Felder in virtuellen Umgebungen ist eine interdisziplinäre Aufgabe, welche Bereiche aus der klassischen Akustik, aus der Raumakustik, aus der Signalverarbeitung und aus der Informatik beinhaltet. Die Modellierung von Akustikfeldern, auch Auralisation genannt, beschäftigt sich mit der Vorhersage des akustischen Verhaltens von einzelnen Komponenten oder sogar kompletten Umgebungen (Vorländer, 2020). Der Modellierungsprozess der Raumakustik in VR lässt sich anhand folgender Abbildung näher betrachten:

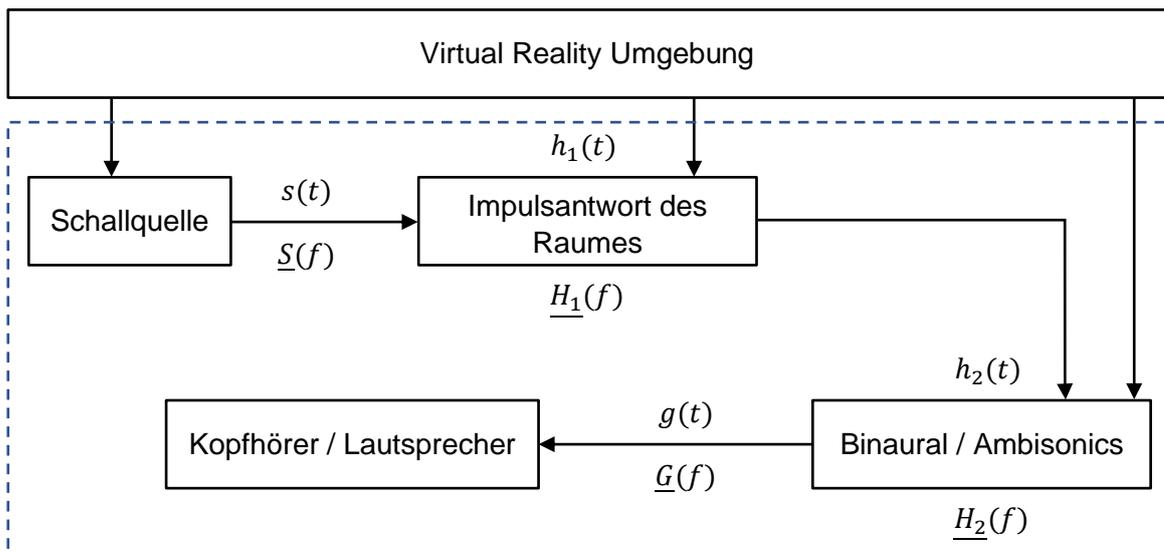


Abbildung 3-1: Modellierung der virtuellen Akustik [nach, (Serapun, et al., 2018), (Vorländer, et al., 2014)]

Die Elemente in dem gestrichelten Rechteck bilden die Grundsteine der Auralisation. In dieser Konfiguration werden sie auch für raumakustische Simulationen eingesetzt. Zusammen mit Informationen aus der virtuellen Realität wird die Modellierung einer virtuellen akustischen Umgebung erst ermöglicht. Die Wichtigkeit der Signalverarbeitung lässt sich in Abbildung 3-1 deutlich machen. Da der Prozess in digitaler Form durchgeführt wird, kommen die Informationsflüsse zwischen den einzelnen Bausteinen als Signale vor. Diese können sowohl im Zeit- als auch im Frequenzbereich betrachtet werden,

allerdings ist die Bearbeitung im Frequenzbereich durch die Konvertierung einer Signalfaltung in eine Signalmultiplikation deutlich einfacher gestaltet (Vorländer, et al., 2014). Für ein besseres Verständnis des Modellierungsprozesses werden nun die einzelnen Komponenten beschrieben.

Die Schallquelle erzeugt das Eingangssignal $s(t)$ für den Auralisationsprozess. Jedes schallerzeugende Vorkommnis, wie zum Beispiel eine Menschenstimme, Naturgeräusche oder musikalische Instrumente, lassen sich als Schallquelle interpretieren. Eine Szene kann auch mehrere Schallquellen beinhalten, die gleichzeitig ein Signal erzeugen. Die Gewinnung der Signale beruht auf unterschiedliche Verfahren. So können diese aus Aufnahmen, Modellen oder Synthesen stammen. Es ist allerdings wichtig, dass die Signalgestalt trocken ist, also keine ungewollten Komponenten, wie zum Beispiel Reflexionen aus der Aufnahmeumgebung, enthält. Bei einer Audioaufnahme kann ein trockenes Signal in einem reflexionsarmen Raum erzeugt werden. Die Generierung des akustischen Signals großer Schallquellen, wie zum Beispiel von Flugzeugen, wird durch eine Synthese erreicht. Da man solche Schallquellen in einer reflexionsarmen Umgebung nicht vermessen kann, wird der vom System generierte Gesamtschall als eine Zusammensetzung einzelnen harmonischen Komponenten betrachtet. Diese können durch separate Messungen oder Simulationen ermittelt werden. Anschließend werden sie durch eine Synthese zusammengebracht. Eine weitere Eigenschaft des Signals $s(t)$, die betrachtet werden muss, ist die dreidimensionale Richtcharakteristik der zu simulierenden Schallquelle. Sie beeinflusst die räumliche Energieausstrahlung der Schallquelle. Für Musikinstrumente, Menschenstimmen oder Lautsprecher werden diese in Datenbanken abgespeichert (Vorländer, 2020).

Eine zentrale Komponente in dem Modellierungsprozess ist die Raumimpulsantwort (RIR) $h_1(t)$ oder die Raumtransferfunktion (RTF) $\underline{H}_1(f)$ der virtuellen Umgebung. Sie beinhaltet die Informationen zu dem akustischen Verhalten der Umgebung. Mit diesem Verhalten wird das Eingangssignal beeinflusst, so dass das resultierende Schallereignis die Akustik der Umgebung widerspiegelt. Die Impulsantwort beinhaltet drei markante Bereiche: den Direktschall, die frühen Reflexionen und den Nachhall (Vorländer, et al., 2014). Je nach Anwendungsfall müssen nicht alle Bereiche mit gleicher Genauigkeit berechnet werden. Das menschliche Gehör benutzt Informationen aus dem Direktschall für die Lokalisierung der Schallquelle, allerdings werden im Nachhall, aufgrund der

Überlappung vieler Schallwellen, nur noch Intensitätsdifferenzen erkannt (Vorländer, et al., 2014). Auch die Rechenzeit, welche von der Berechnungsmethode abhängt, soll in Betracht genommen werden. So liefern numerische Berechnungen der Wellengleichung sehr genaue Ergebnisse, sie benötigen aber lange Berechnungszeiten.

Mit geometrischen Verfahren können hingegen kurze Rechenzeiten erreicht werden, allerdings verliert man an Genauigkeit, vor allem im tieffrequenten Bereich (Svensson, 2002). Diese Erkenntnisse müssen bei der Auswahl des Modellierungsverfahrens der Impulsantworten beachtet werden, sodass ein Optimum zwischen Genauigkeit und Rechenzeit erreicht wird.

Die Unterscheidung der Modellierungsverfahren wird in Abbildung 3-2 gezeigt.

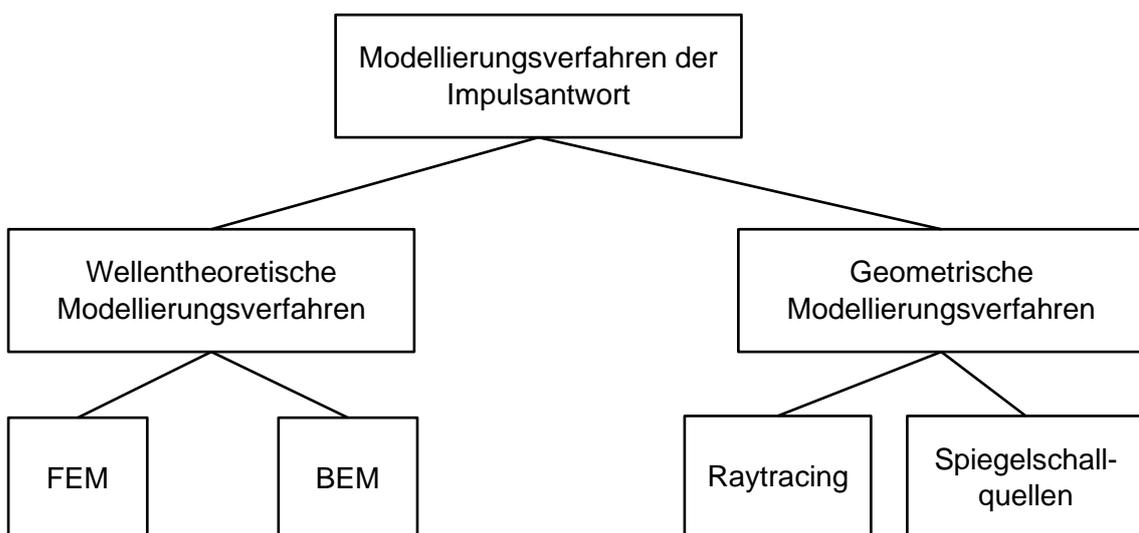


Abbildung 3-2: Berechnungsverfahren der Impulsantwort [nach (Vorländer, 2020, p. 146)]

Die Finite Elemente Methode hat als Grundlage die Diskretisierung des modellierten Volumens in einzelne, dreidimensionale Elemente. Die Ermittlung der Schallgrößen basiert auf der Lösung partieller Differentialgleichungen anhand eines numerischen Verfahrens in jedem Element. Die globalen Randbedingungen werden als Anfangswerte betrachtet. An den Elementübergängen werden die schon bekannten Werte als Randbedingungen für die Berechnung des Schalldruckes in einem benachbarten Element verwendet. Somit erreicht man einen stetigen Verlauf zwischen den Elementen (Vorländer, 2020). Die Berechnungszeit ist abhängig von der Größe des Modells, also der Anzahl der diskreten Elemente, und des analysierten Frequenzbandes (Svensson, 2002).

Bei der Boundary Element Methode (BEM) werden die Flächen der Modellabgrenzungen diskretisiert. Jedem Element wird eine Schallquelle zugewiesen und der Schalldruck und die Schnelle für jede Elementschallquelle berechnet (Svensson, 2002).

Die geometrischen Modellierungsverfahren betrachten das Schallfeld anhand von geradlinigen Schallstrahlen (eng. *sound ray*), die akustische Energie transportieren. Eine Schallquelle sendet Schallstrahlen in alle Raumrichtungen aus. Die initiale Energie der einzelnen Schallstrahlen hängt von der Richtcharakteristik der Schallquelle ab. Die Strahlen werden an den Raumwänden so lange reflektiert, bis ihr Energieniveau unter einen benutzerdefinierten Wert sinkt. Durch die Absorptionsgrade der Wände wird den Strahlen Energie entzogen. Es ist auch möglich, die Absorption des Ausbreitungsmediums zu berücksichtigen, indem eine Abnahme der Schallenergie mit der Vergrößerung des Abstandes zur Schallquelle simuliert wird. Die Impulsantwort an einem Ort im Raum ergibt sich aus der Summe der Energien der einzelnen Schallstrahlen, die sich mit dem Beobachtungsort kreuzen, den Laufzeiten und dem Einfallwinkel der einzelnen Strahlen (Vorländer, et al., 2014).

Es gibt zwei Hauptmethoden zur Simulation der geometrischen Raumakustik: eine stochastische Methode mit Raytracing (engl. Ray-Tracing) und eine deterministische Methode mit Spiegelschallquellen (Vorländer, 2020).

Für die Methode der Spiegelschallquellen (SQ) werden die Wandreflexionen von Spiegelschallquellen auf der anderen Seite der Wand erzeugt (siehe Abbildung 3-3, (rechts)). Mit jeder konsekutiven Reflexion eines Strahls wird auch die Ordnung der Reflexion erhöht und eine neue Spiegelquelle generiert. Die Amplituden der reflektierten Anteile werden so angepasst, dass eine Reduzierung der Schallenergie durch die Wandimpedanz nach jedem Reflexionsvorgang berücksichtigt wird (Svensson, 2002). Mit dieser Methode können nur spiegelnde Reflexionen modelliert werden. Allgemein liefert sie zufriedenstellende Ergebnisse bei vollreflektierenden Oberflächen und großen Räumen. Die Nachteile der Schallspiegelquellen sind aber die fehlenden Berücksichtigungen der Schallbeugungen an Kanten und der diffusen Reflexionskomponenten (Vorländer, et al., 2014). Weiterhin kann diese Methode für Reflexionen hoher Ordnungen sehr rechenintensiv sein, weil sehr viele Spiegelschallquellen generiert werden müssen (Svensson, 2002).

Die Schwächen der SQ-Methode können durch das Einsetzen der Raytracing-Methode umgangen werden. Sie setzt ebenfalls Spiegelschallquellen für die Modellierung der spiegelnden Reflexionen ein, erzeugt aber auch diffuse Reflexionsanteile, wie in Abbildung 3-3 (links) gezeigt. Die hohe Rechenzeit kann durch eine Limitierung der generierten Strahlen korrigiert werden, was aber dazu führen kann, dass die Qualität der Simulation abnimmt, (Svensson, 2002).

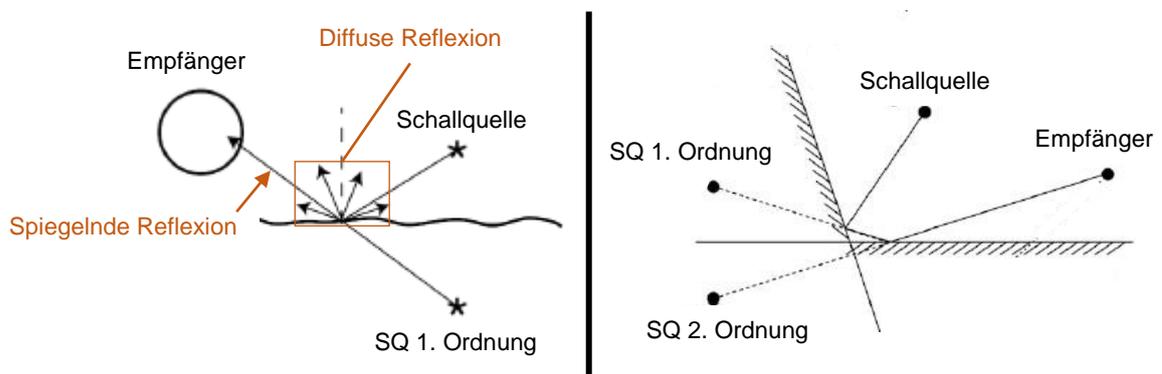


Abbildung 3-3: Simulation der geometrischen Raumakustik mit Raytracing (**links**) [nach (Svensson, 2002)] und Spiegelschallquellen (**rechts**) [nach (Vorländer, 2020)]

Keine der präsentierten geometrischen Modellierungsverfahren können die Schallbeugung simulieren. Es ist allerdings möglich, diese Methoden, um einen Algorithmus zu erweitern, welcher benachbarte Strahlen zusammen bündelt. Trifft ein Bündel auf eine Kante auf, so kann es aufgeteilt werden. Nicht blockierte Strahlen verlaufen weiter im Raum. Für die verhinderten Strahlen werden Schallquellen auf die Kante generiert, welche neue Strahlbündel aussenden.

Die Anwendung hybrider Methoden profitiert von den Vorteilen der wellentheoretischen und der geometrischen Verfahren. So können Systeme entwickelt werden, welche im niederfrequenten Bereich die wellentheoretischen Verfahren einsetzt und bei hohen Frequenzen auf die geometrischen Verfahren wechselt (Vorländer, 2020).

Nachdem ein trockenes akustisches Signal mit der RIR gefaltet wird, enthält es die akustischen Charakteristika des Raumes, wirkt aber aus psychoakustischer Sicht nicht wie ein Schall aus der gleichen realen Umgebung. Da die VR-Umgebung einen dreidimensionalen Eindruck ermittelt, muss auch die Audioausgabe entsprechend erfolgen.

Dies wird erreicht, indem das Signal mit einer weiteren Faltung räumlich platziert wird. Die Wiedergabegeräte spielen dabei eine wichtige Rolle, da sie grundsätzlich das Verfahren der Generierung eines räumlichen Signals festlegen.

Werden für die Wiedergabe Kopfhörer eingesetzt, so wird die räumliche Platzierung mit der binauralen Technologie erreicht. Hierbei wird eine sogenannte Head Related Transfer Function (HRTF) mit dem Signal gefaltet. Die HRTF (Head Related Impulse Response (HRIR) im Zeitbereich) kann als eine Impulsantwort des menschlichen Oberkörpers betrachtet werden. Sie enthält die Informationen über die Übertragung des Schalls zwischen einer Schallquelle und den beiden Ohren aus mehreren Raumrichtungen.

Die Ermittlung der HRTF wird in klassischer Weise durch Messungen erreicht. Menschen oder Kunstköpfe werden in reflexionsarmen Räumen platziert und der erzeugte Schall wird mit zwei Mikrofonen, welche sich in den Ohrkanälen befinden, aus mehreren Richtungen und Elevationen gemessen. Die HRTFs lassen sich auch anhand von FEM-Berechnungen ermitteln. Diese Methode benötigt keine umfangreiche Messtechnik und eine spezielle Messumgebung, verfügt aber nicht über die gleiche Genauigkeit wie eine Messung, weil die Oberkörper-, Kopf- und Ohrgeometrie vereinfacht dargestellt werden (Zhong & Xie, 2014).

Eine räumliche Klanguisgabe über Lautsprecher kann mit dem Ambisonics Format erreicht werden. Dabei wird das Schallfeld in Kugelflächenfunktionen zerlegt. Die Besonderheit dieser Methode ist die Möglichkeit der Platzierung von virtuellen Schallquellen um einen Empfänger. Die Anzahl der zur Verfügung stehenden virtuellen Quellen hängt mit der Auswahl der Ordnung der Kugelflächenfunktion zusammen. Somit wird bei Erhöhung der Ordnung eine bessere räumliche Auflösung erzeugt. Der Vorteil von Ambisonics ist die Möglichkeit der Wiedergabe sowohl über Kopfhörer als auch über jede beliebige Lautsprecherkonfiguration. Es muss dabei beachtet werden, dass das Signal für die eingesetzte Lautsprecherkonfiguration dekodiert wird. Für die Wiedergabe über Kopfhörer wird das Ambisonics-Signal jeder virtuellen Schallquelle mit einer HRTF gefaltet (Frank, et al., 2015).

Der bis jetzt präsentierte Prozess zur Auralisation eines Schallfeldes kann nur dann für die VR eingesetzt werden, wenn die Berechnungen mit ausreichender Geschwindigkeit

aktualisiert werden. Die Interaktion des VR-Anwenders mit der virtuellen Szene führt dazu, dass eine kontinuierliche Änderung der Umgebung stattfinden kann.

Abbildung 3-1 beinhaltet Signalpfade aus der virtuellen Realität zu den Einzelbausteinen des Auralisationsprozesses. Sie transportieren aktualisierte Informationen, wie zum Beispiel die Positionen der Schallquelle und des Empfängers, die Geometrie der Szene oder die Eigenschaften der Oberflächenmaterialien. Dementsprechend ist es erforderlich, die Akustikberechnung schnell zu aktualisieren. Der zeitliche Aspekt ist von hoher Bedeutung, da bei zu großen Verzögerungen die Immersion absinkt. Vorländer M. definiert eine Obergrenze für die Latenz von 50 ms, um einen Klang als synchronisiert und durchgehend wahrnehmen zu können (Vorländer, 2021).

Im Laufe der Jahre entwickelten Forscherteams diverse Simulationssoftware. Viele davon beruhen auf die geometrischen Berechnungsverfahren, weil sich das Problem der hohen Rechenzeiten einer analytischen Berechnung mit den heutigen Technologien nur bedingt umgehen lässt. Nichtsdestotrotz kann die Raumakustik mit optimierten Raytracing Algorithmen mit einer hohen Qualität in einer VR-Umgebung simuliert werden.

RAVEN ist eine Software für die Echtzeitsimulation der Raumakustik und wurde an der RWTH Aachen entwickelt (Lentz, et al., 2007). Die Besonderheit dieser Entwicklung ist die binaurale Wiedergabe über Lautsprecher, welche mittels Cross-Talk Cancellation ermöglicht wird. Die Software wird in der aixCAVE, eine CAVE (CAVE Automatic Virtual Environment) Umgebung, eingesetzt (vgl. Abbildung 3-4) (Schröder, et al., 2010).

Ein Forscherteam der Ruhr-Universität Bochum entwickelte die Software IKA-SIM (Silze, et al., 2004), mit der sich eine virtuelle akustische Umgebung simulieren lässt. Die Software wird in einer Internet-basierten Umgebung eingesetzt. Ein zentraler Rechner übernimmt die akustische Simulation, die von anderen Rechnern aus dem Netz für die binaurale Audiowiedergabe in einer virtuellen Realität verwendet werden kann (Borß, et al., 2008).

DIVA ist die Arbeit eines Forscherteams von der Helsinki University of Technology und bietet auch die Möglichkeit an, die Raumakustik zu simulieren. Anschließend kann die Simulation mit einer virtuellen Umgebung verknüpft werden, in der ein Benutzer sich frei bewegen kann (Savioja, et al., 1999).

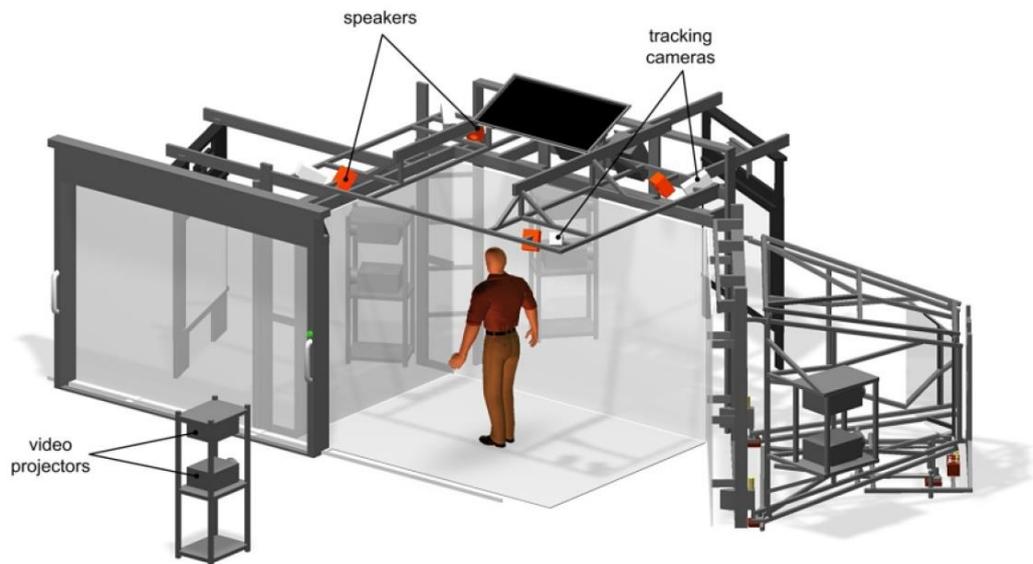


Abbildung 3-4: aixCAVE System der RWTH Aachen [aus (Schröder, et al., 2010)]

EVERTims bietet ein Auralisation-Plug-in für die Grafiksoftware Blender. Es erzeugt eine Echtzeitsimulation der Raumakustik und kann als Add-on in die Blender Spiel-Engine integriert werden, um ein virtuelles akustisches System zu erzeugen (Poirier-Quinot, et al., 2017).

Ein weiteres akustisches virtuelles System beruht auf 360° Aufnahmen, mit welchen die Geometrie eines Raumes erstellt werden kann und die akustischen Eigenschaften der Oberflächen ermittelt werden. Anschließend werden die virtuelle Szene und das akustische Verhalten mit der Spiel-Engine Unity modelliert und simuliert (Kim, et al., 2019).

REVES (Moeck, et al., 2007), NASAs SLAB (National Aeronautics and Space Administration, 2022) oder Spat (IRCAM, 2022) sind weitere Softwarepakete, die sich mit der Virtualisierung der Akustik beschäftigen.

3.1. Steam Audio Plug-in

Das von Valve Corporation® entwickelte Plug-in Steam Audio erweitert die Audio-Engine vieler Softwarepakete aus dem Bereich der Computerspieleentwicklung, um eine physikalisch-basierte Auralisation in Echtzeit zu ermöglichen. Die Haupteigenschaften des Plugins sind (Valve Corporation, 2022):

- Echtzeitsimulation einer physikalisch-basierten Schallausbreitung mit Berücksichtigung der Empfängerposition.
- Schallreflektion und -transmission an geometrischen Objekten aus der Szene.
- Möglichkeit der Implementierung von Ambisonics höherer Ordnung oder binauraler Wiedergabe mittels HRTFs.

Die Modellierung der Schallquelle lässt sich anhand mehrerer Einstellungen (siehe Abbildung 3-5) je nach Bedarf anpassen. So kann eine Faltung mit einer HRTF dafür sorgen, dass das Audiosignal schon bei der Wiedergabe binaural ist. Die abstandsabhängige Dämpfung des Signales kann anhand eines physikalischen Modells simuliert werden. Außerdem kann auch eine frequenz- und abstandsabhängige Luftabsorption, welche höhere Frequenzen stärker als niedrigere Frequenzen dämpft, implementiert werden. Befindet sich eine Schallquelle hinter einem Objekt, so kann man einstellen, ob das Objekt als Hindernis betrachtet wird und wenn das der Fall ist, wie das Plug-in die Ausbreitung der Strahlen modellieren soll. Man kann auch die Transmission des Schallquellensignals durch Objekte einstellen.

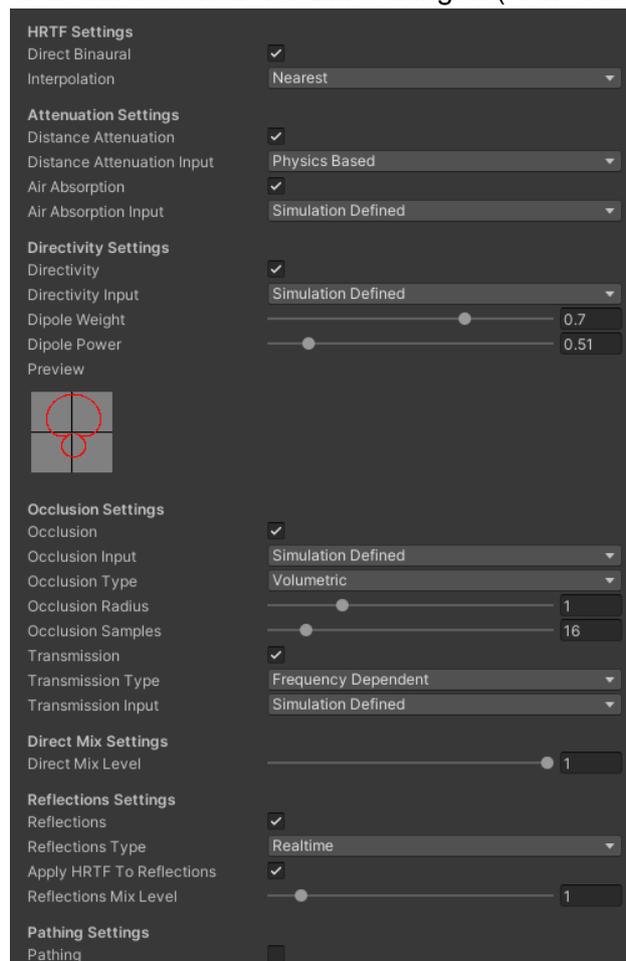


Abbildung 3-5: Einstellungsmöglichkeiten für die Schallquelle in dem Steam Audio Plug-in

Der Transmissionsgrad kann konstant für alle Frequenzen oder frequenzabhängig gewählt werden (Valve Corporation , 2021).

Die Ermittlung der Impulsantwort der Szene erfolgt mit dem geometrischen Verfahren Raytracing. Je nach Komplexität der Szene kann sie in Echtzeit oder im Vorweg berechnet werden. Eine Berechnung der Impulsantwort im Voraus wird dann empfohlen, wenn entweder der Empfänger, die Quelle oder die komplette Szene statisch sind. Dies ermöglicht eine bessere Verwaltung der Computerressourcen, sodass schnellere Rechenzeiten erreicht werden können. Der Einfluss der Objekte (engl. Objects) auf die Impulsantwort kann durch die Verknüpfung eines akustischen Materials mit den Objekten und Wände aus der Szene berechnet werden. Die Materialdatenbank des Plug-ins verfügt über vordefinierte Materialien. Die Eigenschaften der Materialien lassen sich anpassen, indem die Flächenabsorptionsgrade α für Frequenzbänder mit tiefen (0 Hz bis 800 Hz), mittleren (800 Hz bis 8 kHz) und hohen (über 8 kHz) Frequenzen geändert werden können. Der Wertebereich für α liegt zwischen null und eins, wobei $\alpha = 0$ eine komplett reflektierende Fläche und $\alpha = 1$ eine vollständig absorbierende Fläche darstellt. Des Weiteren lässt sich auch die Rauheit des Materials ändern, wodurch die Streuung der reflektierten Welle beeinflusst wird (Valve Corporation (a), 2021).

Die Konfigurationsmöglichkeiten am Empfänger sind aufgrund der umfangreichen Einstellungen der Schallquelle und der Impulsantwortsimulation auf ein Minimum reduziert. Es lässt sich jedoch auswählen, ob die reflektierten Strahlen am Empfänger in Echtzeit ermittelt oder aus einer im Voraus Berechneten Umgebung interpoliert werden (Valve Corporation (c), 2021).

Abbildung 3-6 zeigt die globalen Einstellungen der Steam Audio Engine. Mit den Einstellungen der Strahlenanzahl (Real Time Rays), Anzahl der Reflexionen (Real Time Bounces) und der Länge der Impulsantwort (Real Time Duration) lässt sich die Güte der Simulation aber auch der Ressourcenbedarf und die Rechenzeit einer Echtzeitsimulation des Schallfeldes beeinflussen. (Valve Corporation (b), 2021)

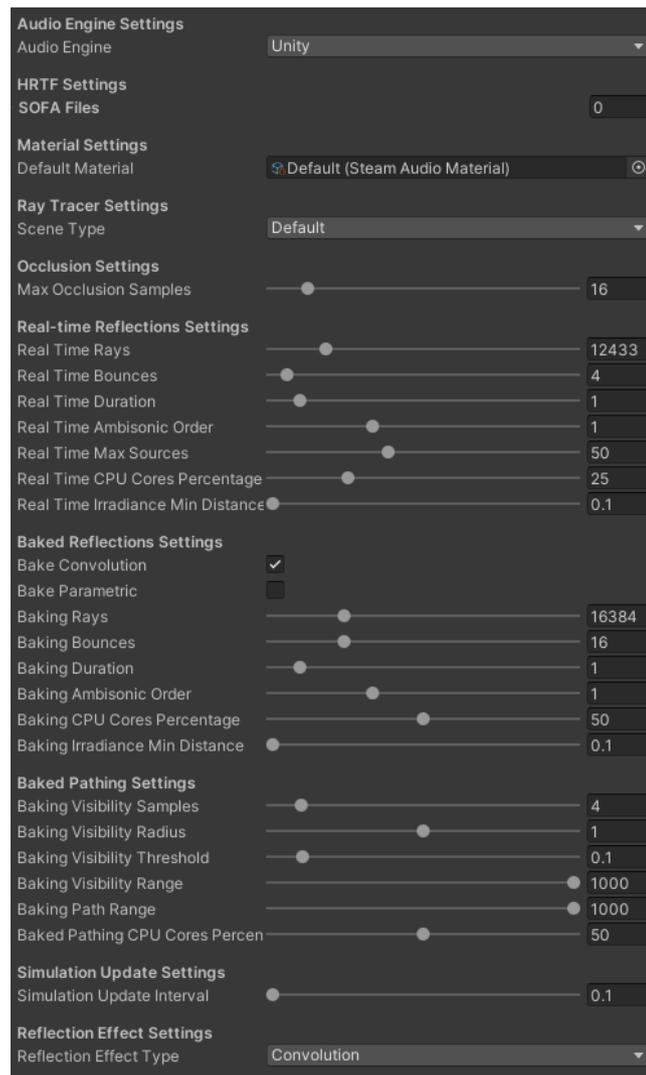


Abbildung 3-6: Globale Einstellungen der Steam Audio Engine

3.2. Resonance Audio Plug-in

Googles Resonance Audio Plug-in bietet eine Erweiterung der Audiogenerierungsalgorithmen unterschiedlicher Game Engines an. Durch die optimierten Algorithmen reduziert sich die benötigte Rechenleistung, sodass auch mobile Geräte, wie zum Beispiel Handys, von den Vorteilen des Plug-Ins profitieren können.

Das Plug-in berechnet die Simulation des Schallfeld mithilfe von geometrischen Verfahren. Das Übertragungsverhalten setzt sich aus drei unterschiedlichen Komponenten zusammen. Ähnlich wie in der Beschreibung der Impulsantwort am Anfang des Kapitels 3, enthält dieses einen direkten Schall, frühe Reflexionen und späte Reflexionen, welche ein diffuses Schallfeld erzeugen (Resonance Audio (a), 2022).

Resonance Audio ermöglicht die Modellierung von Schallquellen mit beliebigen Richtcharakteristiken. Ähnlich wie bei dem Audio Plug-in von Steam wird die Blockierung der Schallquellen durch Objekte aus der Szene von Resonance Audio modelliert. Um eine möglichst reale Schallverteilung bei diesem Vorgang zu erreichen, wird zwischen hochfrequentem und tieffrequentem Schall unterschieden (Resonance Audio (a), 2022).

Resonance Audio führt die Echtzeitberechnung der Impulsantwort in einem benutzerdefinierten rechteckigen Volumen durch. Der Bereich aus der Szene, in dem die akustischen Ereignisse simuliert werden sollen, muss sich in diesem Berechnungsvolumen befinden. Die Abgrenzungen des Volumens werden für die Ermittlung der Reflexionen als Wände interpretiert. Durch eine Wandmaterialzuweisung können die akustischen Eigenschaften der Wände beeinflusst werden.

Für komplexere Geometrien bietet das Plug-in die Möglichkeit an, den vereinfachten Berechnungsbereich mit zusätzlichen Materialangaben zu verbessern. Für die einzelnen Raumkomponenten sowie unterschiedlichen Wandbereiche kann man verschiedene Materialien auswählen, wie in Abbildung 3-7 gezeigt. Die Berechnung der Reflexionen wird deutlich exakter, führt aber dazu, dass die Impulsantwort vor dem Ausführen der Szene berechnet werden muss.

Der Nachteil dieser Verfeinerung ist also eine Reduzierung der Flexibilität, da eine Änderung der Materialien in der aktiven Szene keinen Einfluss auf die Akustik der Umgebung hat, solange man keine neue RIR berechnet (Resonance Audio (b), 2022).

Die menschenorientierte Wiedergabe wird erreicht, indem das Plug-In die Audiosignale in das Ambisonics-Format kodiert und es mit einer HRTF faltet.

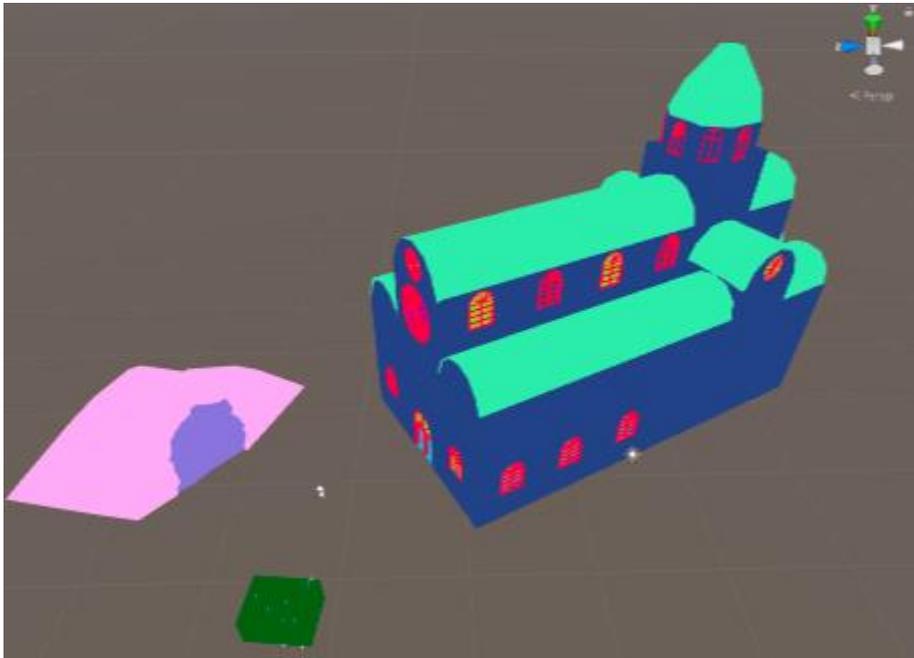


Abbildung 3-7: Materialzuweisung zur Verfeinerung der Impulsantwortberechnung; Jede Farbe repräsentiert ein unterschiedliches Material [nach (Resonance Audio (b), 2022)]

4. Definition und Auswahl der Messumgebungen

Die Auswahl der Umgebung für die Messungen wird anhand folgender Merkmale durchgeführt:

1) Die Geometrie der Räumlichkeit darf nicht zu komplex sein. Es muss möglich sein, eine initiale Berechnung der raumakustischen Eigenschaften mit Hilfe der mathematischen Gleichungen aus Kapitel 2.2 durchzuführen. Ein weiterer Aspekt, der mit der Geometrie der Messumgebung verknüpft ist, repräsentiert die Diskretisierung des Modells, welches für die FEM-Analyse erstellt wird. Bei zu komplexen Geometrien wird die Diskretisierung sehr umfangreich und kann zu unerwünschten Effekten führen. Diese können dann einen negativen Einfluss auf die Ergebnisse der Simulation haben.

2) Die Messumgebung soll ausreichend Platz für die eingesetzte Bewegungseinrichtung zur Verfügung stellen. Eine genauere Beschreibung der Bewegungseinrichtung erfolgt in Kapitel 5.1.

3) Wie in Kapitel 2.2 vorgestellt, kann eine ungünstige Raummodenverteilung zu dem Effekt führen, dass die Raummoden miteinander interferieren und somit ein diffuses Schallfeld entsteht. Die akustische Messung soll bei einer ausgeprägten Mode durchgeführt werden. Die Messungen müssen also in einer Umgebung stattfinden, in der ein Schallfeld mit einer ausgeprägten Raummode erzeugt werden kann. Es sind axiale Raummoden in Längsrichtung angestrebt, weil die Messung in Längsrichtung durchgeführt wird. Die Wellenlänge des Schalls spielt dabei auch eine wichtige Rolle. Es ist erwünscht, mindestens ein Druckmaximum, -minimum oder einen Nulldurchgang der Schallwelle mit der Messung zu erfassen, damit eine klare Schalldruckverteilung erkannt wird. Dadurch ist ein Vergleich der Messungen in den unterschiedlichen Umgebungen vereinfacht. Weil der Verfahrweg der Bewegungseinrichtung nur 0,85 m beträgt, darf die von der Eigenfrequenz abhängige Wellenlänge nicht weniger als 2 m betragen.

Es muss also ein Kompromiss zwischen der Ausbreitungsrichtung der Moden, der Eigenfrequenz und der Modenverteilung in der Messumgebung gefunden werden.

4.1. Mock-up eines Crew Rest Compartments

Die reale Umgebung ist der definierende Faktor bei der Auswahl des Messortes, somit werden die präsentierten Kriterien eingesetzt, um eine geeignete reale Räumlichkeit zu finden. Die Modelle für die FEM-Analyse und für die virtuelle Messung sind die Abbildung der realen Messumgebung. Mit Hilfe der definierten Auswahlkriterien werden drei potenzielle Räumlichkeiten analysiert (siehe Anhang A).

Tabelle 4.1 und Abbildung 4-1 präsentieren die Eigenschaften dieser Räumlichkeiten, die für die Auswahl der geeigneten Messumgebung relevant sind.

Eine erste überschlägige Berechnung der Raummoden kann mit dem Online Tool „*am-roc-THE Room Mode Calculator*“ (Melcher, 2022) durchgeführt werden. Zur Kontrolle, erfolgt die Berechnung der Eigenfrequenz $f_{E,\ddot{u}}$ bei den präsentierten Raummoden mit Hilfe der Gleichung (2-25). Für die Bestimmung der Anzahl der Moden N_{500} nach Gleichung (2-26) ist die Grenzfrequenz auf $f_{Gr} = 500$ Hz festgelegt.

Tabelle 4.1: Eigenschaften der untersuchten Messumgebungen

Messumgebung	$l_x \times l_y \times l_z$ [cm]	Eigenmoden und Eigenfrequenzen ⁴	N_{500}	Bemerkungen
Schallabsorptions- einrichtung	205 x 119,8 x 144,6	(1-0-0) → 83,66 Hz (2-0-0) → 167,32 Hz (3-0-0) → 250,98 Hz (4-0-0) → 334,63 Hz	76,4	Seitenwände aus einem schallabsorbierenden Material → für die Messung nicht geeignet
Klimakammer	314,5 x 313,5 x 254	(1-0-0) → 54,53 Hz (2-0-0) → 109,06 Hz (3-0-0) → 163,59 Hz (4-0-0) → 218,12 Hz	409,9	Aggregate an der Decke angebracht → eventuelle Diskrepanzen zwischen Modell und Realität
Mock-up eines Crew Rest Compartments (CRC)	200 x 70,3 x 65,25	(1-0-0) → 85,75 Hz (2-0-0) → 171,5 Hz (3-0-0) → 257,25 Hz (4-0-0) → 343 Hz	27,9	Keine rechteckige Geometrie

⁴ Es werden nur die ersten vier axialen Eigenmoden in l_x -Richtung betrachtet

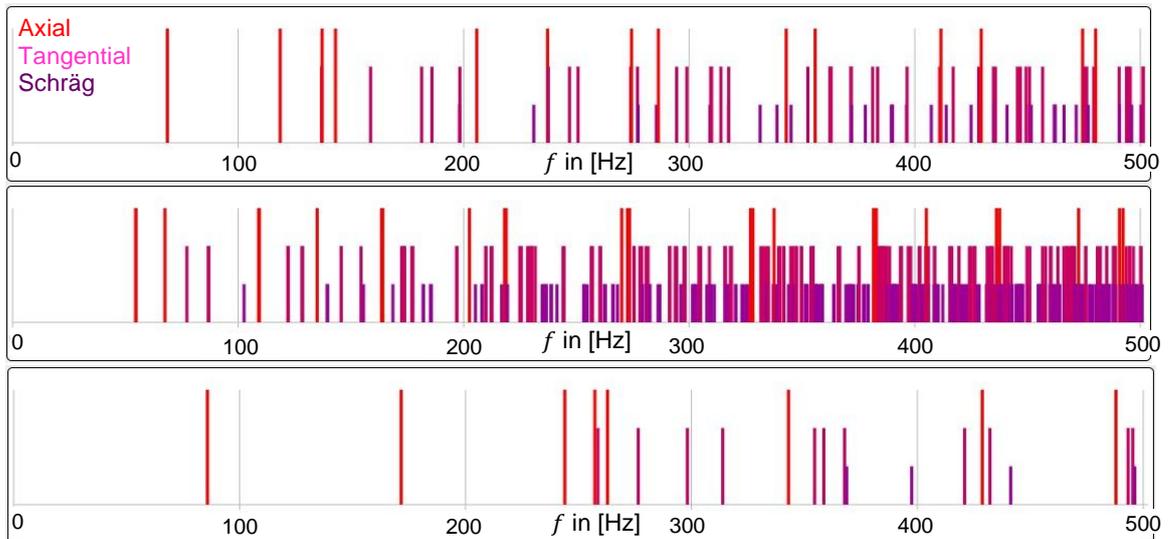


Abbildung 4-1: Verteilung der Raummoden bis $f=500$ Hz in der Schallabsorptionseinrichtung (**Oben**), in der Klimakammer (**Mitte**) und im Mock-up des CRCs (**Unten**) [nach (Melcher, 2022)]

Die Informationen aus Tabelle 4.1 und Abbildung 4-1 zeigen, dass die Klimakammer für die Messung ungeeignet ist. Die hohe Raummodendichte führt zu geringen Abständen zwischen den Eigenfrequenzen, sodass die Erfassung einer einzigen Mode erschwert ist. Es wäre möglich, den Raum ohne Aggregate zu betrachten, allerdings würde diese Annahme zu möglichen Abweichungen zwischen den Messungen in der realen Umgebung und den FEM Ergebnissen führen. Dadurch wäre ein Vergleich zwischen FEM und Realität möglicherweise nicht möglich.

Die Schallabsorptionseinrichtung verfügt über eine geringere Raummodendichte, vor allem bis zu einer Frequenz von $f = 200$ Hz. Eine Messung ist jedoch nicht möglich, weil die Eigenfrequenzen für axiale Raummoden in x -Richtung sich mit den Eigenfrequenzen von anderen Raummoden teilweise überlappen. Es werden also zwei Eigenmoden gleichzeitig angeregt, die miteinander interferieren. Weiterhin müssten die Seitenwände durch andere Wände aus einem schallreflektierenden Material ersetzt werden, was zu einem zusätzlichen Zeit- und Kostenaufwand führen würde.

Das Mock-up eines Crew Rest Compartment ist für die Schallfeldmessungen am besten geeignet. Bis zu der Frequenz von $f = 500$ Hz ist die Raummodendichte geringer als bei den anderen analysierten Räumlichkeiten. Die Raummode (2-0-0) bei $f_{E,\ddot{u}} = 171,5$ Hz hat einen großen Abstand zu den nächsten Moden, somit tritt die Gefahr der gleichzeitigen Anregung mehrerer Moden nicht auf. Mit der Eigenfrequenz von $f_{E,\ddot{u}} = 171,5$ Hz

ergibt sich nach Formel (2-13) und mit der Schallgeschwindigkeit $c = 343 \text{ m/s}$ eine Wellenlänge von 2 m. Somit ist die Erfassung eines Druckminimums oder Druckmaximums mit der eingesetzten Bewegungseinrichtung möglich.

Das Crew Rest Compartment (CRC) ist einen von der Passagierkabine abgetrennten Raum in Passagierflugzeugen, in dem Piloten und Kabinenpersonal die Möglichkeit haben, sich auszuruhen. Diesen Ruhebereich findet man häufig in Langstreckenflugzeugen. Der Ruhebereich besteht aus mehreren einzelnen Abteilen, die jeweils ein Bett enthalten. Abbildung 4-2 zeigt ein Crew Rest Compartment aus dem Labor für Kabine und Kabinensysteme der HAW Hamburg, sowie das eingesetzte Mock-up eines Abteils.

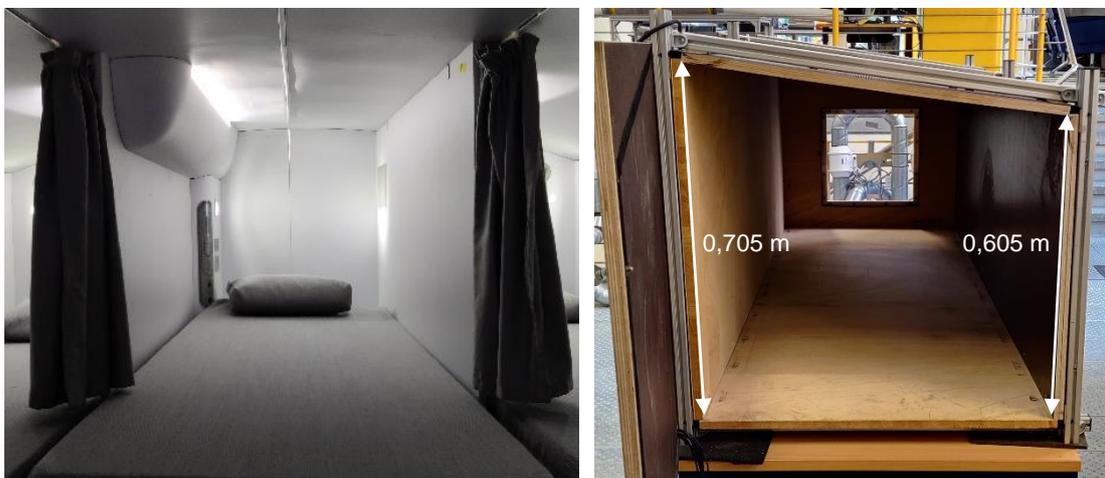


Abbildung 4-2: Crew Rest Compartment und das eingesetzte Mock-up [eigene Darstellung]

Das Mock-up besteht aus einem Aluminiumrahmen und Seitenwänden aus Holzplatten mit einer Materialstärke von 19 mm. Die Endabdeckung verfügt über ein Fenster aus Plexiglas, welches mit zwei Schrauben befestigt wird und bei Bedarf entfernt werden kann. Eine Bohrung in der Tür ermöglicht die Durchführung von Kabeln für die Messtechnik. Die Seitenwände haben unterschiedliche Höhen, wodurch die Geometrie des Mock-ups von der Annahme eines rechteckigen Raumes abweicht. Die Höhen der Seitenwände können der Abbildung 4-2 entnommen werden. Die Eigenmoden aus Tabelle 4.1 sind mit einer Raumhöhe gleich dem Mittelwert der Höhen der beiden Seitenwände berechnet. Mit einer Schalldruckmessung, sowie einer FEM-Simulation wird im späteren Verlauf der Arbeit gezeigt, dass die Abweichung der realen Geometrie zu der angenommenen Geometrie zu keinen großen Unterschieden führt.

Die eingesetzte Messtechnik erfüllt die Aufgabe der Tongenerierung und der Tonaufnahme. Abbildung 4-3 zeigt die eingesetzte Messtechnik und die Verknüpfungen zwischen den einzelnen Komponenten. Die Schallfeldmessungen werden mit Hilfe von vier Mikrofonen mit Kugelcharakteristik durchgeführt. Das Signal der Messmikrofone wird von einem Mikrofonverstärker verstärkt und an die Messkarten myDAQ weitergeleitet.

Das programmierbare Messgerät myDAQ gehört zu der Reihe der DAQ-Geräte (engl. **Data Acquisition**) der Firma National Instruments®. Zusammen mit LabVIEW, eine grafische Programmierumgebung des gleichen Unternehmens, können Messsysteme, Signalverarbeitungssysteme und Signalgeneratoren erstellt werden. Die Vorteile des myDAQs sind die intuitive Handhabung, die standardisierten Anschlüsse und die Signalauffassungsfähigkeiten. Der im späteren Verlauf der Arbeit eingesetzte Messton wird ebenfalls von dem myDAQ generiert, von einem Lautsprecherverstärker verstärkt und mittels eines Lautsprechers wiedergegeben. Die wichtigsten technischen Daten der eingesetzten Messtechnikkomponenten werden in Tabelle 4.2 eingetragen.

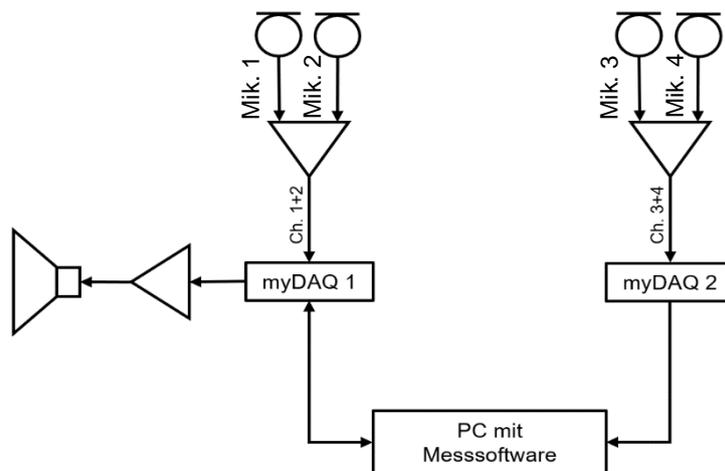


Abbildung 4-3: Komponenten und Aufbau der Messtechnik

Tabelle 4.2: Technische Spezifikationen der Messtechnikkomponenten

Komponente	Anzahl	Hersteller, Modell	Technische Daten
Messmikrofon	4	Img Stageline, ECM-501L/SK	Frequenzbereich: 20 Hz – 20 kHz Empfindlichkeit: 5,6 mV / Pa / 1 kHz Maximaler Schalldruck: 110 dB Stromversorgung: 1,5 – 9 V
Mikrofonverstärker	2	Img Stageline, MPA-202	Mikrofoneingänge: 2 Verstärkung: 15-75 dB Tiefpassfilter: 15-240 Hz
Messkarte	2	National Instruments, myDAQ	Analogeingang Kanäle: 1 Stereoeingang ADU-Auflösung: 16 bit Abtastrate: 200 kS/s Spannungseingang Stereo: ± 2 V Bandpass: 1,5 Hz – 400 kHz Analogausgang Kanäle: 1 Stereoausgang DAU ⁵ -Auflösung: 16 bit Abtastrate: 200 kS/s Spannungsausgang Stereo: ± 2 V
Lautsprecherverstärker	1	Powersoft, LQ2840	4 Ein- und Ausgangskanäle Verstärkung: 11-Stufig ∞ bis 32 dB
Lautsprecher	1	EighteenSound, 6ND430	Membrandurchmesser: 152 mm Leistung: 200 W Impedanz: 8 Ω

4.2. Finite-Elemente Modell eines Crew Rest Compartments

Die numerische Berechnung wird mit der FEM-Simulationssoftware COMSOL Multiphysics[®] durchgeführt. Die Simulationssoftware enthält ein umfangreiches Akustikmodul mit unterschiedlichen Rechenmethoden, wie zum Beispiel FEM, BEM oder Raytracing. COMSOL bietet die Möglichkeit an, mehrere unterschiedliche physikalische Module miteinander zu koppeln, sodass realitätsnahe Simulationen realisierbar sind. Man kann zum Beispiel mit nur einer einzigen Simulation die Vibrationen einer Struktur berechnen und die dadurch induzierten akustischen Störungen in einem umgebenden Schallfeld bestimmen. In dieser Arbeit wird ausschließlich das Akustikmodul betrachtet. Die Simulation kann aber in der Zukunft durch die Kopplung mit anderen Modulen erweitert werden.

⁵ Digital-Analog-Umwandler

COMSOL verfügt über eine Konstruktionsumgebung, in der das Volumen der in dem Mock-up eingeschlossenen Luft modelliert wird. Weil sich der Lautsprecher und die beiden Mikrofonverstärker für die Kartierungsmessungen in der Messumgebung befinden, wird das eingenommene Volumen dieser Komponenten von dem gesamten Luftvolumen abgezogen, sodass mögliche Effekte auf das Schallfeld berücksichtigt werden können (vgl. Abbildung 4-4).

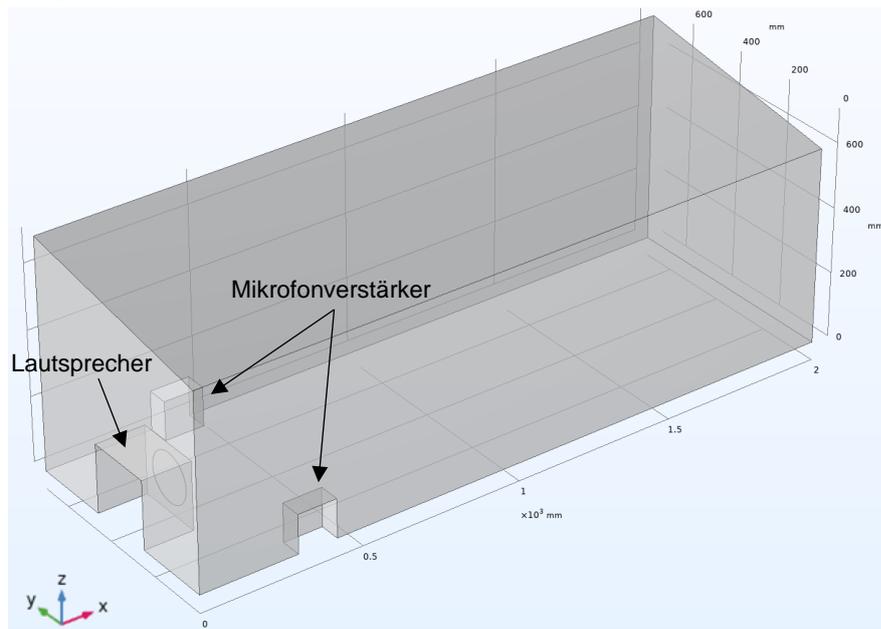


Abbildung 4-4: Modell des CRC Mock-ups in COMSOL Multiphysics

Comsol Multiphysics verfügt über eine umfangreiche Datenbank von Materialien mit vordefinierten Materialeigenschaften. Dem Volumen wird das Material Luft zugewiesen. Je nach Umgebungsbedingungen können die Temperatur T , der Absolutdruck p_{ges} und die Schallgeschwindigkeit c angepasst werden. Durch die komplexe Notation der Schallgeschwindigkeit kann eine Schallabsorption des Ausbreitungsmediums berücksichtigt werden (COMSOL Multiphysics, 2020, p. 227). Für die FEM-Analyse werden folgende Umgebungsbedingungen für das linear-elastische Medium eingestellt:

Tabelle 4.3: Eingestellte Umgebungsbedingungen des Ausbreitungsmediums

Temperatur	28 °C
Absolutdruck	101325 Pa
Schallgeschwindigkeit	$348(1 + i \cdot 0,01)$ m/s

Die Berechnung der Schalldruckverteilung im Comsol Multiphysics erfolgt mit der Lösung der modifizierten inhomogenen Helmholtz Gleichung im Frequenzbereich (COMSOL Multiphysics, 2020, p. 228):

$$\nabla \left(-\frac{1}{\rho} (\nabla p - q_d) \right) - \frac{k^2 p}{\rho} = Q_m \quad (4-1)$$

Die Größen q_d und Q_m repräsentieren Dipole und Monopole. Die Randbedingungen für die Lösung der inhomogenen Differenzialgleichung können nach Bedarf angepasst werden. Standardmäßig sind alle Volumenflächen schallhart. Wie in Kapitel 2.2 erwähnt, ist diese Annahme nur theoretisch möglich. Aus diesem Grund werden die Außenflächen des CRCs, sowie das Lautsprechergehäuse mit einem Absorptionsgrad α versehen. Laut (Smardzewski, et al., 2015) beträgt eine MDF-Platte (MDF T11) mit der Materialstärke von 18,4 mm den Absorptionsgrad $\alpha \approx 0,05$ bei 170 Hz. Aufgrund fehlender Informationen über die Absorption der Holzplatten des CRC Mock-ups wird der Absorptionsgrad aus der Literatur eingesetzt. Für die metallischen Oberflächen des Mikrofonverstärkergehäuses wird keine Impedanzbedingung vorgesehen. Die Schallabsorption des kleinen Volumens, verglichen zu dem Gesamtvolumen des Ausbreitungsmediums, würde nur einen kleinen Einfluss auf das Schallfeld haben, somit kann die Annahme eines schallharten Materials getroffen werden. Die Anregung des Schallfeldes wird über eine frequenzabhängige Beschleunigung $a(f)$ der modellierten Lautsprechermembran erreicht. Abbildung 4-5 stellt die Bereiche mit den unterschiedlichen Randbedingungen dar.

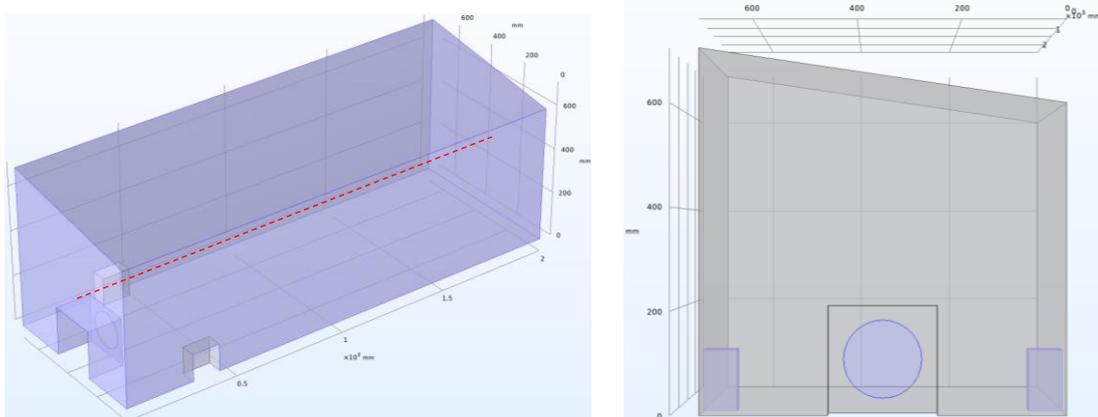


Abbildung 4-5: Randbedingungen des Modells; **Links:** Raumabgrenzungen mit Absorptionsgrad; **Rechts:** Lautsprechermembran mit Beschleunigung und schallhartes Mikrofonverstärkergehäuse

Die Diskretisierung des Modells erfolgt durch ein Tetraedernetz mit Lagrange-Elementen 2. Ordnung. Die Größe der Netz-Elemente orientiert sich an der Wellenlänge der Schallwelle. Um eine Welle mit ausreichender Auflösung zu erfassen, sind mindestens 12 Freiheitsgrade pro Wellenlänge nötig. Dadurch lässt sich ein Aliasing-Effekt, ähnlich wie in der Erfassung analoger Signale, vermeiden. Diese Vorgabe kann erreicht werden, indem die maximale Elementgröße h_{max} folgende Beziehung erfüllt (COMSOL Multiphysics, 2020, p. 187):

$$h_{max} \leq \frac{\lambda}{n} \text{ mit } n \geq 5 \quad (4-2)$$

Dabei ist die Variable n die Anzahl der Knotenpunkte. Weitere Einstellungen, die bei der Erstellung des Netzes durchgeführt sind, betreffen die minimale Elementgröße h_{min} und die maximale Element-Wachstumsrate r_{max} bezogen auf benachbarte Elemente. Der integrierte Mesher berechnet statistische Werte für unterschiedliche Qualitätsmaße des erzeugten Netzes. Ein oft verwendetes Qualitätsmaß für die Beurteilung von FEM-Netzen ist die Schiefe (engl. Skewness) eines einzigen Elements. Weitere ausschlaggebende statistische Indikatoren sind die minimale Elementqualität und die durchschnittliche Elementqualität. Es ist erstrebenswert, eine minimale Elementqualität größer als 0,1 zu erreichen (Griesmer, 2014).

Die Vernetzung des Modells wurde somit mit folgenden Einstellungen durchgeführt:

- Die minimale Elementgröße hat den Wert $h_{min} = 10^{-4} \text{ m}$ für alle Bereiche des Modells. Damit wird sichergestellt, dass die Lautsprechermembran mit ausreichender Auflösung diskretisiert wird.
- Für die maximale Element-Wachstumsrate erreicht man mit dem Wert $r_{max} = 1,1$ ein gutes Verhältnis für die Größen benachbarten Netz-Elementen. Diese wird ebenfalls auf die komplette Geometrie appliziert.
- Die maximale Elementgröße h_{max} auf der Membran des Lautsprechers beträgt 5 mm.
- Die maximale Elementgröße auf den restlichen Flächen ist mit Hilfe einer Netzverfeinerungsstudie bestimmt, indem die Anzahl der Knotenpunkte von fünf bis 25 erhöht wird. Für jede Iteration wird dieser Wert um fünf erhöht. Um eine konservative Lösung zu erzeugen, wird die Wellenlänge mit der Frequenz $f = 500 \text{ Hz}$ und der Schallgeschwindigkeit $c = 343 \text{ m/s}$ berechnet. Die

Lautsprechermembran induziert eine harmonische Anregung des Schallfeldes bei der Frequenz $f = 171.91 \text{ Hz}$. Eine Schnittlinie (siehe gestrichelte Linie in Abbildung 4-5) erstreckt sich durch das komplette Raumvolumen parallel zu der x-Achse.

Für jeden Wert von n wird ein neues Netz generiert und die Schalldrücke entlang der Schnittlinie berechnet. Es werden die Differenzen Δp zwischen den Schalldrücken aller Netze und einem Referenzdruck gebildet, siehe Abbildung 4-6. Als Referenzdruck werden die Ergebnisse des feinsten Netzes mit $h_{max}(25) = 27,44 \times 10^{-3} \text{ m}$ festgelegt.

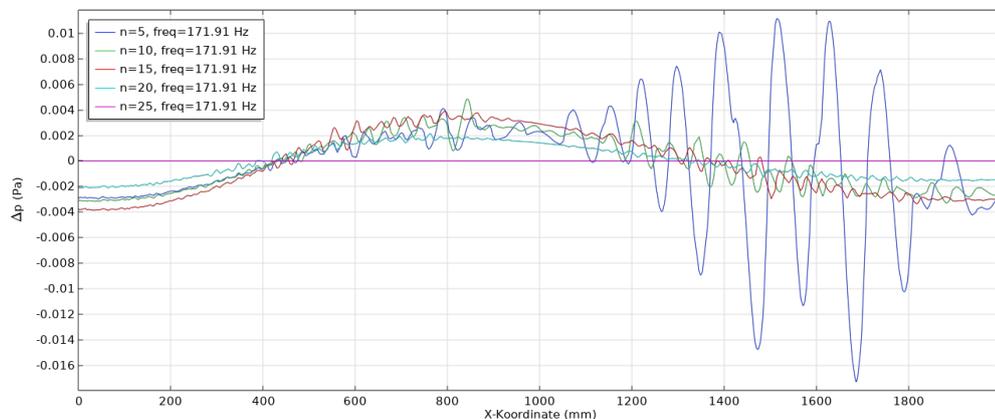


Abbildung 4-6: Schalldruckdifferenzen entlang einer Schnittlinie für Netze mit verschiedenen Feinheitsgraden

Wie erwartet, verbessert sich die Genauigkeit der Simulation mit steigender Auflösung der Diskretisierung. Für $n = 5$ ($h_{max}(5) = 0,14 \text{ m}$) treten starke Schwankungen im hinteren Bereich des Modells auf. Mit der Verfeinerung des Netzes werden die Schwankungen unterdrückt oder sogar eliminiert. Dieses Verhalten lässt sich durch den Einfluss der Vernetzung des Lautsprechers und der Mikrofonverstärker auf das globale Netz erklären. Im vorderen Bereich kann die Elementgröße des globalen Netzes nicht auf die maximal zugelassene Größe wachsen, weil die Größe der Elemente auf der Lautsprechermembran in Kombination mit der Wachstumsrate das nicht erlauben. Erst ab einem bestimmten Abstand erreichen die Elemente ihre maximale Größe, wodurch die Auflösung des Netzes und die Genauigkeit und Qualität der Berechnung abnehmen. Die statistischen Indikatoren der unterschiedlichen Netze zeigen eine tendenzielle Verbesserung der Netzqualität mit steigendem Diskretisierungsgrad (Anhang B). Mit Hilfe der Differenzen der Schalldrücke und der Netzqualitätsindikatoren wird die maximale Elementgröße

auf $h_{max}(20) = 34,3 \times 10^{-3} \text{ m}$ festgelegt. Abbildung 4-7 präsentiert das mit den beschriebenen Einstellungen vernetzte Modell.

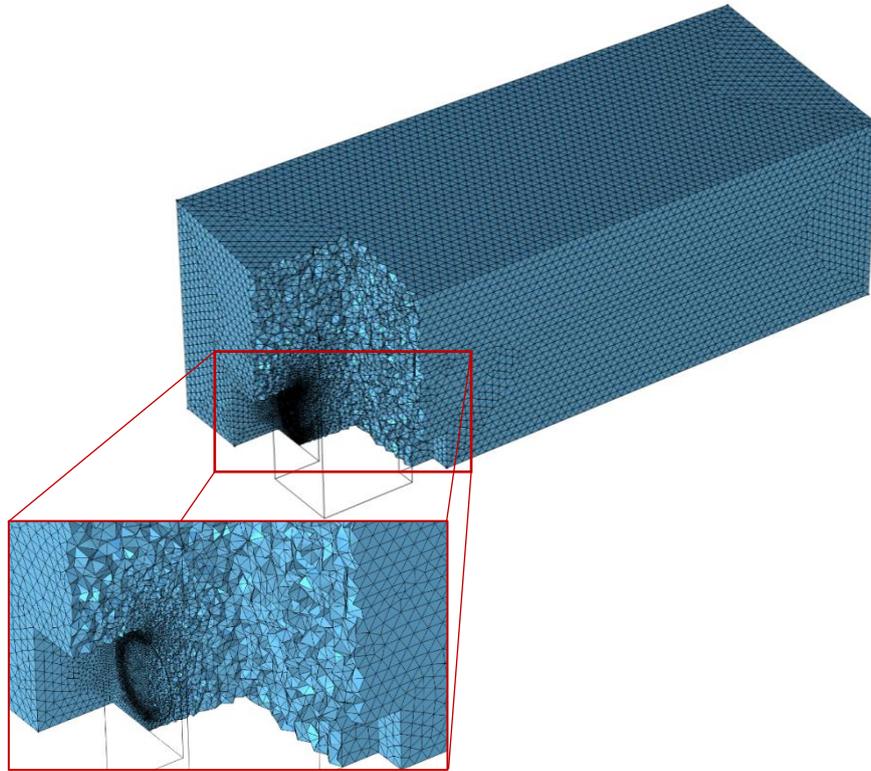


Abbildung 4-7: Vernetztes CRC Mock-up

4.3. Virtual Reality Modell eines Crew Rest Compartments

Das Unity-Modell des CRC Mock-ups besteht aus einzelnen Komponenten, ähnlich wie das reale Mock-up. Die einzelnen Komponenten wurden zuerst in Unitys Konstruktionsumgebung generiert und anschließend an die richtigen Positionen verschoben, damit die Geometrie mit den richtigen Maßen entsteht. Auch der Lautsprecher und die Mikrofonverstärker werden modelliert und an die gleiche Position, wie in der realen Umgebung, positioniert. Da Unity nur einen einzigen aktiven Audioaufnehmer akzeptiert, wird das Schallfeld von einem einzigen Mikrofon aufgenommen. Abbildung 4-8 zeigt das fertige Modell.

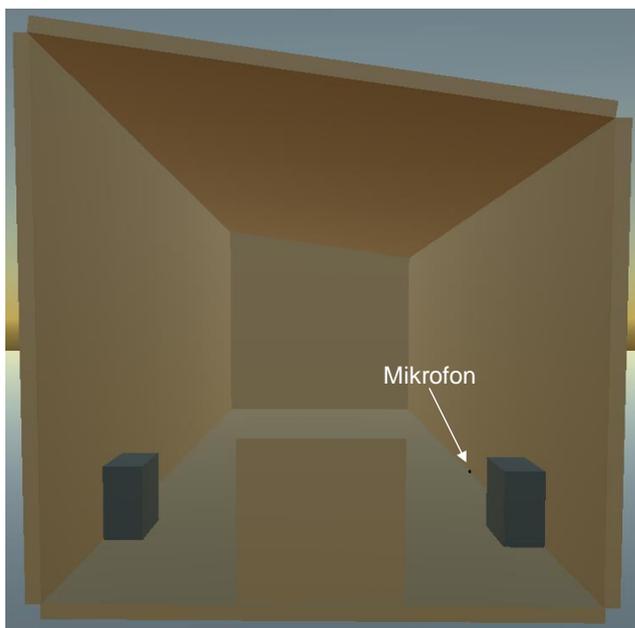


Abbildung 4-8: CRC Mock-up in der virtuellen Umgebung

Das Mikrofon hat die Form einer Kugel⁶ und enthält das Tonaufnahmeskript (*AudioListener*) des jeweiligen Audioplugin.

Ein Skript übernimmt die Bewegung des Mikrofons, sodass die Messung automatisiert wird und die Reproduzierbarkeit der Messungen gewährleistet ist (vgl. Anhang C). Das Bewegungsskript ermöglicht die Definition von Kartierungsrouten mit beliebig vielen Konfigurationen, wie zum Beispiel Anzahl der Messpunkte und Abstand zwischen den Messpunkten in x - und y -Richtung, und die Messdauer an den jeweiligen Messpositionen. Es ist auch möglich, mehrere horizontale Messebenen einzufügen.

⁶ Die Geometrie des virtuellen Mikrofones hat keinen Einfluss auf die Tonaufnahme. Sie dient ausschließlich der Identifizierung des Mikrofones im Raum.

Sowohl der der Ton für die Messung der Frequenzantwort als auch die Messtöne für die Kartierungsmessungen wurden mit einem Tongenerator in LabVIEW erzeugt. Die Abtastfrequenz für die Erstellung der Töne beträgt den standardisierten Wert $f_s = 44100 \text{ Hz}$.

Im nächsten Abschnitt werden die vorgenommenen Einstellungen für die Messung mit den eingesetzten Audio Plug-ins vorgestellt. Alle Messungen wurden mit der Unity Version 2021.2.1f1 durchgeführt.

Wie in Kapitel 3.2 erwähnt, benötigt das Plug-in Resonance Audio die Definition eines Volumens, in dem die akustischen Effekte berechnet werden. Mit dem Objekt *ResonanceAudioRoom* wird ein Quader definiert, welcher das Luftvolumen im CRC Modell umfasst. Die Länge (x -Richtung) und die Breite (y -Richtung) des Quaders entsprechen den Maßen des Mock-ups. Die Höhe wird so eingestellt, dass das gesamte Luftvolumen aus dem Mock-up erfasst werden kann. Den Raumbegrenzungen, sowie dem Lautsprechergehäuse, werden folgende Materialien zugewiesen:

- „*Wood Panel*“ (Holzplatte)
- „*Wood Ceiling*“ (Holzdecke)
- „*Plywood Panel*“ (Sperrholzplatte)

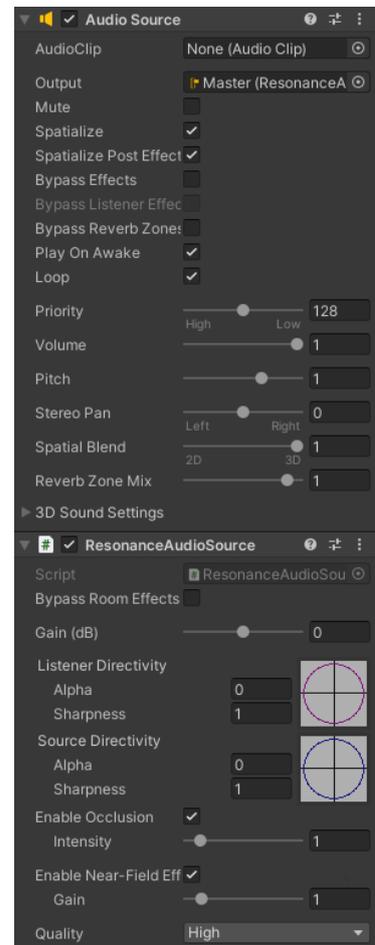


Abbildung 4-9: Einstellungen der Schallquelle und des Mikrofons für das Resonance Audio Plug-in

Da die akustischen Eigenschaften dieser Materialien nicht bekannt sind, werden für jedes einzelne Material Schallfeldmessungen durchgeführt und diese im Kapitel 5.2 präsentiert. Die beiden Mikrofonverstärker sind mit dem Material „*Metal*“ (Metall) versehen. Ähnlich wie in der FEM-Simulation sollen diese einen akustisch reflektierenden Gegenstand im Raum repräsentieren.

Das Lautsprecherobjekt enthält eine Schallquelle, welche das Schallfeld im Mock-up anregt. Die Einstellungen des Lautsprechers wurden nach (Resonance Audio (b), 2022) durchgeführt (vgl. Abbildung 4-9).

Mit den Faktoren „Alpha“ und „Sharpness“ unter dem Unterpunkt Source Directivity wird die Abstrahlform des Lautsprechers angepasst. Da man keine Herstellerdaten für das Abstrahlverhalten des realen Lautsprechers besitzt, wird dieses als omnidirektional angenommen. Die Einstellungen unter dem Unterpunkt Listener Directivity beziehen sich auf den Zuhörer, also das virtuelle Messmikrofon. Mit den eingestellten Werten wird ein Mikrofon mit omnidirektionaler Empfangscharakteristik erzeugt. Sowohl in den Messungen aus der realen Umgebung als auch in den FEM-Berechnungen werden die Effekte einer HRTF nicht berücksichtigt. Um vergleichbare Messergebnisse mit den Plug-ins zu erzeugen, muss die räumliche Platzierung der Audioausgabe deaktiviert werden. Damit das Plug-in die HRTFs ignoriert, wird die Option „Enable Stereo Speaker“ am virtuellen Mikrofon aktiviert. Für die Messungen wurde die Version 1.2.1 des Plug-ins Resonance Audio verwendet.

Als Nächstes werden die Einstellungen für die Messung mit dem Steam Audio Plug-in vorgestellt. Für die in Echtzeit berechneten Reflexionen sind die Materialien der Wände wichtig. Weil die Messungen mit deutlich kleineren Frequenzen als 800 Hz durchgeführt werden, ist nur die Anpassung des Absorptionsgrades α bei tieferen Frequenzen notwendig. Für alle Raumabgrenzungen sowie das Lautsprechergehäuse wird der Wert $\alpha = 0,05$ aus Kapitel 4.2 angewendet. Für die beiden Mikrofonverstärker beträgt der Absorptionsgrad den Wert $\alpha = 0$.

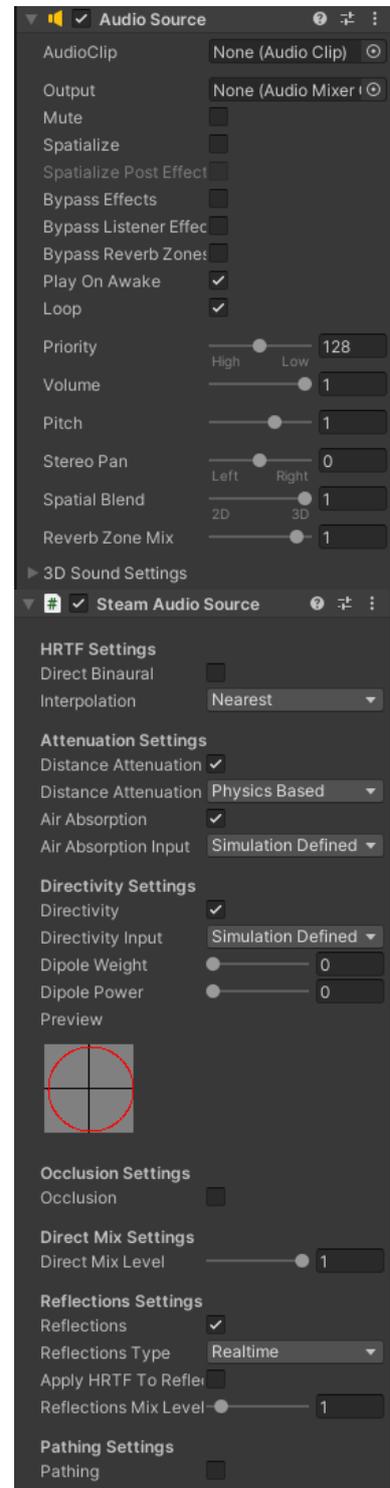


Abbildung 4-10: Lautsprechereinstellungen für das Steam Audio Plug-in

Die Einstellungen für die Schallquelle können der Abbildung 4-10 entnommen werden.

Damit das Plug-in keine HRTF in die Simulation einbezieht, wird die räumliche Ausgabe (Spatialize) deaktiviert. Das Verhalten der Abnahme des Schallpegels mit der Entfernung (Distance Attenuation) und die Luftabsorption (Air Absorption) werden von den eigenen Algorithmen des Plug-ins kontrolliert, sodass eine Vortäuschung durch einen externen Eingriff in das Verhalten des Plug-ins vermieden wird. Das Abstrahlverhalten des Lautsprechers ist identisch mit dem im Resonance Audio Plug-in eingestellten Verhalten.

Die Messungen wurden mit der Version 4.1.1 des Plug-ins Steam Audio durchgeführt.

5. Schallfeldmessungen in den unterschiedlichen Messumgebungen

Das folgende Kapitel befasst sich mit der Durchführung der Messungen in den zuvor vorgestellten Messumgebungen und der Auswertung der Messergebnisse.

Für alle Umgebungen werden hauptsächlich zwei Arten von Messungen durchgeführt. Als Erstes wird der Frequenzantwort der Umgebung untersucht, indem zwei statische Messmikrofone den Schalldruckpegel L_p in dem Frequenzband von $f_m = 50$ Hz bis $f_m = 500$ Hz mit einem Frequenzschritt $\Delta f_m = 5$ Hz erfassen. Die Messdauer t_m beträgt 10 Sekunden für jede einzelne Frequenz f_m . Die Position der Messpunkte wird auf die zwei oberen Raumecken bei $x = 2$ m gesetzt. Auf der gegenüberliegenden Seite des Mock-ups befindet sich die Schallquelle, welche das Schallfeld harmonisch angeregt. Damit der Einfluss der beiden Mikrofonverstärker, welche sich bei der Raumkartierung in dem Mock-up befinden werden, auf die Frequenzantwort des Raumes erfasst wird, positioniert man sie an diesen Positionen, siehe Abbildung 5-1.



Abbildung 5-1: Position der Messmikrofone, des Lautsprechers und der Mikrofonverstärker für die Messung der Frequenzantwort

Diese Messung dient der Identifizierung der Eigenfrequenzen f_E , sodass die getroffenen Annahmen und vorläufig ermittelten Raummoden aus Kapitel 4.1 verifiziert werden können. Aus der Analyse der Frequenzantwort werden auch die Messfrequenzen $f_{m,i}$ für die anschließende Kartierung des Schallfeldes festgelegt.

Mit der zweiten Messung wird die räumliche Schalldruckverteilung für einen Teil des eingeschlossenen Volumens erfasst, indem der Schalldruckpegel L_p in regelmäßigen

Abständen und in zwei Ebenen, E1 bei $z = 0,22 \text{ m}$ und E2 bei $z = 0,52 \text{ m}$ mm ermittelt wird (vgl. Abbildung 5-2). Zusätzlich, werden auch die Phasendifferenzen $\Delta\varphi$ zu einem vordefinierten Referenzpunkt gemessen.

Das Schallfeld wird für diese Messung nur bei den zuvor identifizierten Messfrequenzen $f_{m,i}$ angeregt. Es wird angestrebt, dass sich eine Raummode ausbildet, damit man mit der Kartierung eine stehende Schallwelle erfasst.

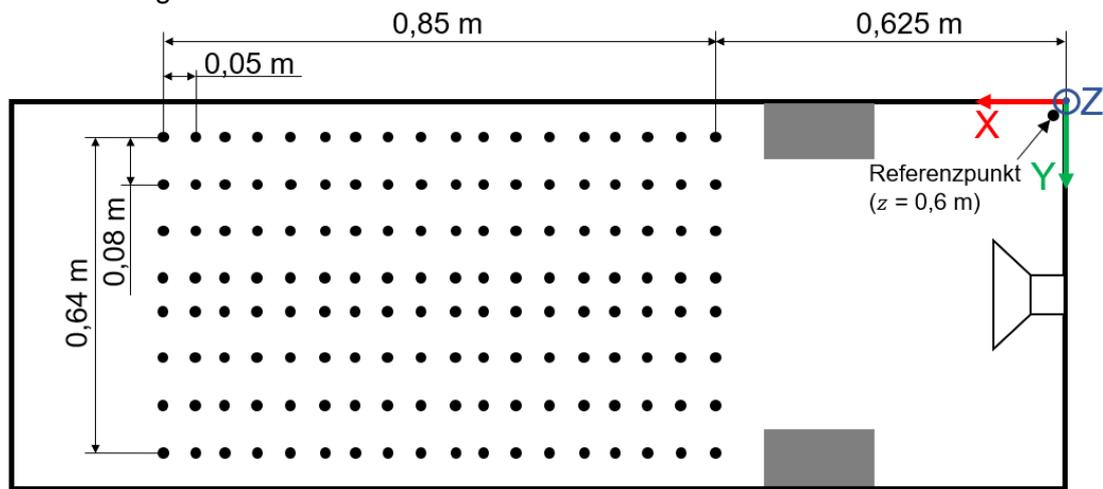


Abbildung 5-2: Messpunkte für die Kartierungsmessung; Position des Lautsprechers und der Mikrofonverstärker bleibt unverändert

5.1. Messvorbereitungen in der realen Messumgebung

Das Übertragungsverhalten der einzelnen Messkomponenten wird vor der Schallfeldmessungen bestimmt. Diese Informationen sind für die richtige Auswertung der Messungen erforderlich.

Eine zentrale Komponente im ganzen Messaufbau sind die Messkarten myDAQ. Diese sind in Kombination mit dem LabVIEW-Messskript für die Generierung des Messsignals, die Signalauffassung, -digitalisierung, -bearbeitung und die Messdatenabspeicherung zuständig. Eine mit LabVIEW erstellte Messsoftware (siehe Anhang D) wertet die gemessenen Mikrofonspannungen $U_{M,i}$ mit einer internen Tonanalyse aus, welche kontinuierlich während der Messung ausgeführt wird. Sie liefert Informationen über die Frequenz, Amplitude und Phase des gemessenen Signales. Die Einstellungen der Abtastfrequenzen der Messkarte beeinflussen die Qualität und Geschwindigkeit der Signalerfassung. Die myDAQ Geräte werden wie folgt konfiguriert:

Tabelle 5.1: Einstellungen der myDAQ Messkarten

	Signalgenerierung	Signalerfassung
Abtastfrequenz f_s	44100 Hz	22050 Hz
Maximale Ein- und Ausgangsspannung	$\pm 2 \text{ V}$	$\pm 2 \text{ V}$

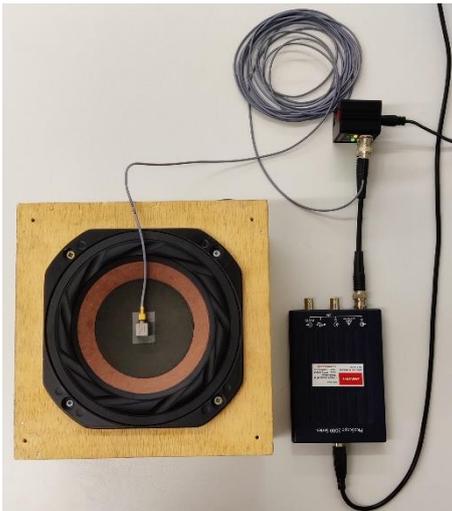


Abbildung 5-3: Messaufbau für die Messung der Übertragungsfunktion des Lautsprechers

Die Übertragungsfunktion des Lautsprechers wird anhand der Membranbeschleunigung $a(f)$ ermittelt. Ein harmonisches Testsignal mit einer Amplitude von $\hat{U}_A = 2 \text{ V}$ wird mit dem Signalgenerator des myDAQ-Gerätes generiert. Der Beschleunigungsaufnehmer Brüel&Kjaer Type 4507 wird im Mittelpunkt der Membran angebracht und registriert die Beschleunigung bei ansteigenden Anregungsfrequenzen, siehe Abbildung 5-3. Mit der Messkarte PicoScope 2204A der Firma Pico Technology wird das Signal des Beschleunigungsaufnehmers digitalisiert und aufgezeichnet. Folgende Übertragungsfunktion lässt sich dadurch aufzeichnen:

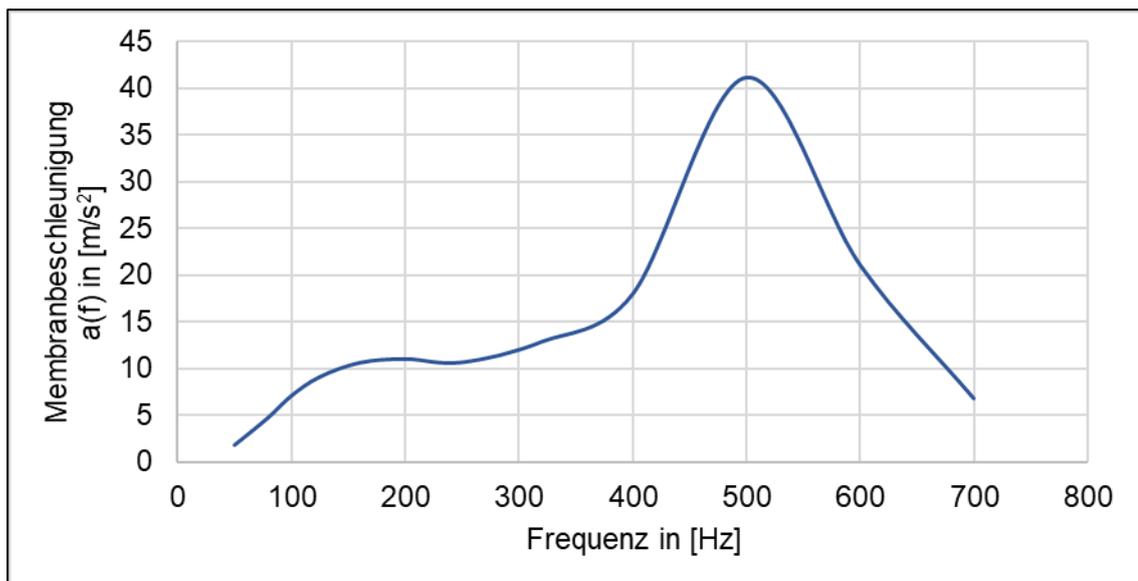


Abbildung 5-4: Gemessene Membranbeschleunigung des Lautsprechers

Die angegebene Empfindlichkeit der Messmikrofone wird durch eine Kalibrierungsmessung angepasst. Mit dem Messaufbau aus Abbildung 5-5 legt man einen Referenzschalldruckpegel mit dem Schallpegelmessgerät NTI XL2 M4261 bei einer harmonischen Anregung im Freifeld fest. Anschließend wird die gleiche Messung mit den Messmikrofonen durchgeführt. Die Empfindlichkeit $E_{M,i}$ der einzelnen Mikrofone lässt sich aus den gemessenen Spannungen $U_{M,i}$ und dem Referenzschalldruckpegel $L_{p,ref}$ wie folgt bestimmen:



Abbildung 5-5: Messaufbau für die Kalibrierung der Messmikrofone

$$E_{M,i}(f) = \frac{U_{M,i}(f) \times 10^3}{2 \times 10^{-5} \times 10^{\frac{26}{20}} \times 10^{\frac{L_{p,ref}(f)}{20}}} \quad (5-1)$$

Der Faktor $10^{\frac{26}{20}}$ ergibt sich aus der eingestellten Verstärkung $V_m = 26$ dB am Mikrofonverstärker. Abbildung 5-6 enthält die Verläufe der gemessenen Empfindlichkeiten über ein Frequenzspektrum von 50 Hz bis 1000 Hz.

Die Bewegungseinrichtung, welche für die Kartierung des Schallfeldes eingesetzt wird, ist Teil einer Messeinrichtung, welche im Rahmen des Projekts *DigiNet.Air* konstruiert und für die Messung der Genauigkeit der Lagesensoren von VR-Brillen eingesetzt wurde. Sie besteht aus einem Aluminiumprofil mit den Maßen 1 x 0,04 x 0,08 m (Länge x Breite x Höhe). Es ist mit einem Schlitten ausgestattet, welcher von einem Schrittmotor um 0,85 m in Längsrichtung verfahren werden kann. Für die Kartierung des Schallfeldes wird nur das Aluminiumprofil mit der Bewegungseinrichtung, sowie die Schrittmotorsteuerung verwendet. Die Ansteuerung des Schrittmotors, welche ebenfalls im Rahmen des oben genannten Projektes erstellt wurde, erfolgt mit einem *Arduino Mega 2560* und einem *Ramps 1.4 Motor Shield*. Mit Hilfe eines Simulink-Modells (siehe Anhang E) kann

der Benutzer den Arduino ansteuern und somit eine gewünschte Position anfahren (Bodolea, 2021).

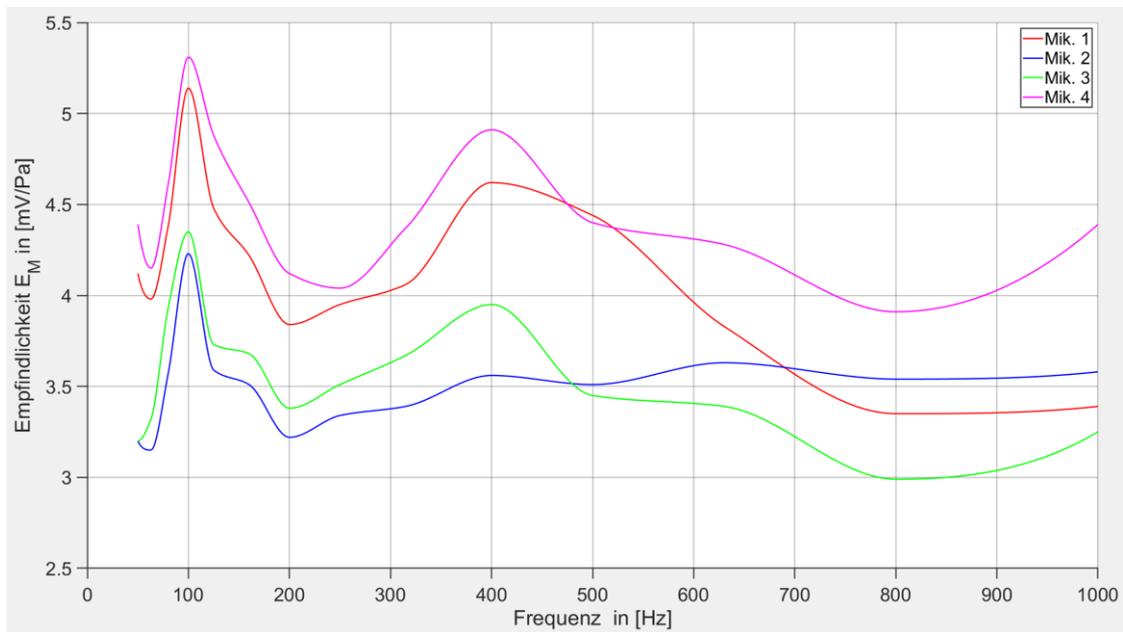


Abbildung 5-6: Gemessene Empfindlichkeiten der Messmikrofone

5.2. Messung der Frequenzantwort der unterschiedlichen Umgebungen

Die Messungen in der realen Umgebung werden mit dem Messaufbau aus Kapitel 4.1, Abbildung 4-3 durchgeführt. Die Auswertung der Messdaten wird mit einem MATLAB-Skript (siehe Anhang F, 10) realisiert. Mit den zuvor ermittelten Mikrofonempfindlichkeiten $E_{M,i}$ für die eingesetzten Messmikrofone lässt sich folgende Frequenzantwort des Mock-ups messen:

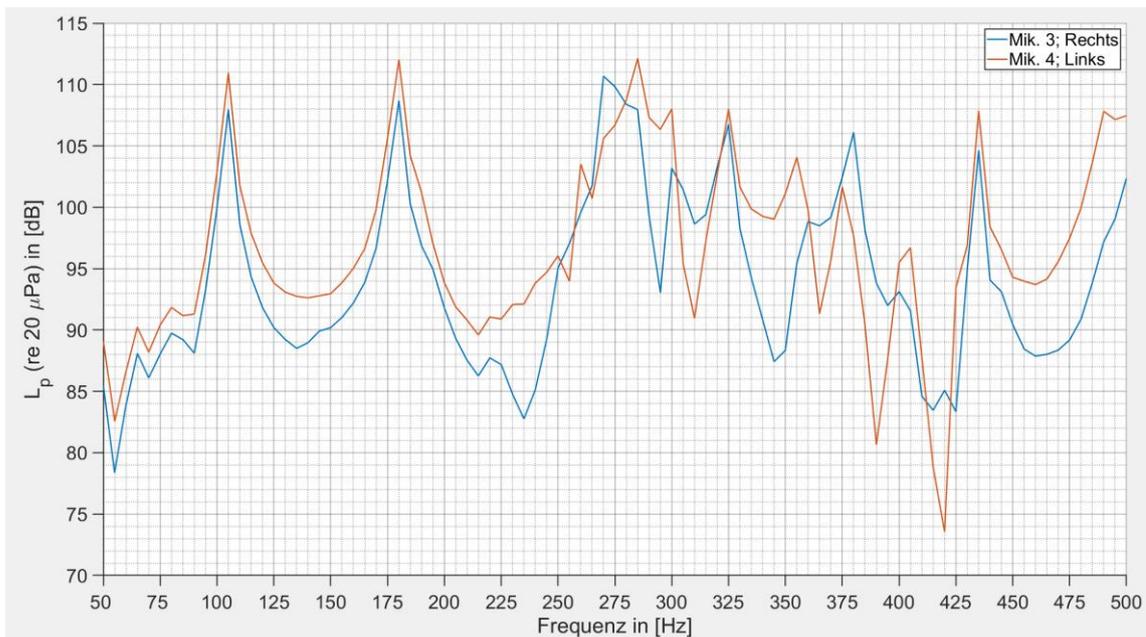


Abbildung 5-7: Gemessene Frequenzantwort des CRC Mock-ups

Man kann aus Abbildung 5-7 zwei distinkte Schallpegelwerte bei $f_m = 105$ Hz und $f_m = 180$ Hz erkennen. Im Allgemeinen können ausgeprägte Extremwerte als die Entstehung einer Raummode interpretiert werden. Ab $f_m \approx 250$ Hz lassen sich einzelne Pegelmaxima und -minima noch erkennen, allerdings bilden sich keine deutlich erkennbare Raummoden mehr aus. Mit den verfügbaren Messdaten ist die Identifikation der Raummoden bei Frequenzen über $f_m \approx 250$ Hz aufgrund der steigenden Raummodendichte nur bedingt realisierbar. Dieses Verhalten spiegelt den Effekt der Überlagerung der Schallwellen, wie in Kapitel 2.2 erläutert, wider.

Anhand der ersten zwei gemessenen Druckmaxima lassen sich die Eigenfrequenzen $f_{E_i,M}$ der ersten Raummoden identifizieren. Die im Kapitel 4.1 überschlägig berechneten Eigenfrequenzen $f_{E_i,\ddot{u}}$ zeigen, dass im Bereich der gemessenen Frequenzen der Druckspitzen zwei Raummoden entstehen. Mit dieser Information wird der Raummode (1-0-0)

die Eigenfrequenz $f_{E_{1,M}} = 105 \text{ Hz}$ und der Raummode (2-0-0) die Resonanzfrequenz $f_{E_{2,M}} = 180 \text{ Hz}$ zugeordnet.

Die Unterschiede zu den zuvor bestimmten Eigenfrequenzen können durch die leicht abweichende Geometrie des realen Mock-ups erklärt werden. Auch die Effekte der Mikrofonverstärker und des Lautsprechergehäuses können bei den Berechnungen aus Kapitel 4.1 nicht berücksichtigt werden.

Die FEM-Berechnung der Frequenzantwort wird mit dem im Kapitel 4.2 präsentierten Modell durchgeführt. Ähnlich wie bei der Messung in der realen Umgebung wird das Luftvolumen über das gleiche Frequenzspektrum angeregt. Die frequenzabhängige Membranbeschleunigung $a(f)$ des virtuellen Lautsprechers entspricht der gemessenen Übertragungsfunktion des realen Lautsprechers. Die Anpassung der Membranbeschleunigung an die Messfrequenz stellt das reale frequenzabhängige Verhalten des modellierten Lautsprechers in der Simulation dar. Mit Hilfe der Messdaten wird eine Funktion für die Beschleunigung in COMSOL definiert. Für jede Anregungsfrequenz f_M wird der entsprechende Beschleunigungswert $a(f_M)$ durch Stückweise-Kubische-Interpolation ermittelt und auf die schwingende Membran appliziert. Abbildung 5-8 zeigt das Ergebnis der FEM-Berechnung.

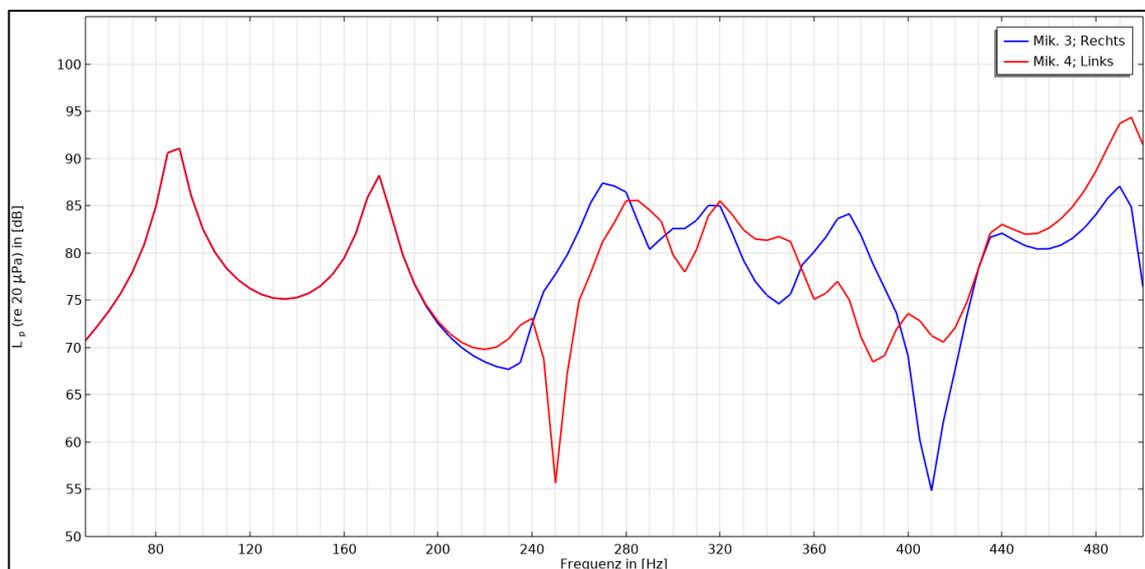


Abbildung 5-8: Berechnete Frequenzantwort des CRC Mock-ups

Die berechnete Frequenzantwort zeigt zwei distinkte Pegelmaxima bei $f_m = 90 \text{ Hz}$ und $f_m = 175 \text{ Hz}$, welche als die ersten Raummoden interpretiert werden. Aus dem leicht

steigenden Kurvenverlauf zwischen $f_m = 85$ Hz und $f_m = 90$ Hz kann festgestellt werden, dass sich die Eigenfrequenz $f_{E_{1,B}}$ der ersten Raummode zwischen diesen beiden Frequenzen befindet. Für die Aufklärung dieses Verlaufes wird für das analysierte Modell eine Eigenfrequenzstudie in COMSOL durchgeführt. Das Ergebnis dieser Studie zeigt für die erste Resonanzfrequenz einen Wert $f_{E_{1,B}} = 87,6$ Hz, wodurch die Interpretation des Pegelmaximums richtig ist. Die zweite Spitze hat einen steileren Verlauf. Die Kontrolle mit der Eigenfrequenzstudie ergibt die zweite Resonanz bei $f_{E_{2,B}} = 174,4$ Hz.

Es wird daraus die Schlussfolgerung gezogen, dass die genaue Erfassung der Eigenfrequenzen nur durch eine Erhöhung der Auflösung des Frequenzbandes realisierbar ist, was dazu führt, dass sich die Rechenzeit verlängert.

Ab der Frequenz $f_m \approx 250$ Hz stellt sich das gleiche Verhalten wie bei der gemessenen Frequenzantwort ein. Es können einzelne Extremwerte beobachtet werden, beispielsweise bei den Frequenzen $f_m = 250$ Hz oder $f_m = 410$ Hz. Aufgrund der wachsenden Raummodendichte bei höheren Frequenzen ist die Feststellung einzelnen Raummoden nur bedingt möglich.

Vergleicht man die berechneten Eigenfrequenzen $f_{E_{i,B}}$ mit den Raummoden aus Kapitel 4.1 so kann man feststellen, dass die überschlägig berechneten Resonanzfrequenzen einen vergleichbaren Wert haben. Die Unterschiede in den Werten können mit den gleichen Argumenten wie bei der Messung in der realen Umgebung erklärt werden.

Für die Messung der simulierten Frequenzantwort des Mock-ups in der virtuellen Umgebung spielt die virtuelle Schallquelle einen Ton mit wechselnden Frequenzen ab. Der Ton enthält das gleiche Frequenzband und Messdauer wie der Messton der Messungen in der realen Umgebung. Die Audioausgabe wird mit der kostenlosen Software Audacity® aufgenommen. Die einzelnen Frequenzen und Amplituden der aufgenommenen Audio-datei werden im MATLAB mit einer FFT-Analyse ermittelt (Anhang F, 11). Die Berechnung der Amplitudenpegel L_A wird auf die maximale gemessene Amplitude bezogen, weil man keine Informationen für eine Umwandlung der gemessenen Größen in akustische Drücke und anschließend in Schalldruckpegel hat. Die Messungen mit dem Plug-in Resonance Audio führt man mit den Wandmaterialien *Holzplatte*, *Holzdecke* und

Sperrholzplatte durch. Abbildung 5-9, Abbildung 5-10 und Abbildung 5-11 zeigen die Messergebnisse.

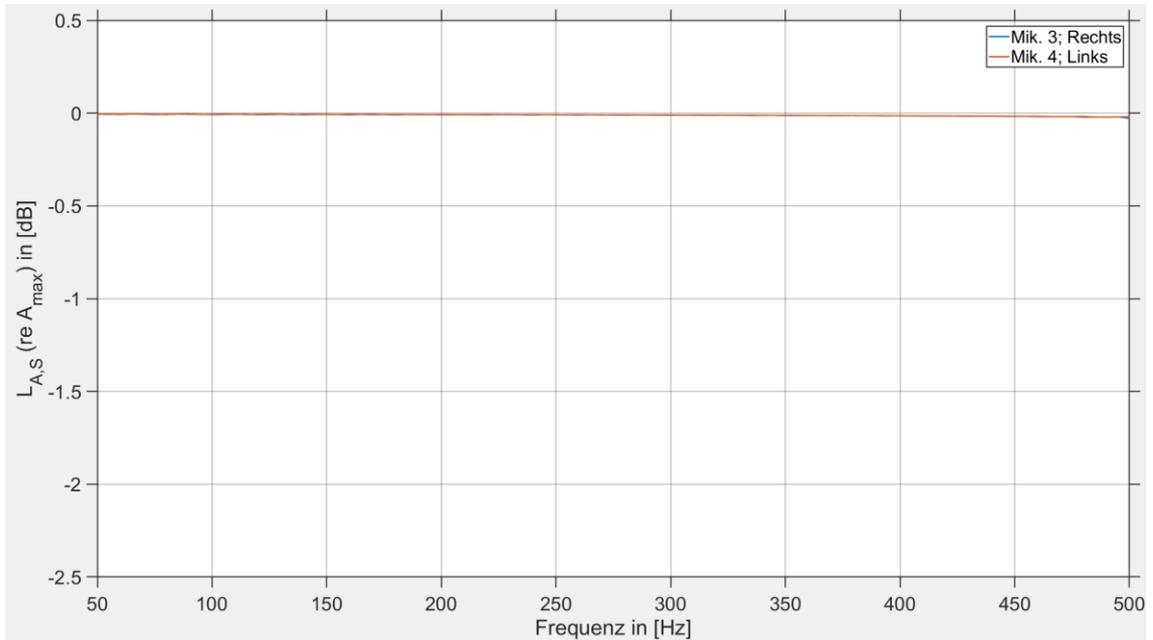


Abbildung 5-9: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial **Holzplatte**; Resonance Audio Plug-in

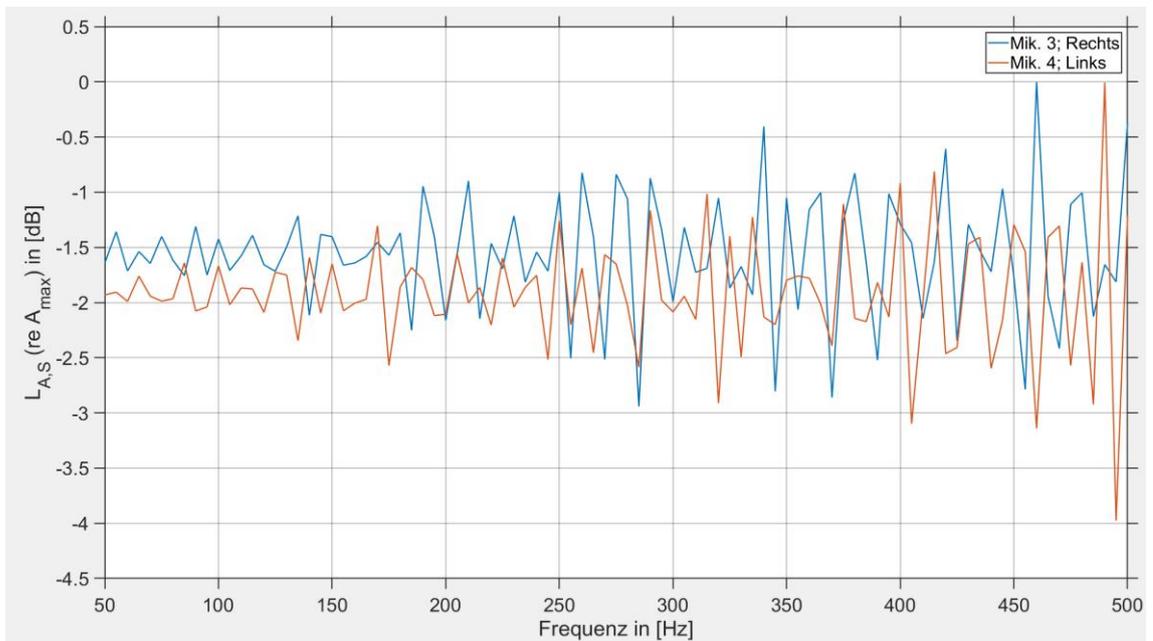


Abbildung 5-10: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial **Holzdecke**; Resonance Audio Plug-in

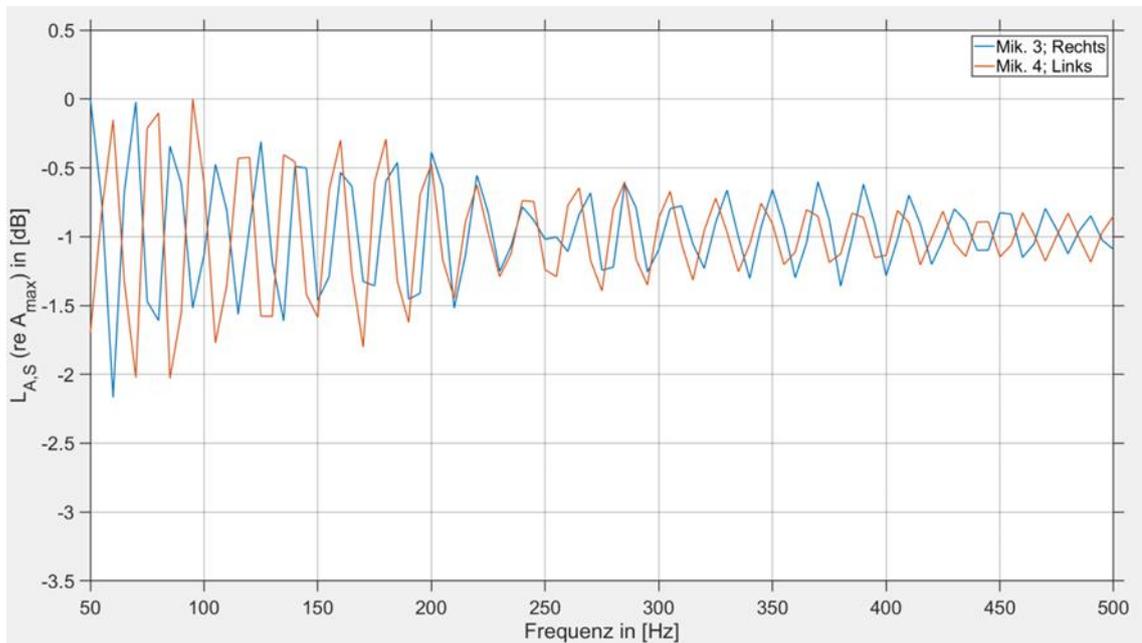


Abbildung 5-11: Simulierte Frequenzantwort des CRC Mock-ups mit dem Wandmaterial **Sperrholz**, Resonance Audio Plug-in

Die Verläufe aller gemessenen Amplitudenpegel zeigen keine Raummodenausprägung. Die wachsenden Amplitudenpegel in Abbildung 5-10 können als eine frequenzabhängige Absorption des Wandmaterials interpretiert werden. Die Simulation zeigt, dass der Absorptionsgrad des Materials Holzdecke mit steigender Frequenz abnimmt. Das Material Sperrholz besitzt das entgegengesetzte Verhalten. Bei niedrigeren Frequenzen wird weniger Energie als bei höheren Frequenzen absorbiert. Dadurch entsteht der im Durchschnitt sinkende Verlauf der Amplitudenpegel, wie in Abbildung 5-11 zu sehen ist. Der Pegelverlauf aus Abbildung 5-9 deutet auf eine nicht vorhandene oder konstant absorbierende Eigenschaft des Wandmaterials hin.

Der ausgewählte Absorptionsgrad der Modellabgrenzungen aus der FEM-Umgebung nimmt mit der Frequenz zu (Smardzewski, et al., 2015). Es kann die gleiche Tendenz in Abbildung 5-10 beobachtet werden. Um das Modell für die folgenden Kartierungsmessungen realitätsnaher zu gestalten, wird Sperrholz als Wandmaterial gewählt.

Die Messung mit dem Steam Audio Plug-in liefert die Ergebnisse aus der folgenden Abbildung:

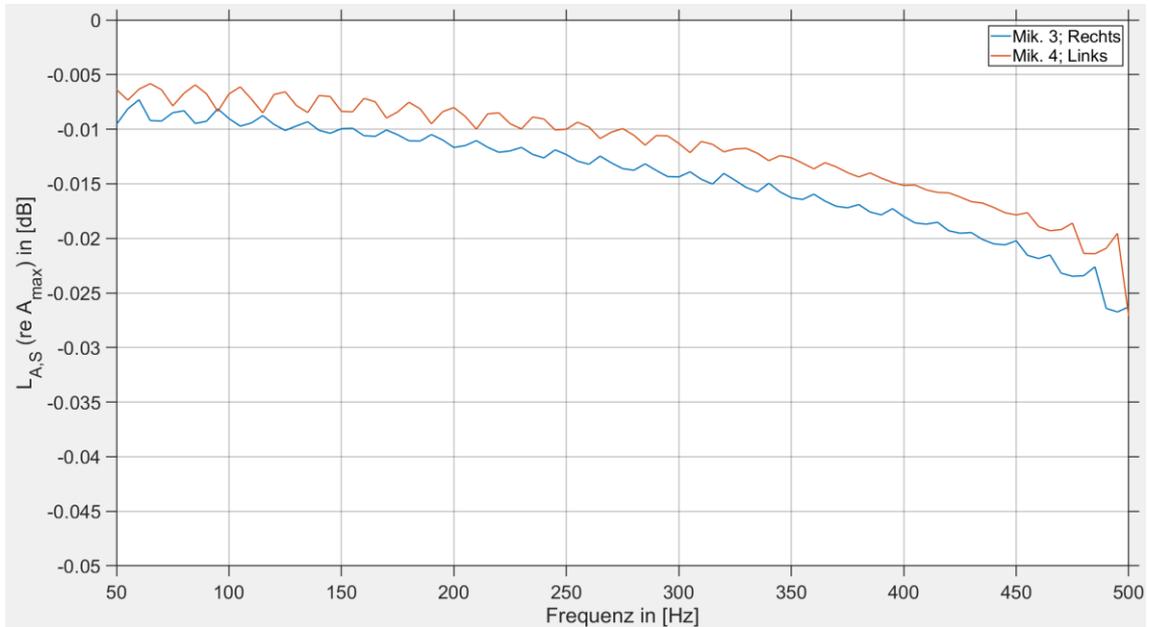


Abbildung 5-12: Simulierte Frequenzantwort der CRC Mock-ups; Steam Audio Plug-in

Ähnlich wie bei den vorherigen Messungen konnte mit dem Plug-in von Steam keine Raummodenverteilung erfasst werden. Der Pegelgang ist nahezu konstant über das gemessene Frequenzband. Da die Wandabsorption für den Frequenzbereich von 0 Hz bis 800 Hz einen konstanten Wert besitzt, kann die leichte Verringerung der Amplitudenpegel nur durch eine frequenzabhängige Luftabsorption hervorgerufen worden sein.

Nachdem die Frequenzgänge aller Umgebungen bestimmt wurden, können anhand der Ergebnisse der Messungen in der realen und der FE-Umgebung die Messfrequenzen $f_{m_i,K}$ für die Kartierung festgelegt werden. Es muss eine oder mehrere Frequenzen definiert werden, die sowohl in der realen Umgebung als auch in der FEM-Berechnung eine Raummode anregen. Weiterhin muss die Wellenlänge der Schallwelle bei der angeregten Raummode nicht unter 2 m sein, sonst kann das festgelegte Kriterium aus Kapitel 4 nicht erfüllt werden. Für die erste Raummode liefern die FEM-Ergebnisse und die Messergebnisse unterschiedliche Resonanzfrequenzen, wodurch die Kartierungen möglicherweise nicht vergleichbare Ergebnisse zeigen würden. Sowohl für $f_{E_1,M} = 105$ Hz als auch für $f_{E_1,B} = 87,6$ Hz beträgt die Wellenlänge bei einer Schallgeschwindigkeit $c = 348$ m/s eine Wellenlänge größer als $\lambda = 2$ m. Für die zweite axiale Eigenmode in

x -Richtung ist die Diskrepanz zwischen der gemessenen und berechneten Eigenfrequenz deutlich geringer. Auch die Wellenlänge nimmt für beide Frequenzen einen Wert unter 2 m ein. Die Messung und die Simulation zeigen, dass die Raummodendichte bei höheren Resonanzfrequenzen sehr stark zunimmt. Somit würde eine Kartierung bei höheren Frequenzen nicht geeignet sein, da das Schallfeld aus mehreren überlagerten Raummoden bestehen würde.

Betrachtet man die Messergebnisse, so werden für die im nächsten Abschnitt geführten Kartierungen des Schallfeldes zwei Messfrequenzen festgelegt, nämlich $f_{m_1,K} = 180$ Hz und $f_{m_2,K} = 170$ Hz. Die Wahl der ersten Frequenz richtet sich nach der gemessenen Eigenfrequenz $f_{E_2,M} = 180$ Hz. Da die reale Umgebung als Referenz betrachtet wird, kann auch das Ergebnis der Kartierung bei dieser Frequenz als Referenz gesehen werden. Mit der zweiten Messfrequenz untersucht man das Verhalten des Schallfeldes bei einer Anregungsfrequenz, welche nah an einer Eigenfrequenz ist. Dadurch kann für jede Umgebung untersucht werden, wie das Schallfeld auf eine zur Resonanzfrequenz benachbarte Anregungsfrequenz reagiert. Daraus lassen sich Rückschlüsse über das dynamische Verhalten des Schallfeldes ziehen.

5.3. Kartierung der Schallfelder in den unterschiedlichen Messumgebungen

Die Kartierung des Schallfeldes in der realen Umgebung erfolgt mit zwei Messmikrofonen. Diese sind, wie in Abbildung 5-13 gezeigt, in eine 3D-gedruckte Mikrofonhalterung montiert. Die erforderliche Verschiebung der Bewegungseinrichtung in z -Richtung um 0,3 m für die Messung in der Ebene $E2$ wird mit Hilfe von zwei weiteren Aluminiumprofilen realisiert.

Mithilfe der Bewegungseinrichtung wird jeder Messpunkt aus Abbildung 5-2 angefahren. Wie man aus der oberen Abbildung erkennen kann, wird die Kartierung mit zwei Messmikrofonen durchgeführt. Somit können jeweils zwei Messbahnen in x -Richtung gleichzeitig gemessen werden. Am Ende einer Messbahn wird der Schlitten in seine Nullstellung gefahren, der Balken um 0,08 m in y -Richtung verschoben und die Messung fortgesetzt. Dieser Vorgang wird bis zur abgeschlossenen Kartierung des gesamten Messraumes wiederholt. Die Position der Referenzmikrofone kann der Abbildung 5-2 entnommen werden. Sie sind für die Berechnung der Phasendifferenzen in der realen Umgebung zuständig. Weil man zwei Messkarten bei der Messung einsetzt, werden zwei Referenzmikrofone benötigt. Die Phasendifferenz $\Delta\varphi_m$ zu dem jeweiligen Referenzmikrofon wird in der Messsoftware gebildet.

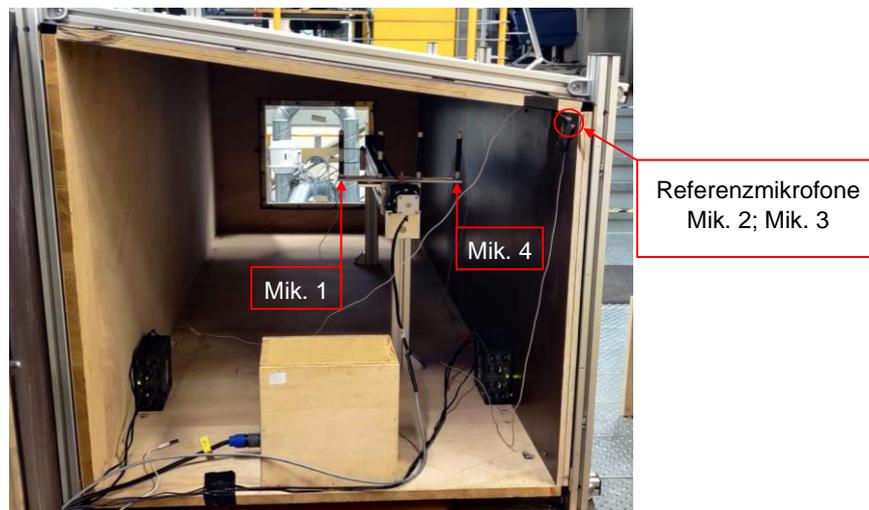


Abbildung 5-13: Messaufbau für die Kartierung des Schallfeldes in der Ebene $E2$

Die Messungen erfolgen bei den zuvor festgelegten Messfrequenzen $f_{m_{1,K}} = 180$ Hz und $f_{m_{2,K}} = 170$ Hz und in den Ebenen E1 und E2. Die Messdaten werden in MATLAB importiert und die räumlichen Verläufe der gemessenen Schalldruckpegel, welche in Abbildung 5-14 und Abbildung 5-15 gezeigt werden, erstellt (siehe Anhang F, 2).

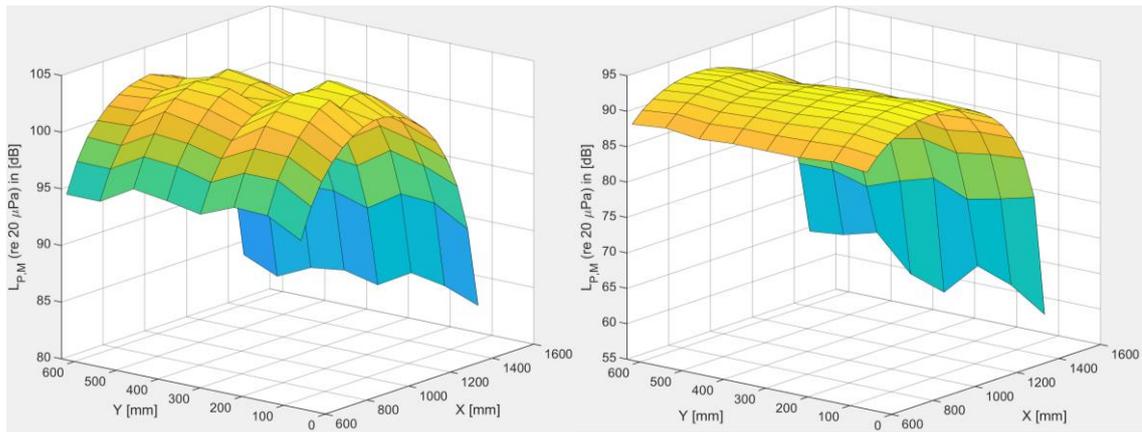


Abbildung 5-14: Gemessene Schalldruckpegelverläufe in der Ebene E1; **linker Verlauf:** $f_{m_{1,K}} = 180$ Hz; **rechter Verlauf:** $f_{m_{2,K}} = 170$ Hz

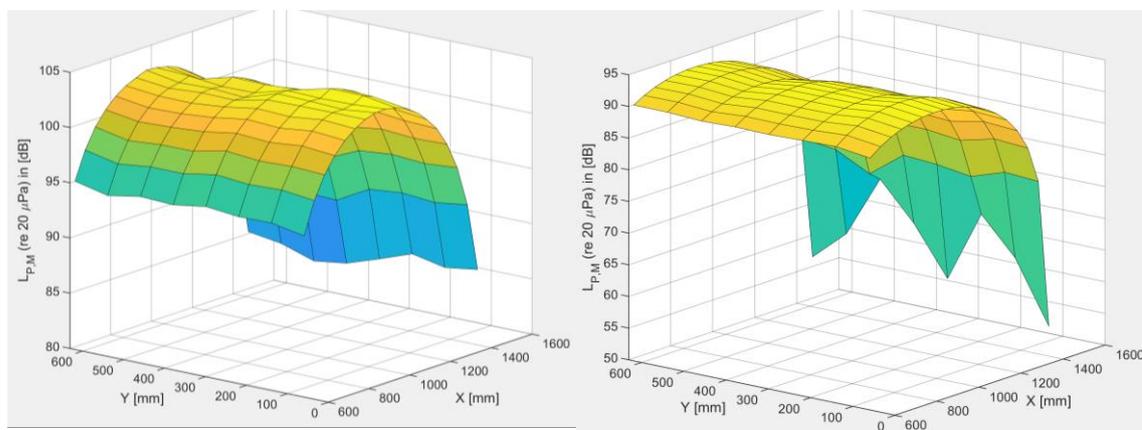


Abbildung 5-15: Gemessene Schalldruckpegelverteilung in der Ebene E2; **linker Verlauf:** $f_{m_{1,K}} = 180$ Hz; **rechter Verlauf:** $f_{m_{2,K}} = 170$ Hz

Die gemessenen Verläufe der Schalldruckpegel zeigen eine eindeutig ausgeprägte Raummode in longitudinaler Richtung. In dem kartierten Bereich kann ein Druckmaximum beobachtet werden. Die räumliche Amplitudenverteilung der Raummode (2-0-0) bei $f_{m_{1,K}} = 180$ Hz zeigt sich in ähnlicher Weise auch bei der geringeren Messfrequenz $f_{m_{2,K}} = 170$ Hz. Daraus wird abgeleitet, dass die stehende Welle der Raummode auch bei Verringerung der Anregungsfrequenz das Schallfeld dominiert.

Bei den Frequenzen $f_{m_1,K} = 180$ Hz erreicht der Schalldruck einen höheren Wert als bei $f_{m_2,K} = 170$ Hz. Dieser Effekt ist erwartet, da bei der Eigenfrequenz das Schallfeld eine stärker ausgeprägte stehende Welle, also einen höheren reaktiven Anteil besitzt. Entlang der y -Koordinate des Raumes können leichte Änderungen in den Schalldruckpegel beobachtet werden. Diese können durch Messungenauigkeiten oder durch den Einfluss der Bewegungseinrichtung im Messraum begründet werden. Am stärksten sind die Unterschiede in der Messung in Ebene 1 und $f_{m_1,K} = 180$ Hz zu sehen. Nichtsdestotrotz, kann behauptet werden, dass die stehende Welle eine ebene Wellenfront besitzt. Die erfasste Schallwelle bei $f_{m_2,K} = 170$ Hz zeigt eine größere Ausdehnung in x -Richtung, die sich durch die größere Wellenlänge erklären lässt.

Die Auswertung der FEM-Berechnung erfolgt mit Hilfe eines MATLAB-Skriptes (siehe Anhang F, 1). Nachdem die Schalldruckverteilung im gesamten Model berechnet wird, können die komplexen Amplituden $\underline{p}_i(\vec{r})$ des Schallfeldes für die in Abbildung 5-2 definierten Messpunkten exportiert werden. COMSOL führt die Berechnung des Schallfeldes im Frequenzbereich durch, sodass eine weitere FFT-Analyse nicht mehr nötig ist. Das eingesetzte MATLAB-Skript ist für die Berechnung der Amplituden $\hat{p}_i(\vec{r})$ laut Gleichung 5-2 (de Vries, 2012) und für die Generierung der Verläufe der Schallpegel $L_{p,B}(\vec{r})$ zuständig. Die Verläufe werden in der Abbildung 5-16 und Abbildung 5-17 präsentiert.

$$\hat{p}_i(\vec{r}) = \sqrt{\left(\operatorname{Re}\{\underline{p}_i(\vec{r})\}\right)^2 + \left(\operatorname{Im}\{\underline{p}_i(\vec{r})\}\right)^2} \quad (5-2)$$

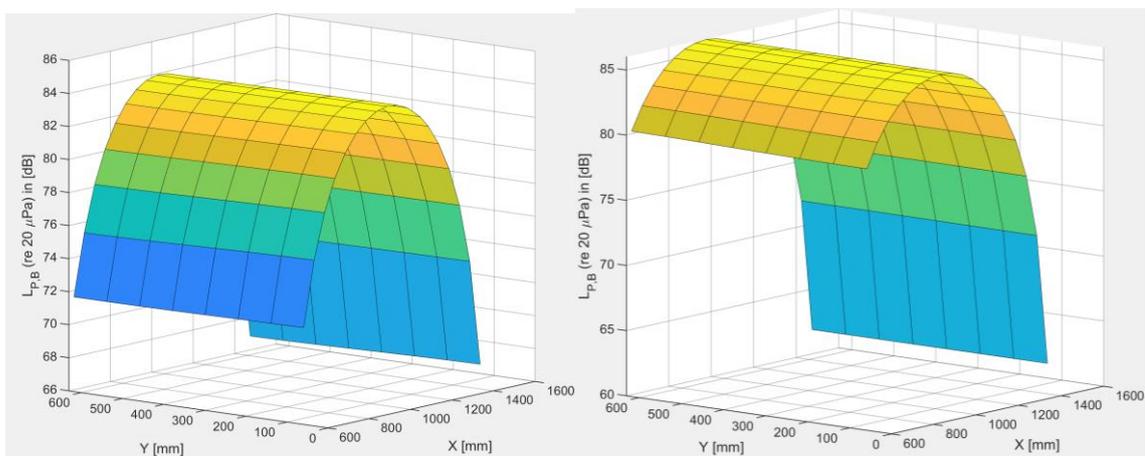


Abbildung 5-16: Berechnete Schalldruckpegelverläufe in der Ebene E1; **linker Verlauf:** $f_{m_1,K} = 180$ Hz; **rechter Verlauf:** $f_{m_2,K} = 170$ Hz;

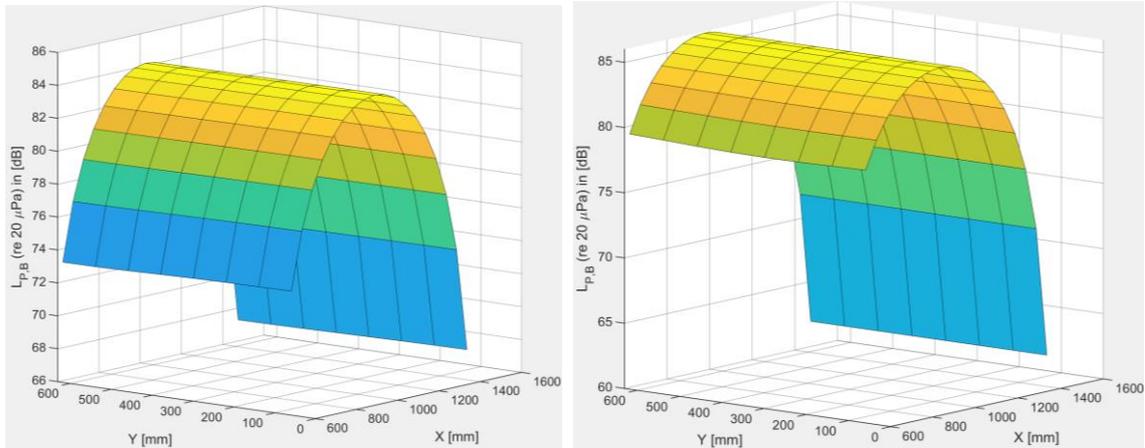


Abbildung 5-17: Berechnete Schalldruckpegelverläufe in der Ebene E2; **linker Verlauf:** $f_{m_1,K} = 180 \text{ Hz}$;
rechter Verlauf: $f_{m_2,K} = 170 \text{ Hz}$

Die berechneten Verläufe zeigen einen Abschnitt aus einer stehenden Welle, welche zu der longitudinalen Raummode (2-0-0) gehört. Vergleicht man die Berechnungsergebnisse in den unterschiedlichen Ebenen aber bei gleichen Messfrequenzen, so können sehr ähnliche Verläufe entlang der x - und y -Achsen beobachtet werden. Die Schalldruckpegel weisen nur kleine Unterschiede in den jeweiligen Messebenen auf. Daraus lässt sich ableiten, dass sich im Raum eine stehende Welle mit ebener Wellenfront bildet.

Die Berechnungen mit $f_{m_1,K} = 180 \text{ Hz}$ erfassen einen größeren Anteil der Schallwelle. Der Grund dafür ist die kleinere Wellenlänge bei der genannten Messfrequenz. Bei beiden Anregungsfrequenzen ist der reaktive Anteil des Schallfeldes dominant.

Zwischen den Messfrequenzen können nur kleine Unterschiede in den maximalen Schalldruckpegel $L_{p,B}$ erkannt werden. Dieses Verhalten lässt sich mit der Betrachtung der Eigenfrequenz $f_{E_2,B} = 174,4 \text{ Hz}$ erklären. Weder die erste noch die zweite Messfrequenz entspricht mit der berechneten Eigenfrequenz, welche sich nahezu in der Mitte zwischen den Messfrequenzen befindet. Somit entsteht nur einen kleinen Unterschied zwischen den berechneten Maximalamplituden.

Die Messdateien der Kartierung in der virtuellen Umgebung wurden in ähnlicher Weise mit der Ermittlung der Frequenzantwort aufgenommen und ausgewertet. Mit einem Bewegungsskript wird das virtuelle Mikrofon auf einer mäanderförmigen Bahn in der x - y -Ebene bewegt. Anschließend führt ein MATLAB-Skript (Anhang F, 4) eine FFT-Analyse durch und erzeugt die Verteilungen der Amplitudenpegel. Für das Plug-in Resonance Audio resultieren die Verteilungen aus Abbildung 5-18 und Abbildung 5-19.

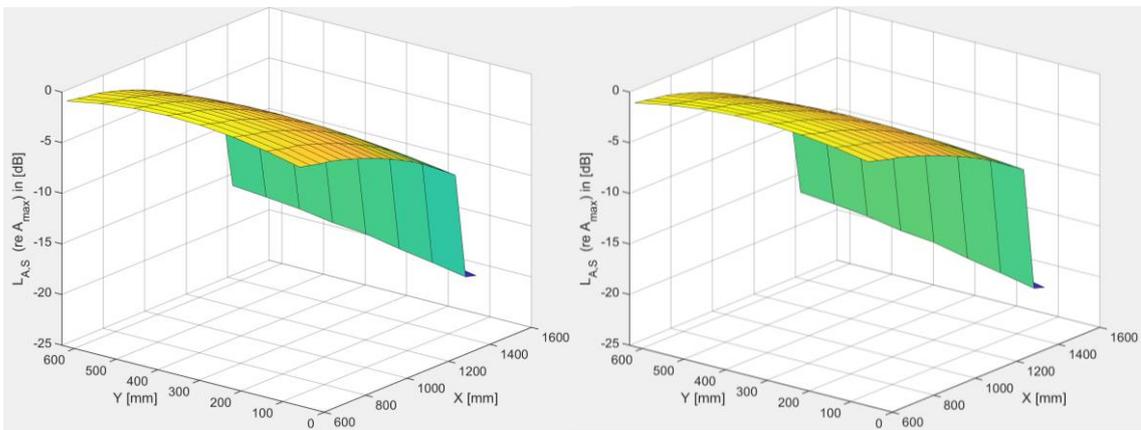


Abbildung 5-18: Simulierte Amplitudenpegelverläufe in der Ebene E1 mit dem Plug-in Resonance Audio; **linker Verlauf:** $f_{m_1,K} = 180$ Hz; **rechter Verlauf:** $f_{m_2,K} = 170$ Hz:

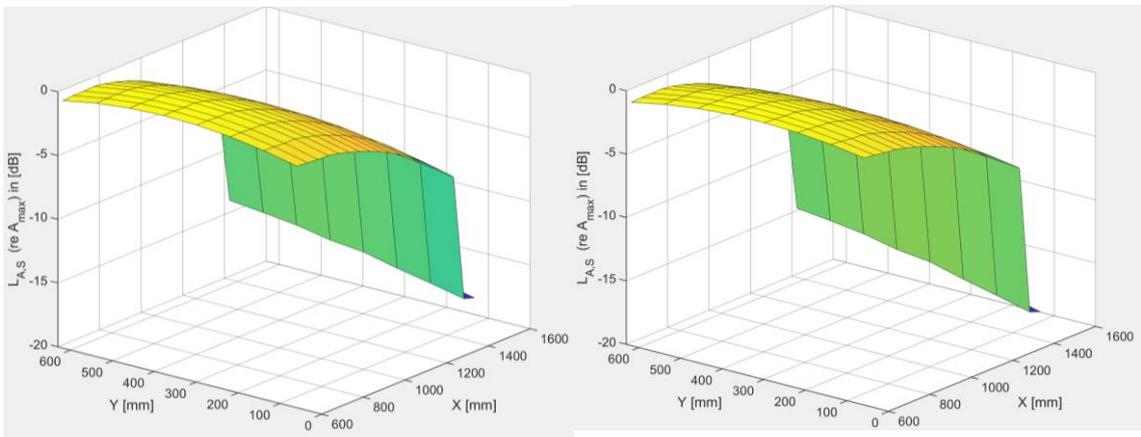


Abbildung 5-19: Simulierte Amplitudenpegelverläufe in der Ebene E2 mit dem Plug-in Resonance Audio; **linker Verlauf:** $f_{m_1,K} = 180$ Hz ; **rechter Verlauf:** $f_{m_2,K} = 170$ Hz

Die Verläufe zeigen einen stetig abnehmenden Amplitudenpegel in x -Richtung. Die sprunghafte Verringerung der Amplitude im hinteren Bereich entsteht durch das

Überschreiten eines festgelegten Abstands zu der Schallquelle, was dazu führt, dass das Plug-in darauf mit einer abrupten Reduzierung der Tonausgabe reagiert. Man kann erkennen, dass die Flächenverläufe, unabhängig von der Messfrequenz oder Messebene, in Breitenrichtung parabelförmig sind und ein globales Maximum in der Mitte bei $x = 625$ mm aufweisen. Weiterhin zeigen die Pegelverläufe aus der Messebene E2 in der Nähe der Schallquelle eine geringere Abnahmerate in Längsrichtung. Zusammen mit dem parabelförmigen Verlauf entlang der y -Koordinate deuten diese Erkenntnisse darauf hin, dass die Abnahme der Amplitude mit der Entfernung zu der Schallquelle zusammenhängt und nicht von resonanzbedingten Effekten beeinflusst wird.

Die Ergebnisse der Messungen mit dem Steam Audio Plug-in können der Abbildung 5-20 und Abbildung 5-21 entnommen werden.

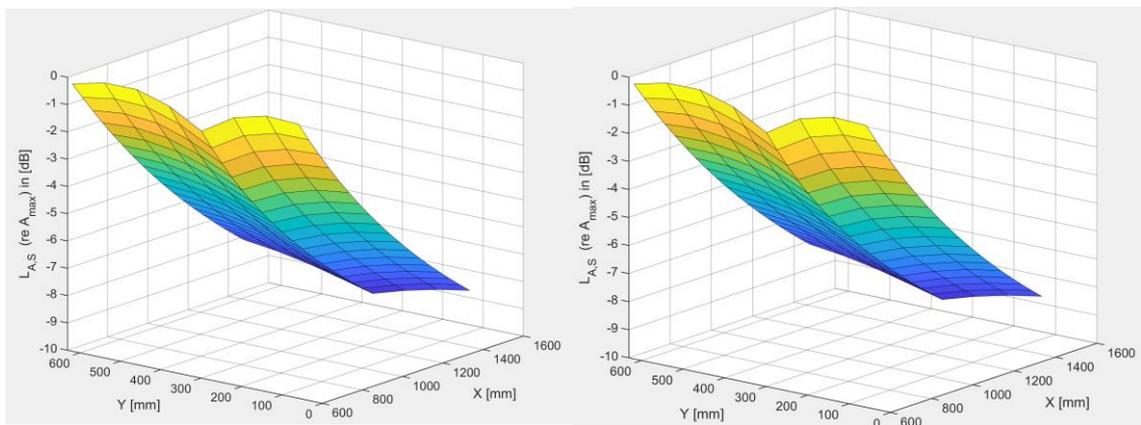


Abbildung 5-20: Simulierte Amplitudenpegelverläufe in der Ebene E1 mit dem Plug-in Steam Audio; **linker Verlauf:** $f_{m_1,K} = 180$ Hz; **rechter Verlauf:** $f_{m_2,K} = 170$ Hz

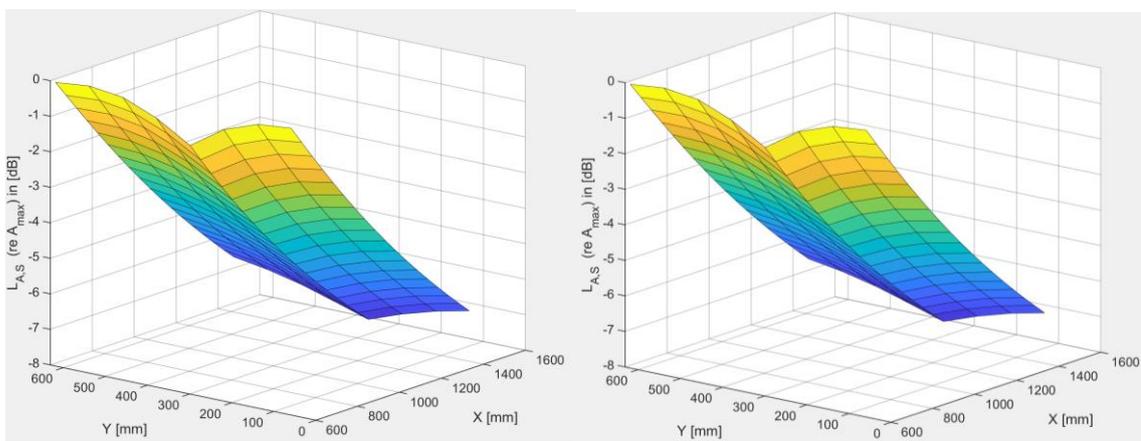


Abbildung 5-21: Simulierte Amplitudenpegelverläufe in der Ebene E2 mit dem Plug-in Steam Audio; **linker Verlauf:** $f_{m_1,K} = 180$ Hz; **rechter Verlauf:** $f_{m_2,K} = 170$ Hz

Die Pegelabnahme mit wachsendem Abstand zu dem Lautsprecher ist auch in diesen Messergebnissen zu sehen. Auch wenn die Messdaten keinen abrupten Amplitudenabfall mehr zeigen, kann ein unerwartetes Verhalten in der vierten Messbahn ($y = 311 \text{ mm}$) erkannt werden, in der die Amplituden einen niedrigeren Wert als in den restlichen Messbahnen aufweisen. Dieses Verhalten wird bei jeder Frequenz und Messebene beobachtet, sodass es nicht auf eine fehlerhafte Messung zurückgeführt werden kann.

Die Pegelabnahme in Längsrichtung ist im Durchschnitt mit der Pegelabnahme aus der Messung mit Resonance Audio vergleichbar.

6. Vergleich der Messergebnisse der Schallfeldmessungen

Das Ziel dieses Kapitels ist eine Aussage über die Einsetzbarkeit der analysierten Audio Plug-ins für die Wiedergabe physikalisch-korrekt akustischer Umgebungen zu formulieren. Dabei werden die betrachteten Umgebungen anhand der Messdaten verglichen. Die Werte aus der realen Umgebung dienen bei dem Vergleich als Referenz. Aus der Messung der Frequenzantwort der jeweiligen Umgebungen kann ermittelt werden, ob Resonanzereignisse im Schallausbreitungsmedium auftreten und wenn es der Fall ist, ob die Modenverteilungen über das Frequenzspektrum Ähnlichkeiten aufweisen. Der Vergleich der durchgeführten Kartierungsmessungen soll eine Aussage über die Druckverteilung im Messbereich liefern. Die Schallpegelverteilungen der in Abschnitt 5.3 gezeigten Diagramme, werden auch mit dem Modal Assurance Criterion untersucht. Mit diesem Verfahren wird eine mathematische Analyse für die Similarität der räumlichen Ausbreitung durchgeführt, sodass die Objektivität des Vergleiches sichergestellt wird.

Der Vergleich der realen Messergebnisse mit den FEM-Berechnungen soll zeigen, wie gut die Ergebnisse des erstellten FE-Modells die Realität widerspiegeln. Der Fokus der Arbeit liegt nicht in der Optimierung des Berechnungsmodells, sodass die Abweichungen minimiert werden. Es wird beabsichtigt, ein FEM-Modell zu erstellen, dessen Resultate mit den realen Ergebnissen vergleichbar sind.

Die Berechnung des MAC-Wertes erfolgt mit der Gleichung (2-27) aus Kapitel 2.2. Für die Ermittlung des Faktors werden die komplexen Amplituden der Kartierungspunkte in die Eigenvektoren ψ eingesetzt. Da die komplexen Amplituden Informationen zu den Phasenverschiebungen enthalten, muss darauf geachtet werden, dass sich die Phaseninformationen aller Messpunkte auf den gleichen Referenzpunkt beziehen. Die Formulierung der komplexen Amplitude wurde bereits in Kapitel 2.1, Gleichung (2-18) präsentiert. Anhand der Eulerschen Formel lässt sich diese folgendermaßen umschreiben (de Vries, 2012):

$$\underline{p}(\vec{r}) = \hat{p}(\vec{r}) \cdot (\cos(\varphi) + i \sin(\varphi)) \quad (6-1)$$

Um die komplexen Amplituden mit dem MAC-Verfahren miteinander zu vergleichen, muss die Phase φ den gleichen Bezugspunkt in alle Umgebungen haben. Damit der neue Referenzpunkt in jeder untersuchten Umgebung gleich ist, wird der erste Messpunkt mit

den Koordinaten $x = 0,625$ m und $y = 0,071$ m in den jeweiligen Messebenen als Bezug gewählt. Die Neureferenzierung der gemessenen Phasen aus der realen Umgebung lässt sich ohne weitere Berechnungen durchführen. Für die Neureferenzierung der Phasen aus der FEM- sowie virtuellen Umgebung, muss diese zuerst aus den komplexen Amplituden extrahiert werden. Mit folgender Formel lässt sich die Phase ermitteln (de Vries, 2012):

$$\varphi_i(\vec{r}) = \tan^{-1} \left(\frac{\text{Im} \{ \underline{p}_i(\vec{r}) \}}{\text{Re} \{ \underline{p}_i(\vec{r}) \}} \right) \quad (6-2)$$

Die neu referenzierten Phaseninformationen und die gemessenen Amplituden werden in Gleichung 6-1 eingesetzt, um die komplexen Amplituden für die MAC-Analyse zu ermitteln.

6.1. Vergleich der realen und der virtuellen Umgebung

Der Vergleich der Ergebnisse der realen und virtuellen Messungen liefert keine Ähnlichkeiten im Frequenzgang der beiden Umgebungen. Die Raummoden (1-0-0) und (2-0-0), welche man in der realen Umgebung bei den Eigenfrequenzen $f_{E_{1,M}} = 105$ Hz und $f_{E_{2,M}} = 180$ Hz beobachten kann, sind aus den Messergebnissen beider Plug-ins nicht ersichtlich. Das Resultat der Aufnahmen mit Resonance Audio und mit Sperrholz als Wandmaterial zeigen einen leichten Anstieg der Amplitudenpegel zwischen den Frequenzen 50 Hz und 100 Hz. Diese Verhaltensweise wird aber nicht mit der Bildung einer Raummode, sondern mit einer Verringerung der frequenzabhängigen Materialabsorption zwischen 50 Hz und 100 Hz assoziiert.

Betrachtet man die maximale Ausdehnung der gemessenen Pegel im untersuchten Frequenzband, so kann man aus den Messergebnissen des realen Mock-ups eine Differenz zwischen dem höchsten und dem niedrigsten Pegel von etwa 37 dB beobachten. Dieser Wert entspricht annähernd einer Erhöhung des Schalldruckes um den Faktor 100. Im virtuellen Modell beträgt die gleiche Differenz, unabhängig vom eingesetzten Plug-in, einen maximalen Wert von etwa 4,3 dB. Der Vergleich zeigt einen deutlichen Unterschied in der Frequenzantwort des realen und virtuellen Mock-ups.

Nach der visuellen Gegenüberstellung der Kartierungsergebnisse für alle Messfrequenzen und für alle Messebenen werden keine Ähnlichkeiten zwischen der virtuellen und realen longitudinalen Ausbreitung identifiziert. Während im realen Verlauf einen Anstieg bis zum Amplitudenmaximum gefolgt von einer Verringerung der Amplitude ab dem maximalen Wert zu sehen ist, erfährt der Pegel in der virtuellen Realität sein Maximum in den ersten Messpunkten vor der Schallquelle, gefolgt von einer kontinuierlichen Abnahme. Dieses Verhalten kann sowohl bei dem Resonance Audio als auch bei dem Steam Audio Plug-in erkannt werden. Das erste Plug-in induziert außerdem eine unrealistische abrupte Pegelreduzierung im hinteren Bereich des Mock-ups, welche keinen Bezug zu den realen Ereignissen hat.

Das Plug-in Resonance Audio zeigt eine Ähnlichkeit mit dem realen Pegelverlauf entlang der Y-Achse. Auch wenn die Verteilungen der gemessenen Amplituden eine Verminderung an den Rändern des Messgebietes erfahren, ist der Verlauf nahezu konstant.

Im Gegensatz dazu, induziert das Audiomodul von Steam wechselnde Amplituden in der Audioausgabe entlang der Breitenachse des Modells. Besonders in der vierten Messachse ($y = 311 \text{ mm}$) zeigen die simulierten Amplituden einen starken Unterschied zu den Amplituden aus den benachbarten Messachsen. Dieses Verhalten kann mit den realen Messungen nicht bestätigt werden.

Die Analyse mit dem MAC wird für die Matrizen ψ_M und ψ_S durchgeführt. Die Spalten der Matrizen enthalten die komplexen Amplituden $\underline{p}_M(\vec{r})$, beziehungsweise $\underline{p}_S(\vec{r})$ der realen und der virtuellen Umgebung. In jeder Spalte werden die komplexen Werte für eine Messung in einer Ebene und bei einer Messfrequenz $f_{m_i,k}$ eingetragen. Die Tabellen 6.1 und 6.2 präsentieren die berechneten MAC-Faktoren für alle Messfrequenzen $f_{m_i,k}$ und Kartierungsebenen. Die Informationen aus den Tabellen werden in den Abbildung 6-2 und Abbildung 6-1 dargestellt.

Tabelle 6.1: MAC-Analyse zwischen der realen Umgebung und dem Resonance Audio Plug-in

Unity	Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$
Reale Umgebung				
Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	$2,35 \cdot 10^{-4}$	$1,71 \cdot 10^{-5}$	$4,03 \cdot 10^{-4}$	$5,31 \cdot 10^{-5}$
Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	$2,36 \cdot 10^{-4}$	$8,75 \cdot 10^{-5}$	$1,10 \cdot 10^{-4}$	$5,07 \cdot 10^{-5}$
Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	$1,39 \cdot 10^{-4}$	$3,63 \cdot 10^{-5}$	$4,13 \cdot 10^{-4}$	$7,40 \cdot 10^{-5}$
Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$	$2,12 \cdot 10^{-4}$	$5,38 \cdot 10^{-5}$	$1,12 \cdot 10^{-4}$	$4,83 \cdot 10^{-5}$

Tabelle 6.2: MAC-Analyse zwischen der realen Umgebung und dem Steam Audio Plug-in

Unity	Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$
Reale Umgebung				
Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	$1,20 \cdot 10^{-3}$	$5,61 \cdot 10^{-6}$	$1,20 \cdot 10^{-3}$	$5,35 \cdot 10^{-5}$
Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	$1,10 \cdot 10^{-3}$	$3,10 \cdot 10^{-5}$	$1,10 \cdot 10^{-3}$	$6,11 \cdot 10^{-5}$
Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	$1,30 \cdot 10^{-3}$	$4,60 \cdot 10^{-6}$	$1,10 \cdot 10^{-3}$	$2,98 \cdot 10^{-5}$
Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$	$1,10 \cdot 10^{-3}$	$1,30 \cdot 10^{-5}$	$8,21 \cdot 10^{-4}$	$9,21 \cdot 10^{-5}$

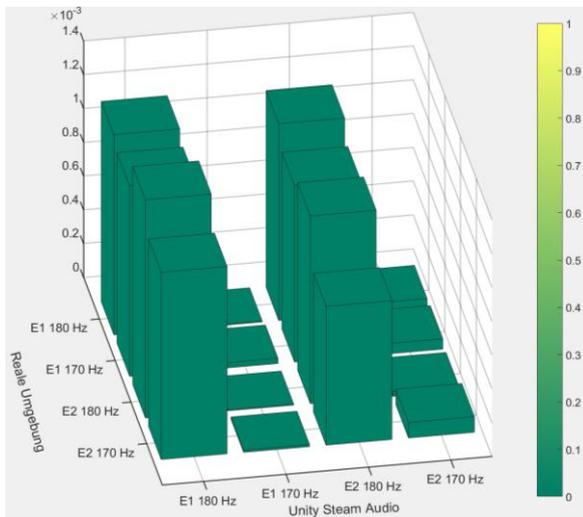


Abbildung 6-1: Graphische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung-Steam Audio Plug-in

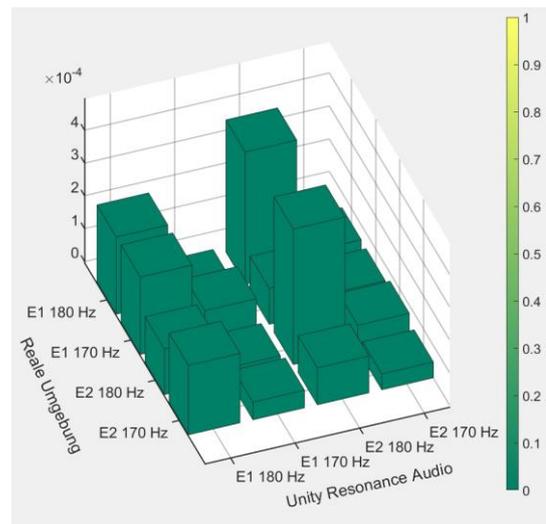


Abbildung 6-2: Graphische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung-Resonance Audio

Die Ermittlung des MAC-Wertes bestätigt die zuvor erkannten Unterschiede zwischen den realen und den virtuellen Verläufen. Vergleicht man die in hellblauer Farbe markierten Felder aus den vorherigen Tabellen, so sieht man, dass für keine der analysierten Amplitudenverteilungen eine Ähnlichkeit existiert.

Fasst man die Ergebnisse des Vergleiches der Frequenzantworten und der Differenzen der Amplitudenverteilungen, welche anhand der Darstellungen identifiziert wurden und der Analyse mit dem MAC-Verfahren zusammen, so wird deutlich, dass die virtuelle Umgebung keine mit der Realität vergleichbare akustische Simulation des Mock-ups mit den eingesetzten Audio Plug-ins im niederfrequenten Bereich realisiert.

6.2. Vergleich der realen Umgebung mit der FEM-Berechnung

Die Messung der Frequenzantwort des CRC-Mock-ups zeigt eine Pegelverteilung, die mit Hilfe des FEM-Programmes COMSOL Multiphysics reproduziert wurde. Bis zu einer Frequenz von 250 Hz lassen sich in beiden Verläufen die Bildung von zwei Raummoden erkennen. Die Resonanzfrequenz der ersten Mode beträgt anhand der realen Messergebnisse den Wert $f_{E_1,M} = 105$ Hz. In COMSOL erscheint die erste Resonanz schon bei $f_{E_1,B} = 87,6$ Hz, woraus ein Unterschied von 17,4 Hz berechnet werden kann. Betrachtet man die nächste Eigenschwingung des Raumes, so reduziert sich der Unterschied zwischen der FEM-Berechnung und den realen Messdaten. Für die Raummode (2-0-0) beträgt die Differenz somit nur noch 5,6 Hz. Die deutlich niedrigere Abweichung ermöglicht einen besseren Vergleich der Raummode (2-0-0) aus der realen und FEM Umgebung. Obwohl die Frequenzantworten einen beinahe unregelmäßigen Verlauf ab $f_m \approx 250$ Hz haben, können einige Gemeinsamkeiten identifiziert werden. Es können beispielweise zwei Extremwerte im Bereich $f_m \approx 250$ Hz und $f_m \approx 420$ Hz beobachtet werden. Die Messergebnisse aus der realen Umgebung zeigen einen starken Anstieg der Schalldruckpegel bei den Frequenzen $f_m \approx 288$ Hz, $f_m \approx 320$ Hz, $f_m \approx 435$ Hz und $f_m \approx 490$ Hz. Die Simulation mit COMSOL liefert auch einen Zuwachs der Amplituden bei den gleichen Frequenzen.

Obwohl die Verläufe für die jeweiligen Mikrofone dem gleichen Trend im Bereich der niedrigen Frequenzen folgen, scheint die Amplitude des rechten Messmikrofons in der realen Messung einen niedrigeren Wert als die Amplitude des anderen Mikrofons zu

haben. Die FEM-Berechnung ergibt die gleiche Amplitude für beide Messpunkte bis zu $f_m \approx 200$ Hz.

Vergleicht man die Werte der Schalldruckpegel, so kann man sehen, dass die Amplituden aus dem FEM-Modell um etwa 20 dB geringer sind. Dadurch dass das Übertragungsverhalten des realen Lautsprechers bei der harmonischen Anregung in der FEM-Umgebung berücksichtigt wurde, lässt sich erschließen, dass sich der Unterschied aus den eingestellten Absorptionsgraden der Volumenabgrenzungen und der Schalldämpfung des Ausbreitungsmediums ergibt.

Die Unterschiede im Frequenzgang zwischen den beiden Messungen können durch eine Verbesserung des FEM-Modells reduziert werden. Es wäre zum Beispiel möglich, das Plexiglasfenster am Ende des Mock-ups zu modellieren und den entsprechenden Absorptionsgrad in die Berechnung einzubeziehen. Des Weiteren könnte eine Studie zur genaueren Bestimmung der Wandabsorptionsgrade durchgeführt werden oder sogar diese messtechnisch in einem Hallraum bestimmen. Solche Verfeinerungen des Modells erzeugen eine sehr realitätstreue Nachbildung des CRC Mock-ups, würden aber den Rahmen dieser Arbeit sprengen. Für das vorgenommene Ziel der Arbeit bietet das hier präsentierte Modell eine ausreichende Genauigkeit im Frequenzgang.

Der Vergleich der Kartierungsmessungen zeigt viele Gemeinsamkeiten. Beiden Kartierungen reflektieren den Bereich des Druckmaximums einer stehenden Welle mit ebenen Wellenfronten. Die unterschiedlichen Wellenlängen aufgrund der abweichenden Messfrequenzen können in beiden Messumgebungen erkannt werden. Während die Kartierung im FEM-Mock-up für alle Frequenzen und Ebenen einen konstanten Verlauf in y -Richtung zeigt, erreicht man in der realen Umgebung nur bei $f_{m_2,K} = 170$ Hz eine ähnliche Amplitudenverteilung. Die Ausnahme bilden die Messpunkte mit der Koordinate $x = 1475$ mm am Ende des Messgebietes, welche entlang der Breitenachse eine sprunghafte Verteilung besitzen. Bei der Messfrequenz $f_{m_1,K} = 180$ Hz erscheinen lokale Steigerungen der Schalldruckpegel entlang des gemessenen Verlaufes, insbesondere im Bereich der maximalen Amplituden. Die letzten Messpunkte weisen ein ähnliches Verhalten zu den Berechnungsergebnissen aus COMSOL auf.

Um die Ähnlichkeiten der Verläufe mathematisch zu untersuchen, wird die MAC-Analyse durchgeführt. Ähnlich wie die Matrix ψ_M wird auch die Matrix ψ_B für das FEM-Modell

erstellt. Die Phaseninformationen der komplexen Amplituden wurde neu referenziert, so dass diese den gleichen Referenzpunkt wie die Daten der Matrix ψ_M haben. Tabelle 6.3 enthält die mit der Gleichung (2-27) errechneten MAC-Faktoren zwischen der realen und der FEM-Umgebung. Die graphische Darstellung der Faktoren kann man der Abbildung 6-3 entnehmen.

Tabelle 6.3: MAC-Analyse zwischen der realen Umgebung und dem FEM-Modell

FEM \ Reale Umgebung	Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$
Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	0,96	0,97	0,94	0,96
Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	0,94	0,98	0,92	0,98
Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	0,97	0,97	0,95	0,96
Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$	0,92	0,97	0,89	0,97

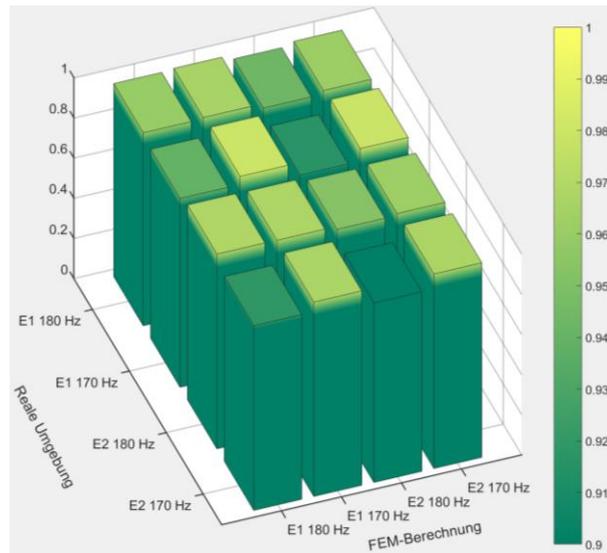


Abbildung 6-3: Graphische Darstellung der MAC-Faktoren für den Vergleich Reale Umgebung - FEM-Modell

Die MAC-Analyse ergibt einen hohen Ähnlichkeitsgrad zwischen den betrachteten Umgebungen. Alle Werte über 0,9 deuten auf eine hohe Übereinstimmung der Amplitudenverteilungen der Kartierungen. Relevant für den Vergleich sind die in blau markierten Felder aus Tabelle 6.3. Alle Faktoren in diesen Felder haben einen Wert größer oder gleich 0,95, was bedeutet, dass die räumlichen Amplitudenverteilungen sich sehr stark ähneln.

Der Vergleich zwischen der realen Umgebung und der FEM-Berechnung des CRC-Modells hat ein zufriedenstellendes Ergebnis geliefert. Die Frequenzgänge präsentieren ein vergleichbares Verhalten, welches in der Zukunft bei Bedarf noch verbessert werden kann. Mit den Kartierungsmessungen konnte der gleiche Bereich einer stehenden Welle erfasst werden, was dazu führt, dass die räumliche Verteilung des Schallfelds sowohl im realen als auch im modellbasierten Mock-up ähnlich ist. Dies konnte auch anhand einer MAC-Analyse bestätigt werden.

6.3. Erkenntnisse

Anhand der durchgeführten Analysen lässt sich die Einsatzbarkeit, der im Kapitel 3 präsentierten Audio Plug-ins für Unity, bewerten. Es wurden Messungen in der realen Umgebung durchgeführt, um eine Referenz für einen Vergleich festzulegen. Das Modell aus der realen Welt wurde im Unity nachgebaut und anschließend wurden Messungen unter vergleichbaren Bedingungen mit den Resonance Audio und Steam Audio Plug-ins durchgeführt. Anhand der Auswertung der Messergebnisse, sowohl aus der realen als auch aus der virtuellen Umgebung können folgende Aussagen formuliert werden:

- Die Frequenzantwort der realen Umgebung kann nicht mit den Audio Plug-ins Resonance Audio und Steam Audio im tieffrequenten Bereich reproduziert werden.
- Die Ergebnisse der Kartierungsmessungen in den untersuchten Umgebungen stimmen nicht überein.

Aus diesen zwei Aussagen wird erkannt, dass eine realitätsnahe Akustik in der virtuellen Realität mit den präsentierten Plug-ins nicht erzeugt werden kann.

Die FEM-Berechnung des CRC-Mock-ups zeigt realitätsnahe Ergebnisse für die Frequenzantwort und Kartierung des gleichen Bereichs. Es existieren noch bemerkbare Unterschiede, nichtdestotrotz hat das präsentierte Modell ein großes Potenzial die Realität nachzubilden. Bei Bedarf kann es weiter verfeinert werden, um exaktere Ergebnisse zu liefern. Somit wird das erstellte FEM Modell für die Erzeugung einer realitätsnahen akustischen Umgebung in der Spiel-Engine Unity eingesetzt.

7. Verbesserung der Akustik in der virtuellen Umgebung

Der Vergleich zwischen den realen und den virtuellen Messergebnissen zeigt, dass mit den präsentierten Audio Plug-ins aus Unity keine mit der Realität vergleichbare Akustik bei tiefen Frequenzen erzeugt werden kann. Das Ziel dieses Kapitels ist die Formulierung, Implementierung und Verifikation einer Optimierung der Akustik in Unity, basierend auf dem erzeugten und analysierten FEM-Modell aus COMSOL Multiphysics. Im ersten Teil des Kapitels wird das angestrebte Konzept für die Verbesserung der Akustik präsentiert. Anschließend wird die Optimierungsstrategie eingesetzt und Kartierungsmessungen durchgeführt. Im letzten Teil des Kapitels werden die Messungen ausgewertet und mit den Messergebnissen aus COMSOL Multiphysics und der realen Umgebung verglichen.

7.1. Beschreibung des Verbesserungskonzeptes

Bevor das Verbesserungskonzept näher erläutert wird, müssen einige Rahmenbedingungen festgelegt werden:

- Die Verarbeitung der FEM-Berechnungen, sowie die Audiowiedergabe soll mit einem gängigen Rechner möglich sein. Damit möchte man eine flexible Implementierung sicherstellen. Ein System, welches eine komplexe Rechnerarchitektur oder mehrere Rechner braucht, würde die Einsetzbarkeit limitieren, was mit dem ursprünglich gesetzten Ziel nicht übereinstimmt. Weiterhin würde die Entwicklung eines solchen Systems den Rahmen dieser Arbeit sprengen.
- Es sollen zunächst nur einzelne Töne ausgegeben werden. Nach erfolgreicher Wiedergabe einzelner Töne, besteht die Möglichkeit in der Zukunft das akustische Umfeld mit komplexeren Signalen zu simulieren.
- Obwohl die Erzeugung eines binauralen Erlebnisses einen realitätsnahen Hörindruck erzeugen würde, ist das in dieser Arbeit nicht angestrebt. Es würde keinen Vergleich zu den anderen Ergebnissen stattfinden können, somit würde eine Verifizierung fehlschlagen. Der Fokus wird zunächst auf die physikalisch korrekte, mit der Realität vergleichbaren Akustik in VR gelegt.

Die vorgeschlagene Verbesserung der Akustik in der virtuellen Realität wird erreicht, indem die Audioausgabe auf Basis der Ergebnisse der FEM-Berechnungen erfolgt und nur der visuelle Teil der VR-Simulation von Unity übernommen wird. Mithilfe von MATLAB wird ein Algorithmus entwickelt, welches ein Audiosignal aus den FEM-Berechnungsdaten erzeugt. Dieses Signal wird anschließend als Ton wiedergegeben.

Damit die auditiven Informationen mit den unterschiedlichen Positionen des virtuellen Empfängers aus Unity synchronisiert werden, müssen seine Positionsdaten extrahiert und dem externen Audioalgorithmus zur Verfügung gestellt werden.

Anhand der Abbildung 7-1 lässt sich das vorgeschlagene Konzept zur Verbesserung der Akustik in Unity erklären.

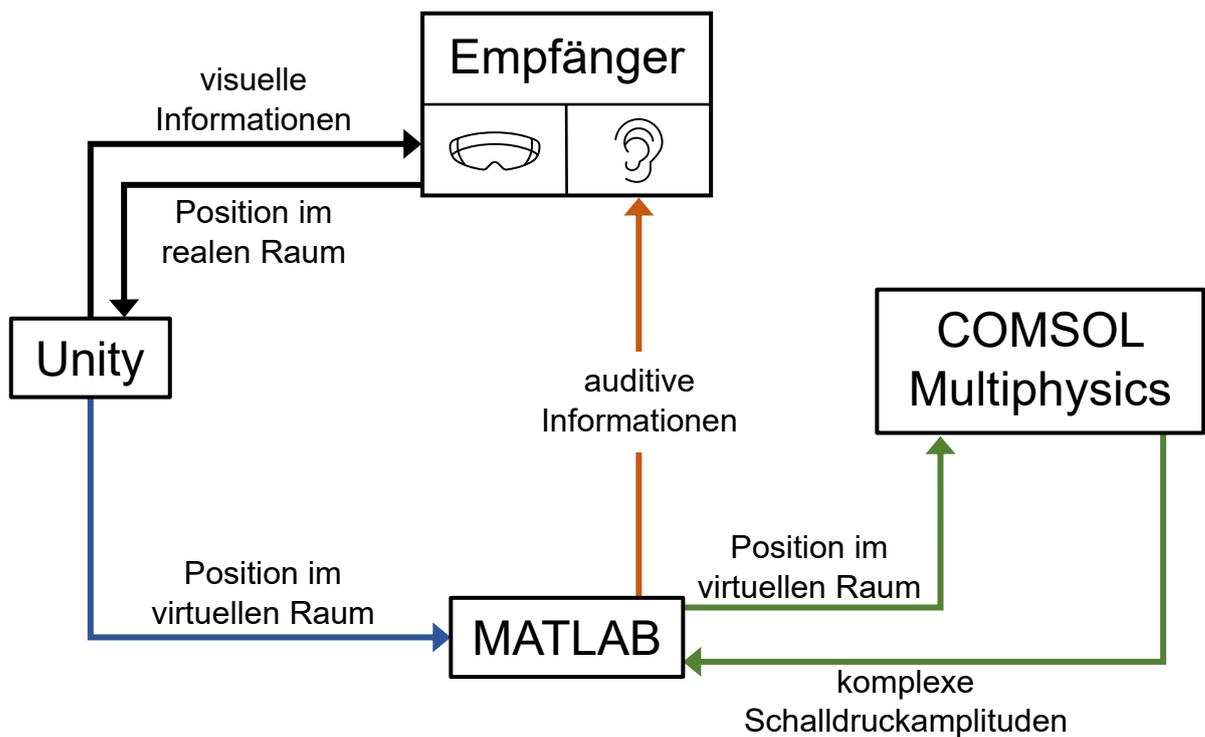


Abbildung 7-1: Konzept zur Verbesserung der Akustik in VR

Die reale Position des Empfängers wird von den eingebauten Lagesensoren der VR-Brille in Echtzeit gemessen und an die Spiel-Engine weitergeleitet. Die internen Algorithmen aus Unity wandeln die reale Position in eine virtuelle um, mit der die visuellen Informationen für den Benutzer generiert werden. Diese werden an die VR-Brille gesendet.

Die normale Funktionsweise sieht auch eine Übertragung von auditiven Informationen an den Empfänger vor, jedoch wird diese, wie weiter oben erläutert, zu Optimierungszwecken deaktiviert.

Im nächsten Schritt werden die virtuellen Positionsdaten für die Erzeugung der akustischen Informationen ausgelesen und an den MATLAB-Algorithmus gesendet. Die Verbindung und der Datentransfer zwischen Unity und MATLAB erfolgen über ein TCP/IP Protokoll. Der Vorteil des TCP/IP Protokolls ist die Möglichkeit der Verbindung mehrerer Rechner aus dem gleichen Netz.

Somit besteht keine Notwendigkeit, dass Unity und MATLAB auf dem gleichen Rechner ausgeführt werden müssen, um einen Datentransfer zu realisieren, jedoch ist es erforderlich, dass sie mit dem gleichen Netz verbunden sind.

Ein MATLAB-Skript enthält den Algorithmus für die Erzeugung der auditiven Informationen. Weiterhin dient es als die Verbindungsstelle zwischen der FEM-Berechnung und der virtuellen Umgebung. Die Verbindung zwischen MATLAB und COMSOL Multiphysics wird mit der in COMSOL eingebauten Funktion namens LiveLink™ for MATLAB® (LLfM) realisiert. Sie ist eine Schnittstelle, welche ebenfalls auf dem TCP/IP Protokoll basiert. Mithilfe von LLfM lässt sich eine Verbindung in Echtzeit zwischen MATLAB und COMSOL herstellen. LLfM ermöglicht nicht nur die Erstellung und Berechnung von FEM-Modellen im COMSOL aus der MATLAB Programmierumgebung, sondern bietet auch den Zugriff auf Berechnungsergebnisse an, welche von MATLAB als Variablen interpretiert werden und für beliebige Zwecke angewendet werden können (COMSOL, 2018). Die Eingangsposition aus Unity soll an Comsol weitergeleitet werden, um die benötigten Informationen zur Erzeugung der realitätsnahen akustischen Signale zu erhalten. Dieser Datenaustausch, welcher durch die grünen Pfeile in Abbildung 7-1 repräsentiert wird, soll ohne Unterbrechung ausgeführt werden.

Das Signal für die Audioausgabe hat als Grundlage eine Kosinusfunktion. Zusammen mit den Informationen aus COMSOL generiert diese eine harmonische Schwingung. Aus den komplexen Schalldruckamplituden $\underline{p}_i(\vec{r})$ werden die realen Amplituden $\hat{p}_i(\vec{r})$ und die Phasen $\varphi_i(\vec{r})$ mithilfe der Gleichungen (5-2) und (6-2) berechnet.

Mit den berechneten Amplituden und Phasen kann man die Funktion für die Signalgenerierung folgendermaßen definieren:

$$S_i(\vec{r}, t) = \hat{p}_i(\vec{r}) \cdot \cos(\omega t + \varphi_i(\vec{r})) \quad (7-1)$$

Die Kreisfrequenz ω kann mit Gleichung (2-12) aus der Frequenz berechnet werden. Mit der MATLAB-Funktion `sound(y,Fs)` wird das Signal y mit der Abtastfrequenz F_s über die Audiokarte des Rechners abgespielt werden (The MathWorks, Inc., 2022).

Somit können die Töne mit den Signalen $S_i(\vec{r}, t)$ erzeugt und für den Empfänger entlang des orangenen Pfades abgespielt werden.

7.2. Implementierung des Verbesserungskonzeptes

Die Implementierung des Verbesserungskonzeptes nach Kapitel 7.1 ist im Rahmen dieser Arbeit nur bedingt möglich. Aufgrund einer technischen Begrenzung ist die Realisierung der Verbindung über LLfM nicht möglich. Der Rechner, auf dem die FEM-Simulation durchgeführt wurde und der Rechner, auf dem die VR-Simulation und das MATLAB-Skript ausgeführt werden, befinden sich auf unterschiedlichen, inkompatiblen Netzwerken, sodass die Verbindung nicht möglich ist.

Damit die Implementierung für diese Arbeit weiterhin realisierbar ist, wird sie angepasst, indem man auf die Verbindung über LLfM verzichtet. Die FEM-Berechnungsergebnisse für das gesamte Model werden aus COMSOL exportiert und für das Hochladen in den MATLAB-Algorithmus vorbereitet (Anhang F, 14,15). Die importierten Variablen sollen nun von dem Algorithmus für die Signalgenerierung wie die über LLfM extrahierten Ergebnissen interpretiert werden. Somit ist es möglich, trotz einer fehlenden COMSOL-MATLAB-Verbindung, die Funktionsweise des MATLAB-Skriptes zu implementieren. (Anhang F, 16)

Die Einrichtung der TCP/IP-Verbindung zwischen Unity und MATLAB beruht auf einer im DigiNet.Air Projekt zur Messung der Positioniergenauigkeit von VR-Brillen vorgeleisteten Arbeit. Sie besteht aus zwei Komponenten, nämlich ein MATLAB-Skript und einen Unity-Code (Anhang G). Das MATLAB-Skript eröffnet einen TCP/IP Server und wartet auf die Positionsdaten des virtuellen Benutzers aus Unity. Der Unity-Code erzeugt einen TCP/IP Client, welcher die ausgelesenen Positionsdaten an den MATLAB-Server sendet,

(Bodolea, 2021). Mit dem Unity-Code ist man in der Lage die Positionsdaten jedes Objekts aus der Szene auszugeben, indem er mit dem Objekt verknüpft wird. Für die Implementierung des Verbesserungskonzeptes wird er mit dem virtuellen Messmikrofon verbunden.

Nun kann das virtuelle Mikrofon in der VR-Umgebung bewegt werden, seine Positionsdaten an MATLAB gesendet und eine auf die FEM-Berechnungen basierende, positionabhängige Tonausgabe erzeugt werden. Für der Verifikation der vorgestellten Optimierungen werden die Kartierungsmessungen mit dem neuen Konzept bei $f_{m_1,K} = 180 \text{ Hz}$ und $f_{m_2,K} = 170 \text{ Hz}$ und in den Messebenen E1 und E2 wiederholt.

Da die Tonausgabe über die Audiokarte des Rechners erfolgt, werden die Messungen erneut mit Audacity® durchgeführt. Die Auswertung der Messungen wird mit einem MATLAB-Skript durchgeführt (Anhang F, 3), welcher auf dem Skript der Kartierungen mit den Audio Plug-ins aus Kapitel 5.3 basiert. Abbildung 7-2 und Abbildung 7-3 präsentiert die Ergebnisse der Kartierungen mit der verbesserten Akustik in VR.

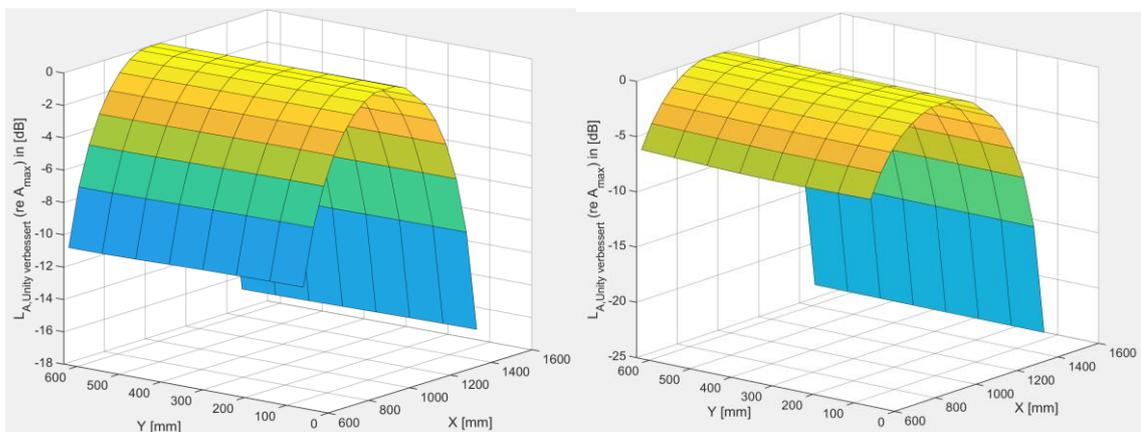


Abbildung 7-2: Simulierte Amplitudenpegelverläufe mit dem Verbesserungskonzept in der Ebene E1;
linker Verlauf: $f_{m_1,K} = 180 \text{ Hz}$; **rechter Verlauf:** $f_{m_2,K} = 170 \text{ Hz}$

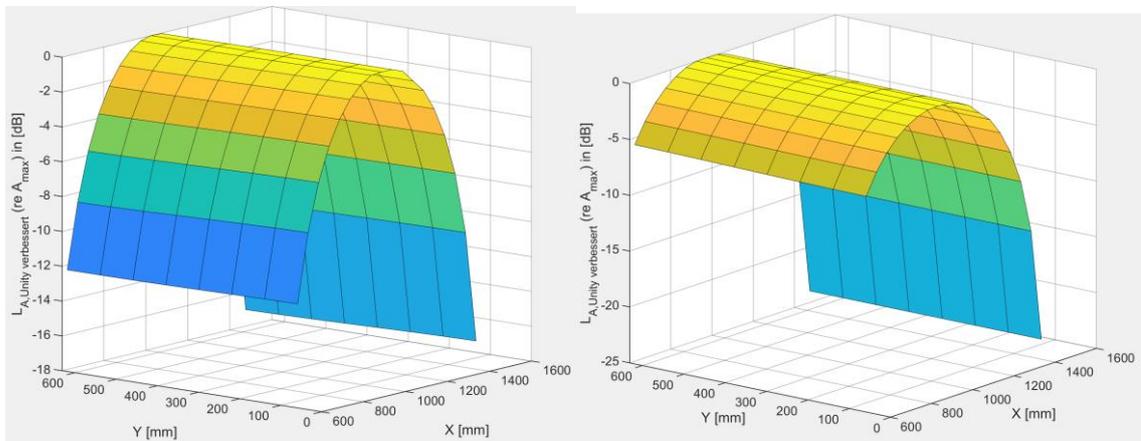


Abbildung 7-3: Simulierte Amplitudenpegelverläufe mit dem Verbesserungskonzept in der Ebene E2;
linker Verlauf: $f_{m_1,K} = 180 \text{ Hz}$; **rechter Verlauf:** $f_{m_2,K} = 170 \text{ Hz}$

Die dargestellten Verläufe der Amplitudenpegel zeigen eine parabelförmige Verteilung der gemessenen Amplituden. Sowohl bei $f_{m_2,K} = 170 \text{ Hz}$ als auch bei $f_{m_1,K} = 180 \text{ Hz}$ wird ein Druckmaximum erfasst. Betrachtet man nur die Verteilungen der Amplituden bei der Messfrequenz $f_{m_1,K} = 180 \text{ Hz}$, so können geringe Unterschiede zwischen den Amplituden in den Messebenen erkannt werden. Für die zweite Messfrequenz tritt dieser Unterschied nicht auf. Erneut können die unterschiedlichen Wellenlängen anhand der Ausdehnung in x -Richtung erkannt werden. Die konstante Pegelverteilung entlang der Breitenachse deutet auf eine ebene Wellenfront hin.

7.3. Vergleich des Verbesserungskonzeptes mit der realen- und FEM-Umgebung

Im Rahmen dieses Abschnittes werden die Messergebnisse aus dem vorherigen Unterkapitel mit den zuvor ermittelten Kartierungsdaten aus der realen und der FEM-Umgebung verglichen. Es soll ermittelt werden, ob die Amplitudenverteilungen aus Unity verbessert wurden. Da das Konzept auf den FEM-Daten basiert, erwartet man eine hohe Ähnlichkeit zwischen diesen Modellen. Es ist aber trotzdem erforderlich, eine objektive Analyse durchzuführen, um eine faktenbasierte Aussage treffen zu können.

Der Vergleich wird anhand der Differenzen zwischen den Amplitudenpegeln $L_{A,FEM}$, $L_{A,real}$ und $L_{A,Unity\ verbessert}$ durchgeführt. Die Bezugspunkte der Pegel sind die globalen

Maximalwerte aus den jeweiligen Umgebungen. Des Weiteren wird die Ähnlichkeit zwischen Unity und den anderen zwei Umgebungen anhand einer MAC-Analyse bewertet.

Aus Gründen der Übersichtlichkeit werden hier nur zwei Diagramme mit den Differenzen der Amplitudenpegel gezeigt. Die restlichen Diagramme können dem Anhang H entnommen werden. Nichtsdestotrotz, werden alle Besonderheiten diskutiert und bei Bedarf auf ein angehängtes Diagramm verwiesen. Abbildung 7-4 präsentiert die Differenzen der Amplitudenpegel $L_{A,FEM} - L_{A,Unity\ verbessert}$ in der Ebene E1 und bei $f_{m_2,K} = 170\text{ Hz}$. Die Kurven repräsentieren die Unterschiede entlang der Messlinien in x -Richtung.

Die gezeigten Differenzen entlang der x -Koordinate sind nicht groß. Im Bereich des Druckmaximums kann eine starke Abnahme der Differenzen beobachtet werden, die für die Messlinie mit der Breitenkoordinate $y = 631\text{ mm}$ sogar den Wert null erreichen. Die Tatsache, dass eine Differenz von null gebildet wird, bedeutet, dass beide Modelle ein Maximum an der gleichen Stelle entlang der Messlinie erreichen. Sowohl in der Abbildung 7-4 als auch in den anderen Verläufen (vgl. Anhang H) beträgt der Unterschied der Amplitudenverläufe vor und nach dem Maximum einen konstanten Wert. Weil die Differenzen sehr klein sind, kann man behaupten, dass die räumlichen Verteilungen der Amplituden zwischen den Modellen nahezu gleich sind.

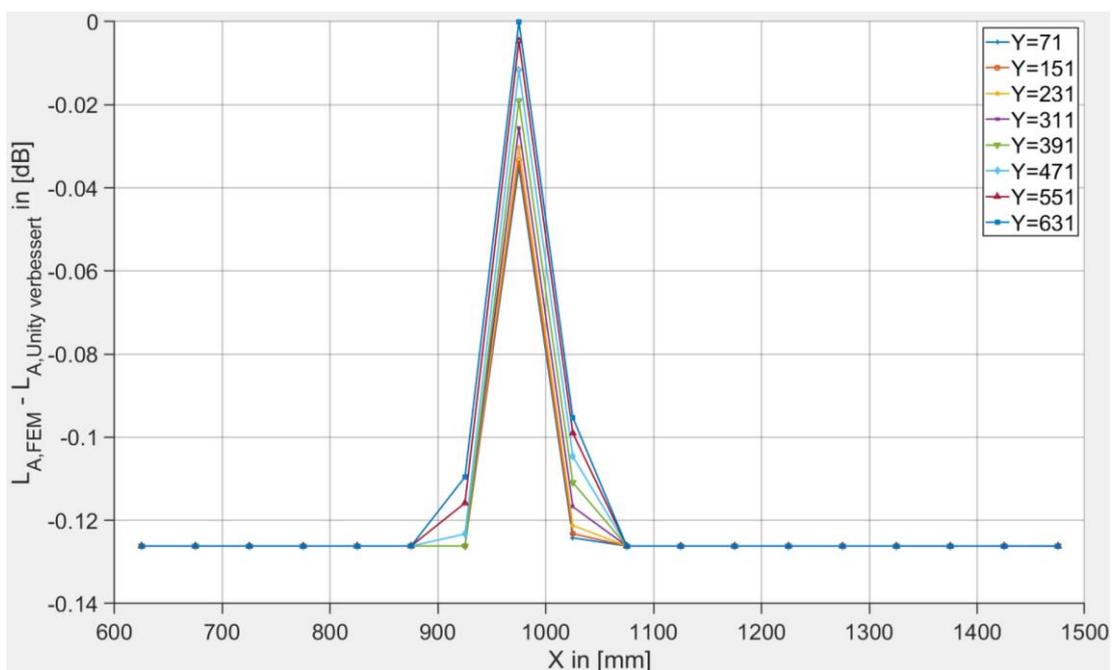


Abbildung 7-4: Amplitudenpegeldifferenzen zwischen COMSOL und Unity verbessert in E1, $f_{m_2,K} = 170\text{ Hz}$

Für den Vergleich der Differenzen der Amplitudenpegel in der realen und der verbesserten virtuellen Umgebung kann Abbildung 7-5 betrachtet werden. Diese enthält die Informationen für die Ebene E2 und die Messfrequenz $f_{m_2,K} = 170 \text{ Hz}$. Die Kurven zeigen deutlich größere Unterschiede, vor allem im hinteren Bereich. Die langsam abfallenden Verläufe der Differenzen sind über die ganze Länge zusammengebündelt. Dies bedeutet, dass sie in Breitenrichtung keine starken Abweichungen aufweisen. Ein ähnlicher Verlauf lässt sich in der tieferen Messebene und mit der gleichen Messfrequenz erzeugen. Bei der höheren Messfrequenz zeigen die Kurven eine steilere Abnahme. Allerdings besitzen die Differenzen im vorderen und hinteren Bereich kleinere Werte (siehe Anhang H).

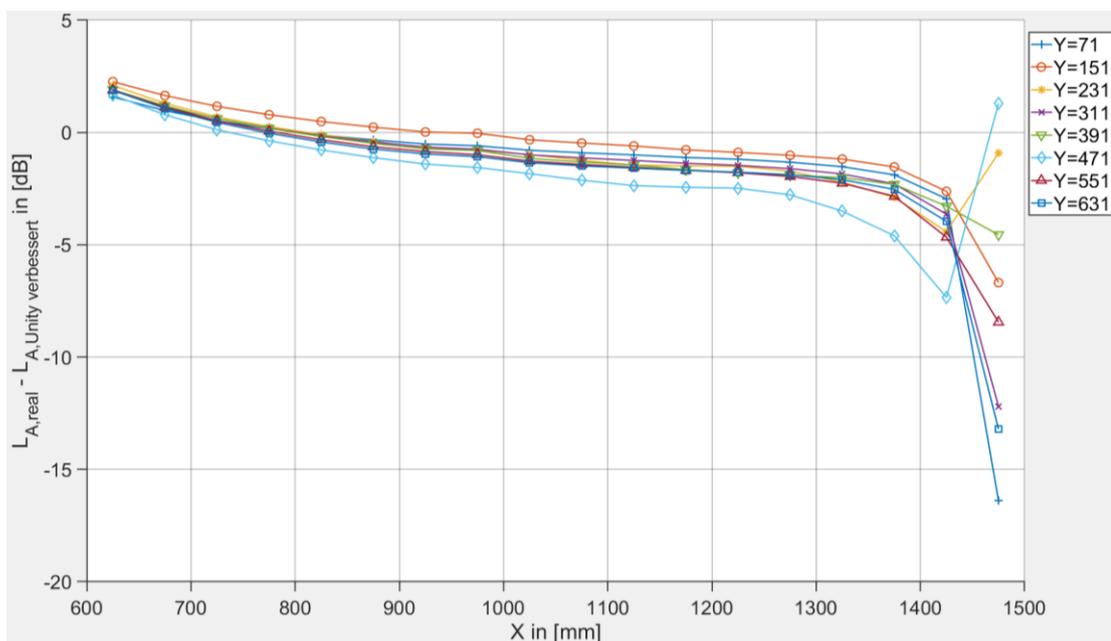


Abbildung 7-5: Amplitudenpegeldifferenzen zwischen der realen Umgebung und Unity verbessert in E2, $f_{m_2,K} = 170 \text{ Hz}$

Die Vergleiche der Messergebnisse aus der realen Umgebung mit der verbesserten virtuellen Umgebung zeigen zunächst große Unterschiede. Es werden erneut lokale Unterschiede von bis zu etwa 16 dB. Wie in dem Vergleich der realen und der FEM-Umgebung aus Kapitel 6.2 erläutert, hängen diese großen Differenzen von den gewählten Randbedingungen aus COMSOL und von den möglichen Messabweichungen aus dem realen Mock-up, welche zum Beispiel von der Bewegungseinrichtung erzeugt werden, ab.

Anhand der MAC-Analyse soll die Ähnlichkeit der Amplitudenverteilungen in dem Raum untersucht werden. Die Berechnung des MAC-Werts wird, wie in Kapitel 6 beschrieben, durchgeführt. Tabelle 7.1 enthält die MAC-Faktoren für den Vergleich zwischen dem COMSOL-Modell und der verbesserten Akustik in Unity.

Tabelle 7.1: MAC-Faktoren FEM-Unity verbessert

Unity verbessert FEM	Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$
Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	1	0,982	0,997	0,976
Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	0,982	1	0,97	0,997
Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	0,997	0,97	1	0,964
Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$	0,978	0,999	0,966	0,998

Die Faktoren in den markierten Feldern zeigen bei allen Messfrequenzen eine vollständige Übereinstimmung zwischen den einzelnen Kartierungen. Dieses Ergebnis wurde, wie in der Einführung des Unterkapitels erwähnt, erwartet. Die grafische Darstellung der MAC-Matrix wird in Abbildung 7-7 präsentiert.

Die Ergebnisse der MAC-Analyse zwischen der Realität und Unity können der Tabelle 7.2 entnommen werden.

Tabelle 7.2: MAC-Faktoren reale Umgebung-Unity verbessert

Unity verbessert Reale Umgebung	Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$
Ebene 1 $f_{m_1,K} = 180 \text{ Hz}$	0,975	0,980	0,965	0,975
Ebene 1 $f_{m_2,K} = 170 \text{ Hz}$	0,946	0,986	0,927	0,985
Ebene 2 $f_{m_1,K} = 180 \text{ Hz}$	0,992	0,99	0,985	0,985
Ebene 2 $f_{m_2,K} = 170 \text{ Hz}$	0,925	0,973	0,903	0,974

Auch für diesen Vergleich zeigen die MAC-Faktoren hohe Ähnlichkeitsgrade zwischen den akustischen Amplitudenverteilungen. Trotz der Differenzen in den Amplitudenpegeln verhalten sich die Schallfelder in ähnlicher Weise. Daraus resultiert, dass die Differenzen nur durch unterschiedliche Skalierungen der Amplituden hervorgerufen werden. Abbildung 7-6 beinhaltet die MAC-Faktoren in grafischer Form.

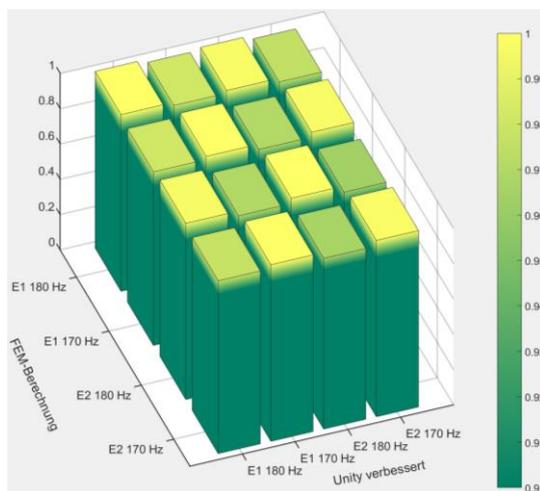


Abbildung 7-6: Grafische Darstellung der MAC-Faktoren für den Vergleich FEM-Unity verbessert

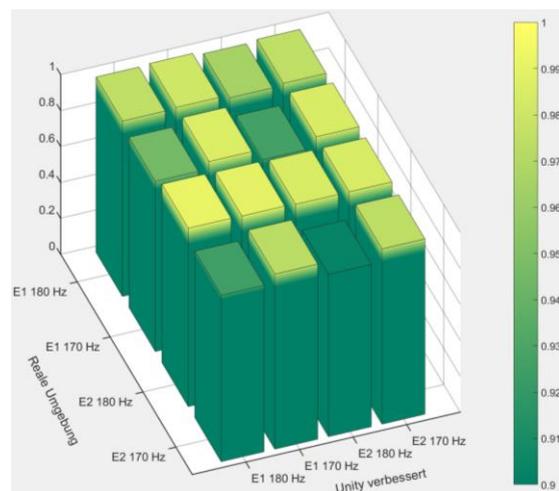


Abbildung 7-7: Grafische Darstellung der MAC-Faktoren für den Vergleich reale Umgebung-Unity verbessert

8. Zusammenfassung und Ausblick

Im Rahmen der verfassten Arbeit wurde ein System entwickelt, mit dem eine realitätsnahe akustische Wiedergabe eines Schallfeldes in der virtuellen Realität ermöglicht wird. Die akustische Simulation beruht auf numerischen Berechnungsergebnissen, welche mit der Finiten Elemente Methode in der Simulationssoftware COMSOL Multiphysics ermittelt wurden. Die Basis für die VR-Umgebung bietet die kommerzielle Spiel-Engine Unity an. Im Mittelpunkt des entwickelten Systems befindet sich ein MATLAB-Skript, welches die Verbindung der virtuellen Umgebung mit der numerischen Berechnung realisiert und die Wiedergabe des akustischen Signals übernimmt.

Eine vordefinierte reale Umgebung wurde in COMSOL Multiphysics und in der virtuellen Umgebung der Spiel-Engine Unity modelliert. Anhand der Messungen der Frequenzantworten bis zu der Frequenz $f = 500 \text{ Hz}$, sowie der Kartierungen der Schallfelder bei der Anregung mit den Frequenzen $f_{m_1,K} = 180 \text{ Hz}$ und $f_{m_2,K} = 170 \text{ Hz}$, konnte die räumliche Schalldruckverteilung erfasst werden. Damit konnte festgelegt werden, dass das erzeugte FEM-Modell das reale Schallfeld wiedergibt.

Des Weiteren wurde gezeigt, dass die kommerziellen Audio Plug-ins Resonance Audio und Steam Audio zur physikalisch-basierten Simulation des Schallfelds das reale Schallfeld bei den zuvor genannten Bedingungen nicht wiedergeben. Dieses Ergebnis begründet die Notwendigkeit der Entwicklung eines Systems zur realitätsnahen Wiedergabe eines niederfrequenten Schallfeldes in der virtuellen Realität.

Mit dem entwickelten System wurde das akustische Verhalten der realen Umgebung erfolgreich simuliert. Vergleiche der Messergebnisse aus der realen-, FEM- und optimierten virtuellen Umgebung haben bewiesen, dass die akustischen Simulationen und die numerischen Berechnungen nahezu identische Ergebnisse liefern. Zwischen der Schallfeldverteilung der realen Umgebung und dem virtuellen Modell liegen noch merkliche Unterschiede. Anhand einer mathematischen Analyse der Ähnlichkeit der räumlichen Amplitudenverteilungen konnte bestätigt werden, dass die Unterschiede nur in der Skalierung der Amplituden und nicht in der räumlichen Verteilung liegen. Da die abweichende Skalierung durch die Einstellungen der Randbedingungen in dem FEM-Modell erzeugt wurden, konnte die erfolgreiche Funktionsweise und Implementierung des Systems nachgewiesen werden.

Das präsentierte System zur akustischen Wiedergabe eines Schallfeldes in der virtuellen Umgebung zeigt ein großes Potenzial. Eine serienmäßige oder kommerzielle Anwendung lässt sich aber in dem aktuellen Entwicklungszustand nur bedingt durchführen. Durch zukünftige Entwicklungsschritte kann das System einen Reifegrad erreichen, der eine breitbandige Anwendung des Konzeptes ermöglicht.

Aus der Sicht des Verfassers sind folgende Weiterentwicklungen notwendig, um ein Grundkonstrukt zu erzeugen, mit dem zukünftige Anwendungen ermöglicht werden.

- Als Erstes ist die erfolgreiche Implementierung der Verbindung zwischen MATLAB und COMSOL über LiveLink for MATLAB von hoher Bedeutung. Die in dieser Arbeit präsentierte Funktionsweise simuliert nur die Verbindung zwischen den beiden Umgebungen. Für die vollständige Validierung und Verifizierung des Konzepts ist eine Interaktion zwischen MATLAB und COMSOL notwendig.
- Ein weiterer Entwicklungsschritt ist die Untersuchung einer multitonalen Anregung des Schallfelds. Anhand der Anregung mit einer einzigen Frequenz wurde bewiesen, dass diese erfolgreich simuliert und wiedergegeben werden kann. Es ist aber selten, dass man in alltäglichen Situationen einzelne Töne wahrnimmt. Somit ist die Betrachtung einer multitonalen Anregung von hoher Bedeutung.
- Bislang wurde die Bewegung des VR-Benutzers in einer kontrollierten Weise betrachtet, um die Reproduzierbarkeit der Messungen zu gewährleisten. Für eine Validierung der Funktionsweise und eine Erweiterung der Einsatzmöglichkeiten muss die realitätsnahe Wiedergabe auch bei einer freien Bewegung im virtuellen Raum stattfinden können.

Die Verbindung zu einer kommerziellen Spiel-Engine bietet die Möglichkeit an, eine Anwendung außerhalb der wissenschaftlichen Umgebung zu realisieren. Bereiche, wie zum Beispiel die Entwicklung von Flugzeugkabinen, können von dieser Lösung enorm profitieren, indem virtuelle Mock-ups mit simulierter Akustik erstellt werden können. Eine weitere Möglichkeit könnte die Implementierung von VR In-flight Entertainment Systemen mit einem hohen Grad an Immersion Dank der realitätsnahen Akustik sein. Die Verwendung eines numerischen Verfahrens für die Berechnung des akustischen Verhaltens kann eine limitierende Wirkung aufgrund hoher Rechenzeiten haben, ermöglicht aber eine hochqualitative Simulation des Schallfeldes, was für viele Anwendungen vorteilhaft ist.

Literaturverzeichnis

- Bodolea, M., 2021. *Konstruktion einer Messeinrichtung für die Bestimmung der Positioniergenauigkeit von VR-Systemen*, Hamburg: HAW Hamburg.
- Borß, C., Silze, A. & Martin, R., 2008. *Internet-Based Interactive Auditory Virtual Environment Generators*. Paris, Proceedings of the 14th International Conference on Auditory Display.
- COMSOL Multiphysics, 2020. *Acoustics Module Users's Guide*. s.l.:s.n.
- COMSOL, 2018. *LiveLink™ for MATLAB® User's Guide*. s.l.:s.n.
- de Vries, A., 2012. *Schwingungen und komplexe Zahlen*, Hagen: FH Sudwestfalen University of Applied Sciences.
- Deutsche Gesellschaft für Akustik e.V., 2006. *DEGA-Empfehlung 101*. Berlin: s.n.
- DEWESoft, 2022. *FFT Spectrum Analysis (Fast Fourier Transform)*. [Online]
Available at: <https://training.dewesoft.com/online/course/fft-spectral-analysis>
[Zugriff am 2022 06 15].
- Frank, M., Zotter, F. & Sontacchi, A., 2015. *Producing 3D Audio in Ambisonocs*. Hollywood, CA, AES 57th International Conference.
- Griesmer, A., 2014. *Size Parameters for Free Tetrahedral Meshing in COMSOL Multiphysics*. [Online]
Available at: <https://www.comsol.com/blogs/size-parameters-free-tetrahedral-meshing-comsol-multiphysics/>
[Zugriff am 23 05 2022].
- Hesselmann, N., 1987. *Digitale Signalverarbeitung*. Würzburg: Vogel Verlag.
- IRCAM, 2022. *IRCAM Forum - Spat*. [Online]
Available at: <https://forum.ircam.fr/projects/detail/spat/>
[Zugriff am 23 05 2022].
- Kim, H., Remaggi, L., Jackson, P. J. & Hilton, A., 2019. *Immersive Spatial Audio Reproduction for VR/AR Using Room Acoustic Modelling from 360° Images*. Osaka, 2019 IEEE Conference on Virtual Reality and 3D User Interfaces.
- Kuttruff, H., 2006. *Acoustics An introduction*. s.l.:Taylor & Francis.
- Kuttruff, H., 2017. *Room Acoustics*. Boca Raton: CRC Press .
- Lentz, T., Schröder, D., Vorländer, M. & Assenmacher, I., 2007. Virtual Reality System with Integrated Sound Field Simulation and Reproduction. *EURASIP Journal on Advances in Signal Processing*, Band 2007.

- Lerch, R., Sessler, G. M. & Wolf, D., 2009. *Technische Akustik Grundlagen und Anwendungen*. Heidelberg: Springer.
- Melcher, A., 2022. *amroc-THE Room Mode Calculator*. [Online]
Available at: <https://amcoustics.com/>
[Zugriff am 20 07 2022].
- Meyer, E. & Neumann, E.-G., 1979. *Physikalische und Technische Akustik*. Wiesbaden: Springer Fachmedien .
- Moeck, T. et al., 2007. *Progressive Perceptual Audio Rendering of Complex Scenes*. Seattle, Proceedings of the 2007 Symposium on Interactive 3D Graphics, SI3D 2007.
- Möser, M., 2015. *Technische Akustik*. Heidelberg: Springer.
- Müller, G. & Möser, M., 2004. *Taschenbuch der Technischen Akustik*. s.l.:Springer.
- National Aeronautics and Space Administration, 2022. *NASA Technology Transfer Program - Sound Lab (SLAB), Version 5*. [Online]
Available at: <https://software.nasa.gov/software/ARC-14991-1>
[Zugriff am 23 05 2022].
- Pastor, M., Binda, M. & Harcarik, T., 2012. *Modal Assurance Criterion*. s.l.:Elsevier Ltd.Selection.
- Pham Vu, T., Rivet, E. & H., L., 2018. *Sound Field Reconstruction In Low-Frequency Room Acoustics: A Benchmark Study With Simulation*. Lausanne: s.n.
- Poirier-Quinot, D., Katz, B. F. & Noistering, M., 2017. *EVERTims: Open source framework for realtime auralization in architectural acoustics and virtual reality*. Edinburgh, 20th International Conference on Digital Audio Effects (DAFx-17).
- Preuß, M., 2021. *Qualitätsprüfung von Plugins für akustische Raumsimulationen in Game-Engines*. Hamburg : Hochschule für Angewandte Wissenschaften Hamburg, Department Medientechnik, Fakultät DMI; Bachelorarbeit.
- Resonance Audio (a), 2022. *Fundamental Concepts*. [Online]
Available at: <https://resonance-audio.github.io/resonance-audio/discover/concepts>
[Zugriff am 12 06 2022].
- Resonance Audio (b), 2022. *Developer Guide for Resonance Audio for Unity*. [Online]
Available at: <https://resonance-audio.github.io/resonance-audio/develop/unity/developer-guide>
[Zugriff am 13 06 2022].
- Savioja, L., Huopaniemi, J., Lokki, T. & Väänänen, R., 1999. Creating Interactive Virtual Acoustic Environmants. *Journal of the Audio Engineering Society*, 47(9), pp. 675-705.

- Schröder, D. et al., 2010. *Virtual Reality System at RWTH Aachen University*. Melbourne, Proceedings of the International Symposium on Room Acoustics, ISRA 2010.
- Serafun, S. et al., 2018. Sonic Interactions in Virtual Reality: State of the Art, Current Challenges, and Future Directions. *IEEE Computer Graphics and Applications*, 38(2), pp. 31-43.
- Silze, A., Novo, P. & Strauss, H., 2004. *IKA-SIM: A System to Generate Auditory Virtual Environments*. Berlin, AES 116th Convention.
- Smardzewski, J. et al., 2015. Sound absorption of wood-based materials. *Holzforschung*, 69(4).
- Svensson, P. U., 2002. *Modelling Acoustic Spaces For Audio Virtual Reality*. Leuven, s.n.
- Technische Universität Chemnitz, 2022. *Audiotools*. [Online]
Available at: <https://www.tu-chemnitz.de/urz/archiv/kursunterlagen/multimedia/audio.html>
[Zugriff am 30 07 2022].
- The MathWorks, Inc., 2022. *MathWorks Help Center sound*. [Online]
Available at: <https://www.mathworks.com/help/matlab/ref/sound.html>
[Zugriff am 09 08 2022].
- Valve Corporation (a), 2021. *Steam Audio User's Guide*. [Online]
Available at: <https://valvesoftware.github.io/steam-audio/doc/unity/guide.html#>
[Zugriff am 28 06 2022].
- Valve Corporation (b), 2021. *Steam Audio Settings*. [Online]
Available at: <https://valvesoftware.github.io/steam-audio/doc/unity/settings.html>
[Zugriff am 20 06 2022].
- Valve Corporation (c), 2021. *Steam Audio Listener*. [Online]
Available at: <https://valvesoftware.github.io/steam-audio/doc/unity/listener.html>
[Zugriff am 25 06 2022].
- Valve Corporation , 2021. *Steam Audio Source*. [Online]
Available at: <https://valvesoftware.github.io/steam-audio/doc/unity/source.html>
[Zugriff am 25 06 2022].
- Valve Corporation, 2022. *Steam Audio-A Benchmark in Immersive Audio Solutions for Games and VR*. [Online]
Available at: <https://valvesoftware.github.io/steam-audio/#learn-more>
[Zugriff am 21 06 2022].
- Vorländer, M., 2020. *Auralization Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. s.l.:ASA Press.
- Vorländer, M., 2021. Virtuelle akustische Welten in. *Akustik Journal*, Issue 2, pp. 18-31.
- Vorländer, M., Schröder, D., Pelzer, S. & Wefers, F., 2014. Virtual Reality for architectural acoustics. *Journal of Building Performance Simulation*, 8(1), pp. 15-25.

Weinzierl, S., 2008. *Handbuch der Audiotechnik*. Heidelberg: Springer.

Zhong, X.-l. & Xie, B.-s., 2014. Head-Related Transfer Functions and Virtual Auditory Display. In: H. Glotin, Hrsg. *Soundscape Semiotics-Localization and Categorization*. s.l.:IntertechOpen.

Anhang

Anhang A

Untersuchte Messumgebungen



Schallabsorptionseinrichtung



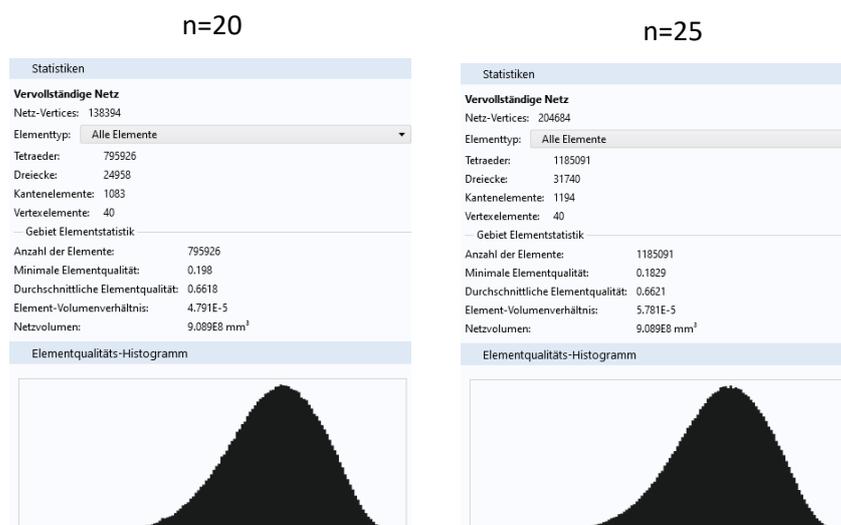
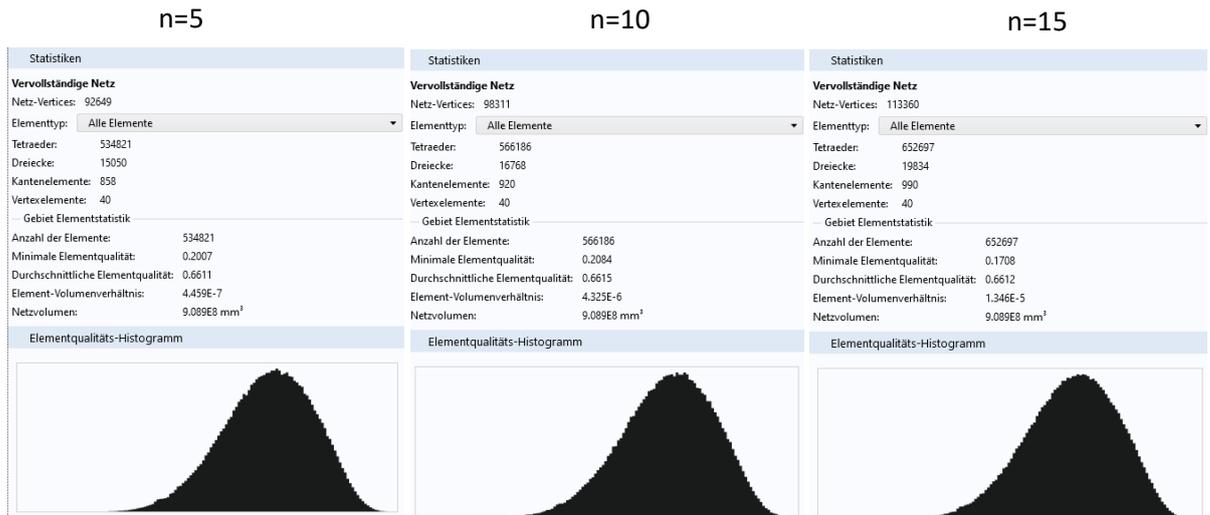
Klimakammer



Crew Rest Compartment
Mock-up

Anhang B

Qualitätsindikatoren für die erzeugten Netze



Anhang C

Skript zur Bewegung des Messmikrofons im Unity

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
[RequireComponent(typeof(AudioSource))]
public class Abkartierung2 : MonoBehaviour
{
    private AudioSource audioSource;

    // Define starting coordinates and add the option to change them in Unity
    public float xStart;
    public float yStart;
    public float zStart;

    // Define horizontal, vertical and longitudinal steps
    private float deltaX;
    private float deltaXneg;
    private float deltaY;
    private float deltaYInit = 0.0113f; // Y-Step only on the inclined side
    private float deltaZ;
    public int XMesspunkte;
    public int Messebenen;
    public int ZMesspunkte;

    // Define size of measurement area ; All values in [m]
    public float XMessbereich;
    public float Height_L = 0.7f;
    public float Height_R = 0.6f;
    public float ZMessbereich;

    public float delay; // Pause between measurement points in seconds
    public float delayAudio; // Delay for audio playback
    private int counter = 0;

    void Awake()
    {
        transform.position = new Vector3(xStart, yStart, zStart); // Move the Mic
        audioSource = GameObject.Find("Lautsprecher").GetComponent<AudioSource>();
        // Compute step size
        deltaX = (XMessbereich - 0.001f) / (XMesspunkte - 1);
        deltaXneg = -deltaX;
        deltaY = (Height_R - 0.001f) / (Messebenen - 1);
        deltaZ = ZMessbereich / (ZMesspunkte - 1);
    }

    IEnumerator Start()
    {
```

```

        // yield return StartCoroutine(DelayedInitialMovement()); //Movement on
the inclined side
        yield return StartCoroutine(FieldMovement()); // Movement inside the
measurement field

        Debug.Log("Measurement is completed");
    }
    // Used Functions

    //IEnumerator DelayedInitialMovement() // Function for initial movement on
inclined side with chosen delay
    //{
    //    for (float m = 0; Math.Abs(m) < Width; m += (Width / (HorPoints - 1)))
    //    {
    //        yield return new WaitForSeconds(delay);
    //        transform.position = new Vector3(transform.position.x + deltaX,
transform.position.y - deltaYInit, transform.position.z);
    //    }
    //}

    IEnumerator FieldMovement() // Function for field movement with chosen delay
w/o first and last row
    {
        for (float m = 0; m < XMesspunkte; m++)
        {
            for (float i = 0; Math.Abs(i) < ZMessbereich; i += (ZMessbereich /
(ZMesspunkte - 1)))
            {
                yield return new WaitForSeconds(delay);
                transform.position = new Vector3(transform.position.x, trans-
form.position.y, transform.position.z + deltaZ);
                audioSource.Pause();
                yield return new WaitForSeconds(delayAudio);
                audioSource.Play();
            }
            counter = ++counter;

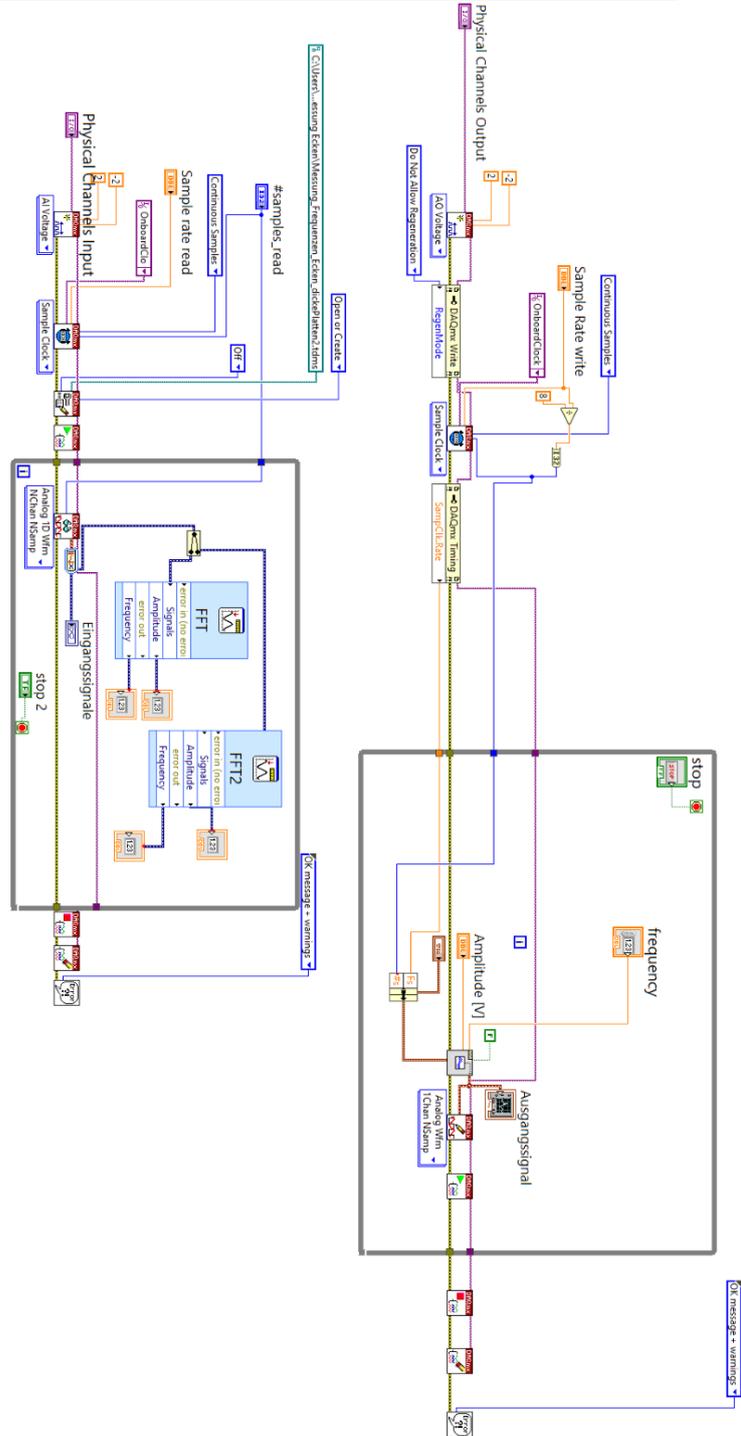
            if (counter < XMesspunkte)
            {
                yield return new WaitForSeconds(delay);
                transform.position = new Vector3(transform.position.x + del-
taXneg, transform.position.y, transform.position.z);
                audioSource.Pause();
                yield return new WaitForSeconds(delayAudio);
                audioSource.Play();

                for (float i = 0; Math.Abs(i) < ZMessbereich; i += (ZMessbereich
/ (ZMesspunkte - 1)))
                {
                    yield return new WaitForSeconds(delay);
                    transform.position = new Vector3(transform.position.x, trans-
form.position.y, transform.position.z - deltaZ);
                    audioSource.Pause();
                    yield return new WaitForSeconds(delayAudio);
                    audioSource.Play();
                }
            }
        }
    }
}

```

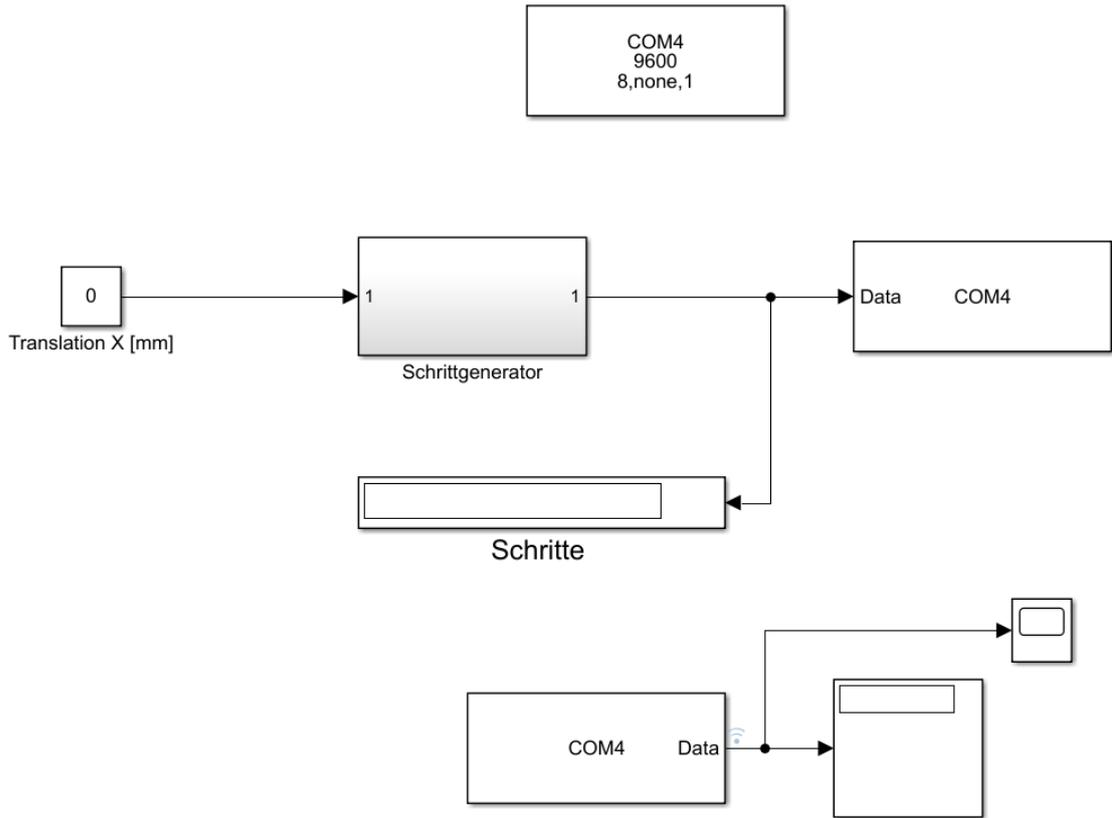

Anhang D

Messsoftware für die Messung der Mikrofonempfindlichkeiten



Anhang E

Simulink Modell zur Steuerung des Schrittmotors



Anhang F

Quellcodes der MATLAB Skripte

1. Kartierung FEM

```
% Skript für die Auswertung der Kartierung des FEM-Modells.
% Es werden die Schalldruckpegel im 3D-Raum dargestellt.
% Für die Auswertung muss jeweils eine Messung importiert werden, indem die
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).
% Die Variable ampFEM_compl_norm_X_Y muss für die Berechnung des
% MAC-Faktors als .m-Datei exportiert; X = Frequenz, Y = Ebene
% z. B. ampFEM_compl_norm_180_E2 = Variable für Frequenz 180 Hz und Messebene
E2

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Importieren der Messdaten aus Comsol

% file1 = load('Kartierung FEM 170Hz E1.txt','-ascii');
file1 = load('Kartierung FEM 180Hz E1.txt','-ascii');
% file1 = load('Kartierung FEM 170Hz E2.txt','-ascii');
% file1 = load('Kartierung FEM 180Hz E2.txt','-ascii');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Auslesen der komplexen Amplituden aus den geladenen Dateien

sz=size(file1);
ampFEM_compl = zeros(sz(1,1), sz(1,2)/2);

for i=1:sz(1,1)
    j=1;
    k=2;
    while k<=sz(1,2)
        ampFEM_compl(i,j) = complex(file1(i,k-1),file1(i,k));
        j=j+1;
        k=k+2;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Absolutdruck berechnen

ampFEM = zeros(sz(1,1),sz(1,2)/2);
sz_ampFEM = size(ampFEM);
Lp_ampFEM = zeros(sz_ampFEM(1,1), sz_ampFEM(1,2));

for i=1:sz_ampFEM(1,1)
```

```

    for j=1:sz_ampFEM(1,2)
        ampFEM(i,j) = abs(ampFEM_compl(i,j));
        Lp_ampFEM(i,j) = 20*log10(ampFEM(i,j)/(sqrt(2)*(2*10^(-5))));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung der X- und Y-Positionvektoren um den Schalldruckpegel darzustellen.

x =
[625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475];
y =
[71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631];
x(:,12) = 1185;

createfigure(x,y,Lp_ampFEM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ermittlung der Phasen für die Neureferenzierung

phase_FEM = zeros(sz_ampFEM);

for i=1:sz_ampFEM(1,1)
    for j=1:sz_ampFEM(1,2)
        phase_FEM(i,j) =
atan2(imag(ampFEM_compl(i,j)),real(ampFEM_compl(i,j)));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neureferenzierung der Phasen bezüglich des Punktes X = 625 mm // Y = 71 mm

phase_FEM_norm = zeros(size(phase_FEM));

for i=1:sz_ampFEM(1,1)
    for j=1:sz_ampFEM(1,2)
        phase_FEM_norm(i,j) = phase_FEM(i,j) - phase_FEM(1,1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bildung der komplexen Amplituden mit den neuen Phasen. Die
% Variable ampFEM_compl_X_Y wird für die MAC-Analyse benötigt.
% Für jede Frequenz/Ebene werden die Werte für X und Y angepasst, damit sie
% wiedererkannt werden

ampFEM_compl_norm_180_E2 = complex(zeros(size(ampFEM_compl)),0);

for i=1:sz_ampFEM(1,1)

```

```

    for j=1:sz_ampFEM(1,2)
        A = ampFEM(i,j);
        ampFEM_compl_norm_180_E2(i,j) = com-
plex(A*cos(phase_FEM_norm(i,j)),A*sin(phase_FEM_norm(i,j)));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung des Schalldruckpegels.
% Bei der Darstellung des Schalldruckpegels muss beachtet werden, dass die
% Grenzen auf der Z-Achse (zlim) richtig gestellt werden, sonst wird der
% Verlauf geschnitten. Gleichzeitig muss darauf geachtet werden, dass
% die Z-Koordinaten für die xlabel und ylabel angepasst werden,
% sodass die xlabel und ylabel sichtbar sind. Dafür sind die Zeilen
% 147 (180 Hz) / 148 (170 Hz), 152 (180 Hz) / 153 (170 Hz) und 162 (180 Hz) /
% 163 (170 Hz) zu aktivieren/deaktivieren.

function createfigure(xdata1, ydata1, zdata1)
%CREATEFIGURE(xdata1, ydata1, zdata1)
% XDATA1: surface xdata
% YDATA1: surface ydata
% ZDATA1: surface zdata

% Create figure
figure1 = figure('WindowState','maximized');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.231041666666667 0.0940478564307074 0.536979166666667
0.902736789631097]);
hold(axes1,'on');

% Create surf
surf(xdata1,ydata1,zdata1,'Parent',axes1);

% Create xlabel
xlabel('L_{P,B} (re 20 \muPa) in [dB]','FontSize',28,...
    'FontName','Arial');

% Create ylabel
ylabel('Y [mm]','Position',[600 350 64],'FontSize',28,'FontName','Arial');
% ylabel('Y [mm]','Position',[600 350 58],'FontSize',28,'FontName','Arial');

% Create xlabel
xlabel('X [mm]','Position',[1130 -25 64],'FontSize',28,'FontName','Arial');
% xlabel('X [mm]','Position',[1130 -25 58],'FontSize',28,'FontName','Arial');

% Uncomment the following line to preserve the X-limits of the axes
xlim(axes1,[600 1600]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 631]);

% Uncomment the following line to preserve the Z-limits of the axes

```

```
zlim(axes1,[66 86]);  
% zlim(axes1, [60 86]);  
  
view(axes1,[-51.3061815993674 10.2315159686687]);  
grid(axes1,'on');  
hold(axes1,'off');  
% Set the remaining axes properties  
set(axes1,'FontName','Arial','FontSize',28, 'LineWidth', 1.7, 'GridAlpha',  
0.3);  
end
```

2. Kartierung reale Umgebung

```
% Skript für die Auswertung der Kartierung der realen Messumgebung.  
% Es werden die Schalldruckpegel im 3D-Raum dargestellt.  
% Für die Auswertung muss jeweils eine Messung importiert werden, indem die  
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).  
% Die Variable kArU wird für die MAC-Analyse erstellt und vom Benutzer  
% als .m-Datei abgespeichert
```

```
clc;  
clear;  
close all;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% %Importieren Dateien Kartierung E1 170 Hz
```

```
pos1 = load('2. Messung Pos.1 E1 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
pos2 = load('2. Messung Pos.2 E1 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
pos3 = load('2. Messung Pos.3 E1 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
pos4 = load('2. Messung Pos.4 E1 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
pos5 = load('2. Messung Pos.5 E1 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');
```

```
% %Importieren Dateien Kartierung E1 180 Hz
```

```
% pos1 = load('3. Messung Pos.1 E1 mit 180 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos2 = load('3. Messung Pos.2 E1 mit 180 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos3 = load('3. Messung Pos.3 E1 mit 180 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos4 = load('3. Messung Pos.4 E1 mit 180 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos5 = load('3. Messung Pos.5 E1 mit 180 Hz_ mit Phasendifferenz.txt', '-  
ascii');
```

```
% %Importieren Dateien Kartierung E2 170 Hz
```

```
% pos1 = load('4. Messung Pos.1 E2 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos2 = load('4. Messung Pos.2 E2 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos3 = load('4. Messung Pos.3 E2 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos4 = load('4. Messung Pos.4 E2 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');  
% pos5 = load('4. Messung Pos.5 E2 mit 170 Hz_ mit Phasendifferenz.txt', '-  
ascii');
```

```

% %Importieren Dateien Kartierung E2 180 Hz

% pos1 = load('5. Messung Pos.1 E2 mit 180 Hz_ mit Phasendifferenz.txt','-
ascii');
% pos2 = load('5. Messung Pos.2 E2 mit 180 Hz_ mit Phasendifferenz.txt','-
ascii');
% pos3 = load('5. Messung Pos.3 E2 mit 180 Hz_ mit Phasendifferenz.txt','-
ascii');
% pos4 = load('5. Messung Pos.4 E2 mit 180 Hz_ mit Phasendifferenz.txt','-
ascii');
% pos5 = load('5. Messung Pos.5 E2 mit 180 Hz_ mit Phasendifferenz.txt','-
ascii');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bearbeitung der importierten Dateien, indem man einen Mittelwert für die
Amplitude, Frequenz und Phase für die
% jeweiligen Positionen berechnet. Die Mittelwerte werden dann in eine
% Variable gespeichert.

sz = [size(pos1),size(pos2),size(pos3),size(pos4),size(pos5)];

for i = 1:2:9
    k = sz(i);
    n = 0;
    amp1 = 0;
    amp2 = 0;
    freq1 = 0;
    freq2 = 0;
    phi1 = 0;
    phi2 = 0;
    p = 1;
    for j = 1:sz(i)
        if i == 1
            if pos1(j,1) ~= 0
                n = n+1;
                amp1 = amp1 + pos1(j,1);
                freq1 = freq1 + pos1(j,2);
                if pos1(j,3) < 0
                    pos1(j,3) = 360 + pos1(j,3);
                end
                phi1 = phi1 + pos1(j,3);
                amp2 = amp2 + pos1(j,4);
                freq2 = freq2 + pos1(j,5);
                if pos1(j,6) < 0
                    pos1(j,6) = 360 + pos1(j,6);
                end
                phi2 = phi2 + pos1(j,6);
            if pos1(j+1,1) == 0
                pos1_mW(p,1) = amp1/n;
                pos1_mW(p,2) = freq1/n;
                pos1_mW(p,3) = phi1/n;
                pos1_mW(p,4) = amp2/n;
                pos1_mW(p,5) = freq2/n;
            end
        end
    end
end

```

```

        pos1_mW(p,6) = phi2/n;
        p = p+1;
        n=0;
        amp1 = 0;
        amp2 = 0;
        freq1 = 0;
        freq2 = 0;
        phi1 = 0;
        phi2 = 0;
    end
end
elseif i == 3
    if pos2(j,1) ~= 0
        n = n+1;
        amp1 = amp1 + pos2(j,1);
        freq1 = freq1 + pos2(j,2);
        if pos2(j,3) < 0
            pos2(j,3) = 360 + pos2(j,3);
        end
        phi1 = phi1 + pos2(j,3);
        amp2 = amp2 + pos2(j,4);
        freq2 = freq2 + pos2(j,5);
        if pos2(j,6) < 0
            pos2(j,6) = 360 + pos2(j,6);
        end
        phi2 = phi2 + pos2(j,6);
        if pos2(j+1,1) == 0
            pos2_mW(p,1) = amp1/n;
            pos2_mW(p,2) = freq1/n;
            pos2_mW(p,3) = phi1/n;
            pos2_mW(p,4) = amp2/n;
            pos2_mW(p,5) = freq2/n;
            pos2_mW(p,6) = phi2/n;
            p = p+1;
            n=0;
            amp1 = 0;
            amp2 = 0;
            freq1 = 0;
            freq2 = 0;
            phi1 = 0;
            phi2 = 0;
        end
    end
end
elseif i == 5
    if pos3(j,1) ~= 0
        n = n+1;
        amp1 = amp1 + pos3(j,1);
        freq1 = freq1 + pos3(j,2);
        if pos3(j,3) < 0
            pos3(j,3) = 360 + pos3(j,3);
        end
        phi1 = phi1 + pos3(j,3);
        amp2 = amp2 + pos3(j,4);
    end
end

```

```

freq2 = freq2 + pos3(j,5);
if pos3(j,6) < 0
    pos3(j,6) = 360 + pos3(j,6);
end
phi2 = phi2 + pos3(j,6);
if pos3(j+1,1) == 0
    pos3_mW(p,1) = amp1/n;
    pos3_mW(p,2) = freq1/n;
    pos3_mW(p,3) = phi1/n;
    pos3_mW(p,4) = amp2/n;
    pos3_mW(p,5) = freq2/n;
    pos3_mW(p,6) = phi2/n;
    p = p+1;
    n=0;
    amp1 = 0;
    amp2 = 0;
    freq1 = 0;
    freq2 = 0;
    phi1 = 0;
    phi2 = 0;
end
end
elseif i == 7
if pos4(j,1) ~= 0
    n = n+1;
    amp1 = amp1 + pos4(j,1);
    freq1 = freq1 + pos4(j,2);
if pos4(j,3) < 0
    pos4(j,3) = 360 + pos4(j,3);
end
phi1 = phi1 + pos4(j,3);
if pos4(j+1,1) == 0
    pos4_mW(p,1) = amp1/n;
    pos4_mW(p,2) = freq1/n;
    pos4_mW(p,3) = phi1/n;
    p = p+1;
    n=0;
    amp1 = 0;
    freq1 = 0;
    phi1 = 0;
end
end
else
if pos5(j,1) ~= 0
    n = n+1;
    amp1 = amp1 + pos5(j,1);
    freq1 = freq1 + pos5(j,2);
if pos5(j,3) < 0
    pos5(j,3) = 360 + pos5(j,3);
end
phi1 = phi1 + pos5(j,3);
if pos5(j+1,1) == 0
    pos5_mW(p,1) = amp1/n;

```

```

        pos5_mW(p,2) = freq1/n;
        pos5_mW(p,3) = phi1/n;
        p = p+1;
        n=0;
        amp1 = 0;
        freq1 = 0;
        phi1 = 0;
    end
end
end
end
end

gemessene_datei_mW =
[pos1_mW(:,4:6),pos2_mW(:,4:6),pos3_mW(:,4:6),pos1_mW(:,1:3),pos2_mW(:,1:3),p
os3_mW(:,1:3),pos4_mW(:,1:3), pos5_mW(:,1:3)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Interpolation für die Berechnung der Empfindlichkeiten der
%% Messmikrofone für die jeweiligen Messfrequenzen.

file=load('E_Mikro.txt', '-ascii');
freq=file(:,1);
E_M1=file(:,2);
E_M2=file(:,3);
E_M3=file(:,4);
E_M4=file(:,5);
E = zeros(4,1);

for i=1:4
    E(i,1)=pchip(freq,file(:,i+1),180);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Berechnung des Schalldrucks aus den importierten Dateien

for i=1:3:24
    for j=1:18
        if i<=3
            sdruck(j,i) = (gemessene_datei_mW(j,i))*10^3/(E(4,1)*10^(26/20));
        else
            sdruck(j,i) = (gemessene_datei_mW(j,i))*10^3/(E(1,1)*10^(26/20));
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Bestimmung des Schalldruckpegels

for i=1:3:24
    for j=1:18
        L_p(j,i) = 20*log10((sdruck(j,i)/sqrt(2))/(2*10^(-5)));
    end
end

```

```

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung der X- und Y-Positionvektoren sowie der Variable L_kompact,
% um die räumliche Verteilung des Schalldruckpegels darzustellen.

x =
[625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:50:1475;625:
50:1475;625:50:1475];
y =
[71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:
631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:80:631;71:
80:631;71:80:631;71:80:631]';
x(:,12) = 1185;

L_kompact = [L_p(:,1) L_p(:,4) L_p(:,7) L_p(:,10) L_p(:,13) L_p(:,16)
L_p(:,19) L_p(:,22)];

createfigure(x,y,L_kompact')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neureferenzierung der Phasen der Messpunkten aus der realen Umgebung bezüglich des
% Punktes x = 625 mm // y = 71 mm

for i = 1:18
    for j = 3:3:24
        gemessene_datei_mW_norm(i,j-2) = gemessene_datei_mW(i,j-2);
        gemessene_datei_mW_norm(i,j-1) = gemessene_datei_mW(i,j-1);
        gemessene_datei_mW_norm(i,j) = deg2rad(gemessene_datei_mW(i,j) - gemessene_datei_mW(1,3));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung der komplexen Amplituden für die reale Umgebung (kArU) mit neuen Phasen. Die
% Variable kArU wird für das MAC-Verfahren benötigt.

for i = 1:18
    kk = 1;
    for j = 3:3:24
        A = sdruck(i,j-2);
        kArU(i,kk) = complex(A*cos(gemessene_datei_mW_norm(i,j)),A*sin(gemessene_datei_mW_norm(i,j)));
        kk = kk+1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung des Schalldruckpegels.
% Bei der Darstellung des Schalldruckpegels muss beachtet werden, dass die
% Grenzen auf der Z-Achse (zlim) richtig gestellt werden, sonst wird der

```

```

% Verlauf geschnitten. Gleichzeitig muss darauf geachtet werden, dass
% die Z-Koordinaten für die xlabel und ylabel angepasst werden,
% sodass die xlabel und ylabel sichtbar sind. Dafür sind die Zeilen
% 340 (180Hz) / 341 (170Hz), 344 (180Hz) / 345 (170Hz) und 352 (180 Hz) / 353
% (170 Hz) zu aktivieren/deaktivieren.

```

```

function createfigure(xdata1, ydata1, zdata1)
%CREATEFIGURE(xdata1, ydata1, zdata1)
% XDATA1: surface xdata
% YDATA1: surface ydata
% ZDATA1: surface zdata

% Create figure
figure1 = figure('WindowState','maximized');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.231041666666667 0.0940478564307074 0.536979166666667
0.902736789631097]);
hold(axes1,'on');

% Create surf
surf(xdata1,ydata1,zdata1,'Parent',axes1);

% Create xlabel
xlabel('L_{P,M} (re 20 \muPa) in [dB]','FontSize',28,...
    'FontName','Arial');

% Create ylabel
% ylabel('Y [mm]','Position',[600 400 73],'FontSize',28,'FontName','Arial');
ylabel('Y [mm]','Position',[600 400 47],'FontSize',28,'FontName','Arial');

% Create xlabel
% xlabel('X [mm]','Position',[1150 -37 73],'FontSize',28,'FontName','Arial');
xlabel('X [mm]','Position',[1150 -37 48],'FontSize',28,'FontName','Arial');

% Uncomment the following line to preserve the X-limits of the axes
xlim(axes1,[600 1600]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 631]);
% Uncomment the following line to preserve the Z-limits of the axes
% zlim(axes1, [75 105]);
zlim(axes1,[50 95]);

view(axes1,[-51.684945235731 17.4646540439825]);
grid(axes1,'on');
hold(axes1,'off');
% Set the remaining axes properties
set(axes1,'FontName','Arial','FontSize',28, 'GridAlpha',0.3,'LineWidth',
1.7);
end

```

3. Kartierung Unity optimiert

```
% Skript für die Auswertung der Kartierung der optimierten Simulation
% und der Berechnung der Differenzen zwischen den Amplitudenpegel.
% Es werden die Amplitudenpegel im 3D-Raum dargestellt.
% Für die Auswertung muss jeweils eine Messung importiert werden, indem die
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).
% Die Variable ampFEM_compl_norm_X_Y muss für die Berechnung des
% MAC-Faktors als .m-Datei exportiert; X = Frequenz, Y = Ebene
% z. B. ampFEM_compl_norm_180_E2 = Variable für Frequenz 180 Hz und Messebene
E2

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ablesen der Audiodateien

% [data, fs] = audioread('Kartierung_180Hz_E1.wav');
[data, fs] = audioread('Kartierung_170Hz_E1.wav');
% [data, fs] = audioread('Kartierung_180Hz_E2.wav');
% [data, fs] = audioread('Kartierung_170Hz_E2.wav');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Importieren der Amplitudenpegel aus der realen und der FEM Umgebung

load('La_ampFEM_170E1.mat');
load('La_ampFEM_180E1.mat');
load('La_ampFEM_170E2.mat');
load('La_ampFEM_180E2.mat');
load('L_a_real_170E1.mat');
load('L_a_real_180E1.mat');
load('L_a_real_170E2.mat');
load('L_a_real_180E2.mat');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bearbeitung der abgelesenen Audiodateien, indem man die Messdateien in Blö-
cke
% für die jeweiligen Positionen separiert. Die aufgespalteten Messdateien
% werden zunächst in eine Variable gespeichert.

i = 1;
luecke = 1;
interval = 4;
war_luecke = true;
zaehler = 1;
datapos = 1;
datapos2 = 1;
while i<=length(data)
    if i==1
        i = 0.5*fs;
```

```

end
if war_luecke == true
    while zaehler <= 4*fs
        data_sec(datapos) = data(i);
        datapos = datapos+1;
        i = i+1;
        zaehler = zaehler+1;
    end
    war_luecke = false;
    zaehler = 1;
else
    while zaehler <= luecke*fs && i<=length(data)
        data_sec(datapos) = 0;
        datapos = datapos+1;
        i = i+1;
        zaehler = zaehler+1;
    end
    war_luecke = true;
    zaehler = 1;
end
end

data_sec = data_sec';

i = 1;
col = 1;
war_luecke = true;
data_matrix = zeros(interval*fs,144);
while (i<=length(data_sec)-1)
    if war_luecke == true
        for k = 1:interval*fs
            data_matrix(k,col) = data_sec(i);
            i = i+1;
        end
        col = col+1;
        war_luecke = false;
    else
        for k=1:luecke*fs
            i=i+1;
        end
        war_luecke = true;
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT-Analyse, Erstellung der Positionsvektoren X und Y

data_matrx_fft = fft(data_matrix);
L_matrix = length(data_matrix(:,1));
size_dmf = size(data_matrx_fft);
amp_fft = zeros(1,size_dmf(1,2));
phase_fft = zeros(1,size_dmf(1,2));
f_matrix = fs*(0:(L_matrix/(fs/500)))/L_matrix;

```

```

for i=1:size_dmf(1,2)
    P2 = abs(data_matrx_fft(:,i))/(L_matrix/2);
    P1 = P2(1:L_matrix/(fs/500)+1);
    pos_complex_max = find(P1 == max(P1));
    amp_fft(i) = P1(pos_complex_max);
    phase_fft(i) = atan2(imag(data_matrx_fft(pos_complex_max,i)),real(data_matrx_fft(pos_complex_max,i)));
end

X = zeros(1,size_dmf(1,2));
Y = zeros(1,size_dmf(1,2));
X(1) = 625;
Y(1) = 71;
i=2;
k=2;
for j=1:4
    while k<=18
        Y(i) = Y(i-1);
        X(i) = X(i-1)+50;
        i=i+1;
        k=k+1;

    end
    k=k-1;
    Y(i) = Y(i-1)+80;
    X(i) = X(i-1);
    i=i+1;
    while k>=2
        Y(i) = Y(i-1);
        X(i) = X(i-1)-50;
        i=i+1;
        k=k-1;

    end
    k=2;
    if i<=size_dmf(1,2)
        Y(i) = Y(i-1)+80;
        X(i) = X(i-1);
        i=i+1;

    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der Amplitudenpegel bezogen auf die maximale Amplitude
amp_max = max(amp_fft,[],"all");
amp_fft_norm = amp_fft/amp_max;
L_amp_fft_norm = 20*log10(amp_fft_norm);
amp_matrix = [L_amp_fft_norm(1:18); flip(L_amp_fft_norm(19:36));
L_amp_fft_norm(37:54); flip(L_amp_fft_norm(55:72));L_amp_fft_norm(73:90);
flip(L_amp_fft_norm(91:108)); L_amp_fft_norm(109:126);
flip(L_amp_fft_norm(127:144))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3D Plot

```

```

X_plot =
[X(1:18);flip(X(19:36));X(37:54);flip(X(55:72));X(73:90);flip(X(91:108));X(10
9:126);flip(X(127:144))];
Y_plot =
[Y(1:18);Y(19:36);Y(37:54);Y(55:72);Y(73:90);Y(91:108);Y(109:126);Y(127:144)]
;
createfigure(X_plot,Y_plot,amp_matrix)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der Unterschiede der Amplitudenpegel

% L_diff_FEM_Opt=La_ampFEM_170E1-amp_matrix;
% L_diff_FEM_Opt=La_ampFEM_180E1-amp_matrix;
% L_diff_FEM_Opt=La_ampFEM_170E2-amp_matrix;
% L_diff_FEM_Opt=La_ampFEM_180E2-amp_matrix;
% L_diff_FEM_Opt=L_a_real_170E1'-amp_matrix;
% L_diff_FEM_Opt=L_a_real_180E1'-amp_matrix;
L_diff_FEM_Opt=L_a_real_170E2'-amp_matrix;
% L_diff_FEM_Opt=L_a_real_180E2'-amp_matrix;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot der Differenzen

createfigure2(X_plot, L_diff_FEM_Opt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neureferenzierung der Phasen der Messpunkten bezüglich des
% Punktes x = 625 mm // y = 71 mm

phase_fft_norm = zeros(1,length(phase_fft));
for i=1:length(phase_fft)
    phase_fft_norm(i) = phase_fft(i)-phase_fft(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bildung der komplexen Amplituden mit den neuen Phasen. Die
% Variable amp_matrix_norm_X_Y wird für die MAC-Analyse benötigt.
% Für jede Frequenz/Ebene werden die Werte für X und Y angepasst, damit sie
% wiedererkannt werden

amp_compfft_norm = zeros(1,length(phase_fft));

for i=1:length(phase_fft_norm)
    A = amp_fft_norm(i);
    amp_compfft_norm(i) = com-
plex(A*cos(phase_fft_norm(i)),A*sin(phase_fft_norm(i)));
end

amp_matrix_norm_180_E2 = [amp_compfft_norm(1:18);
flip(amp_compfft_norm(19:36)); amp_compfft_norm(37:54);
flip(amp_compfft_norm(55:72)); amp_compfft_norm(73:90);
flip(amp_compfft_norm(91:108)); amp_compfft_norm(109:126);
flip(amp_compfft_norm(127:144))];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung des Amplitudenpegels.
% Bei der Darstellung des Amplitudenpegels muss beachtet werden, dass die
% Grenzen auf der Z-Achse (zlim) richtig gestellt werden, sonst wird der
% Verlauf geschnitten. Gleichzeitig muss darauf geachtet werden, dass
% die Z-Koordinaten für die xlabel und ylabel angepasst werden,
% sodass die xlabel und ylabel sichtbar sind. Dafür sind die Zeilen
% 252 (180Hz) / 253 (170Hz), 256 (180Hz) / 257 (170Hz) und 264 (180Hz) / 265
% (170Hz) zu aktivieren/deaktivieren.

function createfigure(xdata1, ydata1, zdata1)
%CREATEFIGURE(xdata1, ydata1, zdata1)
% XDATA1: surface xdata
% YDATA1: surface ydata
% ZDATA1: surface zdata

% Create figure
figure1 = figure('WindowState','maximized');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.231041666666667 0.0940478564307074 0.536979166666667
0.902736789631097]);
hold(axes1,'on');

% Create surf
surf(xdata1,ydata1,zdata1,'Parent',axes1);

% Create xlabel
xlabel('L_{A,Unity verbessert} (re A_{max}) in [dB]','FontSize',28,...
    'FontName','Arial');

% Create ylabel
% ylabel('Y [mm]','Position',[600 350 -20],'FontSize',28,'FontName','Arial');
ylabel('Y [mm]','Position',[600 350 -27],'FontSize',28,'FontName','Arial');

% Create xlabel
% xlabel('X [mm]','Position',[1130 -29 -20],'FontSize',28,'FontName','Ari-
al');
xlabel('X [mm]','Position',[1130 -29 -27],'FontSize',28,'FontName','Arial');

% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[600 1600]);
% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[0 631]);
% Uncomment the following line to preserve the Z-limits of the axes
% zlim(axes1,[-18 0]);
% zlim(axes1,[-25 0]);

view(axes1,[-52.3477815993674 15.3980431653214]);
grid(axes1,'on');
hold(axes1,'off');
% Set the remaining axes properties

```

```

set(axes1, 'FontName', 'Arial', 'FontSize', 28, 'LineWidth', 1.7, 'GridAlpha',
0.3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung der Pegeldifferenzen

function createfigure2(X1, YMatrix1)
%CREATEFIGURE(X1, YMatrix1)
% X1: vector of x data
% YMATRIX1: matrix of y data

% Create figure
figure2 = figure('WindowState', 'maximized');

% Create axes
axes1 = axes('Parent', figure2);
hold(axes1, 'on');

% Create multiple lines using matrix input to plot
Marker_Counter=1;
Markers = {'+', 'o', '*', 'x', 'v', 'd', '^', 's'};
for i=1:8
    plot(X1(i,:), YMatrix1(i,:), 'LineWidth', 2, 'Marker', Markers{Marker_Coun-
ter}, 'MarkerSize', 12);
    Marker_Counter=Marker_Counter+1;
end

% Create ylabel
ylabel('L_{A,real} - L_{A,Unity verbessert} in [dB]', 'FontSize', 28, 'Font-
Name', 'Arial');

% Create xlabel
xlabel('X in [mm]', 'FontSize', 28, 'FontName', 'Arial');

hold(axes1, 'off');

% Set the remaining axes properties
set(axes1, 'FontName', 'Arial', 'FontSize', 28, 'GridAlpha', 0.3, 'Tick-
Dir', 'out', ...
'XGrid', 'on', 'YGrid', 'on', 'LineWidth', 1.7);

% Create legend
legend('Y=71', 'Y=151', 'Y=231', 'Y=311', 'Y=391', 'Y=471', 'Y=551', 'Y=631', 'Font-
Name', 'Arial', 'FontSize', 28);
end

```

4. Kartierung Unity Resonance Audio

```
% Skript für die Auswertung der Kartierung der simulierten Messumgebung
% Unity Resonance Audio
% Für die Auswertung muss jeweils eine Messung importiert werden, indem die
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ablesen der Audiodateien

% [data, fs] = audioread('Kartierung_180Hz_E1.wav');
[data, fs] = audioread('Kartierung_170Hz_E1.wav');
% [data, fs] = audioread('Kartierung_180Hz_E2.wav');
% [data, fs] = audioread('Kartierung_170Hz_E2.wav');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bearbeitung der abgelesenen Audiodateien, indem man die Messdateien in Blö-
cke
% für die jeweiligen Positionen separiert. Die aufgespalteten Messdateien
% werden zunächst in eine Variable gespeichert.

luecke = 2.002;
interval = 4;
i = 1;
war_luecke = true;
zaehler = 1;
datapos = 1;
datapos2 = 1;

while i<=length(data)
    if i==1
        i = 0.5*fs;
    end
    if war_luecke == true
        while zaehler <= 4*fs
            data_sec(datapos) = data(i);
            datapos = datapos+1;
            i = i+1;
            zaehler = zaehler+1;
        end
        war_luecke = false;
        zaehler = 1;
    else
        while zaehler <= luecke*fs && i<=length(data)
            data_sec(datapos) = 0;
            datapos = datapos+1;
            i = i+1;
            zaehler = zaehler+1;
        end
    end
end
```



```

k=2;
for j=1:4
    while k<=18
        Y(i) = Y(i-1);
        X(i) = X(i-1)+50;
        i=i+1;
        k=k+1;
    end
    k=k-1;
    Y(i) = Y(i-1)+80;
    X(i) = X(i-1);
    i=i+1;
    while k>=2
        Y(i) = Y(i-1);
        X(i) = X(i-1)-50;
        i=i+1;
        k=k-1;
    end
    k=2;
    if i<=size_dmf(1,2)
        Y(i) = Y(i-1)+80;
        X(i) = X(i-1);
        i=i+1;
    end
end

amp_matrix = [amp_fft(1:18); flip(amp_fft(19:36)); amp_fft(37:54);
flip(amp_fft(55:72));amp_fft(73:90); flip(amp_fft(91:108)); amp_fft(109:126);
flip(amp_fft(127:144))];
amp_matrix_max = max(amp_matrix, [], 'all');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der Amplitudenpegel bezogen auf die maximale Amplitude

for i=1:8
    for j=1:18
        amp_matrix_Lp(i,j)= 20*log10(amp_matrix(i,j)/amp_matrix_max);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3D Plot

X_plot =
[X(1:18);flip(X(19:36));X(37:54);flip(X(55:72));X(73:90);flip(X(91:108));X(10
9:126);flip(X(127:144))];
Y_plot =
[Y(1:18);Y(19:36);Y(37:54);Y(55:72);Y(73:90);Y(91:108);Y(109:126);Y(127:144)]
;
createfigure(X_plot,Y_plot,amp_matrix_Lp)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neureferenzierung der Phasen der Messpunkte bezüglich des

```

```

% Punktes x = 625 mm // y = 71 mm

phase_fft_norm = zeros(1,length(phase_fft));
for i=1:length(phase_fft)
    phase_fft_norm(i) = phase_fft(i)-phase_fft(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bildung der komplexen Amplituden mit den neuen Phasen. Die
% Variable amp_matrix_norm_X_Y wird für die MAC-Analyse benötigt.
% Für jede Frequenz/Ebene werden die Werte für X und Y angepasst, damit sie
% wiedererkannt werden

amp_fft_norm = zeros(1,length(phase_fft));

for i=1:length(phase_fft_norm)
    A = amp_fft(i);
    amp_fft_norm(i) = complex(A*cos(phase_fft_norm(i)),A*sin(phase_fft_norm(i)));
end

amp_matrix_norm_170_E1 = [amp_fft_norm(1:18); flip(amp_fft_norm(19:36));
amp_fft_norm(37:54); flip(amp_fft_norm(55:72)); amp_fft_norm(73:90);
flip(amp_fft_norm(91:108)); amp_fft_norm(109:126);
flip(amp_fft_norm(127:144))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung des Amplitudenpegels.
% Bei der Darstellung des Amplitudenpegels muss beachtet werden, dass die
% Grenzen auf der Z-Achse (zlim) richtig gestellt werden, sonst wird der
% Verlauf geschnitten. Gleichzeitig muss darauf geachtet werden, dass
% die Z-Koordinaten für die xlabel und ylabel angepasst werden,
% sodass die xlabel und ylabel sichtbar sind. Dafür sind die Zeilen
% 218 (180Hz) / 219 (170Hz), 222 (180Hz) / 223 (170Hz) und 230 (180Hz) / 231
% (170Hz) zu aktivieren/deaktivieren.

function createfigure(xdata1, ydata1, zdata1)
%CREATEFIGURE(xdata1, ydata1, zdata1)
% XDATA1: surface xdata
% YDATA1: surface ydata
% ZDATA1: surface zdata

% Create figure
figure1 = figure('WindowState','maximized');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.231041666666667 0.0940478564307074 0.536979166666667
0.902736789631097]);
hold(axes1,'on');

% Create surf

```

```

surf(xdata1,ydata1,zdata1,'Parent',axes1);

% Create zlabel
xlabel('L_{A,S} (re A_{max}) in [dB]','FontSize',22,...
      'FontName','Arial');

% Create ylabel
% ylabel('Y [mm]','Position',[600 400 -22],'FontSize',28,'FontName','Arial');
ylabel('Y [mm]','Position',[600 400 -27],'FontSize',28,'FontName','Arial');

% Create xlabel
% xlabel('X [mm]','Position',[1150 -37 -22],'FontSize',28,'FontName','Arial');
xlabel('X [mm]','Position',[1150 -37 -27],'FontSize',28,'FontName','Arial');

% Uncomment the following line to preserve the X-limits of the axes
xlim(axes1,[600 1600]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 631]);
% Uncomment the following line to preserve the Z-limits of the axes
% zlim(axes1,[-20 0]);
zlim(axes1,[-25 0]);

view(axes1,[-51.6849452852915 23.147833914725]);

grid(axes1,'on');
hold(axes1,'off');

% Set the remaining axes properties
set(axes1,'FontName','Arial','FontSize',28,'GridAlpha',0.3,'LineWidth',
1.7);
end

```

5. Kartierung Unity Steam Audio

```
% Skript für die Auswertung der Kartierung der simulierten Messumgebung
% Unity Steam Audio.
% Für die Auswertung muss jeweils eine Messung importiert werden, indem die
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ablesen der Audiodateien

[data, fs] = audioread('Kartierung_180Hz_E1.wav');
% [data, fs] = audioread('Kartierung_170Hz_E1.wav');
% [data, fs] = audioread('Kartierung_180Hz_E2.wav');
% [data, fs] = audioread('Kartierung_170Hz_E2.wav');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bearbeitung der abgelesenen Audiodateien, indem man die Messdateien in Blö-
cke
% für die jeweiligen Positionen separiert. Die aufgespalteten Messdateien
% werden zunächst in eine Variable gespeichert.

luecke = 2.013;
interval = 4;
i = 1;
war_luecke = true;
zaehler = 1;
datapos = 1;
datapos2 = 1;
while i<=length(data)
    if i==1
        i = 0.5*fs;
    end
    if war_luecke == true
        while zaehler <= 4*fs
            data_sec(datapos) = data(i);
            datapos = datapos+1;
            i = i+1;
            zaehler = zaehler+1;
        end
        war_luecke = false;
        zaehler = 1;
    else
        while zaehler <= luecke*fs && i<=length(data)
            data_sec(datapos) = 0;
            datapos = datapos+1;
            i = i+1;
            zaehler = zaehler+1;
        end
    end
end
```

```

        war_luecke = true;
        zaehler = 1;
    end
end

data_sec = data_sec';
i = 1;
col = 1;
war_luecke = true;
data_matrix = zeros(interval*fs,144);
while (i<=length(data_sec)-1)
    if war_luecke == true
        for k = 1:interval*fs
            data_matrix(k,col) = data_sec(i);
            i = i+1;
        end
        col = col+1;
        war_luecke = false;
    else
        for k=1:luecke*fs
            i=i+1;
        end
        war_luecke = true;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT-Analyse, Erstellung der Positionsvektoren X und Y

data_matrix_fft = fft(data_matrix);
L_matrix = length(data_matrix(:,1));
size_dmf = size(data_matrix_fft);
amp_fft = zeros(1,size_dmf(1,2));
phase_fft = zeros(1,size_dmf(1,2));
f_matrix = fs*(0:(L_matrix/(fs/500)))/L_matrix;

for i=1:size_dmf(1,2)
    P2 = abs(data_matrix_fft(:,i))/(L_matrix/2);
    P1 = P2(1:L_matrix/(fs/500)+1);
    pos_complex_max = find(P1 == max(P1));
    amp_fft(i) = P1(pos_complex_max);
    phase_fft(i) = atan2(imag(data_matrix_fft(pos_complex_max,i)),real(data_matrix_fft(pos_complex_max,i)));
end

X = zeros(1,size_dmf(1,2));
Y = zeros(1,size_dmf(1,2));

X(1) = 625;
Y(1) = 71;
i=2;
k=2;
for j=1:4

```

```

while k<=18
    Y(i) = Y(i-1);
    X(i) = X(i-1)+50;
    i=i+1;
    k=k+1;
end
k=k-1;
Y(i) = Y(i-1)+80;
X(i) = X(i-1);
i=i+1;
while k>=2
    Y(i) = Y(i-1);
    X(i) = X(i-1)-50;
    i=i+1;
    k=k-1;
end
k=2;
if i<=size_dmf(1,2)
    Y(i) = Y(i-1)+80;
    X(i) = X(i-1);
    i=i+1;
end
end

amp_matrix = [amp_fft(1:18); flip(amp_fft(19:36)); amp_fft(37:54);
flip(amp_fft(55:72));amp_fft(73:90); flip(amp_fft(91:108)); amp_fft(109:126);
flip(amp_fft(127:144))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der Amplitudenpegel bezogen auf die maximale Amplitude

amp_matrix_max = max(amp_matrix, [], 'all');
for i=1:8
    for j=1:18
        amp_matrix_Lp(i,j)= 20*log10(amp_matrix(i,j)/amp_matrix_max);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3D Plot
X_plot =
[X(1:18);flip(X(19:36));X(37:54);flip(X(55:72));X(73:90);flip(X(91:108));X(10
9:126);flip(X(127:144))];
Y_plot =
[Y(1:18);Y(19:36);Y(37:54);Y(55:72);Y(73:90);Y(91:108);Y(109:126);Y(127:144)]
;

createfigure(X_plot,Y_plot,amp_matrix_Lp)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neureferenzierung der Phasen der Messpunkte bezüglich des
% Punktes x = 625 mm // y = 71 mm

```

```

phase_fft_norm = zeros(1,length(phase_fft));
for i=1:length(phase_fft)
    phase_fft_norm(i) = phase_fft(i)-phase_fft(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bildung der komplexen Amplituden mit den neuen Phasen. Die
% Variable amp_matrix_norm_X_Y wird für die MAC-Analyse benötigt.
% Für jede Frequenz/Ebene werden die Werte für X und Y angepasst, damit sie
% wiedererkannt werden

amp_fft_norm = zeros(1,length(phase_fft));

for i=1:length(phase_fft_norm)
    A = amp_fft(i);
    amp_fft_norm(i) = complex(A*cos(phase_fft_norm(i)),A*sin(phase_fft_norm(i)));
end

amp_matrix_norm_180_E1 = [amp_fft_norm(1:18); flip(amp_fft_norm(19:36));
amp_fft_norm(37:54); flip(amp_fft_norm(55:72)); amp_fft_norm(73:90);
flip(amp_fft_norm(91:108)); amp_fft_norm(109:126);
flip(amp_fft_norm(127:144))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die Darstellung des Amplitudenpegels.

function createfigure(xdata1, ydata1, zdata1)
%CREATEFIGURE(xdata1, ydata1, zdata1)
% XDATA1: surface xdata
% YDATA1: surface ydata
% ZDATA1: surface zdata

% Create figure
figure1 = figure('WindowState','maximized');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.231041666666667 0.0940478564307074 0.536979166666667
0.902736789631097]);
hold(axes1,'on');

% Create surf
surf(xdata1,ydata1,zdata1,'Parent',axes1);

% Create xlabel
xlabel('L_{A,S} (re A_{max}) in [dB]','FontSize',28,...
    'FontName','Arial');

% Create ylabel

```

```

ylabel('Y [mm]', 'Position', [600 400 -11], 'FontSize', 28, 'FontName', 'Arial');

% Create xlabel
xlabel('X [mm]', 'Position', [1300 -25 -11], 'FontSize', 28, 'FontName', 'Arial');

% Uncomment the following line to preserve the X-limits of the axes
xlim(axes1, [600 1600]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1, [0 631]);
% Uncomment the following line to preserve the Z-limits of the axes
zlim(axes1, [-10 0]);

view(axes1, [-51.684945235731 17.4646540439825]);

grid(axes1, 'on');
hold(axes1, 'off');

% Set the remaining axes properties
set(axes1, 'FontName', 'Arial', 'FontSize', 28, 'GridAlpha', 0.3, 'LineWidth',
1.7);
end

```

6. MAC reale Umgebung mit FEM-Modell

% Skript für die MAC-Analyse zwischen den Messdaten aus der realen Messumgebung und dem FEM-Modell.

```
clc
clear
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Importieren der komplexen Amplituden aus der FEM-Messumgebung beziehungsweise
% aus der realen Messumgebung.
```

```
load('ampFEM_compl_norm_170_E1.mat');
load('ampFEM_compl_norm_180_E1.mat');
load('ampFEM_compl_norm_170_E2.mat');
load('ampFEM_compl_norm_180_E2.mat');
```

```
kArU_E1_170 = load('kArU_E1_170Hz.mat');
kArU_E1_180 = load('kArU_E1_180Hz.mat');
kArU_E2_170 = load('kArU_E2_170Hz.mat');
kArU_E2_180 = load('kArU_E2_180Hz.mat');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Erstellung der 144x4 Matrizen phi1 (mit Werten aus der FEM-Messumgebung) und
```

```
% phi2 (mit Werten aus der realen Messumgebung).
```

```
% Die Variablen "ampFEM_compl_norm_X_Y" (X = Frequenz bzw. Y = Ebene)
```

```
% wurden aus dem Skript "Kartierung_FEM" exportiert.
```

```
% Die Variablen "kArU_Y_X" wurden aus dem Skript
```

```
% "Kartierung_reale_Umgebung" exportiert.
```

```
% Spalte 1: Frequenz 180 Hz, Ebene E1
```

```
% Spalte 2: Frequenz 170 Hz, Ebene E1
```

```
% Spalte 3: Frequenz 180 Hz, Ebene E2
```

```
% Spalte 4: Frequenz 170 Hz, Ebene E2
```

```
phi1(:,1) = [ampFEM_compl_norm_180_E1(1,:) ampFEM_compl_norm_180_E1(2,:)
ampFEM_compl_norm_180_E1(3,:) ampFEM_compl_norm_180_E1(4,:) ...
ampFEM_compl_norm_180_E1(5,:) ampFEM_compl_norm_180_E1(6,:)
ampFEM_compl_norm_180_E1(7,:) ampFEM_compl_norm_180_E1(8,:)];
phi1(:,2) = [ampFEM_compl_norm_170_E1(1,:) ampFEM_compl_norm_170_E1(2,:)
ampFEM_compl_norm_170_E1(3,:) ampFEM_compl_norm_170_E1(4,:) ...
ampFEM_compl_norm_170_E1(5,:) ampFEM_compl_norm_170_E1(6,:)
ampFEM_compl_norm_170_E1(7,:) ampFEM_compl_norm_170_E1(8,:)];
phi1(:,3) = [ampFEM_compl_norm_180_E2(1,:) ampFEM_compl_norm_180_E2(2,:)
ampFEM_compl_norm_180_E2(3,:) ampFEM_compl_norm_180_E2(4,:) ...
ampFEM_compl_norm_180_E2(5,:) ampFEM_compl_norm_180_E2(6,:)
ampFEM_compl_norm_180_E2(7,:) ampFEM_compl_norm_180_E2(8,:)];
phi1(:,4) = [ampFEM_compl_norm_170_E2(1,:) ampFEM_compl_norm_170_E2(2,:)
ampFEM_compl_norm_170_E2(3,:) ampFEM_compl_norm_170_E2(4,:) ...
```

```

    ampFEM_compl_norm_170_E2(5,:) ampFEM_compl_norm_170_E2(6,:)
ampFEM_compl_norm_170_E2(7,:) ampFEM_compl_norm_170_E2(8,:)]];

phi2(:,1) = [kArU_E1_180.kArU(:,1); kArU_E1_180.kArU(:,2);
kArU_E1_180.kArU(:,3); kArU_E1_180.kArU(:,4); ...
    kArU_E1_180.kArU(:,5); kArU_E1_180.kArU(:,6); kArU_E1_180.kArU(:,7);
kArU_E1_180.kArU(:,8)];
phi2(:,2) = [kArU_E1_170.kArU(:,1); kArU_E1_170.kArU(:,2);
kArU_E1_170.kArU(:,3); kArU_E1_170.kArU(:,4); ...
    kArU_E1_170.kArU(:,5); kArU_E1_170.kArU(:,6); kArU_E1_170.kArU(:,7);
kArU_E1_170.kArU(:,8)];
phi2(:,3) = [kArU_E2_180.kArU(:,1); kArU_E2_180.kArU(:,2);
kArU_E2_180.kArU(:,3); kArU_E2_180.kArU(:,4); ...
    kArU_E2_180.kArU(:,5); kArU_E2_180.kArU(:,6); kArU_E2_180.kArU(:,7);
kArU_E2_180.kArU(:,8)];
phi2(:,4) = [kArU_E2_170.kArU(:,1); kArU_E2_170.kArU(:,2);
kArU_E2_170.kArU(:,3); kArU_E2_170.kArU(:,4); ...
    kArU_E2_170.kArU(:,5); kArU_E2_170.kArU(:,6); kArU_E2_170.kArU(:,7);
kArU_E2_170.kArU(:,8)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Durchführen das MAC-Verfahren

MAC_Matrix = zeros(4,4);

for i=1:4
    for j=1:4
        MAC_Matrix(i,j) = Mac(phi2(:,i),phi1(:,j));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung des Balkendiagramms für die Veranschaulichung des
% MAC-Faktores. Der farbliche Verlauf kann aus dem Property Inspector
% angepasst werden

figure1 = figure;
axes1 = axes('Parent',figure1,...
    'Position',[0.135902855836662 0.195107414480986 0.674886678996049
0.78921717041918]);
hold(axes1,'on');

b = bar3(MAC_Matrix);
colorbar

for k = 1:length(b)
    zdata = b(k).ZData;
    b(k).CData = zdata;
    b(k).FaceColor = 'interp';
end

% Create ylabel
ylabel('Reale Umgebung','Rotation',-63);

```

```

% Create xlabel
xlabel('FEM-Modell','Rotation',10);

view(axes1,[-20.4718063223464 38.943630715022]);

grid(axes1,'on');
axis(axes1,'ij');
hold(axes1,'off');

% Set the remaining axes properties
set(axes1,'CLim',[0.9 1],'FontSize',20,'GridAlpha',0.3,'LineWidth',1.7,'Plot-
BoxAspectRatio',...
    [1.22970583144992 1.61803398874989 1],'XTick',[1 2 3 4],'XTickLabel',...
    {'E1 180 Hz','E1 170 Hz','E2 180 Hz','E2 170 Hz'},'YTick',[1 2 3 4],...
    'YTickLabel',{'E1 180 Hz','E1 170 Hz','E2 180 Hz','E2 170 Hz'});

% Create colorbar
colorbar(axes1,'Position',...
    [0.915531943566933 0.194189983288325 0.0218355510064824
    0.78921717041918]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die MAC-Analyse

function mAc=Mac(Phi1,Phi2)
    mAc= (abs(Phi1'*Phi2))^2/((Phi1'*Phi1)*(Phi2'*Phi2));
end

```

7. MAC reale Umgebung / FEM-Modell mit Unity optimiert

% Skript für die MAC-Analyse zwischen den Messdaten aus der optimierten Umgebung, realen Messumgebung und FEM-Modell.

```
clc
clear
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Importieren der komplexen Amplituden aus der verbesserten Messumgebung
Unity,
% aus der realen Messumgebung und aus FEM.
```

```
% FEM Umgebung
```

```
F_170E1 = load("F_170E1.mat");
F_170E2 = load('F_170E2.mat');
F_180E1 = load("F_180E1.mat");
F_180E2 = load('F_180E2.mat');
```

```
% Reale Umgebung
```

```
kArU_E1_170=load('kArU_E1_170Hz.mat');
kArU_E1_180=load('kArU_E1_180Hz.mat');
kArU_E2_170=load('kArU_E2_170Hz.mat');
kArU_E2_180=load('kArU_E2_180Hz.mat');
```

```
% Unity optimiert
```

```
load('Uo_170E1.mat');
load('Uo_170E2.mat');
load('Uo_180E1.mat');
load('Uo_180E2.mat');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Erstellung der 144x4 Matrizen phi1 (mit Werten aus der Messumgebung Unity
Resonance Audio) und
```

```
% phi2 (mit Werten aus der realen Messumgebung).
```

```
% Die Variablen "amp_matrix_norm_X_Y" (X = Frequenz bzw. Y = Ebene)
```

```
% wurden aus dem Skript "Kartierung_Unity_opt" exportiert.
```

```
% Die Variablen "kArU_Y_X" wurden aus dem Skript
```

```
% "Kartierung_reale_Umgebung" exportiert.
```

```
% Die Variablen "F_YX" wurden aus dem Skript
```

```
% "Kartierung_FEM" exportiert.
```

```
% die Matrix phi2 muss entsprechend dem gewollten Vergleich aktiviert/deakti-
viert werden
```

```
% Spalte 1: Frequenz 180 Hz, Ebene E1
```

```
% Spalte 2: Frequenz 170 Hz, Ebene E1
```

```
% Spalte 3: Frequenz 180 Hz, Ebene E2
```

```
% Spalte 4: Frequenz 170 Hz, Ebene E2
```

```

phi1(:,2) = [amp_matrix_norm_170_E1(:,1); amp_matrix_norm_170_E1(:,2);
amp_matrix_norm_170_E1(:,3); amp_matrix_norm_170_E1(:,4); ...
amp_matrix_norm_170_E1(:,5); amp_matrix_norm_170_E1(:,6); amp_mat-
rix_norm_170_E1(:,7); amp_matrix_norm_170_E1(:,8)];
phi1(:,1) = [amp_matrix_norm_180_E1(:,1); amp_matrix_norm_180_E1(:,2);
amp_matrix_norm_180_E1(:,3); amp_matrix_norm_180_E1(:,4); ...
amp_matrix_norm_180_E1(:,5); amp_matrix_norm_180_E1(:,6); amp_mat-
rix_norm_180_E1(:,7); amp_matrix_norm_180_E1(:,8)];
phi1(:,4) = [amp_matrix_norm_170_E2(:,1); amp_matrix_norm_170_E2(:,2);
amp_matrix_norm_170_E2(:,3); amp_matrix_norm_170_E2(:,4); ...
amp_matrix_norm_170_E2(:,5); amp_matrix_norm_170_E2(:,6); amp_mat-
rix_norm_170_E2(:,7); amp_matrix_norm_170_E2(:,8)];
phi1(:,3) = [amp_matrix_norm_180_E2(:,1); amp_matrix_norm_180_E2(:,2);
amp_matrix_norm_180_E2(:,3); amp_matrix_norm_180_E2(:,4); ...
amp_matrix_norm_180_E2(:,5); amp_matrix_norm_180_E2(:,6); amp_mat-
rix_norm_180_E2(:,7); amp_matrix_norm_180_E2(:,8)];

```

% phi2 mit Daten aus FEM

```

% phi2(:,1) = [F_180E1.ampFEM_compl(1,:)'; F_180E1.ampFEM_compl(2,:)';
F_180E1.ampFEM_compl(3,:)'; F_180E1.ampFEM_compl(4,:)'; ...
% F_180E1.ampFEM_compl(5,:)'; F_180E1.ampFEM_compl(6,:)';
F_180E1.ampFEM_compl(7,:)'; F_180E1.ampFEM_compl(8,:)'];
% phi2(:,2) = [F_170E1.ampFEM_compl(1,:)'; F_170E1.ampFEM_compl(2,:)';
F_170E1.ampFEM_compl(3,:)'; F_170E1.ampFEM_compl(4,:)'; ...
% F_170E1.ampFEM_compl(5,:)'; F_170E1.ampFEM_compl(6,:)';
F_170E1.ampFEM_compl(7,:)'; F_170E1.ampFEM_compl(8,:)'];
% phi2(:,3) = [F_180E2.ampFEM_compl(1,:)'; F_180E2.ampFEM_compl(2,:)';
F_180E2.ampFEM_compl(3,:)'; F_180E2.ampFEM_compl(4,:)'; ...
% F_180E2.ampFEM_compl(5,:)';
F_180E2.ampFEM_compl(6,:)'; F_180E2.ampFEM_compl(7,:)';
F_180E2.ampFEM_compl(8,:)'];
% phi2(:,4) = [F_170E2.ampFEM_compl(1,:)'; F_170E2.ampFEM_compl(2,:)';
F_170E2.ampFEM_compl(3,:)'; F_170E2.ampFEM_compl(4,:)'; ...
% F_170E2.ampFEM_compl(5,:)'; F_170E2.ampFEM_compl(6,:)';
F_170E2.ampFEM_compl(7,:)'; F_170E2.ampFEM_compl(8,:)'];

```

% phi2 mit Daten aus realer Umgebung

```

phi2(:,1) = [kArU_E1_180.kArU(:,1); kArU_E1_180.kArU(:,2);
kArU_E1_180.kArU(:,3); kArU_E1_180.kArU(:,4); ...
kArU_E1_180.kArU(:,5); kArU_E1_180.kArU(:,6); kArU_E1_180.kArU(:,7);
kArU_E1_180.kArU(:,8)];
phi2(:,2) = [kArU_E1_170.kArU(:,1); kArU_E1_170.kArU(:,2);
kArU_E1_170.kArU(:,3); kArU_E1_170.kArU(:,4); ...
kArU_E1_170.kArU(:,5); kArU_E1_170.kArU(:,6); kArU_E1_170.kArU(:,7);
kArU_E1_170.kArU(:,8)];
phi2(:,3) = [kArU_E2_180.kArU(:,1); kArU_E2_180.kArU(:,2);
kArU_E2_180.kArU(:,3); kArU_E2_180.kArU(:,4); ...
kArU_E2_180.kArU(:,5); kArU_E2_180.kArU(:,6); kArU_E2_180.kArU(:,7);
kArU_E2_180.kArU(:,8)];
phi2(:,4) = [kArU_E2_170.kArU(:,1); kArU_E2_170.kArU(:,2);
kArU_E2_170.kArU(:,3); kArU_E2_170.kArU(:,4); ...

```

```

    kArU_E2_170.kArU(:,5); kArU_E2_170.kArU(:,6); kArU_E2_170.kArU(:,7);
    kArU_E2_170.kArU(:,8)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Durchführen der MAC-Analyse

MAC_Matrix = zeros(4,4);

for i=1:4
    for j=1:4
        MAC_Matrix(i,j) = Mac(phi2(:,i),phi1(:,j));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung des Balkendiagramms für die Veranschaulichung des
% MAC-Faktors. Der farbliche Verlauf kann aus dem Property Inspector
% angepasst werden

figure1 = figure;
axes1 = axes('Parent',figure1,...
    'Position',[0.135902855836662 0.195107414480986 0.674886678996049
0.78921717041918]);
hold(axes1,'on');

b = bar3(MAC_Matrix);
colorbar

for k = 1:length(b)
    zdata = b(k).ZData;
    b(k).CData = zdata;
    b(k).FaceColor = 'interp';
end

% Create ylabel
ylabel('Reale Umgebung','Rotation',-65);

% Create xlabel
xlabel('Unity verbessert','Rotation',12);

view(axes1,[-20.4718063223464 38.943630715022]);

grid(axes1,'on');
axis(axes1,'ij');
hold(axes1,'off');

% Set the remaining axes properties
set(axes1,'CLim',[0.9 1],'FontSize',20,'GridAlpha',0.3,'LineWidth',1.7,'Plot-
BoxAspectRatio',...
[1.22970583144992 1.61803398874989 1],'XTick',[1 2 3 4],'XTickLabel',...
{'E1 180 Hz','E1 170 Hz','E2 180 Hz','E2 170 Hz'},'YTick',[1 2 3 4],...
'YTickLabel',{'E1 180 Hz','E1 170 Hz','E2 180 Hz','E2 170 Hz'});

```

```

% Create colorbar
colorbar(axes1, 'Position', ...
    [0.915531943566933 0.194189983288325 0.0218355510064824
    0.78921717041918]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die MAC-Analyse

function mAc=Mac(Phi1,Phi2)
    mAc= (abs(Phi1'*Phi2))^2/((Phi1'*Phi1)*(Phi2'*Phi2));
end

```

8. MAC reale Umgebung mit Unity Resonance Audio

% Skript für die MAC-Analyse zwischen den Messdaten aus der realen Messumgebung und der simulierten Umgebung Resonance Audio.

```
clc
clear
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Importieren der komplexen Amplituden aus der simulierten Messumgebung Unity Resonance Audio beziehungsweise % aus der realen Messumgebung.
```

```
load("Amp_matrix_norm_170_E1.mat");
load('Amp_matrix_norm_170_E2.mat');
load("Amp_matrix_norm_180_E1.mat");
load('Amp_matrix_norm_180_E2.mat');
```

```
kArU_E1_170 = load('kArU_E1_170Hz.mat');
kArU_E1_180 = load('kArU_E1_180Hz.mat');
kArU_E2_170 = load('kArU_E2_170Hz.mat');
kArU_E2_180 = load('kArU_E2_180Hz.mat');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Erstellung der 144x4 Matrizen phi1 (mit Werten aus der Messumgebung Unity Resonance Audio) und
```

```
% phi2 (mit Werten aus der realen Messumgebung).
```

```
% Die Variablen "amp_matrix_norm_X_Y" (X = Frequenz bzw. Y = Ebene)
```

```
% wurden aus dem Skript "Kartierung_Resonance_Audio" exportiert.
```

```
% Die Variablen "kArU_Y_X" wurden aus dem Skript
```

```
% "Kartierung_reale_Umgebung" exportiert.
```

```
% Spalte 1: Frequenz 180 Hz, Ebene E1
```

```
% Spalte 2: Frequenz 170 Hz, Ebene E1
```

```
% Spalte 3: Frequenz 180 Hz, Ebene E2
```

```
% Spalte 4: Frequenz 170 Hz, Ebene E2
```

```
phi1(:,1) = [amp_matrix_norm_180_E1(:,1); amp_matrix_norm_180_E1(:,2);
amp_matrix_norm_180_E1(:,3); amp_matrix_norm_180_E1(:,4); ...
amp_matrix_norm_180_E1(:,5); amp_matrix_norm_180_E1(:,6); amp_matrix_norm_180_E1(:,7); amp_matrix_norm_180_E1(:,8)];
phi1(:,2) = [amp_matrix_norm_170_E1(:,1); amp_matrix_norm_170_E1(:,2);
amp_matrix_norm_170_E1(:,3); amp_matrix_norm_170_E1(:,4); ...
amp_matrix_norm_170_E1(:,5); amp_matrix_norm_170_E1(:,6); amp_matrix_norm_170_E1(:,7); amp_matrix_norm_170_E1(:,8)];
phi1(:,3) = [amp_matrix_norm_180_E2(:,1); amp_matrix_norm_180_E2(:,2);
amp_matrix_norm_180_E2(:,3); amp_matrix_norm_180_E2(:,4); ...
amp_matrix_norm_180_E2(:,5); amp_matrix_norm_180_E2(:,6); amp_matrix_norm_180_E2(:,7); amp_matrix_norm_180_E2(:,8)];
phi1(:,4) = [amp_matrix_norm_170_E2(:,1); amp_matrix_norm_170_E2(:,2);
amp_matrix_norm_170_E2(:,3); amp_matrix_norm_170_E2(:,4); ...
```

```

amp_matrix_norm_170_E2(:,5); amp_matrix_norm_170_E2(:,6); amp_mat-
rix_norm_170_E2(:,7); amp_matrix_norm_170_E2(:,8)];

phi2(:,1) = [kArU_E1_180.kArU(:,1); kArU_E1_180.kArU(:,2);
kArU_E1_180.kArU(:,3); kArU_E1_180.kArU(:,4); ...
kArU_E1_180.kArU(:,5); kArU_E1_180.kArU(:,6); kArU_E1_180.kArU(:,7);
kArU_E1_180.kArU(:,8)];
phi2(:,2) = [kArU_E1_170.kArU(:,1); kArU_E1_170.kArU(:,2);
kArU_E1_170.kArU(:,3); kArU_E1_170.kArU(:,4); ...
kArU_E1_170.kArU(:,5); kArU_E1_170.kArU(:,6); kArU_E1_170.kArU(:,7);
kArU_E1_170.kArU(:,8)];
phi2(:,3) = [kArU_E2_180.kArU(:,1); kArU_E2_180.kArU(:,2);
kArU_E2_180.kArU(:,3); kArU_E2_180.kArU(:,4); ...
kArU_E2_180.kArU(:,5); kArU_E2_180.kArU(:,6); kArU_E2_180.kArU(:,7);
kArU_E2_180.kArU(:,8)];
phi2(:,4) = [kArU_E2_170.kArU(:,1); kArU_E2_170.kArU(:,2);
kArU_E2_170.kArU(:,3); kArU_E2_170.kArU(:,4); ...
kArU_E2_170.kArU(:,5); kArU_E2_170.kArU(:,6); kArU_E2_170.kArU(:,7);
kArU_E2_170.kArU(:,8)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Durchführen der MAC-Analyse

MAC_Matrix = zeros(4,4);

for i=1:4
    for j=1:4
        MAC_Matrix(i,j) = Mac(phi2(:,i),phi1(:,j));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung des Balkendiagramms für die Veranschaulichung des
% MAC-Faktores. Der farbliche Verlauf kann aus dem Property Inspector
% angepasst werden

figure1 = figure;
axes1 = axes('Parent',figure1,...
'Position',[0.135902855836662 0.195107414480986 0.674886678996049
0.78921717041918]);
hold(axes1,'on');

b = bar3(MAC_Matrix);
colorbar

for k = 1:length(b)
    zdata = b(k).ZData;
    b(k).CData = zdata;
    b(k).FaceColor = 'interp';
end

% Create ylabel
ylabel('Reale Umgebung','Rotation',-63);

```

```

% Create xlabel
xlabel('Unity Resonance Audio','Rotation',10);

view(axes1,[-20.4718063223464 38.943630715022]);

grid(axes1,'on');
axis(axes1,'ij');
hold(axes1,'off');

% Set the remaining axes properties
set(axes1,'FontSize',20,'GridAlpha',0.3,'LineWidth',1.7,'PlotBoxAspectRa-
tio',...
    [1.22970583144992 1.61803398874989 1], 'XTick',[1 2 3 4], 'XTickLabel',...
    {'E1 180 Hz', 'E1 170 Hz', 'E2 180 Hz', 'E2 170 Hz'}, 'YTick',[1 2 3 4],...
    'YTickLabel',{'E1 180 Hz', 'E1 170 Hz', 'E2 180 Hz', 'E2 170 Hz'});

% Create colorbar
colorbar(axes1,'Position',...
    [0.915531943566933 0.194189983288325 0.0218355510064824
    0.78921717041918]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für das MAC-Verfahren

function mAc=Mac(Phi1,Phi2)
    mAc= (abs(Phi1'*Phi2))^2/((Phi1'*Phi1)*(Phi2'*Phi2));
end

```

9. MAC reale Umgebung mit Unity Steam Audio

% Skript für die MAC-Analyse zwischen den Messdaten aus der realen Messumgebung und der simulierten Umgebung Steam Audio.

```
clc
clear
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Importieren der komplexen Amplituden aus der simulierten Messumgebung Unity
Steam Audio beziehungsweise
% aus der realen Messumgebung.
```

```
load("Amp_matrix_norm_170_E1.mat");
load('Amp_matrix_norm_170_E2.mat');
load("Amp_matrix_norm_180_E1.mat");
load('Amp_matrix_norm_180_E2.mat');
```

```
kArU_E1_170 = load('kArU_E1_170Hz.mat');
kArU_E1_180 = load('kArU_E1_180Hz.mat');
kArU_E2_170 = load('kArU_E2_170Hz.mat');
kArU_E2_180 = load('kArU_E2_180Hz.mat');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Erstellung der 144x4 Matrizen phi1 (mit Werten aus der simulierten Messum-
gebung Unity Steam Audio) und
% phi2 (mit Werten aus der realen Messumgebung).
% Die Variablen "Amp_matrix_norm_X_Y" (X = Frequenz bzw. Y = Ebene)
% wurden aus dem Skript "Kartierung_Steam_Audio" exportiert.
% Die Variablen "kArU_Y_X" wurden aus dem Skript
% "Kartierung_reale_Umgebung" exportiert.
% Spalte 1: Frequenz 180 Hz, Ebene E1
% Spalte 2: Frequenz 170 Hz, Ebene E1
% Spalte 3: Frequenz 180 Hz, Ebene E2
% Spalte 4: Frequenz 170 Hz, Ebene E2
```

```
phi1(:,1) = [amp_matrix_norm_180_E1(:,1); amp_matrix_norm_180_E1(:,2);
amp_matrix_norm_180_E1(:,3); amp_matrix_norm_180_E1(:,4); ...
amp_matrix_norm_180_E1(:,5); amp_matrix_norm_180_E1(:,6); amp_mat-
rix_norm_180_E1(:,7); amp_matrix_norm_180_E1(:,8)];
phi1(:,2) = [amp_matrix_norm_170_E1(:,1); amp_matrix_norm_170_E1(:,2);
amp_matrix_norm_170_E1(:,3); amp_matrix_norm_170_E1(:,4); ...
amp_matrix_norm_170_E1(:,5); amp_matrix_norm_170_E1(:,6); amp_mat-
rix_norm_170_E1(:,7); amp_matrix_norm_170_E1(:,8)];
phi1(:,3) = [amp_matrix_norm_180_E2(:,1); amp_matrix_norm_180_E2(:,2);
amp_matrix_norm_180_E2(:,3); amp_matrix_norm_180_E2(:,4); ...
amp_matrix_norm_180_E2(:,5); amp_matrix_norm_180_E2(:,6); amp_mat-
rix_norm_180_E2(:,7); amp_matrix_norm_180_E2(:,8)];
phi1(:,4) = [amp_matrix_norm_170_E2(:,1); amp_matrix_norm_170_E2(:,2);
amp_matrix_norm_170_E2(:,3); amp_matrix_norm_170_E2(:,4); ...
```

```

amp_matrix_norm_170_E2(:,5); amp_matrix_norm_170_E2(:,6); amp_mat-
rix_norm_170_E2(:,7); amp_matrix_norm_170_E2(:,8)];

phi2(:,1) = [kArU_E1_180.kArU(:,1); kArU_E1_180.kArU(:,2);
kArU_E1_180.kArU(:,3); kArU_E1_180.kArU(:,4); ...
kArU_E1_180.kArU(:,5); kArU_E1_180.kArU(:,6); kArU_E1_180.kArU(:,7);
kArU_E1_180.kArU(:,8)];
phi2(:,2) = [kArU_E1_170.kArU(:,1); kArU_E1_170.kArU(:,2);
kArU_E1_170.kArU(:,3); kArU_E1_170.kArU(:,4); ...
kArU_E1_170.kArU(:,5); kArU_E1_170.kArU(:,6); kArU_E1_170.kArU(:,7);
kArU_E1_170.kArU(:,8)];
phi2(:,3) = [kArU_E2_180.kArU(:,1); kArU_E2_180.kArU(:,2);
kArU_E2_180.kArU(:,3); kArU_E2_180.kArU(:,4); ...
kArU_E2_180.kArU(:,5); kArU_E2_180.kArU(:,6); kArU_E2_180.kArU(:,7);
kArU_E2_180.kArU(:,8)];
phi2(:,4) = [kArU_E2_170.kArU(:,1); kArU_E2_170.kArU(:,2);
kArU_E2_170.kArU(:,3); kArU_E2_170.kArU(:,4); ...
kArU_E2_170.kArU(:,5); kArU_E2_170.kArU(:,6); kArU_E2_170.kArU(:,7);
kArU_E2_170.kArU(:,8)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Durchführen der MAC-Analyse

MAC_Matrix = zeros(4,4);

for i=1:4
    for j=1:4
        MAC_Matrix(i,j) = Mac(phi2(:,i),phi1(:,j));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung des Balkendiagramms für die Veranschaulichung des
% MAC-Faktores. Der farbliche Verlauf kann aus dem Property Inspector
% angepasst werden

figure1 = figure;
axes1 = axes('Parent',figure1,...
'Position',[0.135902855836662 0.195107414480986 0.674886678996049
0.78921717041918]);
hold(axes1,'on');

b = bar3(MAC_Matrix);
colorbar

for k = 1:length(b)
    zdata = b(k).ZData;
    b(k).CData = zdata;
    b(k).FaceColor = 'interp';
end

% Create ylabel
ylabel('Reale Umgebung','Rotation',-63);

```

```

% Create xlabel
xlabel('Unity Steam Audio','Rotation',10);

view(axes1,[-20.4718063223464 38.943630715022]);

grid(axes1,'on');
axis(axes1,'ij');
hold(axes1,'off');

% Set the remaining axes properties
set(axes1,'FontSize',20,'GridAlpha',0.3,'LineWidth',1.7,'PlotBoxAspectRa-
tio',...
    [1.22970583144992 1.61803398874989 1], 'XTick',[1 2 3 4], 'XTickLabel',...
    {'E1 180 Hz', 'E1 170 Hz', 'E2 180 Hz', 'E2 170 Hz'}, 'YTick',[1 2 3 4],...
    'YTickLabel',{'E1 180 Hz', 'E1 170 Hz', 'E2 180 Hz', 'E2 170 Hz'});

% Create colorbar
colorbar(axes1,'Position',...
    [0.915531943566933 0.194189983288325 0.0218355510064824
    0.78921717041918]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion für die MAC-Analyse

function mAc=Mac(Phi1,Phi2)
    mAc= (abs(Phi1'*Phi2))^2/((Phi1'*Phi1)*(Phi2'*Phi2));
end

```

10. Frequenzgang reale Umgebung

% Skript für die Berechnung des Frequenzganges für die reale Umgebung

```
clc
clear;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Importieren der Messwerte
Messungen = load('Messung mit Mikroverstärker.txt', '-ascii');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialisierung der Variablen
sz = size(Messungen);

k = 2.5;
amp1 = 0;
amp2 = 0;
freq1 = 0;
freq2 = 0;
mW_freq_ch_1 = 0;
mW_freq_ch_2 = 0;
n = 1;
j = 1;
mW = zeros(90,4);

% For-Schleife um die durchschnittliche Frequenz und Spannung für eine Fre-
quenz zu bestimmen.
% Hier wird gezählt wie viele Messungen / Frequenz gemessen wurden. Die Fre-
quenzen und
% Spannungen werden summiert und anhand des arithmetischen Mittels werden
% die durchschnittlichen Werte berechnet.

for i = 1:sz(1,1)
    amp1 = amp1 + Messungen(i,3);
    amp2 = amp2 + Messungen(i,5);
    freq1 = freq1 + Messungen(i,2);
    freq2 = freq2 + Messungen(i,4);
    if i+1<= sz(1,1)
        m = abs(Messungen(i+1,2) - Messungen(i,2));
        if m < k
            n = n+1;
        else
            mW_freq_ch_1 = freq1 / n;
            mW_freq_ch_2 = freq2 / n;
            mW_amp_1 = amp1 / n;
            mW_amp_2 = amp2 / n;
            mW(j,1) = mW_freq_ch_1;
            mW(j,2) = mW_amp_1;
            mW(j,3) = mW_freq_ch_2;
            mW(j,4) = mW_amp_2;
        end
    end
end
```

```

        j = j+1;
        amp1 = 0;
        amp2 = 0;
        freq1 = 0;
        freq2 = 0;
        n = 1;
    end
else
    mW_freq_ch_1 = freq1 / n;
    mW_freq_ch_2 = freq2 / n;
    mW_amp_1 = amp1 / n;
    mW_amp_2 = amp2 / n;
    mW(j,1) = mW_freq_ch_1;
    mW(j,2) = mW_amp_1;
    mW(j,3) = mW_freq_ch_2;
    mW(j,4) = mW_amp_2;
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Interpolation für die Berechnung der Empfindlichkeiten der
% Messmikrofone für die jeweiligen Messfrequenzen.

file=load('E_Mikro.txt', '-ascii');
freq=file(:,1);
E_M1=file(:,2);
E_M2=file(:,3);
E_M3=file(:,4);
E_M4=file(:,5);

for j=1:91
    for i=1:1:4
        E(i,j)=pchip(freq,file(:,i+1),mW(j,1));
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung des Schalldrucks aus den importierten Dateien
% Mik. 3 und Mik. 4

for i=1:91
    p(1,i) = (rms(mW(i,2))*10^3)/(E(4,i)*10^(26/20)); %Mik. 3
    p(2,i) = (rms(mW(i,4))*10^3)/(E(3,i)*10^(26/20)); %Mik. 4
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bestimmung der Schalldruckpegel
% Mik. 3 und Mik. 4

for i=1:91
    L_p(1,i) = 20*log10(p(1,i)/ (2*10^(-5))); %Mik. 3
    L_p(2,i) = 20*log10(p(2,i)/ (2*10^(-5))); %Mik. 4
end

```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Darstellung der Schalldruckpegel
```

```
figure1 = figure('WindowState','maximized');  
axes1 = axes('Parent',figure1);  
hold(axes1,'on');  
h=plot(mW(:,1),L_p(1,:), mW(:,3),L_p(2,:));  
hold off  
h(1).LineWidth=2;  
h(1).Color=[0 0.4470 0.7410];  
h(2).LineWidth=2;  
h(2).Color=[0.8500 0.3250 0.0980];  
xlabel('Frequenz in [Hz]')  
ylabel('L_p (re 20 \muPa) in [dB]')  
set(axes1,'FontName','Arial','FontSize',28,'GridAlpha',0.3,'Tick-  
Dir','out',...  
      'XGrid','on','YGrid','on','LineWidth',1.7);  
legend('Mik. 3; Rechts','Mik. 4; Links')  
grid on  
grid minor  
ax = gca;  
ax.XTick=50:25:500;  
ax.XTickLabel=50:25:500;  
ax.TickDir='out';  
ax.MinorGridAlpha = 0.3;  
hold on;
```

11. Frequenzgang Unity Resonance Audio

```
% Skript für die Berechnung des Frequenzganges für die simulierte Umgebung
% Unity Resonance Audio
% Für die Berechnung muss jeweils eine Messung importiert werden, indem die
% entsprechenden Codezeilen aktiviert werden (Kommentar entfernen).
```

```
clc
clear
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Auslesen der Audiodateien, rechts und links
% Zu beachten: Für die Darstellung des Frequenzganges für "FreqSwipe_WoodC"
% muss Zeile 104 aktiviert werden.
% Für die Darstellung des Frequenzganges für "FreqSwipe_WoodP" muss Zeile
% 105 aktiviert werden.
```

```
[data, fs] = audioread('FreqSwipe_Plywood_rechts.wav');
[data2, fs_2] = audioread('FreqSwipe_Plywood_links.wav');
% [data, fs] = audioread('FreqSwipe_WoodC_rechts.wav');
% [data2, fs_2] = audioread('FreqSwipe_WoodC_links.wav');
% [data, fs] = audioread('FreqSwipe_WoodP_rechts.wav');
% [data2, fs_2] = audioread('FreqSwipe_WoodP_links.wav');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% FFT Analyse
```

```
T = 1/fs;
T_2 = 1/fs_2;
L = length(data);
L_2 = length(data2);
t = (0:L-1)*T;
t_2 = (0:L_2-1)*T_2;
f = fs*(0:(L/(fs/500)))/L;
f_2 = fs_2*(0:(L_2/(fs_2/500)))/L_2;
data_fft = fft(data);
data2_fft = fft(data2);
```

```
P2 = abs(data_fft)/(L/2);
P2_2 = abs(data2_fft)/(L_2/2);
P1 = P2(1:L/(fs/500)+1);
P1_2 = P2_2(1:L_2/(fs_2/500)+1);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Normierung der Amplituden, amp_i/amp_max

amp_max = max(P1);
amp_max_2=max(P1_2);

for i=1:length(P1)
    P1_Norm(i) = P1(i)/amp_max;
    L_P1_Norm(i)=20*log10(P1_Norm(i));
end
for i=1:length(P1_2)
    P1_2_Norm(i) = P1_2(i)/amp_max_2;
    L_P1_2_Norm(i)=20*log10(P1_2_Norm(i));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot Amplituden-Verlauf.
% Zu beachten: Für die Darstellung des Frequenzganges für "FreqSwipe_WoodC"
muss Zeile 104 aktiviert werden.
% Für die Darstellung des Frequenzganges für "FreqSwipe_WoodP" muss Zeile 105
aktiviert werden.

f_ver(1) = f(1);
f_ver_2(1) = f_2(1);
amp_ver(1) = P1_Norm(1);
amp_ver_2(1) = P1_2_Norm(1);
k=5;
f= f';
f_2=f_2';
ff = find(f <=5,1,'last');
ff_2 = find(f_2 <=5,1,'last');

for i=1:500
    f_position(i) = find(f<=k,1,'last');
    f_position_2(i) = find(f_2<=k,1,'last');
    f_ver(i) = f(f_position(i));
    f_ver_2(i) = f_2(f_position_2(i));
    k = k+5;
end

figure(1)
for i=1:length(f_position)
    amp_ver(i) = L_P1_Norm(1,f_position(i));
end
for i=1:length(f_position_2)
    amp_ver_2(i) = L_P1_2_Norm(1,f_position_2(i));
end

h=plot(f_ver,amp_ver, f_ver_2,amp_ver_2);

h(1).LineWidth=2;

```

```

h(1).Color=[0 0.4470 0.7410];
h(2).LineWidth=2;
h(2).Color=[0.8500 0.3250 0.0980];

axis([50 500 -3.5 0.5])
% axis([50 500 -4.5 0.5])
% axis([50 500 -0.5 0.5])

xlabel ('Frequenz in [Hz]', 'FontSize',28,'FontName','Arial')
ylabel('L_{A,S} (re A_{max}) in [dB]', 'FontSize',28,'FontName','Arial')

legend('Mik. 3; Rechts','Mik. 4; Links')

grid on

ax = gca;
set(ax, 'FontName', 'Arial', 'FontSize', 28, 'GridAlpha', 0.3, 'LineWidth', 1.7);
ax.TickDir='out';

```

12. Frequenzgang Unity Steam Audio

% Skript für die Berechnung des Frequenzganges für die simulierte Umgebung
% Unity Steam Audio

```
clc
clear
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Auslesen der Audiodateien, rechts und links
```

```
[data, fs] = audioread('FreqSwipe_rechts.wav');
[data2, fs_2] = audioread('FreqSwipe_links.wav');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FFT Analyse
```

```
T = 1/fs;
T_2 = 1/fs_2;
L = length(data);
L_2 = length(data2);
t = (0:L-1)*T;
t_2 = (0:L_2-1)*T_2;
f = fs*(0:(L/(fs/500)))/L;
f_2 = fs_2*(0:(L_2/(fs_2/500)))/L_2;
data_fft = fft(data);
data2_fft = fft(data2);
```

```
P2 = abs(data_fft)/(L/2);
P2_2 = abs(data2_fft)/(L_2/2);
P1 = P2(1:L/(fs/500)+1);
P1_2 = P2_2(1:L_2/(fs_2/500)+1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Normierung der Amplituden, amp_i/amp_max
```

```
amp_max = max(P1);
amp_max_2 = max(P1_2);
```

```
for i=1:length(P1)
    P1_Norm(i) = P1(i)/amp_max;
    L_P1_Norm(i)=20*log10(P1_Norm(i));
end
for i=1:length(P1_2)
    P1_2_Norm(i) = P1_2(i)/amp_max_2;
    L_P1_2_Norm(i)=20*log10(P1_2_Norm(i));
end
End
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot Amplituden-Verlauf

f_ver(1) = f(1);
f_ver_2(1) = f_2(1);
amp_ver(1) = P1_Norm(1);
amp_ver_2(1) = P1_2_Norm(1);
k=5;
f= f';
f_2=f_2';
ff = find(f <=5,1,'last');
ff_2 = find(f_2 <=5,1,'last');

for i=1:500
    f_position(i) = find(f<=k,1,'last');
    f_position_2(i) = find(f_2<=k,1,'last');
    f_ver(i) = f(f_position(i));
    f_ver_2(i) = f_2(f_position_2(i));
    k = k+5;
end

figure(3)
for i=1:length(f_position)
    amp_ver(i) = L_P1_Norm(1,f_position(i));
end
for i=1:length(f_position_2)
    amp_ver_2(i) = L_P1_2_Norm(1,f_position_2(i));
end
h=plot(f_ver,amp_ver, f_ver_2,amp_ver_2);

h(1).LineWidth=2;
h(1).Color=[0 0.4470 0.7410];
h(2).LineWidth=2;
h(2).Color=[0.8500 0.3250 0.0980];

axis([50 500 -0.05 0])

xlabel ('Frequenz in [Hz]', 'FontSize',28,'FontName','Arial')
ylabel('L_{A,S} (re A_{max}) in [dB]', 'FontSize',28,'FontName','Arial')

legend('Mik. 3; Rechts','Mik. 4; Links')

grid on

ax = gca;
set(ax,'FontName','Arial','FontSize',28, 'LineWidth', 1.7,'GridAlpha',0.3);
ax.TickDir='out';

```

13. Empfindlichkeit Messmikrofone

% Das Skript stellt die Empfindlichkeit der 4 Messmikrofonen in dem
% Frequenzbereich 50 bis 1000 Hz dar

```
clc  
clear  
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Importieren Werte Empfindlichkeit der 4 Messmikrofonen  
file=load('E_Mikro.txt', '-ascii');  
freq=file(:,1);  
E_M1=file(:,2);  
E_M2=file(:,3);  
E_M3=file(:,4);  
E_M4=file(:,5);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Graphen erstellen
```

```
M1=fit(freq,E_M1,'pchip');
```

```
figure1 = figure('WindowState','maximized');  
axes1 = axes('Parent',figure1);  
hold(axes1,'on');  
M1_g=plot(M1,freq,E_M1);  
hold on  
M2=fit(freq,E_M2,'pchip');  
M2_g=plot(M2,freq,E_M2);  
M3=fit(freq,E_M3,'pchip');  
M3_g=plot(M3,freq,E_M3);  
M4=fit(freq,E_M4,'pchip');  
M4_g=plot(M4,freq,E_M4);  
hold off
```

```
set([M1_g(2)], 'color', 'r', 'LineWidth',2)  
set([M1_g(1)], 'Marker','none')  
set([M2_g(2)], 'color', 'b', 'LineWidth',2)  
set([M2_g(1)], 'Marker','none')  
set([M3_g(2)], 'color', 'g', 'LineWidth',2)  
set([M3_g(1)], 'Marker','none')  
set([M4_g(2)], 'color', 'magenta', 'LineWidth',2)  
set([M4_g(1)], 'Marker','none')  
xlabel('Frequenz in [Hz]')  
ylabel('Empfindlichkeit E_M in [mV/Pa]')  
set(axes1, 'FontName', 'Arial', 'FontSize', 28, 'GridAlpha', 0.3, 'Tick-  
Dir', 'out', ...  
    'XGrid', 'on', 'YGrid', 'on', 'LineWidth', 1.7);  
legend([M1_g(2) M2_g(2) M3_g(2) M4_g(2)], 'Mik. 1', 'Mik. 2', 'Mik. 3', 'Mik.  
4')
```

14. Auslesen FEM Datei

```
% Skript zum Auslesen der FEM-Datei und Ermittlung der Amplituden und
% Phasen.

clc
clear
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Importieren der FEM-Dateien

% file1 = textread('Kartierung FEM 170Hz E1.txt','%s');
% file1 = textread('Kartierung FEM 180Hz E1.txt','%s');
file1 = textread('Kartierung FEM 170Hz E2.txt','%s');
% file1 = textread('Kartierung FEM 180Hz E2.txt','%s');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erstellung der Positionsvektoren x,y,z sowie der Vektoren "amp" und
% "phase", die die Werte für die Amplitude und Phase enthalten. Diese
% Vektoren werden vom Benutzer gespeichert, um im Skript
% "Erstellung_dataFEM" ausgelesen zu werden. Der Vektor z wird als Z.mat ge-
% speichert
% wenn die Messungen der Ebene 1 entsprechen, beziehungsweise Z2.mat wenn die
% Messungen der Ebene 2 entsprechen.

k = 1;

for i=4:4:length(file1)
    x(k) = str2double(file1(i-3));
    y(k) = str2double(file1(i-2));
    z(k) = str2double(file1(i-1));
    comp(k) = str2double(file1(i));
    k=k+1;
end

amp = zeros(1,length(comp));
phase = zeros(1,length(comp));

for i=1:length(comp)
    amp(i) = abs(comp(i));
    phase(i) = atan2(imag(comp(i)),real(comp(i)));
end
```

15. Erstellung dataFEM

% Skript für die Erstellung der "data_FEM" Datei mit den Dateien
% x,y,z,amp,phase, die aus dem Skript "Auslesen_FEM_Datei" exportiert wurden.

```
clc  
clear  
close all
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Je nachdem welche Ebene gemessen wurde, muss man entweder "load('Z.mat')"  
% oder "load('Z2.mat')" aktivieren.
```

```
% load('Z.mat') wenn z=220  
% load('Z2.mat') wenn z=520
```

```
load("X.mat");  
load('Y.mat');  
% load('Z.mat');  
load('Z2.mat');
```

```
% load('amp_170_E1.mat');  
% load('phase_170_E1.mat');
```

```
% load('amp_180_E1.mat');  
% load('phase_180_E1.mat');
```

```
% load('amp_170_E2.mat');  
% load('phase_170_E2.mat');
```

```
load('amp_180_E2.mat');  
load('phase_180_E2.mat');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Die Variable "data_FEM" muss für die Generierung des Tones als  
% "data_FEM_X_Y.mat" abgespeichert werden. (X = Frequenz, Y = Ebene)
```

```
data_FEM = [x;y;z;amp;phase]';
```

16. Verbindung Matlab und Unity

% Skript für die Live-Verbindung von Unity mit Matlab

```
clear;
clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TCP/IP Server erstellen
tcpipServer = tcpip('0.0.0.0',1234,'NetworkRole','Server');
tcpipServer.ReadAsyncMode = 'continuous';
tcpipServer.InputBufferSize=32;
tcpipServer.Timeout = 30;
fopen(tcpipServer);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dateiinitialisierung
position = zeros(1,3);
X = 0;
Y = 0;
Z = 0;
k=1;
% w_170=2*pi*170;
w_180=2*pi*180;

secs = 5;
FS = 44100;
t = linspace(0, secs, FS*secs+1);
t(end) = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Importieren der Dateien für die Erstellung des Tones. Je nachdem welcher Ton
% abgespielt werden soll, müssen die entsprechenden Zeilen (41,42,43,44) und (26,27) aktiviert
% werden.

% load('dataFEM_170_E1.mat');
load('dataFEM_180_E1.mat');
% load('dataFEM_170_E2.mat');
% load('dataFEM_180_E2.mat');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Daten empfangen und bearbeiten

while(1)

    while(tcpipServer.BytesAvailable>0)
        rawData = fread(tcpipServer);
        rawData = char(rawData);
        i=1;
        z=1;
        tempMatrix="";

        while (i<32)
            a=any(rawData(i)==char(32)); %ASCII 32 <=> "SPACE"

            if(a==1)
                position(z) = str2double(tempMatrix);
                tempMatrix="";
                z=z+1;
            end
        end
    end
end
```

```

end

b=any(rawData(i)==newline);

if (b==1)
    position(z) = str2double(tempMatrix);
    tempMatrix="";
    z=1;
end
if (a==0 && b==0)
    tempMatrix = append(tempMatrix,rawData(i));
end
i=i+1;
end
end
clear rawData;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Aktuelle Position anzeigen / Ton abspielen

if (X~= position(1)) || (Y~= position(2)) || (Z~= position(3))
    clear sound;
    X = position(1);
    y_comsol = 351-X*1000;
    Y = position(2);
    z_comsol = (Y-0.02)*1000;
    Z = position(3);
    x_comsol = (1+Z)*1000;
    T = table(X,Y,Z);
    T_Comsol = table(x_comsol, y_comsol, z_comsol)
    k = find(round(data_FEM(:,1)) == round(x_comsol) & round(data_FEM(:,2)) == round(y_comsol));
    s = data_FEM(k,4)*cos(w_180*t+data_FEM(k,5));
    sound(s,FS);
end
end

```

Anhang G

Unity Code für die Erstellung der TCP/IP Verbindung

```
using UnityEngine;
using System.IO;
using System.Net.Sockets;
using System.Threading.Tasks;
using JetBrains.Annotations;
using System;

public class ComClient : MonoBehaviour
{
    double X, Y, Z;
    double dataX;
    double dataY;
    double dataZ;
    TcpClient client;
    NetworkStream theStream;
    readonly String Host = "localhost";
    readonly Int32 Port = 1234;

    void Awake()
    {
        client = new TcpClient(Host, Port);
        try
        {
            theStream = client.GetStream();
        }
        catch (Exception e)
        {
            Debug.Log("Socket error: " + e);
        }
    }

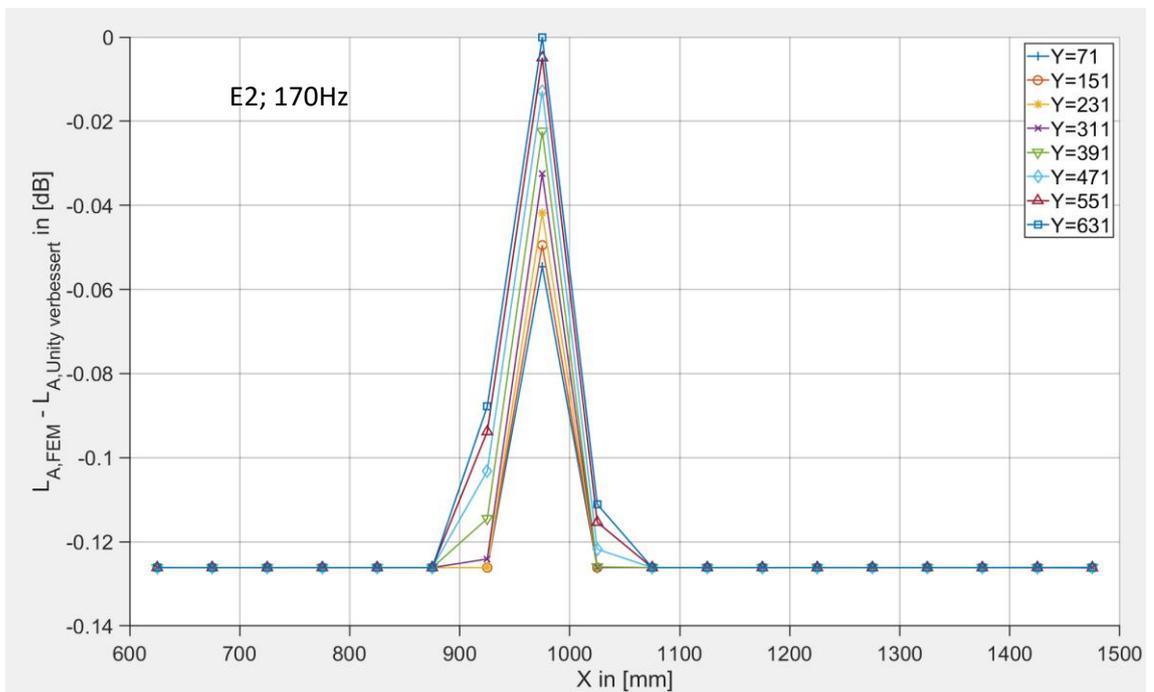
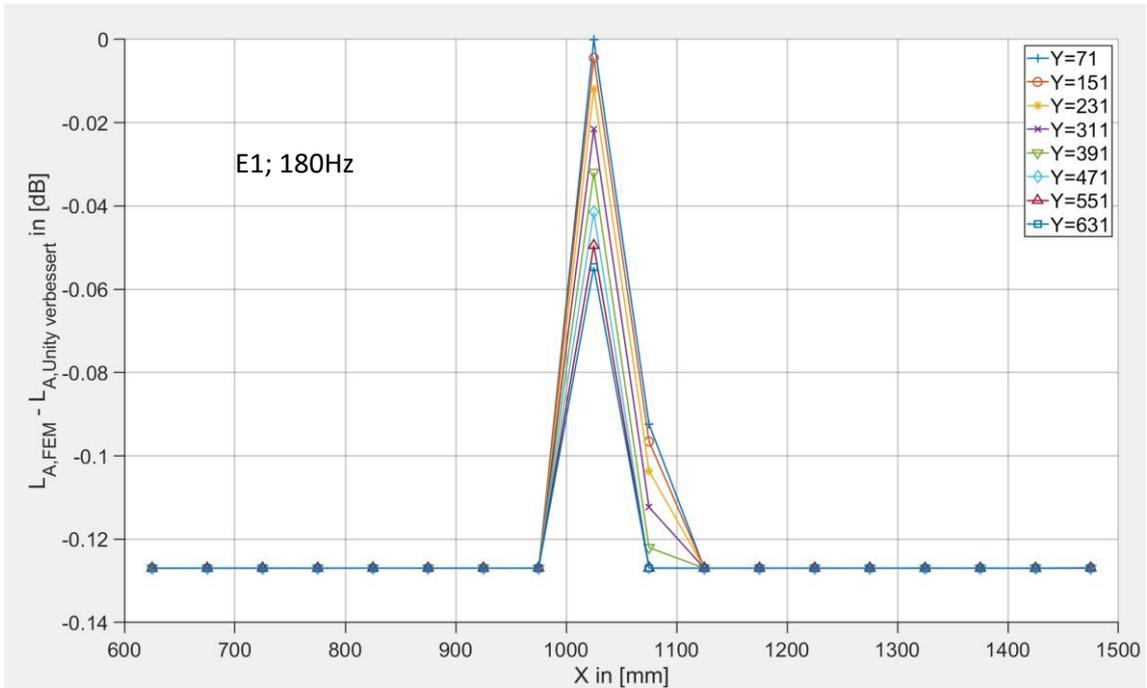
    public void Update()
    {
        dataX = Math.Round(transform.position.x, 2);
        dataY = Math.Round(transform.position.y, 2);
        dataZ = Math.Round(transform.position.z, 2);

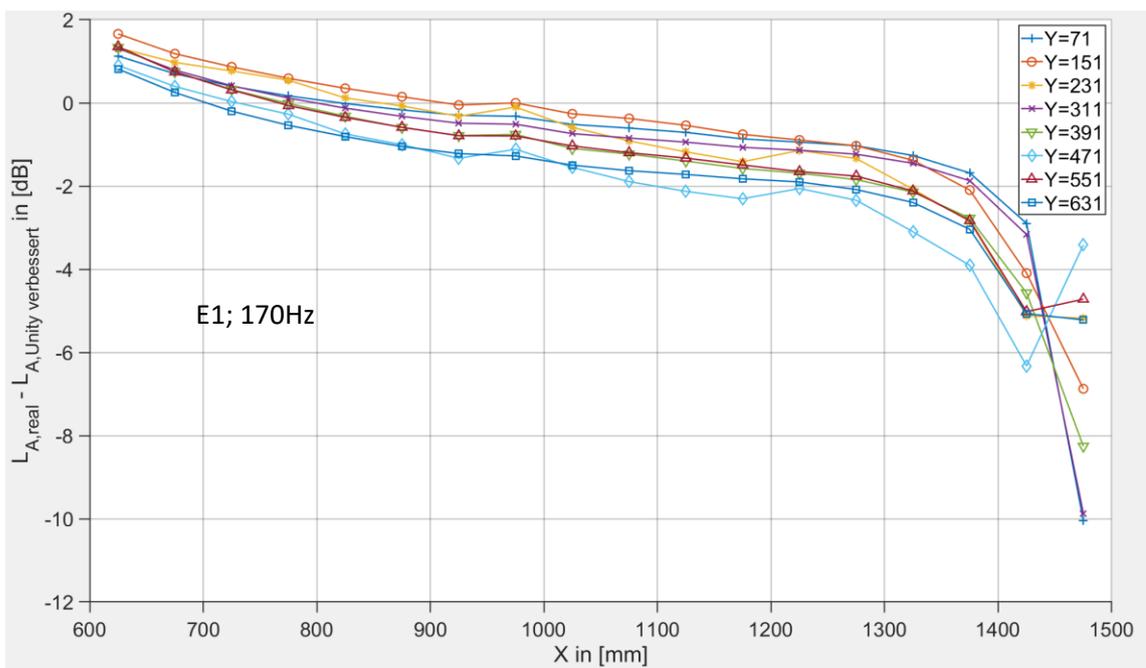
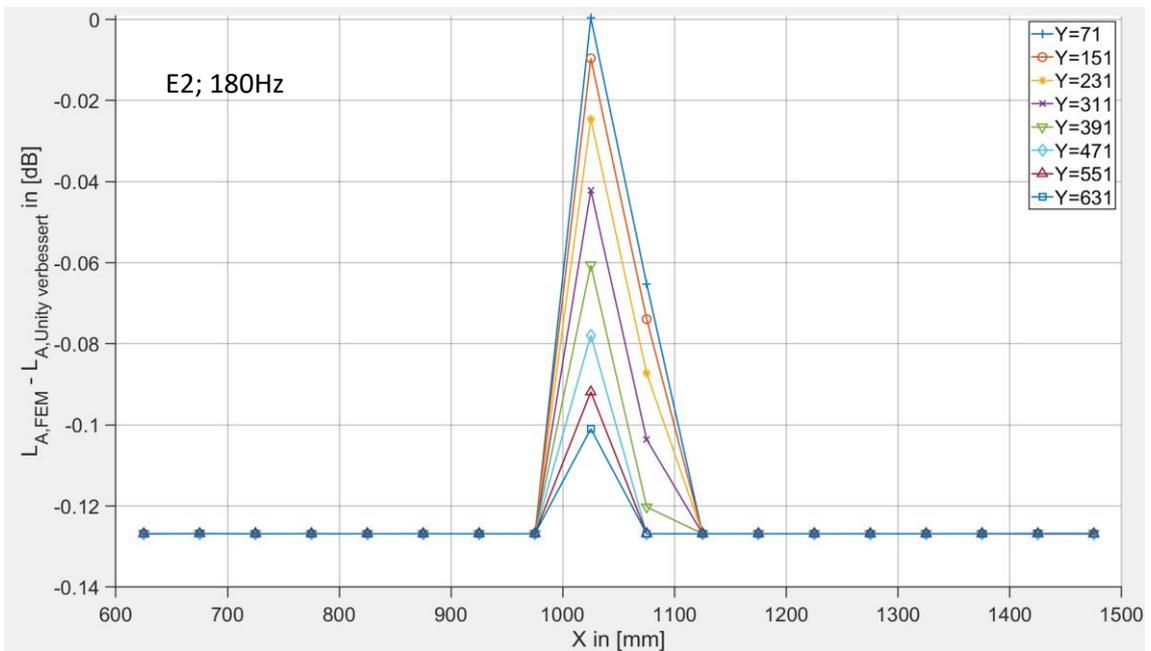
        print("X:" + dataX.ToString() + " " + " Y:" + dataY.ToString() + " " + "
Z:" + dataZ.ToString());
        try
        {
            Byte[] sendBytesdataX = System.Text.Encoding.UTF8.Get-
Bytes(string.Concat(dataX.ToString(), ' '));
            Byte[] sendBytesdataY = System.Text.Encoding.UTF8.Get-
Bytes(string.Concat(dataY.ToString(), ' '));
            Byte[] sendBytesdataZ = System.Text.Encoding.UTF8.Get-
Bytes(string.Concat(dataZ.ToString(), '\n'));

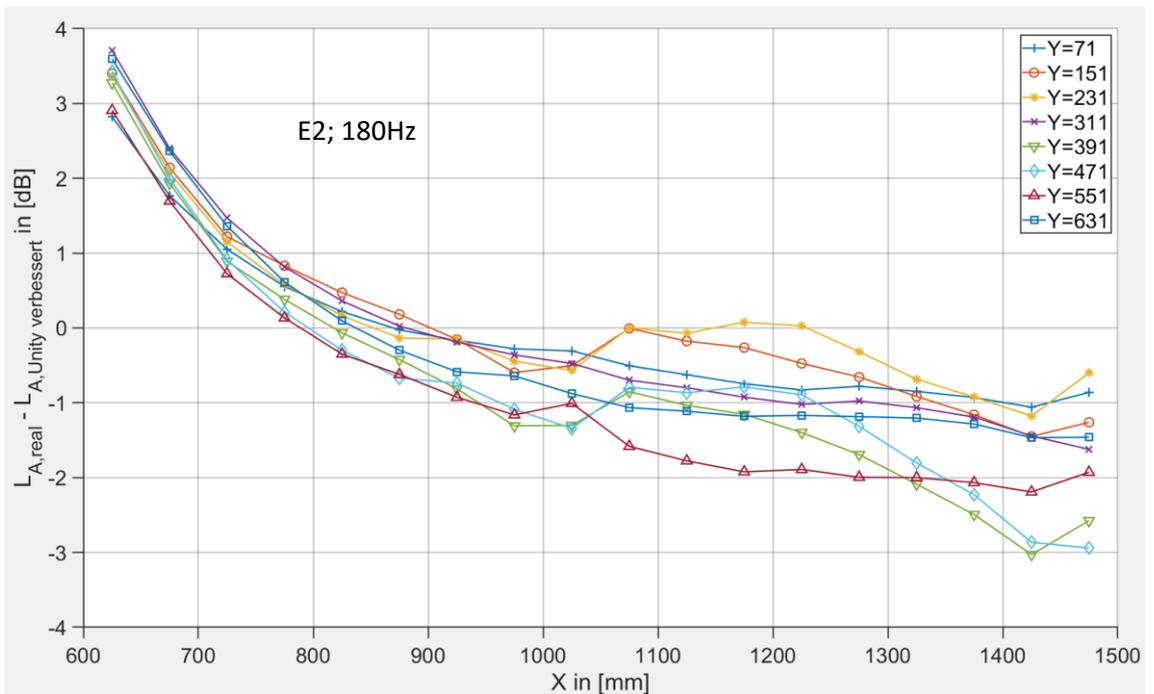
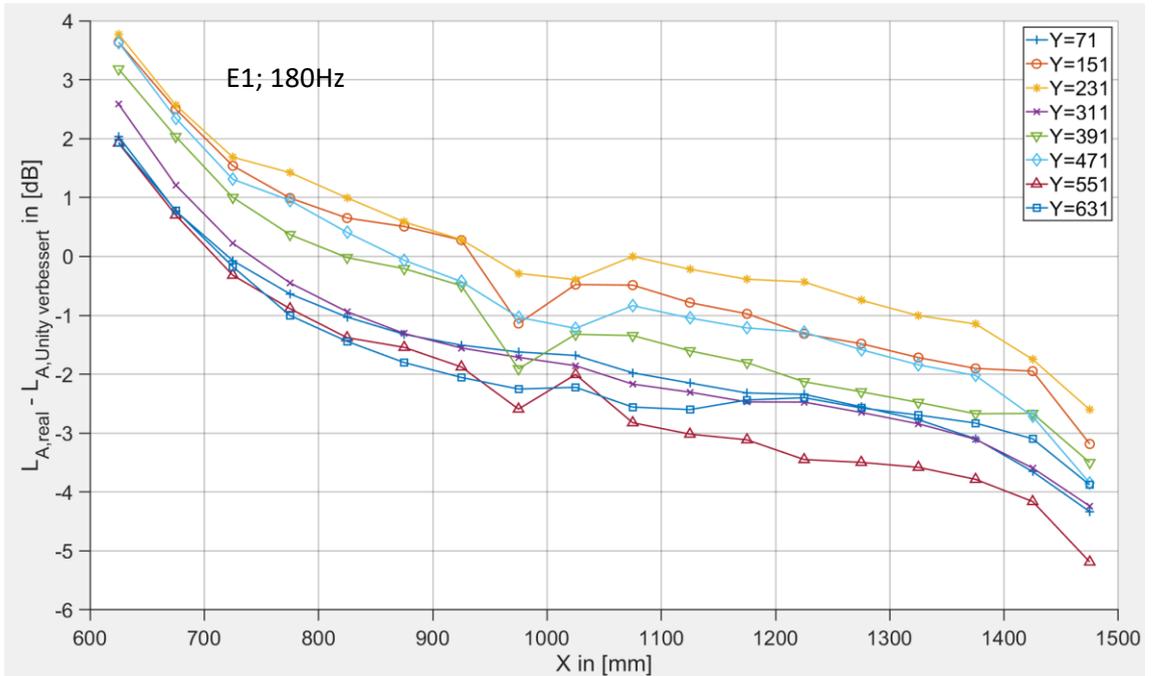
```

```
        theStream.Write(sendBytesdataX, 0, sendBytesdataX.Length);
        theStream.Write(sendBytesdataY, 0, sendBytesdataY.Length);
        theStream.Write(sendBytesdataZ, 0, sendBytesdataZ.Length);
    }
    catch (Exception e)
    {
        Debug.Log("Socket error: " + e);
    }
}
```

Anhang H









Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Bodolea

Vorname: Mihai

dass ich die vorliegende Masterarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Entwicklung einer Virtual Reality Umgebung für einen realitätsnahen Höreindruck

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Hamburg

Ort

06.09.2022

Datum

[Signature]
Unterschrift im Original