'Zur Hörbarkeit und Interpolation von Kopf-über-Torso-Orientierungen bei Aufnahmen mit einem Kopf-und-Torso-Simulator'

'Audibility and interpolation of head-above-torso-orientation for recordings made with a head and torso simulator'

Gutachter:
Prof. Dr. Matthias Blau (Jade-HS Oldenburg)
Prof. Dr. Stefan Weinzierl (TU Berlin)
Fabian Brinkmann M.A. (TU Berlin)

27. September 2013

Bachelorarbeit vorgelegt von: Reinhild Popko

[redacted]

Technische Universität Berlin
Fakultät I - Fachgebiet Audiokommunikation
Einsteinufer 17c
D-10587 Berlin

Institut für Hörtechnik und Audiologie
Jade Hochschule Oldenburg/Wilhelmshaven/Elsfleth
Fachbereich Bauwesen und Geoinformation (B+G)
Ofener Str. 16-19
D-26121 Oldenburg

Eidesstattliche Erklärung

Hiermit versichere ich gegenüber der Jade-Hochschule Oldenburg durch meine Unterschrift die vorliegende Arbeit selbstständig ohne fremde Hilfe und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst zu haben. Alle Ausführungen, die anderen veröffentlichten oder nicht veröffentlichten Schriften wörtlich oder sinngemäß entnommen wurden, habe ich kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Fassung noch keiner anderen Prüfungsbehörde vorgelegen.

Berlin, den 27. September 2013

.....
Reinhild Popko

Danksagung

Herrn Prof. Matthias Blau, Prof. Stefan Weinzierl, Alexander Lindau und vor allem Fabian Brinkmann möchte ich für die fachliche Unterstützung durch ihre große Expertise danken; Professoren, Mitarbeitern und Kommilitonen des Instituts für Hörtechnik und Audiologie möchte ich für eine interessante, lehrreiche und schöne Studienzeit in Oldenburg danken, sowie dem Fachgebiet der Audiokommunikation und -technologie der TU-Berlin für die freundliche Zusammenarbeit in der Abschlussphase des Studiums.

Ein besonderer Dank gilt meiner Familie.

Zusammenfassung

Kopfbezogene Übertragungsfunktionen (HRTFs) sind die Grundlage für ein räumliches Hören und deshalb Fokus der Simulation von akustischen Umgebungen wie z.B. durch die Binauraltechnik. Deren Bestreben ist eine realitätstreue und artefaktfreie, aber gleichzeitig effiziente Messung und Synthese von HRTF-Daten. Eine oft gemachte Vereinfachung ist das Auralisieren von Kopfbewegungen mittels HRTFs verschiedener Quellpositionen bei statischer Kopf-über-Torso-Orientierung anstelle von HRTFs dynamischer Kopf-über-Torso-Orientierung (HATO) und gleichbleibender Quellposition. Der dabei unberücksichtigte Einfluss des Torsos, der in einer natürlichen Kopfbewegung seine Position bezüglich Außenohr und damit das Schallfeld ändert, wurde in vorliegender Studie physikalisch und hinsichtlich seiner Hörbarkeit perzeptiv in einem ABX-Vergleich evaluiert. Ausgangspunkt dafür war der gezeigte Kammfilter des Torsos und der Schultern, der die Lokalisation elevierter Quellen unterstützt [AAD01].

Dem ABX-Versuch dieser Studie lag ein fein aufgelöster HRTF-Datensatz zugrunde, der mit einem Kopf-und-Torso-Simulator gemessen wurde. Es konnte der hörbare Unterschied zwischen dynamischer HATO und benannter Vereinfachung nachgewiesen werden, was die Notwendigkeit dem Hörenden möglichst exakte HRTFs darzubieten zeigte.

Um den Messaufwand von HRTFs verschiedener HATOs zu verringern, wurden verschiedene Interpolationsalgorithmen angewendet und physikalisch evaluiert. Hervorzuheben ist der Ansatz einer Interpolation zwischen zwei HRTFs unterschiedlicher Torso-zu-Quell- und gleicher Kopf-zu-Quell-Orientierung, der sich als vorteilhafter als der umgekehrte Fall (gleiche Torso-zu-Quell- und unterschiedliche Kopf-zu-Quell-Orientierung) erwies.

Stichworte: auditive Wahrnehmung; Binauraltechnik; räumliches Hören; kopfbezogene Übertragungsfunktion (HRTF); Kopf-und-Torso-Simulator; Kopf-über-Torso-Orientierung; ABX-Test; HRTF-Interpolation;

Abstract

Head-related transfer functions (HRTFs) are the basis of spatial hearing and therefore focused by the simulation of acoustical environments e.g. by binaural synthesis. Its effort is to reproduce HRTF data without audible artifacts with simultaneous consideration of efficiency. Convinient simplification is replacing HRTFs with varying head-above-torso orientation (HATO) by HRTFs with static HATOs and including a corresponding source movement. However, the torso changes the acoustic field by changing its position relative to the outer ear in a natural head movement. Therefore, the present investigation observes the disregarded influence of the torso, from two differnt views: physically and perceptively in regard to its audibility. The evaluation is based on a fine-resolved dataset of HRTFs measured with a head and torso simulator. This approach refers to the results of Algazi et. al [AAD01] showing the comb filter caused by torso and shoulders, which supports localization of elevated sources.

In the current study a conducted ABX listening test showed evidence for an audible difference between dynamic HATOs and the simplification given above. This result endorses the providing accurate HRTFs to the listener. The interpolation of HRTFs with varying head-above-torso orientation were investigated physically in order to reduce an extra effort of measuring. Algorithmns interpolating between two HRTFs

with varying torso to source and same head to source orientation are more precisely than vice versa (same torso to source and varying head to source orientation).

KEYWORDS: auditive perception; binaural synthesis; spatial hearing; Head related transfer function (HRTF); head and torso simulator; head above torso orientation; ABX listening test; HRTF interpolation;

Inhaltsverzeichnis

1 Einleitung und Forschungsfragen **1**

2 Stand der Forschung **4**
2.1 Räumlich Hören 4
2.2 Binauraltechnik 6
2.3 Einfluss des Torsos bei statischer HATO 8
2.4 Zu berücksichtigende Bewegungsbereiche 9
2.5 Kunstkopf 10
2.6 Interpolationsalgorithmen 10
2.7 Zusammenfassung - Stand der Forschung 12

3 HRTF Messung und Processing **13**
3.1 HRTF Messungen 13
3.1.1 Ort, Zeit der Messung 13
3.1.2 Koordinatenkonvention, Messpunkte 13
3.1.3 Messaufbau 15
3.1.3.1 Vorbereitung 15
3.1.3.2 Geräteliste 16
3.1.3.3 Aufbau FABIAN 18
3.1.3.4 Temperatur 18
3.1.3.5 Quellpositionen 18
3.1.3.6 Aufbau Referenzmessung 20
3.1.4 Verlässlichkeit des Aufbaus 20
3.1.4.1 SNR-Level, Reflexionstest 20
3.1.4.2 Mechanische Verlässlichkeit/Reliabilität 22
3.1.4.3 Soundkartendelay 23
3.1.4.4 Lautsprecherbelastungstest 23
3.1.5 Testsignal 23
3.2 HRTF Processing 25
3.2.1 Pre-Processing 25
3.2.2 Entfaltung 25
3.2.3 Radius-Schätzung 26
3.2.4 Post-Processing 27
3.2.4.1 Schätzung der interauralen Zeitdifferenzen ITD 27
3.2.4.2 Korrektur der Temperaturveränderung 27
3.2.4.3 Kürzung der Dateien und Fensterung 28
3.2.4.4 Generierung der Minimalphase 28
3.3 Zusammenfassung - Messung und Processing 28

4 Torsoeinfluss in HRTFs **29**
4.1 Physikalische Evaluation . . . 30
4.1.1 Differenzspektren . . . 31
4.1.2 Physikalisch-perzeptives Einzahlmaß . . . 32
4.1.3 Einfluss auf ILD und ITD . . . 34
4.2 Perzeptive Evaluation . . . 34
4.2.1 Grundlegendes zum ABX-Test . . . 34
4.2.2 Wahl der Stimuli . . . 35
4.2.3 Stichprobengröße . . . 35
4.2.4 Probanden . . . 36
4.2.5 Geräte, Signalpfad . . . 37
4.2.6 Ablauf und Instruktion des Probanden . . . 40
4.2.7 Erhobene Daten . . . 41
4.2.8 Statistische Methode und Auswertung . . . 41
4.2.8.1 Hörbarkeitshypothese - individuell betrachtet . . . 41
4.2.8.2 Hörbarkeitshypothese - über die Summe aller Ergebnisse betrachtet . . . 43
4.2.8.3 Ablehnung einer Gleichverteilungsannahme für Sprache und Rauschen . . . 44
4.2.8.4 Ablehnung einer Gleichverteilungsannahme für Quellpositionen . . . 44
4.3 Zusammenfassung der Evaluation des Torsoeinflusses . . . 45

5 HRTF Interpolation **46**
5.1 Stützstellen . . . 47
5.2 Algorithmen . . . 48
5.2.1 Nearest neighbour „via Kopf“ . . . 49
5.2.2 Linear „via Kopf“ im Zeitbereich . . . 50
5.2.3 Spline „via Kopf“ im Zeitbereich . . . 50
5.2.4 Linear „via Kopf“ im Frequenzbereich . . . 51
5.2.5 Spline „via Kopf“ im Frequenzbereich . . . 51
5.2.6 Linear „via Torso“ im Zeitbereich . . . 51
5.2.7 Linear „via Torso“ im Frequenzbereich . . . 52
5.3 Physikalische Evaluation der Interpolationsalgorithmen . . . 52
5.3.1 Minimalphasige HRTFs bei Kopfbewegung innerhalb der Horizontalebene . . . 52
5.3.1.1 Spektraler Vergleich . . . 52
5.3.1.2 Differenzspektren . . . 55
5.3.1.3 Physikalisch-perzeptives Einzahlmaß . . . 56
5.3.1.4 ILD . . . 57
5.3.1.5 ITD . . . 60
5.3.2 Minimalphasige HRTFs bei Kopfbewegungen innerhalb der Median- und Frontalebene . . . 62
5.3.3 Originalphasige HRTF . . . 63
5.4 Zusammenfassung der Interpolation . . . 63

6 Fazit und Ausblick **64**

A Abbildungen **74**
i zur Hörbarkeit . . . 74
ii zur Interpolation . . . 87

B Unterlagen ABX-Test **99**

C Tabellenanhang **105**

D Quellcode Matlab **107**
i HRTF Processing . . . 107
a path_definition_and_setup . . . 107
b pre_processing . . . 111
c deconvolution . . . 114
d radius_estimation . . . 117
e post_processing . . . 119
ii abx_analyse . . . 124
iii Interpolation . . . 130
a hrtf_interpolation . . . 130
b catch2interpol . . . 135
c interpolate_head_freq . . . 140
d interpolate_torso_freq . . . 144
e interpolate_head_time . . . 147
f interpolate_torso_time . . . 152
g in_time . . . 155
h in_freq . . . 159

E Datenträger **162**
i Arbeit im PDF-Format . . . 162
ii I_Problemvisualisierung . . . 162
iii II_Messung_Processing . . . 162
iv III_Torsoeinfluss . . . 162
v IV_Interpolation . . . 162

Abkürzungsverzeichnis

BRIR binaurale Raumimpulsantwort/Binaural Romm Impulse Response

BRTF binaurale Raumübertragungsfunktion/Binaural Room Transfer Function

DFT diskrete Fouriertransformation/Discrete Fourier Transformation

ERB äquivalente Rechteckbandbreite/Equivalent Rectangular Bandwidth

FABIAN verwendeter Kopf-Torso-Simulator: Fast and Automatic Binaural Impulse response AcquisitioN [LW06]

HATO Kopf-über-Torso-Orientierung/Head Above Torso Orientation

HATS Kopf-und-Torso-Simulator/Head-And-Torso-Simulator

HRTF kopfbezogene Übertragungsfunktion/Head Related Transfer Function

HRIR kopfbezogene Impulsantwort/Head Related Impulse Response

ILD interauraler Pegelunterschied/Interaural Level Difference

IPD interauraler Phasenunterschied/Interaural Phase Difference

ITD interauraler Zeitunterschied/Interaural Time Difference

JND Schwellwert des eben wahrnehmbaren Unterschieds/Just Noticeable Difference

SEACEN Simulation und Evaluation akustischer Umgebungen/Simulation and Evaluation of Acoustical EnvironmeNts

Kapitel 1

Einleitung und Forschungsfragen

Die Binauraltechnik ist ein Verfahren reale akustische Umgebungen nachzubilden und sie virtuell erfahrbar zu machen. Sie umfasst die Aufnahme bzw. Synthetisierung und das Auralisieren bzw. Resynthetisieren binauraler Signale am Ohr oder auch direkt am Trommelfell des Zuhörers [Nic10, vgl. S. xii]. Diese Auralisierung einer im virtuellen Raum lokalisierbaren Schallquelle ist in vielen konkreten Anwendungen wie etwa bei der Simulation von Räumen zur besseren Planung von Raumakustik, in assistiven Technologien für sehbehinderte Menschen oder aber im multimedialen Zusammenhang gefragt.

Die Qualität des Syntheseergebnisses ist maßgeblich abhängig von der Aufnahme und Bearbeitung jener Signale, die zur Synthese verwendet werden, und stellt sich in der Abweichung der gehörten Reproduktion des Schallfelds zu einer Referenz dar. Neben Eigenschaften des zu simulierenden Raumes sind vor allem kopfbezogene Übertragungsfunktionen (HRTFs) das grundlegende Teilstück der Schallausbreitung von der Quelle zum linken und rechten Trommelfell in der Binauraltechnik. Im Falle von modellbasierter binauraler Synthese von Räumen können die binauralen Raumübertragungsfunktionen (BRTFs) aus der Überlagerung von kopfbezogenen HRTFs konstruiert und somit die Ergebnisse daraufhin verallgemeinert werden (HRTFs, BRTFs, siehe 2.2). HRTFs enthalten alle inter- und monauralen Informationen für ein räumliches Hören. Neben zeitlichen Signalmodifizierungen (durch z.B. Laufzeitunterschiede) werden die Spektren vor der physiologischen Weiterleitung zur auditiven Wahrnehmung durch Abschattungs-, Reflexions- und Diffusionseffekte an Pinna, Kopf, Schultern, Torso, Nase und Augenhöhlen gefiltert; auch die individuelle Form des Gehörgangs und die akustische Impedanz des Trommelfells nehmen Einfluss [Mø92]. Hier vorgestellte Untersuchungen beschränken sich auf den Schallweg im Freifeld bis zu den geblockten Ohrkanälen. Da je nach Anwendung eine verbesserte Schallquellenlokalisierung und z.T. größtmögliche Authentizität und Realitätsnähe bzw. Plausibilität der virtuellen Nachbildung angestrebt wird [LW12], ist es notwendig jeden auf die Simulation Einfluss nehmenden Parameter genau zu untersuchen.

Die Simulation von Kopfbewegungen ist für eine authentische und plausible Nachbildung des Schallfelds maßgeblich und bedingt auch die Lokalisation von Quellen. Werden Kopfbewegungen berücksichtigt, ist von einer dynamischen Binauralsynthese zu sprechen. Hier knüpfte die erste Frage des behandelten Forschungsthemas an. Es sollte der Einfluss der Kopf-über-Torso-Orientierung (HATO) und damit die Wirkung der Torsoposition auf das Schallfeld am rechten und linken geblockten

Ohrkanal[1] hinsichtlich der Hörbarkeit untersucht werden. Bisher ist es gängige Praxis eine Kopfbewegung aus Messungen einer Kopf-zu-Torso-Orientierung bei mehreren Quellen zu modellieren; der menschliche oder künstliche Oberkörper bleibt statisch an seiner Position, während entlang einem Lautsprecher-Array Messsignale abgespielt und am Ohrkanal oder Trommelfell aufgenommen werden. Dieses Verfahren ist etwas genauer bezüglich der Winkeländerung bei Messung mit Menschen, da es leichter ist die Versuchsperson in einer Position verharren zu lassen als ihre Kopf-über-Torso-Position mehrmals neu auszurichten, und eignet sich für Kunstköpfe mit Torso ohne Halsgelenk. Dabei bleiben jedoch die möglichen Auswirkungen durch die sich ändernde Position von Kopf zu Torso einer natürlichen Kopfdrehung unbeachtet. Schallreflexionen oder Abschattungseffekte durch den Torso bzw. an den Schultern sind in der Literatur zu kopfbezogenen Übertragungsfunktionen (HRTF) für statische Kopf-zu-Torso-Orientierungen gezeigt (dessen und dem weiteren wissenschaftlichen Kontext, an den diese Arbeit gegliedert ist, widmet sich das Kapitel 2). Hier vorgestellte Betrachtungen auf physikalischer und perzeptiver Ebene untersuchen hinsichtlich einer dynamische Binauralsynthese Kopfbewegungen fein aufgelöster Kopf-über-Torso-Orientierungen (HATOs) vor allem in der Horizontalebene, sowie der Median- und Frontalebene. Hierzu wurde ein ABX-Hörversuch zum Vergleich der Kopf-über-Torso-Bewegung in der Horizontalebene mit entsprechender Quellbewegung bei statischem Halsgelenk durchgeführt, der die Hörbarkeit diesen Unterschieds zeigen konnte. Damit ist die Notwendigkeit einer Torsoberücksichtigung für eine getreue Reproduktion gegeben (Kap. 4).

Da neben der Realitätstreue auch stets ein möglichst effizientes Verfahren für die Virtualisierung angestrebt wird, beschäftigte sich ein zweiter Forschungsgegenstand mit der Interpolationsproblematik aufwendig zu messender HRTFs benachbarter Kopf-zu-Torso-Orientierungen (Kap. 5). Zwei Herangehensweisen wurden dabei verfolgt, die mit verschiedenen Algorithmen im Zeit und Frequenzbereich umgesetzt wurden. Zum einen können zwei HRTFs unterschiedlicher Kopf-zu-Quell-Winkel bei gleicher Torso-zu-Quell-Ausrichtung interpoliert werden („Interpolation über den Kopf“). Zum anderen ist es denkbar die gesuchte HRTF bzw. HRIR einer bestimmten HATO aus zwei Stützstellen bei gleichem Kopf- und unterschiedlichem Torso-zu-Quell-Winkel zu errechnen. Allgemein reduziert sich der Messaufwand maßgeblich je gröber das Raster von notwendigen Messpunkten sein kann. Es galt demnach im Weiteren den sich ergebenden physikalischen Fehler implementierter Interpolationsalgorithmen bei unterschiedlichen Rasterweiten zu ermitteln. Anhand dieser Betrachtung können bereits Tendenzen für Ergebnisse einer zukünftigen perzeptiven Evaluation hinsichtlich des Interpolationsverfahrens und -intervalls an der Schwelle der Hörbarkeit von Artefakten geschätzt werden.

Vorliegender Text orientiert sich an der Abfolge notwendig gewesener Arbeitsschritte - hinführend zur Beantwortung der Forschungshypothesen. Nach dem Stand der Forschung wird daher zunächst auf die Messung und die Signalverarbeitung eines HRTF-Datensatzes eingegangen (Kap. 3), der für eine eingeschränkte Anzahl von Schallquellen und eine feine Auflösung von Kopf-zu-Torso-Ausrichtungen in drei Freiheitsgraden mittels einem Kopf-und-Torso-Simulator angelegt wurde. Daten und Ergebnisse basieren daher auf nicht individuellen HRTFs hoher Reliabilität.
Diese Studie ist Teil des von der Deutschen Forschungsgemeinschaft (DFG) geförderten

[1] Messungen mit Simulator FABIAN am geblockten Ohrkanal und nicht am Trommelfell

Verbundprojektes zur Simulation und Evaluation akustischer Umgebungen (SEACEN) [hb00] [2] - angesiedelt im Fachgebiet für Audiokommunikation der Technischen Universität Berlin.

[2] www.seacen.tu-berlin.de/

Kapitel 2

Stand der Forschung

Bevor Effekte der Torsoposition auf binaurales Hören, sowie die Interpolation von HRTFs untersucht werden können, ist es notwendig Grundlegendes über räumliches Hören, physiologische Parameter, Grenzen bzw. die Sensibilität des auditorischen Systems und die anhand dessen sich entwickelnde Binauraltechnik herauszustellen.

2.1 Räumlich Hören

Das binaurale Signalpaar bestehend aus dem Signal am rechten und linken Trommelfell ist die Grundlage des räumlichen Hörens. Durch Auswertung zeitlicher und spektraler Strukturen ist es möglich Eigenschaften des Raumes und der Schallquelle wahrzunehmen [BB08]. Dreidimensionales Hören beruht maßgeblich auf der Fähigkeit des auditorischen Systems Schallquellen an ihrer Position lokalisieren zu können. Im sphärischen Koordinatensystem beschreibt der Azimuthwinkel ϑ die horizontale Lage, der Elevationswinkel φ die vertikale Auslenkung und der Radius r die Entfernung der Quelle bezüglich des interauralen Zentrums [3]. Bei Ankunft des Schalls am rechten und linken Ohrkanal ist er vom Quellpunkt bereits zum einen durch das System des Raumes und zum anderen durch Reflexionen, Abschattungseffekte und Laufzeitunterschiede des binauralen Systems bestehend aus Pinnae, Kopf und Torso verändert worden. Eigenschaften diesen binauralen Filters sind wie ein akustischer „Fingerabdruck“ der Hörermorphologie [Nic10, vgl. S. 11] wegen unterschiedlicher Pinna-, Kopf- und Torsobeschaffenheit individuell. Aufgeprägte bin- und monaurale Informationen werden wie im Abgleich mit einem Referenzsystem durch das auditorische System ausgewertet.

Møller [Mø92] stellt vier Parameter des räumlichen Hörens heraus: monaurale spektrale Färbung sowie interaurale Phasen- (IPD), Zeit- (ITD) und Pegeldifferenzen (ILD).

[3] Mittelpunkt der interauralen Achse $\neq$ Kopfmittelpunkt

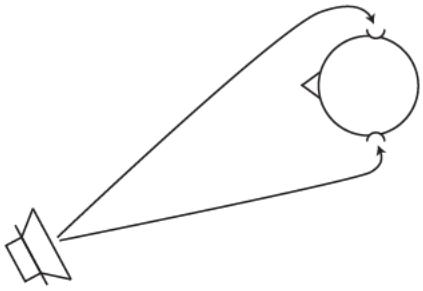


Abbildung 2.1: Skizze des Schallwegs zum Trommelfell

In Abbildung 2.1 ist der Ausbreitungsweg von einer Schallquelle zum rechten und linken Trommelfell angedeutet. Da sich der Lautsprecher nicht in der Medianebene des Kopfes befindet ist der Weg und damit auch die benötigte Zeit zum rechten Ohr länger. Für tiefe Frequenzen werden interaurale Phasenunterschiede, für mittlere Frequenzen Zeitdifferenzen wahrgenommen. Des Weiteren kann sich der Schall für kurze Wellenlängen nicht um den Kopf beugen, so entsteht ein Abschattungseffekt am schallabgewandten Ohr und so eine interaurale Pegeldifferenz für höhere Frequenzen. Befindet sich die Quelle in der Medianebene und sind Ausbreitungswege gleich lang, orientiert sich das auditorische System anhand spektraler Färbungen, die durch die von Quellentfernung und Schalleinfallsrichtung abhängigen Reflexionen und Resonanzen an Pinna, Kopf und Torso, sowie durch deren Diffusor- und Abschattungswirkung entstehen.

Tabelle 2.1: JNDs - Lateralisation/Lokalisation

Lateralisation		
ITD	2 – 10 μs	Klickreize [Kle20]
ILD	0.6 dB	2 kHz Sinuston [For42]
IPD	2 – 4 °	Sinustöne bei 150 - 1000 Hz [Mil58]
Lokalisation		
Horizontalebene	0.75 - 2°	Klickreize [Kle20]
Medianebene	4°	weißes Rauschen [Wet70]

Eine anschließende Frage stellt sich zum Auflösungsvermögen genannter Reize bzw. zur Empfindlichkeit des Ohres auf Abweichungen. Blauert [Bla97, Kap. 2] zitiert viele Experimente, in denen die kleinsten eben merkliche Änderungen (JND) untersucht wurden. Mittels über Kopfhörer wiedergegebener, reizspezifisch manipulierter Signale wurden Lateralisations-JNDs[4] für diese Reize ermittelt. Es ist auf die Frequenz-

[4]Lateralisation meint die Wahrnehmung der seitlichen Auslenkung eines Signals bei

und Lautheitsabhängigkeit dieser Ergebnisse hinzuweisen. Im Weiteren untersuchte man in einer Vielzahl von Hörversuchen den eben hörbaren Unterschied für die Richtungswahrnehmung von Schallquellen. In Tabelle 2.1 sind die kritischsten, jeweils kleinsten bei Blauert genannten Werte zusammengetragen.

2.2 Binauraltechnik

Binauralsynthese ist die Wiedergabe gemessener oder simulierter binauraler Signale via Lautsprecher oder Kopfhörer, sodass die nachgebildete akustische Umgebung die gleichen Wahrnehmungen bezüglich der Schallrichtung und Raumeigenschaften etc. einer natürlichen Umgebung evoziert [Mø92, vgl. S. 176]. Im Fokus der Binauraltechnik sind daher die kopfbezogenen Übertragungsfunktionen, die unter der Verwendung in dieser Arbeit den Hals, den Torso und die Oberarme mit einbeziehen. Sie bestehen aus zwei Impulsantworten bzw. Übertragungsfunktionen für jeweils rechtes und linkes Ohr und beschreiben das System unter der Annahme von Linearität und Zeitinvarianz (LTI-System) vollständig [Mü08, vgl. S. 1094]. Es gilt zu unterscheiden zwischen kopfbezogenen *Freifeld*übertragungsfunktionen (HRTF mit dem Zeitbereichsäquivalent HRIR) und binauralen *Raum*übertragungsfunktionen (BRTFs mit dem Zeitbereichsäquivalent BRIR). Die binaurale Raumimpulsantwort BRIR setzt sich aus dem am Hörerort eintreffenden Direktschall und weiteren dem Direktschall überlagerten Reflexionen zusammen, die als einzelne Impulse betrachtet werden können. Bei der modellbasierten Auralisation von Räumen werden daher unter Annahme eines LTI-Systems die binaurale Raumübertragungsfunktion durch die Überlagerung von gewichteten und phasenverzögerten HRTFs für linkes und rechtes Ohr gebildet [Vor08, Kap. 15].

$$BRTF = \sum_{n=1}^{N} e^{-j\omega t_n} \cdot g_n \cdot HRTF_n \tag{2.1}$$

Wobei $j\omega t_n$ die Phasenverzögerung entsprechend der Verzögerung der n-ten Reflexion ist und g_n als frequenzabhängige Gewichtungsfunktion die raumakustischen Eigenschaften, die geometrische Ausbreitungsdämpfung über den zurückgelegten Weg der Reflexion, die Luftabsorption sowie die Richtcharakteristik der Schallquelle enthält. Komponenten werden rechnerisch mittels Raum- und Kopfmodellen nach verschiedenen Ansätzen (z.B. Spiegelschallquellen), bei Vorländer [Vor08] oder auch Berkhout et al. [BDVBVdO99] beschrieben, bestimmt.

Kopfhörerwiedergabe. Der Begriff Lokalisation ist davon abzugrenzen und meint die Auswertung von Richtung und Entfernung einer Schallquelle in der Umgebung beim natürlichen Hören.

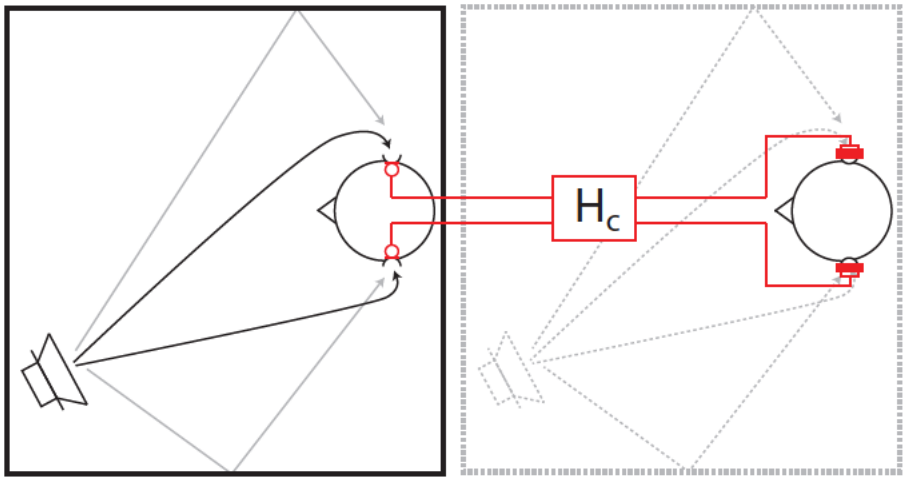


Abbildung 2.2: Skizze zur datenbasierten Binauralsynthese
Elektroakustische Übertragung von der BRIR-Messung (links) über die Kompensation mit Filter H_c zur Wiedergabe der Reproduktion per Kopfhörer (rechts)

Abbildung 2.2 zur datenbasierten Binauralsynthese skizziert Schallwellen des Direktschalls und der Reflexionen, die bei der Freifeldübertragung wegfallen, eines am geblockten Ohrkanal aufgenommenen Messsignals. Nach Entfaltung der Aufnahme mit dem Messsignal ist die BRIR ermittelt. Diese als akustische Situation gefaltet mit reflexionsarmen Audiomaterial wird nach Kompensation H_c der in der Übertragungsstrecke enthaltenen Störfaktoren (hier Mikrophon, Kopfhörer) in dieser Skizze per Kopfhörer wiedergegeben. Kompensation kann beispielsweise durch eine regularisierte Invertierung der Übertragungsfunktion erfolgen. Regularisiert bzw. begrenzt werden müssen dabei schmalbandige Senken in der Übertragungsfunktion, wie sie vor allem bei circum- und supraauralen Kopfhörern gemessen werden, die nach dem Ab- und wieder Aufsetzen spektral verschoben sein können bzw. bei anderen Frequenzen liegen, weil Übertragungsfunktionen nicht individuell für den Hörenden gemessen werden konnten. Ohne Regularisierung würden diese Einbrüche nach Invertierung dem erwünschten Signal störend als tonale Peaks hinzugefügt [SL09].

Ziel einer realitätsnahen Simulation ist es dem Hörenden darüber hinaus Bewegungen zu ermöglichen unter Beibehaltung des Eindrucks sich in der akustischen Szene zu befinden - die Schalleinfallsrichtung einer festen Quelle bezüglich des Trommelfells soll sich bei Bewegung entsprechend ändern. Grundlage dessen sind perzeptiv kontinuierliche HRTF-Datensätze (für jede Kopf-über-Torso-Orientierung) und eine dynamische Signalverarbeitung, um zunächst Kopfbewegungen in der simulierten Umgebung zu gewährleisten. Durch das begrenzte Auflösungsvermögen des auditorischen Systems können unterschwellige Artefakte ohne eine perzeptive Kontinuität zu gefährden zugelassen werden. Den Toleranzbereich dafür geben experimentell erhobene JNDs, die bereits auf die Beschaffenheit des der Synthese zu Grunde liegenden HRTF-Datensatzes auch unter dem Gesichtspunkt der Effizienz bei seiner Erstellung angewendet werden können. Am Beispiel der Lokalisation ist zu erkennen, dass Datensätze für perzeptiv kontinuierliche Quellbewegungen mit viel feinerer Auflösung als 0.7° (siehe JND-Tabelle 2.1) unnötig groß sind. So stehen sich auf dem Gebiet der Binauraltechnik Plausibilität/Realitätstreue und Effizienz

gegenüber. Ein Gleichgewicht dieser Qualitäten soll in dieser Studie hinsichtlich der Kopf-über-Torso-Orientierung untersucht werden.

2.3 Einfluss des Torsos bei statischer HATO

Die benötigte Winkelauflösung eines HRTF-Datensatzes bei der Balance von perzeptiver Kontinuität und Effizienz ist durch Minnaar et al. [MPC05] am Kopf-und-Torso-Simulator (HATS) für frontale Ausrichtung des Kopfes untersucht und danach durch die hauptsächlich von der Pinna verursachten Feinstruktur in den HRTFs gegeben. Das deckt sich mit Erkenntnissen der im Abschnitt 2.1 angesprochenen monauralen spektralen Modifizierungen, die vor allem bei vertikaler Lokalisation hilfreich sind. Diese liegen bei Frequenzen höher als 3 kHz und sind der Ohrmuschel zuzuweisen [Bla97] [MG91]. Der Pinna kommt damit eine große Bedeutung zu.

Die Relevanz von Schulterreflexionen ist für unterschiedliche Quellpositionen bei statischer HATO beschrieben worden. Ein physikalischer Evaluationsansatz ist bei Algazi et al. [ADD$^+$02] nachzulesen, die versuchten HRTFs mittels einfacher geometrischer Modelle von Kopf und Torso anzunähern. Bei der Betrachtung des tieferen Frequenzbereichs sprechen die Autoren von potentiell signifikanten Elevations-Cues durch Torsoreflexionen bei Quellen oberhalb und Torsoabschattungen bei Quellen unterhalb der Horizontalebene. Effekte seien schwach aber nicht von der Hand zu weisen. Sie schließen, dass auf eine Modellierung des Torsos nicht verzichtet werden sollte, allerdings mit einfachen geometrischen Formen bereits gute Ergebnisse erzielt werden können.

In welcher Größenordnung sich dieser Effekt der Schulter bzw. des gesamten Torsos physikalisch auswirkt ist auch bei Gardner [Gar73, Abb. 6] gezeigt. Methodisch wurde zwei mal die Differenz der Schalldrücke am Trommelfell in dB für eine Quellelevation von 18° und -18° ermittelt. Im ersten Fall war es ein Kunstkopf mit abgedeckter Pinnastruktur auf einem Torso. Im zweiten Fall ist der Torso zusätzlich entfernt worden. Die ermittelte Differenz aus den Messungen bei Anwesenheit des Torsos ist relativ glatt außer im Bereich von ca. 700 Hz bis 3.5 kHz, dort weist das Diagramm ein Minimum von ca. -5 dB auf. Ohne Torso glättet sich auch dieser Frequenzbereich. Demnach verfärbt der Torso das HRTF-Betragsspektrum innerhalb der Medianebene, im Bereich von 0.7 und 3.5 kHz winkelabhängig. Diesen Frequenzbereich würde man auch aufgrund der größeren Abmaße des Torsos erwarten. Gardner hat jedoch nicht untersucht, ob gefundene Unterschiede durch das auditorische System für Lokalisierung auswertbar seien. Searle et al. [SBDC76] ordnen die Schultereffekte mit Bezug zu Gardner et al. als geringste aller genannten Schlüsselreize der Lokalisation ein.

Algazi et al. [AAD01] stellten die resultierenden Spektren der torsobezogenen Übertragungsfunktion in Abhängigkeit der Quellelevation dar und konnten auf Gardner aufbauend belegen, dass auch diese geringen Modifizierungen durch den Torso die Lokalisationsleistung beeinflussen. Dazu sollten Probanden bei 3 kHz tiefpassgefilterte Signale orten. Ergebnisse fielen zwar weniger sicher aus im Vergleich zu breitbandigeren Stimuli, jedoch konnte ein Zusammenhang zwischen genanntem und tatsächlichem Quellort statistisch eindeutig nachgewiesen werden.

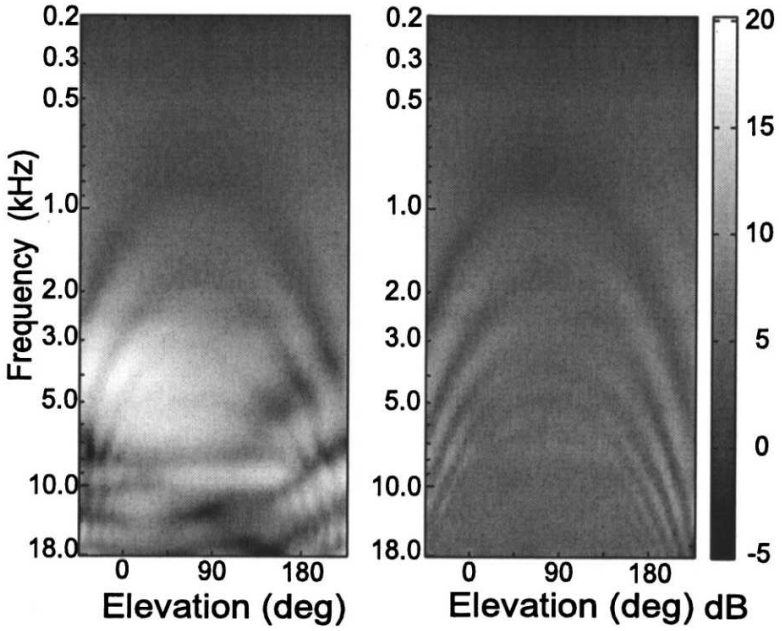


Abbildung 2.3: durch Torso verursachter Kammfilter (Bildquelle: [AAD01])
links: torsobezogene Übertragungsfunktion mit Effekten der Pinna über die Elevation der Quellposition bei einem linksseitigen Azimuth von 45°, dargestellt ist die ipsilaterale Seite; rechts: ohne Pinna

Abbildung 2.3 ist der selben Studie des eben beschriebenen Hörexperiments entnommen. Spektren der torsobezogenen Übertragungsfunktion über dem Winkel der Quellposition zum Torso (rechts) zeigen einen Kammfilter, der sich von ca. 700 Hz bis zu oberen Frequenzen des Hörbereichs hin reicht (hier ersichtlich bis 18 kHz) und sich mit dem Winkel der Elevation spektral verfärbt. Die Anwesenheit der Pinna überlagert diesen Effekt (links).

Guldenschuh et al. [GSZ08] extrahierten den Torsoeffekt ebenfalls und zeigten, dass bei seitlichen Quellpositionen, vor allem mit Elevationen unterhalb des Kopfes Reflexionen und Abschattungen der Schultern deutlicher hervortreten. (Für diese kritischen Quellen ändern sich die torsobezogenen Übertragungsfunktionen am stärksten mit dem Winkel.) Das entspricht auch den Ergebnissen nach Minnaar et al. [MPC05]. Weniger kritische Quellen, für die sich die torsobezogenen Übertragungsfunktion weniger stark über dem Winkel ändern, liegen oberhalb des Kopfes, innerhalb der Horizontalebene ventral und dorsal. Folglich wäre für kritische Quellposition eine Hörbarkeit des Torsoeinflusses auf das Schallfeld wahrscheinlicher.

2.4 Zu berücksichtigende Bewegungsbereiche

Soll der Einfluss des Torsos auf das Schallfeld am Trommelfell bei sich ändernder HATO analysiert werden, ist die Betrachtung im Winkelbereich einer natürlichen Bewegung sinnvoll. Die Anatomie der Halswirbelsäule mit ihrem oberen Abschluss durch die Kopfgelenke erlaubt Kopfbewegungen in den folgenden drei Freiheitsgraden:

1. Rotation in Horizontalebene (rechts/links-Bewegung)
2. Ante- und Retroflexion (nicken) in Sagital- bzw. Medianebene
3. Lateralflexion (seitliche Kopfneigung) in Frontalebene

Tausendfreund [Tau04] legt in seiner Dissertation verschiedene Ergebnisse von Bewegungsanalysen dar. Darin lassen sich Angaben zum maximalen Bewegungsausmaß für einzelne Freiheitsgrade finden. Weitere Angaben finden sich bei Schöps et al. [SSS+97], wo die in Tabelle 2.2 aufgeführten maximal möglichen Bewegungsbereiche zu entnehmen sind, die als Mittelwerte unter Verwendung eines Goniometers für 20 männlich wie weibliche beschwerdefreie Probanden erhoben wurden. Die Standardabweichung betrug ca. 5°. Thurlow et al. [TMR67] untersuchten Kopfbewegung bei der Schallquellenlokalisierung und schildern den hörtypischen Bewegungsbereich, der ebenfalls in der Tabelle gelistet ist. Zusätzlich wurde gezeigt, dass Probanden stets versuchen durch Kopfbewegung in Kombination aller Freiheitsgrade die Quelle zu fokussieren.

Tabelle 2.2: hörtypischer und maximaler Bewegungsbereich

Mittelw.	hörtypisch [TMR67]	Maximalbereich [SSS+97]
Rotation	±42°	±81.8°
Ante-/Retroflexion	±15.2°	74.7°/71.3°
Lateralflexion	±11.6°	±45.9°

2.5 Kunstkopf

Aus pragmatischen Gründen werden Kunstköpfe an Stelle realer Personen verwendet, da so bei zeitaufwendigen Messungen weniger durch Bewegung bedingte Messfehler auftreten. Entwickler verschiedener Kunstköpfe streben oft nach unterschiedlichen Kriterien eine Art Durchschnitt an. Sowohl schematisch geformte Köpfe als auch Nachformungen realer durchschnittlicher Köpfe, mit und ohne Torso, statisch und beweglich sind gebaut worden [BS75] [CJM00] [Sch95] [Mol02] [LW06].

Darunter ist der Kopf-Torso-Simulator FABIAN (Fast and Automatic Binaural Impulse response AcquisitioN) besonders hervorzuheben, da dessen Kopf-über-Torso-Orientierung in drei Freiheitsgraden über eine Software steuerbar ist, während der gesamte Torso mittels Motor gedreht werden kann [LW06] [LHW07]. Es handelt sich beim Kopf um einen Abguss eines männlichen Kopfes, wobei das Außenohr bis zum Eingang des Ohrkanals aus einem Material ähnlicher Impedanz der natürlichen Pinna nachgebildet ist. Der vertikale Winkel der Pinna zum Kopf beträgt 8°. Am geblockten Ohrkanal befindet sich ein Miniaturmikrofon. Der Torso ist anhand durchschnittlicher Werte männlicher US-Soldaten entworfen, besitzt ebenso Oberarme und ist mit einem höhenverstellbaren Beingestell erweiterbar.

2.6 Interpolationsalgorithmen

Perzeptiv artefaktfreie Simulation von Kopfbewegungen in virtueller akustischer Umgebung erfordert entweder eine sehr feine Auflösung vorliegender HRTF-Datensätze oder aber die Interpolation von Daten gröberer Auflösung. Im laufenden

Forschungsprojekt SEACEN [hb00] existiert ein mit 2° aufgelöster sphärischer HRTF-Datensatz für elf verschiedene Kopf-über-Torso-Orientierungen in einer Auflösung von 10° (Azimuth ±50°), der für modellbasierte Auralisation verwendet werden soll. Es ist bisher nicht geprüft, ob diese Auflösung von Kopf-über-Torso-Orientierungen fein genug ist, um eine artefaktfreie Wahrnehmung einer Kopfbewegung nach der Resynthese zu ermöglichen. Lindau und Weinzierl [LW09] nehmen für die Bewegung des Kopfes bei statischem Körper eine Schrittweite von 2°[5] an, um von einem perzeptiv unterschwelligen Einfluss bei der Auralisierung von BRIRs ausgehen zu können bzw. sprechen von der Interpolation der Daten, mit der möglicherweise bei richtiger Wahl des Algorithmus eine gröbere Rasterweite für die perzeptiv artefaktfreie Wiedergabe genüge.

Frühere Studien befassten sich mit der Interpolation von Quellpositionen [HBS99] [MPC05]. Dort aufgeführte Ansätze sollten nach Möglichkeit auf die Interpolation von Kopf-über-Torso-Orientierungen und deren spätere Evaluation übertragen werden. Eine Interpolation der Daten kann entweder im Zeitbereich oder aber im Frequenzbereich stattfinden. Je nach Datensatz ist ein sphärischer oder zweidimensionaler Ansatz möglich. Minnaar et al. [MPC05] repräsentierten die HRTFs durch ihre Minimum-Phasen-Komponenten (siehe 3.2.4.4) und der zugehörigen interauralen Zeitdifferenz, um für verschiedene Auflösungen zwischen den Komponenten linear im Zeitbereich zu interpolieren. Es wurden zu den interpolierten Werten nachträglich die interauralen Zeitdifferenzen der Originalmesswerte der feinen Auflösung ergänzt. So kann lediglich ein interpolierter Parameter zum physikalischen Fehler führen. Die Wahl einer linearen Interpolation der Impulsantwort wurde dabei aufgrund der Einfachheit getroffen. Für kritische Quellpositionen wurden im perzeptiven Hörtest dieser Studie kleinere Interpolationsintervalle an der Schwelle hörbarer Artefakte ermittelt.

Es werden weitere Modellierungs- und Interpolationsarten angeführt - die Modellierung von HRTFs über ihre Pol- und Nullstellen mit dem Verweis auf Blommer et al. [BW97] und Runkle et al. [RBW95], die Analyse nach Kistler et al. [KW92] (principal component analysis) und nach Larcher et al. [LJGW12] (independent component analysis) und schließlich die Interpolation mit sphärischen Harmonischen nach Evans et al. [EAT98] (surface spherical harmonics). Robeson [Rob97] erläuterte und evaluierte sphärische Methoden der räumlichen Interpolation - darunter die nicht-statistische inverse Distanzwichtung und die sphärische Spline-Interpolation über sphärische Harmonische.

Hartung et al. [HBS99] verglichen vier Interpolationsverfahren für HRTFs, inverse Distanzwichtung und sphärische Spline-Interpolation jeweils für die Darstellung im Zeitbereich (als FIR-Koeffizienten) und im Frequenzbereich nach diskreter Fouriertransformation (DFT) der Impulsantwort. Es wurden die besten Ergebnisse mittels sphärischer Spline-Interpolation im Frequenzbereich erzielt. Eine Übertragung ist jedoch nur möglich, wenn die verwendeten Datensätze die selbe Struktur aufweisen. Im Abgleich der Möglichkeiten mit vorliegenden gemessenen HRTFs (siehe 3) kommen für diese Arbeit nur zweidimensionale Verfahren in Betracht. Kapitel 5 stellt die in dieser Arbeit letztlich verwendeten, zweidimensionalen Ansätze genauer vor, darunter die inverse Distanzwichtung bzw. die einfache lineare Interpolation und die Splinemethode in Gegenüberstellung zu „nearest neighbour“ und der fein aufgelösten

[5] Das gilt für die Beschränkung auf ±30° für Lateralbewegungen

Referenz ohne Interpolation.

2.7 Zusammenfassung - Stand der Forschung

Die Fähigkeit zum räumlichen Hören in einer natürlichen Umgebung wird durch Binauraltechnik in virtuellen akustischen Umgebungen ermöglicht. Ein plausibler bzw. realistischer Höreindruck innerhalb der Simulation lässt sich nur durch möglichst umfassende Berücksichtigung aller Einflüsse auf das Schallfeld erreichen. JNDs bilden dabei die Grenzen des auditorischen Systems ab Unterschiede bzw. Artefakte hören zu können; gleichzeitig geben sie die notwendige Genauigkeit einer Simulation wieder. Der Einfluss des Torsos in HRTF-Daten, dem zwar gegenüber den Filtereigenschaften des Kopfes eine untergeordnete Rolle zukommt, ist für unterschiedliche Quellpositionen bei statischer Kopf-zu-Torso-Orientierung gezeigt worden. Daraus ergibt sich die Fragestellung nach der Hörbarkeit des Torsoeinflusses bei dynamischer Kopf-zu-Torso-Orientierung und statischer Quellposition.

Mit Kopf-Torso-Simulatoren sollen Oberkörper einbeziehend HRTFs, die durchschnittlichen HRTFs an Individuen entsprechen, reliabel und fehlerarm gemessen werden. Um Messaufwand zu reduzieren werden in der Literatur Ansätze von Interpolationsalgorithmen besprochen.

Kapitel 3

HRTF Messung und Processing

3.1 HRTF Messungen

Um den physikalischen und perzeptiven Einfluss von Kopf-über-Torso-Orientierungen (HATOs) klären zu können, sowie um Rohdaten für die Interpolation von HRTFs hinsichtlich dieser Orientierungen zu erhalten, galt es einen HRTF-Datensatz mit variabler HATO in den drei Hauptebenen (Horizontal-, Median-, Frontalebene) anzulegen. Im Folgenden soll der genaue Messablauf und -aufbau, sowie zugehörige Reliabilitäts- und SNR-Tests zur Messung beschrieben werden[6].

3.1.1 Ort, Zeit der Messung

Die Messungen zur Frontal- und Medianebene fanden im Zeitraum vom 06.-15.04.2013 und die Messungen innerhalb der Horizontalebene vom 17.-21.05.2013 im reflexionsarmen Raum des Instituts für Strömungsmechanik und Technische Akustik der TU-Berlin statt. Dieser fasst ein Volumen von 1070 m^3 bei einer unteren Grenzfrequenz von 63 Hz. Zum Betreten des Raumes ist ein Netz gespannt, auf dem ebenso leichte Teile der Messausrüstung platziert werden können. Für schwere und bewegungsempfindliche Geräte kann ein vom Netz entkoppeltes Gerüst auf in den Boden ragende Stahlstangen aufgebaut werden. Die Größe und Form kann dabei variieren. Öffnungen in der Decke machen ein Abhängen von Teilen des Versuchsaufbaus möglich.

3.1.2 Koordinatenkonvention, Messpunkte

Abbildung 3.1 zeigt das räumliche Koordinatensystem. Mittelpunkt ist das interaurale Zentrum, sprich die Mitte der interauralen Achse. Der Torso mit einem Beingestell konnten auf das VariSphear-System gestellt in der Horizontalebene gedreht werden - dieser Drehwinkel ist der Torso-Azimuth. Kopf und Quelle änderten sich im horizontalen und vertikalen Drehwinkel und bekamen daher neben einem Azimuth- auch einen Elevationswinkel zugewiesen. (Ob es sich bei der Kopfelevation um eine Seitneigung oder eine Retro- bzw. Anteflexion handelte, ist mit der Ebene kodiert, die den Messdaten zugewiesen wurden. In der Horizontalen ist die Kopfelevation stets 0°, in der Medianebene entspricht die Kopfelevation dem Winkel der Ante-/Retroflexion, in

[6]Speicherort, -struktur und -format ist in einer etwas ausführlicheren dem Datensatz beiliegenden Dokumentation erklärt.

der Frontalebene der Seitneigung.)

Nach Festlegung eines Referenzpunktes von 0° Azimuth und 0° Elevation bei neutraler Torso- und Kopfposition sind Bereiche oberhalb und linkerseits vom Torso positiv, unterhalb sowie rechtsseitig negativ kodiert. Bsp.: Ist der Kopf auf 0° Azimuth und 0° Elevation gerichtet und die HATO 0°, befindet sich sowohl Kopf als auch Torso auf den Referenzpunkt des Koordinatensystems zeigend in neutraler Position [0°; 0°]. Drehen sich Torso und Kopf z.B. um 90° nach links ist die Position [90°; 0°] erreicht bei einer bleibenden HATO von 0°. Sind Torso und Kopf wieder in neutraler Haltung und bewegt sich lediglich der Kopf um 90° nach rechts blickt er auf den Punkt [-90°; 0°], wodurch die HATO -90° beträgt.
Quellen werden ebenfalls vom Referenzpunkt ausgehend in ihren Koordinaten angeordnet. Die Quellpositionen werden noch in 3.1.3.5 genauer erläutert.

Messpunkte sind durch die Angabe folgender Variablen eindeutig beschrieben:

- Quellazimuth, Quellelevation
- Torsoazimuth
- Kopf-Azimuth, Kopf-Elevation
- HATO (redundant: Differenz aus Torso- und Kopfazimuth)

In der **Horizontalebene** wurden je Quelle Kopf-über-Torso-Orientierungen im Bewegungsbereich von ±82° in 0.5° Auflösung bei statischer Torsoausrichtung von 0° Azimuth gemessen. Diese Bewegung sollte im Weiteren interpoliert werden. Die Daten in feiner Auflösung dienten dazu als Referenz (siehe Kap. 5). Sollen zwei HRTFs „via Kopf“ interpoliert werden, weisen diese bei gleicher Torso-Ausrichtung, unterschiedliche Kopf-Orientierungen auf. Es handelt sich in dem Fall um Stützstellen, die der Referenzmessung im gewünschten Intervall zu entnehmen sind.

Des Weiteren waren im Messablauf Punkte enthalten für Torso-Azimuthwinkel zwischen ±82° bei stets statischer HATO von 0°, um den Torsoeinfluss, wie in der Einleitung bereits angeführt, zu untersuchen und dabei statische mit dynamischer HATO vergleichen zu können. Für das Ermöglichen verschiedener Interpolationsintervalle von Kopf-über-Torso-Orientierungen (siehe Tab. 5.1), wurden außerdem je Quelle und je Interpolationsintervall alle benötigten Stützstellen gemessen, um jede Kopf-über-Torso-Orientierung im Bewegungsbereich über den Torso, bei Torso 0°, interpolieren zu können. ”Torso-Interpolation”meint die Interpolation zweier HRTFs die bei gleicher Kopf-Ausrichtung, unterschiedliche Torso-Orientierungen aufweisen. Um beispielsweise den Punkt Torso 0° und Kopf 7° Azimuth im Intervall 30° aus dem Torso interpolieren zu können, werden einmal Torso 7° mit Kopf-Az. 7° benötigt und dazu Torso -23° mit Kopf-Az. 7°.

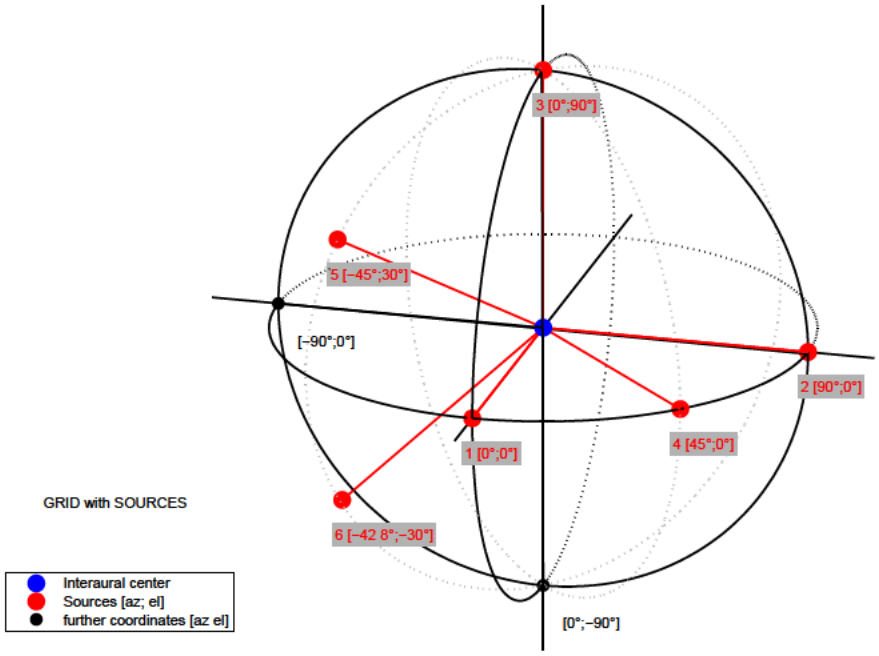


Abbildung 3.1: Koordinatensystem und Quellpositionen in der Horizontalebene (verzichtet auf die Darstellung der Radii, da diese im Processing als Laufzeit entfernt wurden)

In der **Medianebene** bewegte sich die Kopfelevation für die Kopfazimuthwinkel von -60°, -30°, 0°, 30° und 60° zwischen ±45° wobei der Torso stets 0°-Ausrichtung beibehielt. Gleiches wurde für die **Frontalebene** gemessen - allerdings nur für die Kopfazimuthwinkel von -30°, 0° und 30°. Die Auflösung betrug auch in diesen Ebenen 0.5°. (Die Interpolation „via Kopf“ ist damit auch in der Median- und Frontalebene möglich. Für die Interpolation „via Torso“ hätte der Messroboter für verschiedene Elevationen bei statischer HATO gekippt werden müssen - was nicht möglich war.)

3.1.3 Messaufbau

3.1.3.1 Vorbereitung

Zunächst wurde ein Gerüst auf Stahlstangen im reflexionsarmen Raum stabil verstrebt. Dabei wurde darauf geachtet, dass möglichst wenige Reflexionsflächen geschaffen wurden. Entstandene Flächen wurden nachträglich mit Absorbermaterial eingeschäumt bzw. abgedeckt.

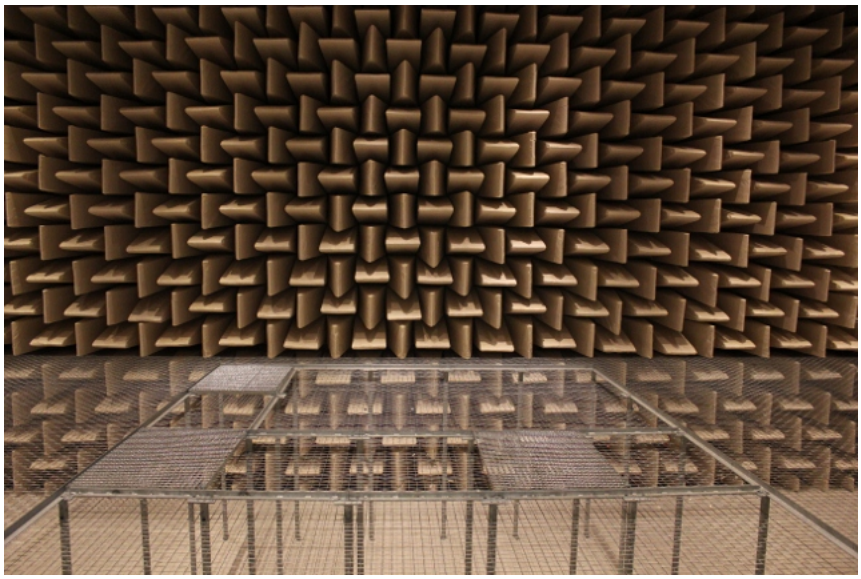


Abbildung 3.2: Gerüst auf Stahlstangen im reflexionsarmen Raum

3.1.3.2 Geräteliste

- 3 Lautsprecher: GENELEC 8030A (2-Wege-Aktivbox, Bassreflex); Volume stets maximal
- Lautsprecherstative
- FABIAN
  - Torso, Kopf mit PowerCube-Halsgelenk, Beingerüst
  - Spannungsversorgung des Halsgelenkes: Voltcraft PS 3620
  - 2 DPA 4060 Miniaturmikrofone
- VariSphear Array Measurement System + Netzteil
- zwei-kanaliger Lake People Mic Preamp C360
- Soundkarte RME Multiface II
- Thinkpad T61
- `Matlab` 2011a
- patch-, XLR-Kabel etc.
- HYGROSENS USB Thermometer

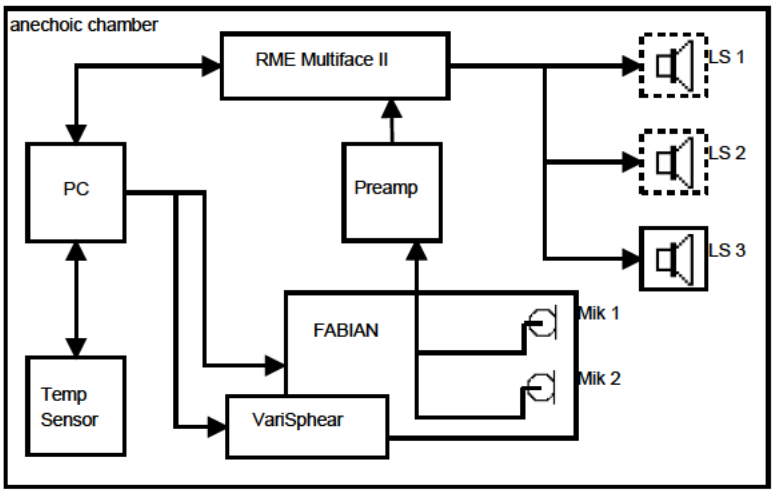


Abbildung 3.3: Signalpfad bei Messung

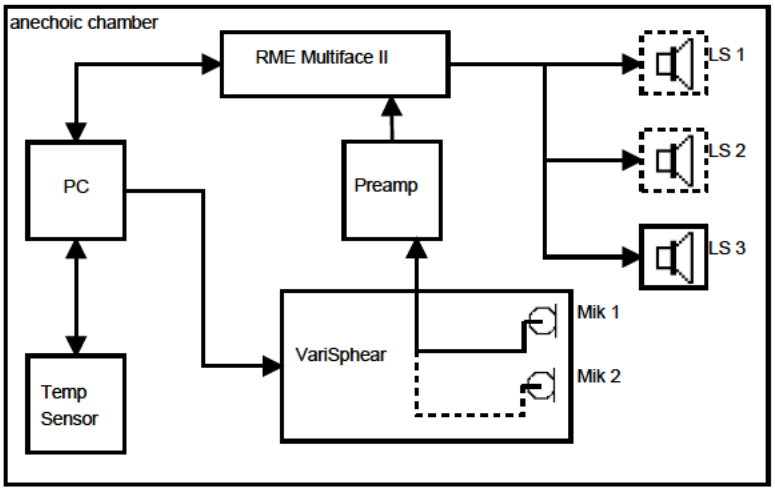


Abbildung 3.4: Signalpfad bei Referenzmessung

3.1.3.3 Aufbau FABIAN

Etwa mittig im Raum auf vorgesehenem Gitter wurde zunächst das VariSphear-System platziert und die Drehachse mit einem Metallpin markiert. Anschließend sind darauf das mit Absorbermaterial umhüllte FABIAN-Beingestell und der mit einem Baumwoll-T-Shirt bekleidete Torso montiert und mit Klemmen befestigt worden. Das Halsgelenk konnte für die Messungen zur Vermeidung harter Reflexionen leider nicht abgedeckt werden, da sich der dafür vorgesehene Lederstoff bei Bewegung des Halsgelenkes in verschiedene Falten legte, wodurch Messdaten selber Messpositionen nicht reliabel gewesen wären. Für die zu Grunde liegende Forschungsfrage sind diese harten, systematischen Reflexionen nicht problematisch, da nicht eine besondere Natürlichkeit später auralisierter Daten erreicht werden sollte, sondern eine hohe Vergleichbarkeit innerhalb des Datensatzes. Kabelzuleitungen zum Roboter wurden vorsichtig befestigt und von der Decke abgehängt zur sicheren Kabelführung bei Rotation des VariSphear-Systems um die vertikale Achse. Kontakte zwischen empfindlichen Mikrofonkabeln und stromleitenden Verbindungen wurden weitgehend verhindert.

3.1.3.4 Temperatur

Zur Aufzeichnung der Temperatur während den Messungen wurde ein Temperatursensor in der Nähe des Messaufbaus installiert.

3.1.3.5 Quellpositionen

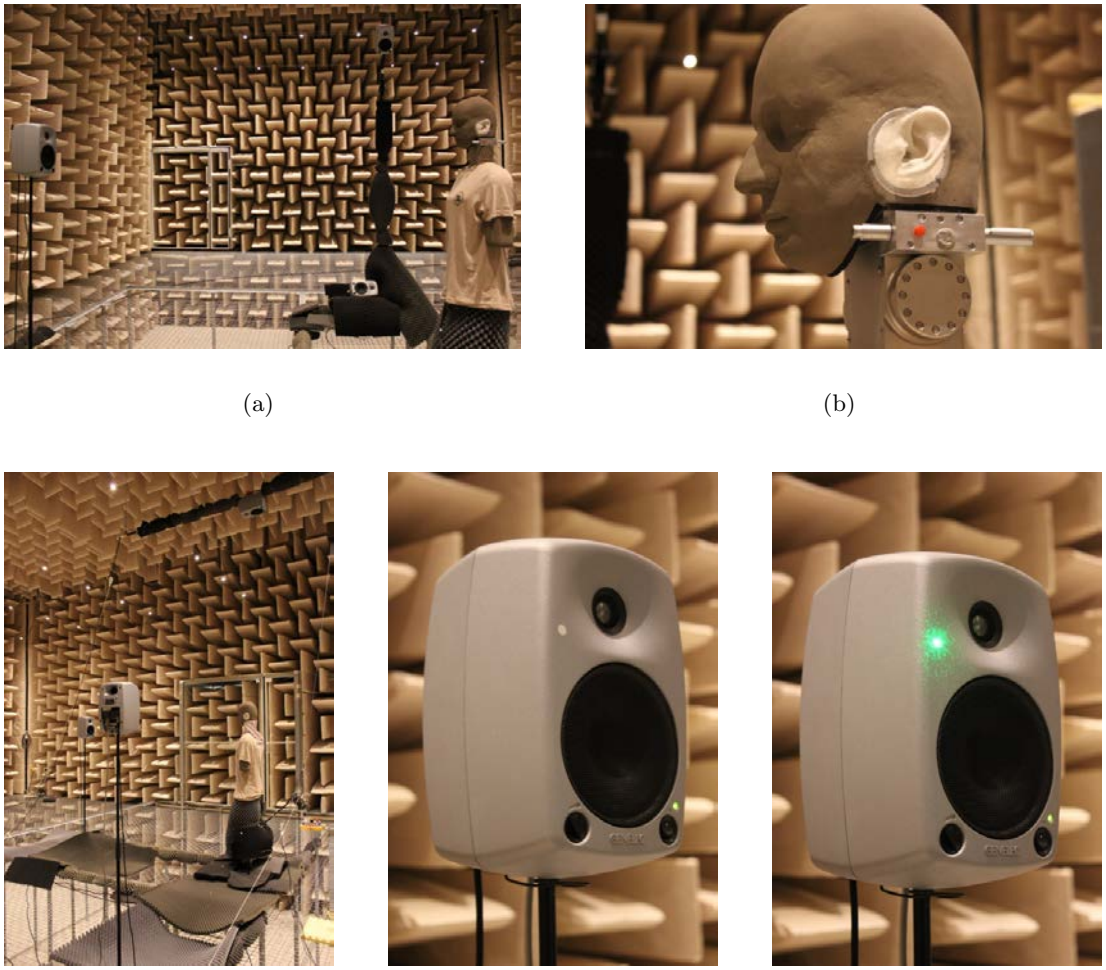
Bezogen auf eine Referenzachse ausgehend vom interauralen Zentrum des Kunstkopfes zum Punkt [Azimuth Elevation]= [0° 0°] sind mit folgenden Quellpositionen 1 bis 3 (Setup a) in einem ersten und mit den Quellen 4 bis 6 (Setup b) in einem zweiten Messaufbau gemessen worden. Die genauen Positionen sind mit Hilfe eines Lasers am Halsgelenk, der Entfernung (Quelle zu linkem Ohr, bei Kopf-zu-Quell-Orientierung von 0°) und `FABIAN_reference_offset.m` ermittelt worden.

Median- und Frontalebene (Setup 1a 1b): Quellnummer - Quellkoordinaten - Quellentfernung

1. [0° 0°] - 2.10 m
2. [90° 0°] - 2.13 m
3. [0° 90°] - 2.18 m
4. [45° 0°] - 2.19 m
5. [-45.2° 29.8°] - 2.47 m
6. [-41.5° -30°] - 2.59 m

Für die Horizontalebene wurde noch einmal erneut aufgebaut (Setup 2a 2b), sodass sich die Positionen etwas unterschieden. Die Frage nach der Vergleichbarkeit ist dabei irrelevant, da Betrachtungen über Messdaten immer innerhalb einer Ebene angestellt werden.

1. [0° 0°] - 2.10 m
2. [90° 0°] - 2.17 m
3. [0° 90°] - 2.07 m
4. [45° 0°] - 2.13 m
5. [-45° 30°] - 2.63 m
6. [-42.8° -30°] - 2.53 m



(a) (b)

(c) (d) (e)

Abbildung 3.5: Setup 1a, 1b
a) Quellen 1-3; Torso und Kopf auf Quelle [0° 0°] ausgerichtet; b) Laser am Hals neben und unterhalb interauralen Zentrum; c) Quellen 4-6; Torso und Kopf auf Quelle [45° 0°] ausgerichtet; d) weißer Punkt markiert Laser-Offset zum akustischen Zentrum des Lautsprechers (Tweeter); e) optimale Position des Lautsprechers: Laser trifft Offset-Punkt

3.1.3.6 Aufbau Referenzmessung

Das interaurale Zentrum des Messroboters befand sich 138.75 cm oberhalb des VariSphear-Systems. Bei der Referenzmessung wurde jeweils eines der Miniaturmikrofone an die Stelle des interauralen Zentrums fixiert und auf den Azimuth der jeweiligen Quelle gedreht, um dann wie bei der Messung zwei zu mittelnde Sweeps pro Quellposition aufzunehmen. Außerdem wurde bei der Referenzmessung darauf geachtet, dass die Mikrofonmembran in etwa die gleiche Orientierung im Raum hatte, wie sie auch im geblockten Ohrkanal lag. Somit ist davon auszugehen, dass sich die konstruktionsbedingte einfallswinkelabhängige Gruppenlaufzeit des Mikrofons bei der Entfaltung im späteren Dataprocessing aufhebt. Abbildung 3.6 zeigt eines der beiden Mikrofone ohne Kapsel. Die vier Reihen à 3 Löchern befinden sich nur auf einer Seite, was Ursache der genannten Gruppenlaufzeit ist.

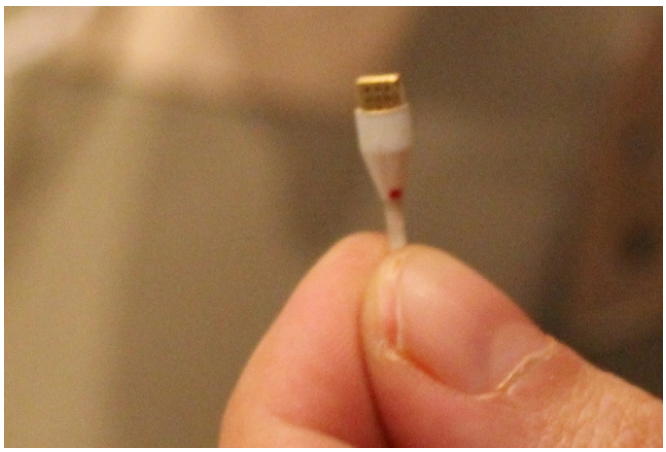
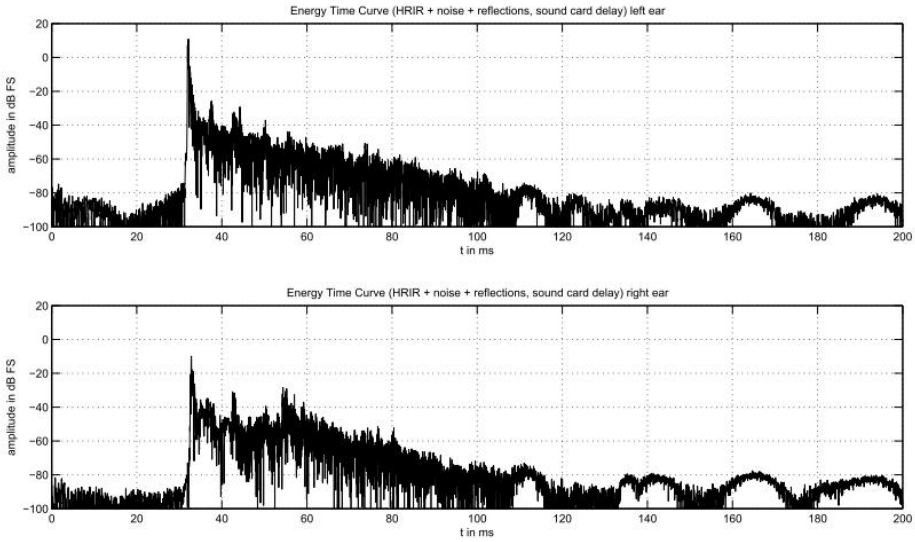


Abbildung 3.6: DPA 4060 Miniaturmikrofon ohne Kapsel

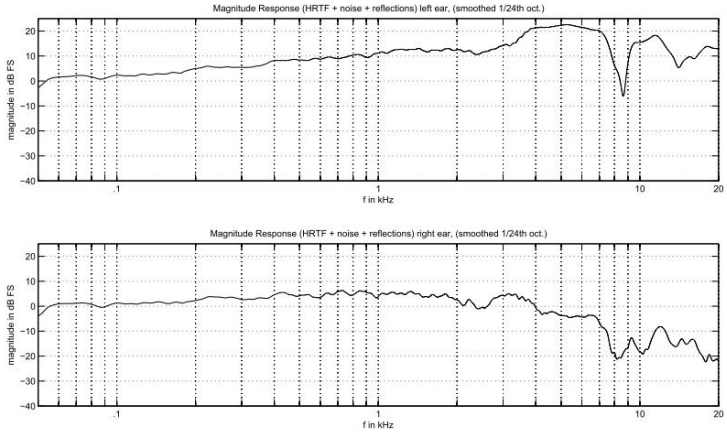
3.1.4 Verlässlichkeit des Aufbaus

3.1.4.1 SNR-Level, Reflexionstest

Brummeinstreuungen wurden durch Auflage der Netzgeräte und Kabelverbindungen auf Schaumstoff verhindert, sowie die Eingangssignale über die Mikrofone stets qualitativ per Kopfhörer geprüft. Zusätzlich ist das SNR-Level bestimmt worden unter Verwendung der Quelle 4, die während der SNR-Messung seitlich auf den Torso strahlte. Das linke Ohr war dabei der Quelle direkt zugewandt, das rechte abgewandt (Kopf-zu-Torso=0°). Bestimmt wurde das SNR bei Mittlung des über den Lautsprecher zweimal abgespielten und durch die Mikrofone wieder aufgenommenen Sweeps. (Weitere Einstellungen (für alle Messungen) waren: Lautsprecher stets ganz aufgedreht, Vorverstärker 30 dB, Sweep-Gain in Matlab 0.04 bzw. -27.96 dB Fs.) Auf der schallabgewandten Seite lag das SNR (Peak zu Rauschen) bei einem guten Wert von 88 dB. Die schallzugewandte Seite lieferte erwartungsgemäß einen noch besseren Wert von 108 dB. Abbildung 3.8 zeigt den frequenzspezifischen Signal-Rausch-Abstand. Diese Überprüfung wurde vor jedem Messstart durchgeführt, um grobe Störungen im Signalpfad ausschließen zu können.



(a) Rauschen und Reflexionen enthaltende HRIR (Faltungsprodukt aus aufgenommenem Signal und invertiertem Testsweep)



(b) Rauschen und Reflexionen enthaltende HRTF

Abbildung 3.7: Ergebnisse des vorab durchgeführten SNR- und Reflexionstests

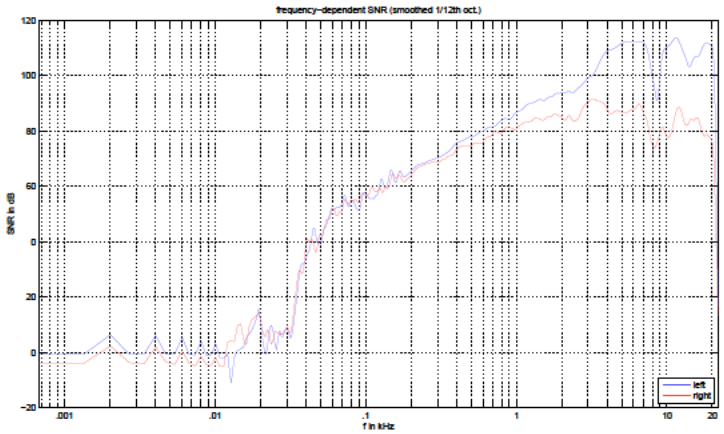


Abbildung 3.8: Frequenzspezifisches SNR

Nicht gewünschte Reflexionen durch den Messaufbau traten erst nach dem Abklingen der HRIR, später als 5 ms nach dem Peak auf (Abb. 3.7). Das Pegelmaximum lag bei -6 dB FS für die schallzugewandte Seite. Für den unerwarteten Fall von Clipping wurde jedoch eine automatische Reduktion des Signalpegels im Messablauf implementiert.

3.1.4.2 Mechanische Verlässlichkeit/Reliabilität

Der Aufbau entsprach dem des SNR-Tests (Quelle 4 bei gleicher VariSphear- und FABIAN-Orientierung). Es wurde je eine HRTF vor und nach mechanischer Höchstbelastung des VariSphears (zehn mal von 180° auf 80° und zurück mit einer Geschwindigkeit von 50°/s und einer Beschleunigung von $50°/s^2$) gemessen. Abweichungen traten bei hohen Frequenzen ab 10 kHz auf, lagen bei ca. 19 kHz bei -0.6 dB, sonst jedoch in einem sehr kleinen Intervall von nur ±0.2 dB (Abb. 3.9).

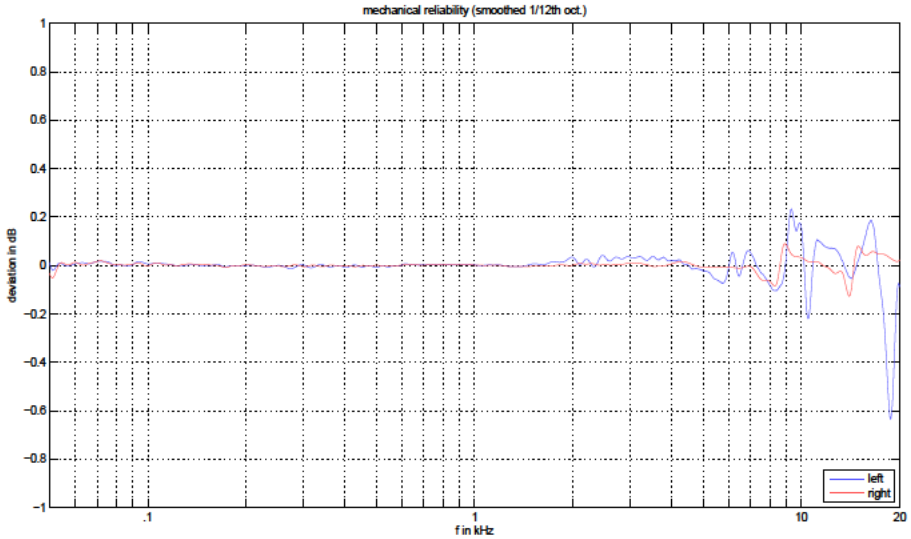


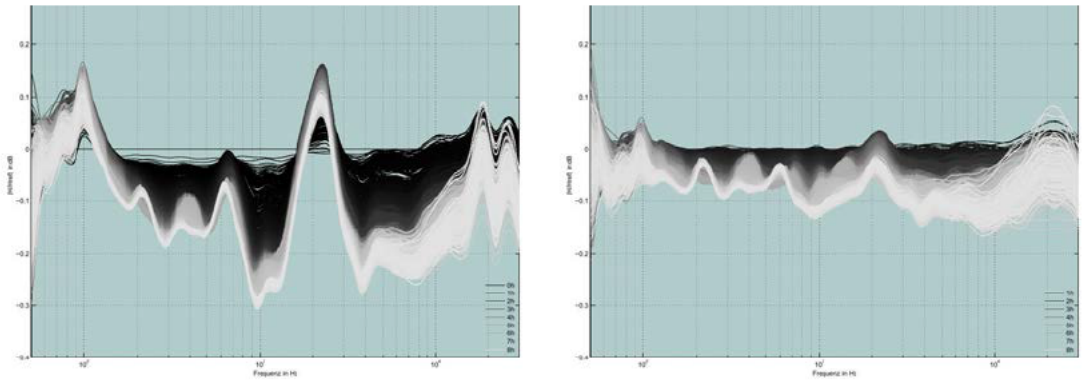
Abbildung 3.9: mechanische Reliabilität

3.1.4.3 Soundkartendelay

Die Soundkarte wurde mittels Matlabskript geprüft. Dabei wurde von Matlab eine wav-Datei ausgegeben, die lediglich einen Impuls enthielt. Die wieder aufgenommene Datei enthielt den Impuls stets 25.74 ms später; bei einer Samplingrate von 44100 Samples pro Sekunde entsprach das einer konstanten Verzögerung von 1135 Samples.

3.1.4.4 Lautsprecherbelastungstest

Um die Kontinuität der Lautsprecher (GENELEC 8030A) über mehrere Messstunden hinweg zu gewährleisten, wurde vorab ein Belastungstest mit einem der Lautsprecher durchgeführt. In einem Raum wurden dieser auf einem Stativ stehend und ein Mikrofon (Behringer ECM 8000) in einem Abstand von ca. 2.2 m aufeinander ausgerichtet. Acht Stunden lang folgte auf zwei Sweeps (siehe Abschnitt 3.1.5) stets eine 10 Sekunden lange Pause, was der Belastung während der Messung entsprach. Die Spektren der vom Mikrofon aufgenommenen Signale sind in Abbildung 3.10 dargestellt. Da Abweichungen schon nach kurzer Zeit geringer wurden (±0.2 dB), begann jede Messung mit einer einstündigen Aufwärmzeit mit selbigem Signalintervall für die Lautsprecher.

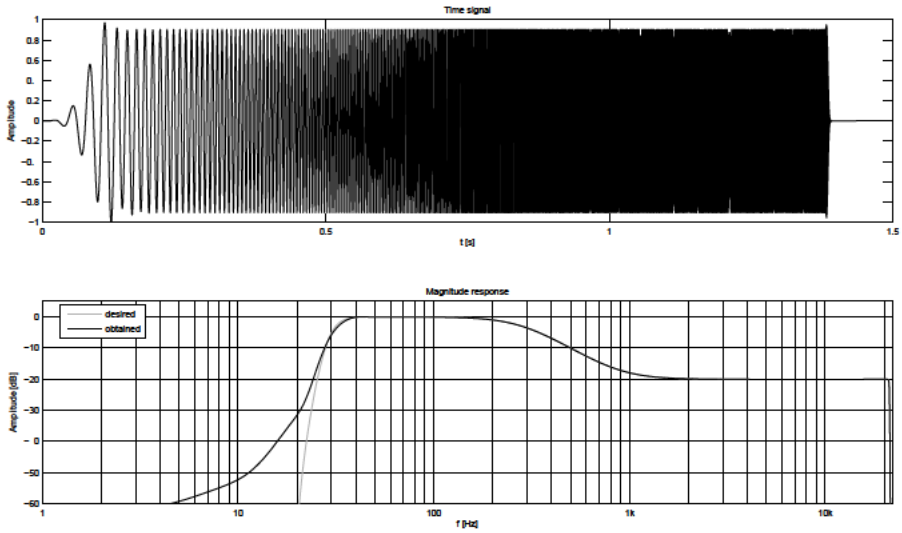


(a) (b)

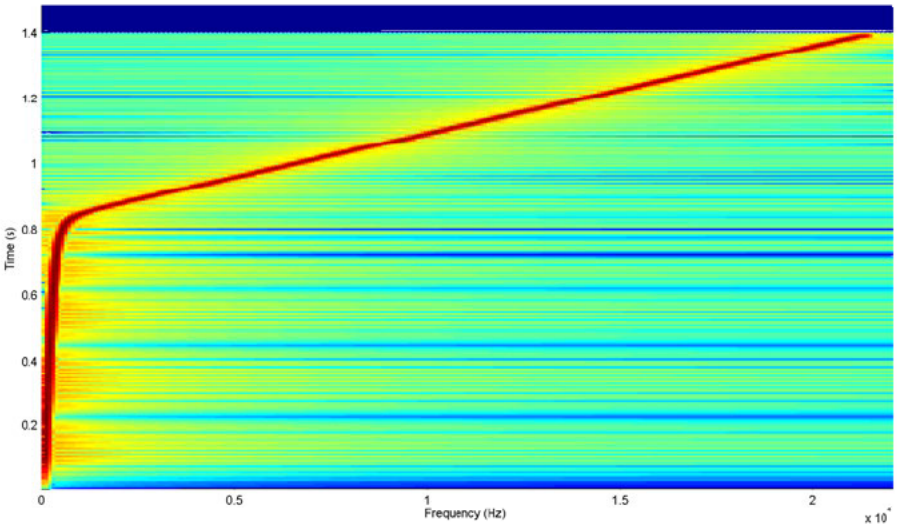
Abbildung 3.10: Ergebniss Lautsprecher-Belastungstest
a) Spektrale Abweichungen des Lautsprechers über 8 Stunden Belastung; b) Verlauf zur Referenz nach einer Stunde

3.1.5 Testsignal

Über die Messsteuerung, die sich mit ca. 3 m Entfernung zu den Mikrofonen innerhalb des reflexionsarmen Raumes befinden musste, wurde ein 2^{16} Samples langer Sweep (Abb. 3.11) bei einer Samplingfrequenz von 44100 Hz generiert. Die Bandbreite wurde zwischen 50 Hz und 21 kHz gewählt, wobei für ein besseres SNR die zusätzliche Bassemphase die Energie des Signals bei tiefen Frequenzen erhöht.



(a) Sweep im Zeit- und Frequenzbereich



(b) Spektrogramm

Abbildung 3.11: Eigenschaften des Testsignals

3.2 HRTF Processing

Im Folgenden werden Schritte der Signalverarbeitung geschildert, die HRTFs aus den Rohdaten „herauslösten“[7].

3.2.1 Pre-Processing

Vorbereitend wurde (mit `b_pre_processing`, siehe Anhang Dib) die mittlere Temperatur aller Messungen der jeweiligen Ebene für eine spätere Temperaturkompensation berechnet. Es wurden Histogramme der Temperaturverteilung gespeichert. Außerdem wurde hier der später für die Entfaltung verwendete Hoch- und Tiefpassfilter angelegt und gesichert, wofür die erstellte und nicht matlabinterne Funktion `deconv` bereitstand. Die Filterparameter basierten auf der Betrachtung bzw. Analyse einzelner spektraler Verläufe der Referenzmessungen, in denen die abfallende Abstrahlleistung verwendeter Lautsprecher unterhalb 70 Hz zu entnehmen war. Dabei wurden für den Hochpass folgende Werte gewählt und letztlich innerhalb `deconv` an `fdesign.highpass('Fst,Fp,Ast,Ap'...)` übergeben: Stopband-Frequenz Fp: 20 Hz; Passband-Frequenz Fst: 70 Hz; Stopband-Dämpfung Ast: 60 dB; Passband-Toleranz Ap: 0.1 dB. Der Hochpass gewährleistete einen identischen Verlauf aller HRTFs unterhalb von 70 Hz. Auf eine Extrapolation der Daten unterhalb dieser Frequenz wurde verzichtet, da hinsichtlich der Fragestellung eines Torsoeinflusses auf das Schallfeld am Trommelfell davon ausgegangen werden konnte, dass dieser Frequenzbereich bei Wellenlängen mehr als 5 m nicht mehr zu einem Einfluss beiträgt.

Der Tiefpass wurde mit diesen Werten erstellt: Stopband-Frequenz Fp: 20 kHz[8]; Passband-Frequenz Fst: 21.5 kHz; Stopband-Dämpfung Ast: 60 dB; Passband-Toleranz Ap: 0.1 dB. Designparameter wurden jeweils über `design` und `filter` zu einem Tschebyscheff-Filter vom Typ 2 umgesetzt und gespeichert.

3.2.2 Entfaltung

Das Matlabskript `c_deconvolution` (Dic) entfaltete aufgenommene Sweeps mit der jeweiligen Referenzmessung (Quelle-ID, Seite) bzw. die aufgenommenen Sweeps der Referenzmessung mit dem Anregungssignal. Darüber hinaus wurden die erhaltenen kopfbezogenen Impulsantworten initial auf 2^{11} Samples gekürzt, um weiteres Processing zu vereinfachen, sowie hoch- und tiefpassgefiltert. Die entfaltete Referenzmessung wird um den Wert des Soundkartendelays (1135 Samples) zeitlich nach vorn verschoben. Um Akausalität der HRIRs (verursacht durch kürzere Strecken von Quelle zu Ohr als zu Referenzmikrofon) zu vermeiden, wurden diese 48 Samples verzögert, was der Laufzeit von 0.0011 s entspricht und worin Zeit für ein 20 Samples langes Fade-In berücksichtigt wurde.

Zur Kontrolle beim Einlesen aufgenommener Sweeps führte ein Log-File fehlende Dateien auf, allerdings gab es wie erwartet keine Einträge.

[7] Wie zur Messung gibt es auch hier die Schilderungen bezüglich der Datenablage in einer separaten Dokumentation.

[8] als obere Grenze des hörbaren Bereichs

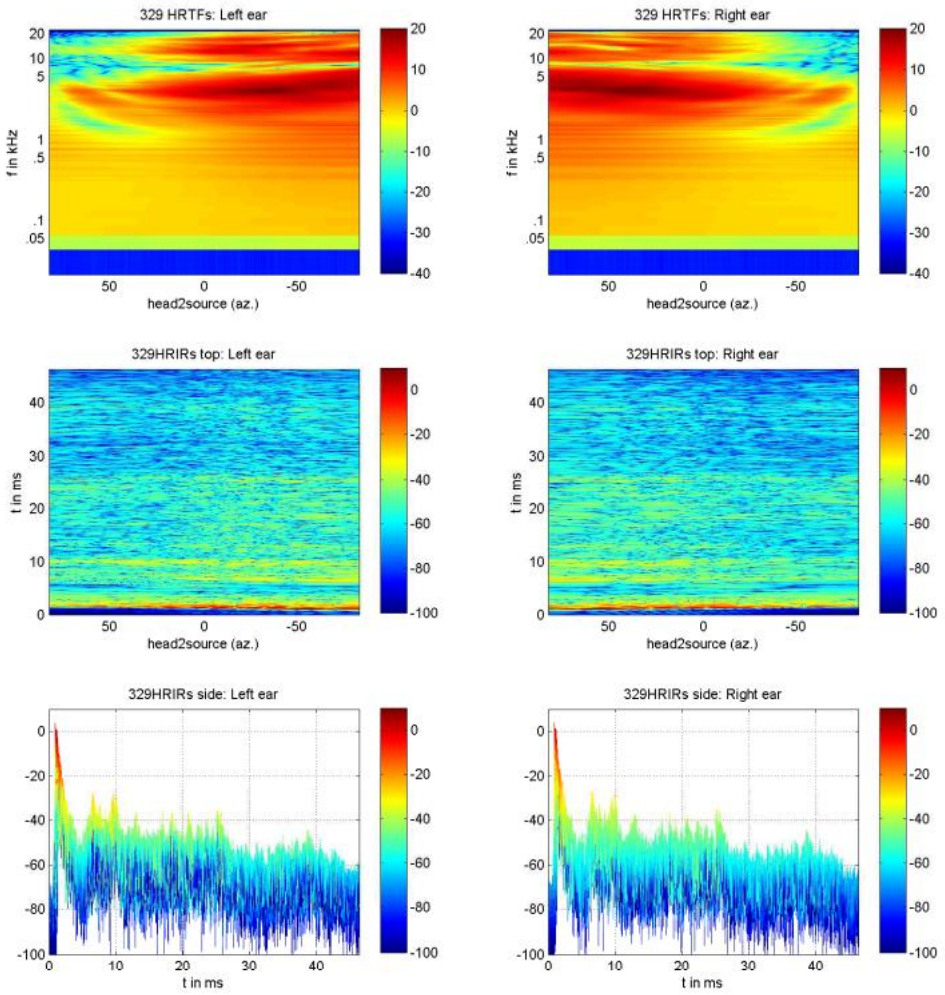


Abbildung 3.12: Beispiel eines Kontrollplots: zeigt HRTFs der Quelle 1 [0°; 0°] in der Horizontalebene über alle Kopf-zu-Quellazimuthwinkel (da Quelle 1 gleich dem Kopfazimuth) bei steter HATO von 0°

3.2.3 Radius-Schätzung

Skript `d_radius_estimation` (Did) schätzt die Entfernung der Mikrofone zur jeweiligen Quelle innerhalb der Referenzmessung unter Berücksichtigung der Temperatur und der davon abhängigen Schallgeschwindigkeit. Die über Onset-Detektion ermittelte Laufzeit in Samples dividiert mit der Samplingfrequenz, multipliziert mit der ermittelten Schallgeschwindigkeit ergibt so den Radius. Die aufgerufene Funktion `onset_detect` arbeitet dabei mit Upsamplingfaktor von 10 und suchte das erste Sample -3 dB unterhalb dem Maximum der ersten 400 Samples. Die Funktion `speed_of_sound` nimmt Luft als ideales Gas an und berechnet Schallgeschwindigkeit c wie folgt:

$$c = \sqrt{\frac{\kappa \cdot R}{M \cdot T}} \tag{3.1}$$

wobei T = Temperatur in Kalvin, M = molare Masse von Luft $28.8 \cdot 10^{-3}$ kg/mol; Adiabatenexponent $\kappa = 1.4$, die universelle Gaskonstante R = 8.314 J/(mol K).

Geschätzte Radii sind bereits neben den Quellpositionen im Abschnitt 3.1.2 vermerkt und entsprachen den grob Ermittelten während der Messung.

3.2.4 Post-Processing

Das Postprocessing wurde mittels dem Skript `e_post_processing` (Die) durchgeführt. Zunächst werden $\sin^2$-Flanken der als Setupparameter festgelegten Länge setup.N_fade (15 Samples) generiert und die Radii sowie Temperaturdaten geladen. Weitere Schritte sollen im Weiteren genauer erläutert werden.

3.2.4.1 Schätzung der interauralen Zeitdifferenzen ITD

Der interaurale Zeitunterschied ergibt sich aus der Differenz der detektierten Peaks des rechten und linken Kanals der ersten 100 Samples der HRIRs. Die ITDs wurden in μs berechnet und über dem Winkel der Kopfbewegung aufgetragen. Aufgrund des leicht unsymmetrischen Kunstkopfes wird für frontalen Schalleinfall eine leicht von 0μs abweichende ITD beobachtet (Abb. 3.13).

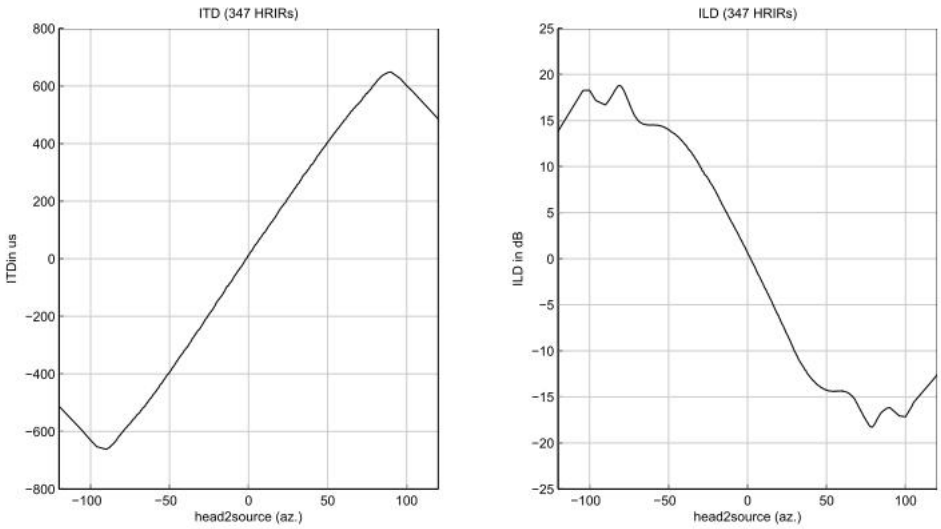


Abbildung 3.13: Kontrollplot der HRTF-ITD und -ILD der Quelle 1 in der Horizontalebene über alle Kopf-zu-Quellazimuthwinkel bei stetem Torsoazimuth von 0°

3.2.4.2 Korrektur der Temperaturveränderung

Da nicht zu jedem aufgenommenen Sweep eine Temperatur gemessen wurde, ordnete man die zeitlich dichteste Temperatur zu. Weicht diese Temperatur von der Temperatur

während der Referenzmessung ab, ist die Schallgeschwindigkeit zu diesen Zeitpunkten ebenfalls unterschiedlich. Jede HRIR wurde daher an den Temperatur-Median angeglichen, sodass Peaks der Impulsantworten benachbarter Winkel mit etwa selber Verzögerung auftreten. Dazu wurde zunächst die Abweichung der Referenzmessung zur mittleren Temperatur in Samples berechnet um davon die Abweichung der HRIR in Samples abzuziehen.

$$Abweichung_{ref} = (\frac{1}{c_{median}} - \frac{1}{c_{ref}}) \cdot radius_{scID} \cdot fs \tag{3.2}$$

$$Abweichung_{hrir} = (\frac{1}{c_{median}} - \frac{1}{c_{hrir}}) \cdot radius_{scID} \cdot fs \tag{3.3}$$

$$Abweichung = Abweichung_{ref} - Abweichung_{hrir} \tag{3.4}$$

Die Funktion `fractional_delay` verschiebt jede HRIR um ihren Wert bei vorheriger Filterung mit einer Kaiser-Fenster gewichteten Sinc-Funktion.

3.2.4.3 Kürzung der Dateien und Fensterung

Gekürzt wurde indem nur der relevante vordere Teil der HRIRs weitergeführt wurde. Die Länge ist in den Setupparametern auf 425 Samples festgelegt worden. Nach der Kürzung folgte das Ein- und Aus-Faden mit vorbereiteten $\sin^2$-Flanken.

3.2.4.4 Generierung der Minimalphase

Bis zu diesem Punkt des Processing, ist die Phase noch nicht berücksichtigt worden. Als Originalphase wird die Phase benannt, die das Spektrum jeder HRIR nach der Entfaltung, Kürzung, Fensterung etc. hat. Sämtliche HRIRs wurden nochmals mit Minimalphase abgelegt, die mittels der Funktion `make_phase` den Beträgen der HRIRs hinzugefügt wurden. Dazu wurden den bis zu dem Punkt gekürzten HRIRs noch einmal 4000 Nullen angehängt, um Minimalphasen-HRTFs betraglich den Originalphasen-HRTFs auch im Tieffrequenzbereich sehr genau anzugleichen. (Minimalphase bedeutet, dass ein System und seine Inverse stabil und kausal sind und per Definition das logarithmierte Betragsspektrum und Phasen-Spektrum Hilberttransformierte von einander sind [KIC95] nach [OSB+89].) Die ITD wird dadurch aus den HRIRs entfernt und separat neben der HRIR im .mat-File gespeichert. Minimalphasige HRTFs werden anschließend ebenfalls im Zeitbereich auf die finale Länge gekürzt, aber nicht mehr mit dem $\sin^2$ multipliziert.

3.3 Zusammenfassung - Messung und Processing

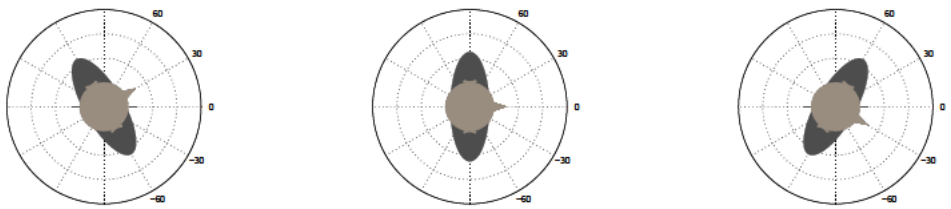
In diesem Kapitel ist die Entstehung des HRTF-Datensatzes dokumentiert worden, der den zu beantwortenden Forschungsfragen zugrunde liegt. Datenreihen enthalten u.a. für sechs feste Quellpositionen die Kopfbewegung bei statischem Torso in Horizontal-, Median- und Frontalebene, sowie die Kopfbewegung bei mitgeführtem Torso in der Horizontalebene. Die mittels Kopf-Torso-Simulator gemessenen Daten durchliefen dabei das beschriebene, im Anhang ersichtliche Processing, um letztlich in zweifacher Ausführung minimal- und originalphasig vorzuliegen.

Kapitel 4

Torsoeinfluss in HRTFs

Der Torsoeinfluss auf das Schallfeld am Trommelfell wird offensichtlich im Vergleich folgender zwei HRTF-Datenreihen, die physikalisch sowie perzeptiv einander gegenübergestellt wurden. Im Weiteren werden diese als Bewegungen a) und b) geführt:

- a) horizontale Bewegung des Torsos zwischen ±82° bei stets neutraler, statischer Kopf-über-Torso-Orientierung (Abb. 4.1)
- b) horizontale Bewegung des Kopfes zwischen ±82° bei steter Torsoausrichtung von 0° und sich dynamisch ändernder HATO (Abb. 4.2)



(a) (b) (c)

Abbildung 4.1: statische HATO: Kopfbewegung bei Mitführung des Torsos hier angedeutet zwischen 30 und -30°

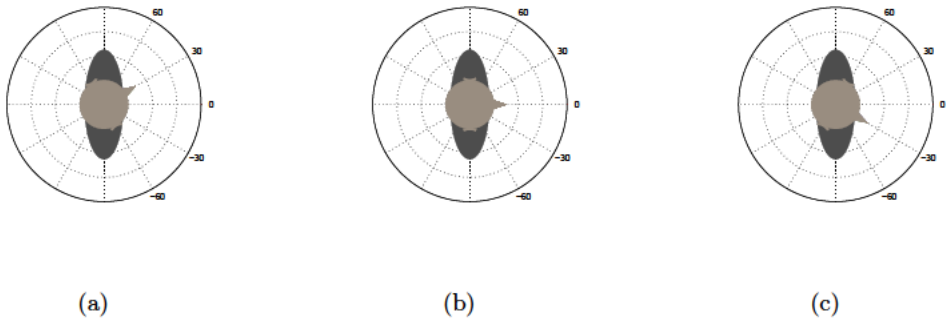


Abbildung 4.2: dynamische HATO: Kopfbewegung bei statischem Torso hier angedeutet zwischen 30 und -30°

4.1 Physikalische Evaluation

Zur Ermittlung des physikalischen Unterschieds des Schalldrucks am geblockten Ohrkanal sind Impulsantworten sowie Spektren aufgenommener Signale untersucht worden. Beispielhaft sei hier die Bewegung der dynamischen HATO bei statischer Torsoausrichtung gezeigt (Abb. 4.3). Durch die Symmetrie linker und rechter Seite ist zu erkennen, dass es sich um eine Quelle in der Medianebene handelt.

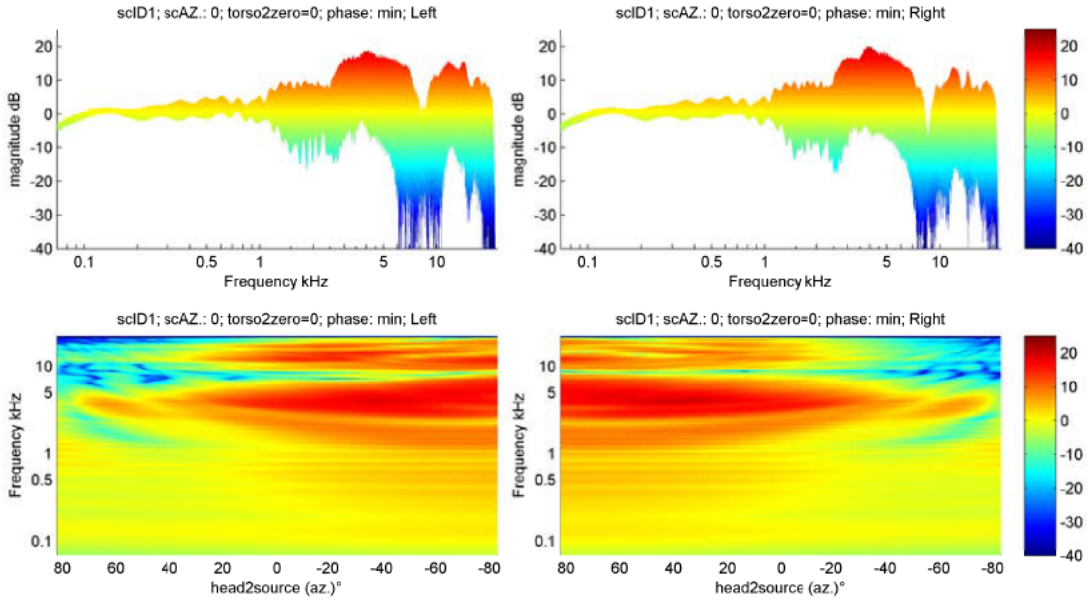


Abbildung 4.3: dynam. HATO bei Torsoazimuth=0°, Quelle 1
oben: seitlicher Blick auf die Daten, unten: Blick von oben auf die Daten

Zunächst wurden dafür frequenzabhängige Beträge der HRTFs beider Bewegungen über dem Kopf-zu-Quell-Winkel für alle sechs Quellen aufgetragen. Dafür wurden alle benötigten HRIRs in kleinster vorliegender Auflösung von 0.5° aus dem Datensatz extrahiert und fouriertransformiert. Die logarithmierten Beträge dessen sind für

Bewegung a) in den Bildgruppen A.1 und für b) in A.2 im Anhang zu sehen. Es sind viele Gemeinsamkeiten zu erkennen. Der ausgeprägteste Notch um 8 kHz ist beiden Arten der Bewegung zu entnehmen, sowie die Kammfiltereigenschaft bis zu hohen Frequenzen. Die schallzugewandte Seite besitzt erwartungsgemäß stets höhere Beträge.

4.1.1 Differenzspektren

Um den physikalischen Unterschied zwischen beiden Bewegungen sichtbar zu machen, wurden die Betragsspektren der HRTFs beider Bewegungen (HRTF_a, HRTF_b) dividiert und anschließend logarithmiert ($20 \cdot \log_{10} \frac{|HRTF_a|}{|HRTF_b|}$). Für Quelle 1 ergibt sich in Abbildung 4.4 zu sehendes Differenzspektrum.

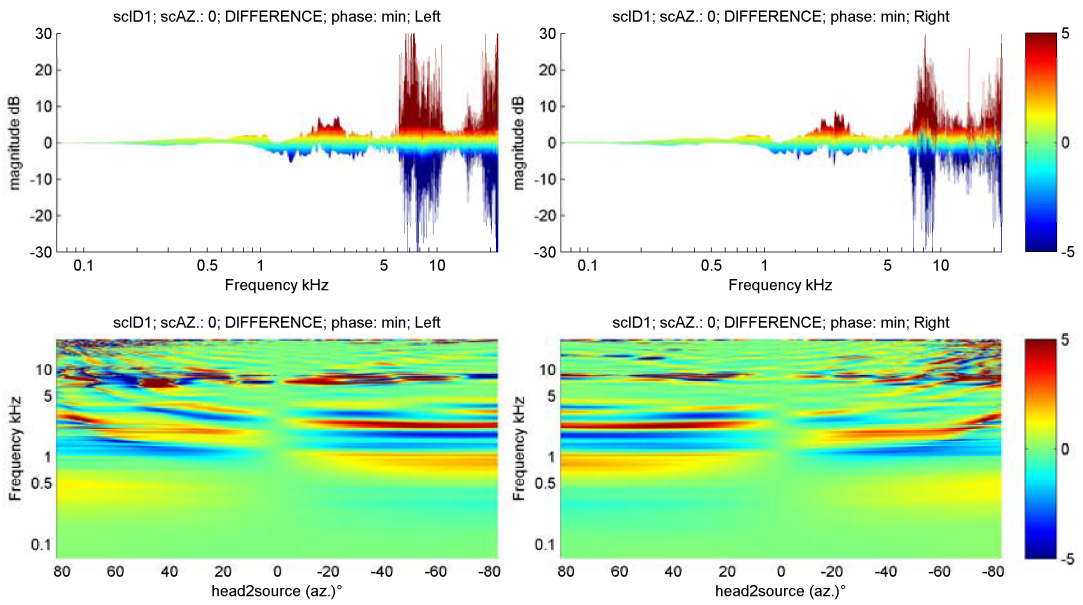


Abbildung 4.4: Differenz (stat. - dynam.), Quelle 1

Werden gezeigte Ergebnisse (siehe auch Abb.-gruppe A.3 im Anhang) negativ ist entsprechend das Betragsspektrum der Bewegung b) bei dynamischer HATO größer, der Quotient demnach kleiner eins und der Wert nach dem Logarithmieren kleiner als 0 dB. Für eine geeignete Darstellung sind Unterschiede zwischen ±5 dB farblich skaliert - Werte über dieses Intervall hinaus, die für die Wahrnehmung bereits als erheblich einzustufen sind[9], bleiben rot bzw. blau. In der Aufsicht (Frequenz über Winkel) werden daher die Maxima des betraglichen Unterschieds nicht deutlich. Zur Ergänzung des Betrags wurde die Seitenansicht (Betrag über Frequenz) erstellt, mit der der exakte Einfluss des Torsos in dB (teilweise weit über ±5 dB hinaus) für jede einzelne Quelle ersichtlich wird. Im Bereich der neutralen Kopf- und Torso-Position sollte es bis auf mögliche Messungenauigkeiten physikalisch keine Differenz geben, da beide Bewegungen bei Ausrichtung auf [0°;0°] die identische HATO und auch Torsoposition aufweisen. Für

[9] Für weißes Rauschen bei einem Darbietungspegel von 30 dB SL ermittelte Miller in einer grundlegenden Studie die eben merkliche Schwelle einer Lautstärkeänderung von 0.49 dB [Mil47].

diesen Winkel wurde daher die selbe einmal gemessene HRTF verwendet. Es bestätigt sich aber, dass die Differenzen auch um den Azimuth 0° des Koordinatensystems (stets mittig in den Abbildungen) gegen Null gehen und nur bei hohen Frequenzen durch Ungenauigkeiten von Null abweichen. Zu größeren Auslenkungen hin werden Differenzen sehr deutlich. Diese lassen sich als Verstimmungen der Notches und des Kammfilters der Schultern beschreiben. Kapitel 2.3 zeigte den Torsoeinfluss anhand der Literatur ab 700 Hz; hier sind leichte Änderungen bereits ab ca. 500 Hz zu beobachten. Wird nach optischen Gesichtspunkten die Stärke aller Abweichungen geordnet, ändert sich für die von oben auf den Kopf strahlende Quelle am wenigsten, da bei Betrachtung des Schallwegs Reflexionen und Abschattungen der Schultern eher irrelevant für die Position des Ohrkanals sind. Quellen innerhalb und oberhalb der Horizontalebene scheinen sich vom Ausmaß zu ähneln. Anhand dieser Abbildungen sind Differenzen für die negativ elevierte Quelle (Nr. 6) am stärksten.

4.1.2 Physikalisch-perzeptives Einzahlmaß

Um die Quellen hinsichtlich ihrer kritischen Positionen in eine Rangfolge einordnen zu können, wurde ein Einzahlwert je Winkel entwickelt, der sich an hörphysiologischen Eigenschaften der Basilarmembran orientiert. Unendlich viele, sich überlappende auditorische Filter der tonotopen Basilarmembran werden im Modell der ERB-Skala in endlich viele, sich nicht überlappende äquivalente Rechteckfilter eingeteilt und durchnummeriert [Moo95b]. Das Einzahlmaß bedient sich dem Modell der ERB-Skala und lehnt sich an die Ausführungen von Schärer und Lindau [SL09] an, die für die Berechnung des spektralen, betraglichen Unterschieds zweier Signale innerhalb eines auditorischen Filterbands folgende Gleichung ebenfalls auf Moore [Moo95a] und im Weiteren auf Salomons [Sal95] zurückgehend aufstellten. Die Gleichung wurde im genutzten Skript Schärers `erb_error` implementiert:

$$E(f_c) = 10 \log_{10} \left(\frac{\int_{-\infty}^{\infty} C(f, f_c) |HRTF_a|^2 df}{\int_{-\infty}^{\infty} C(f, f_c) |HRTF_b|^2 df} \right) \tag{4.1}$$

Der spektrale Fehler $E(f_c)$ in Gleichung 4.1, abhängig vom Filterband einer charakteristischen Frequenz auf der Basilarmembran, ergibt sich als logarithmierte Angabe des Quotients aus den integrierten, mit dem jeweiligen auditorischen Filter frequenzweise gewichteten quadrierten Betragsspektren zu vergleichender Signale. Im Digitalen ist es die Summe über die Frequenzbins.

Für die auditorischen Filter wurde die Form eines Gammatone-Filters nach Patterson und Holdworth zur Simulation der Cochlea verwendet. Koeffizienten wurden mit dem Skript `MakeERBFilters` von Malcolm Slaney mit genauer Literatur und Beschreibung zu finden in [Sla93] generiert.

Erstreckt sich der interessierende Frequenzbereich zwischen der unteren und oberen Grenzfrequenz f_{unten} und f_{oben} über mehrere Filter der ERB-Filterbank, so errechnet sich die Anzahl n zu berücksichtigender auditorischer Filter der jeweiligen Mittenfrequenz f_c aus der Differenz zugewiesener Bandnummern der auditorischen

Filter, die sich um die Frequenzen f_{oben} und f_{unten} in Hz[10] befinden.

$$n = \underbrace{21.4\log_{10}(0.004367 f_{oben} + 1)}_{ERBBandnummer f_{oben}} - \underbrace{21.4\log_{10}(0.004367 f_{unten} + 1)}_{ERBBandnummer f_{unten}} \tag{4.2}$$

Da wie im Abschnitt 3.2.1 bereits angeführt, die HRTF-Daten aufgrund geringer Abstrahlleistung der bei der Messung verwendeten Lautsprecher bei 70 Hz hochpassgefiltert wurden und darüber hinaus der Torsoeinfluss (siehe 2.3) nicht im Tieffrequenzbereich vermutet wird, wurden je Winkel alle $E(f_c)$ für die auditorischen Filter im Bereich von 70 Hz bis zur oberen Grenze des Hörbereichs 20 kHz betrachtet. Übergeben wurden für jedes n die HRTF der Bewegung a) und b) für rechtes und auch linkes Ohr[11]. Damit lagen zunächst für jeden Winkel zwischen ±82° Vektoren mit $E(f_c)$-Werten der Länge n der rechten und linken Seite vor. Beträge der Seiten in dB wurden aufaddiert und anschließend über alle n Mittenfrequenzen der auditorischen Filter gemittelt. In den Abbildungen unter A.3 sind nicht nur die Differenzspektren aufgetragen, sondern unterhalb dessen das Einzahlmaß über dem Winkel in der Auflösung von 0.5 dB Schritten wie der in Matlab erstellten „Colormap“ entnommen werden kann. Die Art des Diagramms ist der Darstellung eines physikalischen Interpolationsfehlers bei Minnaar et. al [MPC05] mit zusätzlicher Berücksichtigung der auditorischen Filter nachempfunden, die mittels diesen Maßes richtige Tendenzen für ein durchgeführtes Hörexperiment zur Hörbarkeit des Interpolationsfehlers vorhersagen konnten. Der Wertebereich liegt bei Minnaar et al. ähnlich zu hier gezeigten Werten zwischen 0 und 2.5 dB in 0.1 dB Stufen vor. Hinsichtlich der Interpolation wird das Fehlermaß in dieser Arbeit noch einmal im Kapitel 5 aufgegriffen.

Das Differenzspektrum für Quelle 1 führt zu den Einzahlwerten je Winkel, wie sie in Abbildung 4.5 dargestellt sind. Wie der Farbkodierung zu entnehmen ist, können dieser Quellposition die drei untersten Stufen des Fehlers zugewiesen werden. Das Einzahlmaß liefert Werte zwischen 0 dB bei mittleren Winkeln und 1.5 dB zu äußeren Winkeln.

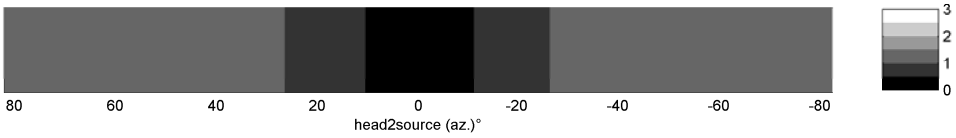


Abbildung 4.5: Differenz (stat. - dynam.), Quelle 1

Noch einmal die Gruppe von Abbildungen A.3 hinsichtlich des Einzahlwertes betrachtet ergibt sich eine Rangfolge kritischer Quellen, die die Tendenz aus Betrachtungen der bloßen Differenz bestätigen. Nach wie vor ist der Einfluss des Torsos für Quelle 3 am geringsten, gefolgt von Quelle 5 (oberhalb der Horizontalebene), die gegen ±82° hin etwas unauffälliger ist als Quelle 2 (innerhalb der Horizontalebene). Die Rangfolge setzt sich mit Quelle 4, dann 1 fort. Ebenfalls bestätigt sich Quelle 6 als kritischste Position. Die Vermutung hier am ehesten Bewegungen unterscheiden zu können, festigt sich.

[10] Gleichungen bei Moore [Moo95a] basieren auf Frequenzangaben in kHz

[11] Ein Signal z.B. Bewegung a)/rechtes Ohr wurde n mal gefiltert nach Glg. 4.1

4.1.3 Einfluss auf ILD und ITD

Klangfärbende Unterschiede beider Bewegungen sind in den Spektren ersichtlich. Um den Unterschied beider Bewegungen hinsichtlich des räumlichen Hörens genauer zu erfassen wurden ILD und ITD (Abb.-gruppen A.4 und A.5 im Anhang) beider Bewegungen über dem Winkel bestimmt. Die interauralen Zeitdifferenzen wurden für diesen Zweck wie im Abschnitt 3.2.4.1 über Onset-Detektierung ermittelt. Die interaurale Pegeldifferenz errechnete sich für jeden Winkel jeder Bewegung aus der Differenz logarithmierter RMS-Werte linker und rechter HRIR (einer finalen Länge von 425 Samples). (in Matlab berechnet: Glg. 4.3).

$$ILD = 20 \cdot \log 10(sqrt(mean(HRIR_left\hat{.}2))) - 20 \cdot \log 10(sqrt(mean(HRIR_right\hat{.}2))); \tag{4.3}$$

Abweichungen halten sich für alle Quellen im geringen Ausmaß. Die ILD-Differenz bewegt sich für die Quellen 1, 2 und 5 ausschließlich im unterschwelligen Bereich (unterhalb 0.6 dB). Für die Quellen 3, 4 und 6 erreichen die Werte in manchen Winkelregionen den Schwellwert. Am stärksten ist dieser Effekt für die unterhalb der Horizontalebene platzierte Quelle 6, bei Winkeln, an denen sich die rechte Pinna auf die Quelle ausrichtet. Bei einer späteren Auralisation wäre damit ein wahrzunehmender Lokalisationsunterschied der selben Quellen in Bewegung a) und b) durch die ILD-Abweichung möglich. Effekte durch ITD und ILD ordnen sich im Ganzen eindeutig spektralen Verfärbungen unter und sind eher unempfindlich für die Position des Torsos, denn auch ITD-Differenzen halten sich gering. Ein Sonderfall ist für die ITD aber hervorzuheben. Das Eintreffen des Direktschalls am Ohr kann sich erwartungsgemäß nur im Fall eines direkten Hindernisses durch den Torso verzögern. Bei einem Kopf-zu-Quellwinkel von ca. 90° in der ITD-Abbildung zur Quelle 6 tritt diese Verzögerung für statische HATO ein - die Schulter schiebt sich zwischen Ohr und Quelle. Die Differenz der ITDs erhöht sich über einen kleinen Winkelbereich von wenigen Grad auf über 40μs (!).

4.2 Perzeptive Evaluation

In Kap. 4.1 gezeigter physikalischer Torsoeinfluss wurde in einem ABX-Vergleich auf Hörbarkeit hin untersucht. Im Weiteren werden Auralisierung des Unterschieds, Methode und Verfahren des ABX-Vergleichs, sowie dessen Auswertung geschildert. Zu prüfende Hypothese H_1 lautet: Der Unterschied zwischen HRTF-Daten für Bewegung a) und b) ist hörbar.

4.2.1 Grundlegendes zum ABX-Test

Der ABX-Test ist ein gängiges Verfahren im Bereich der Audiokommunikation und -signalverarbeitung eine Hörbarkeitshypothese zu überprüfen oder aber zu verwerfen. Etwa bei neu entwickelten Algorithmen der Audiodatenverarbeitung ist es wichtig zu erfahren, ob diese Artefakte im hörbaren, überschwelligen Bereich hinterlassen. Im Grunde handelt es sich bei dem Verfahren um ein Bernoulli-Experiment, das je Versuch zwei mögliche Ausgänge, richtig und falsch, und damit eine Ratewahrscheinlichkeit von 50% hat. Innerhalb eines Versuchs hört sich der Proband bzw. die Probandin die Signale A und B an, entscheidet des Weiteren ob ein Signal X wie A oder B klingt. Üblicherweise und so auch in dieser Studie wird keine zeitliche Begrenzung

festgelegt - der Teilnehmende hört sich die Stimuli beliebig oft und lang an. Das Verfahren und seine statistischen Größen ist bei Leventhal [Lev86] und Burstein [Bur88] beschrieben. Es wird die Wahrscheinlichkeit berechnet, ob die Anzahl richtiger Antworten sich möglicherweise zufällig ergab und ob anhand dessen die Alternativhypothese angenommen oder verworfen werden kann.

Ist die Hörbarkeit für den Probanden sehr offensichtlich, handelt es sich um große Effekte, die schon bei wenigen Versuchen, auch Trials genannt, statistisch gesichert werden können. Die Nichthörbarkeit ist statistisch nicht nachweisbar, da das Experiment nicht unendlich oft wiederholt werden kann, um auch den kleinsten Effekt ausschließen zu können. Der Test kann hinsichtlich einer Nichthörbarkeit demnach nie die notwendige Empfindlichkeit zeigen. Im Konzept Bursteins und Leventhals ist die Effektgröße gleichzusetzen mit dem resultierenden Prozent-Korrekt-Wert, der bei Nichthörbarkeit wie die Ratewahrscheinlichkeit 50% bzw. 0.5 betragen würde.

4.2.2 Wahl der Stimuli

Es konnten im Vergleich nicht alle sechs Quellen einbezogen werden, da das Hörexperiment in einer dem Probanden zumutbaren Länge absolvierbar sein sollte. Ergebnisse der physikalischen Evaluation und ein informeller Vortest stellten drei Quellen heraus, für die es vermutlich unterschiedlich schwer sein würde im ABX-Test die zwei HRTF-Datensätze zu erkennen. Quelle 3 zeigte am wenigsten Veränderungen im physikalisch-perzeptiven Einzahlmaß aufgrund geringster spektraler Differenzen im Vergleich beider Bewegungen. Quelle 2 schien eine mittlere Schwierigkeit aufzuweisen und Quelle 6 eine besonders leichte Detektierbarkeit, da sie wie auch in der Literatur beschrieben als kritischste Quelle angesehen werden kann.

HRTF-Daten bzw. HRIR-Daten jeder dieser drei Quellposition wurde für die Faltung mit einem rosa Rauschen und einem Sprachausschnitt eines männlichen Redners hinterlegt. Ein durchgängiges rosa Rauschen eignete sich als breitbandiges Signal am ehesten die aufgezeigten spektralen Unterschiede der Bewegungen a) und b) hörbar zu machen und stellt damit den kritischsten Fall (pro Hörbarkeit) dar. Um die Hörbarkeitshypothese auf einen alltäglichen und weniger kritischen Fall (contra Hörbarkeit) anzuwenden, wurde der Sprachausschnitt gewählt. Es ergaben sich daraus sechs Testpaare (3 Quellen à 2 Inhalte) und damit sechs Einheiten des ABX-Tests. In jeder dieser Einheiten wurde der Proband für mehrere Trials gebeten das Signal X dem Signal A oder B zuzuordnen.

4.2.3 Stichprobengröße

Die Anzahl der Trials einer Test-Einheit für einen Versuchsteilnehmenden ergab sich aus der mit dem Programm GPower 3.1 [FELB07] apriori berechneten Stichprobengröße bei zunächst angenommener Effektgröße p_2 von 75%, einem α-Fehler von 5%, einem β-Fehler von 20% und der Ratewahrscheinlichkeit p_1 von 50%. Abbildung 4.6 verdeutlicht den Zusammenhang der Bernoulli-Verteilungen über p_1 der Nullhypothese und p_2 der Alternativhypothese und bildet den α-Fehler (Wahrscheinlichkeit Nullhypothese fälschlicherweise zurückzuweisen), sowie β-Fehler (Wahrscheinlichkeit Nullhypothese fälschlicherweise beizubehalten) ab. GPower errechnete über diesen Zusammenhang der Bernoulliverteilungen, dass bei 16 richtigen Antworten von n=23 Trials sich das Ergebnis signifikant von der Ratewahrscheinlichkeit unterscheide.

Unter Berücksichtigung der Alphafehler-Kumulierung, die durch multiples Testen aller Probanden zu jeder Test-Einheit schnell groß wird, wurde die Anzahl an Probanden auf ca. zehn, letztlich genau elf Versuchsteilnehmende beschränkt, um nach Korrektur dieser Inflation nicht von einem zu kleinen α-Niveau ausgehen zu müssen (nähere Erläuterungen im Abschnitt 4.2.8). Bei elf Probanden à 23 Abfragen je Einheit ergibt sich die Stichprobengröße von 253 nach Leventhal [Lev86] als unabhängig anzunehmende Versuche für jedes der sechs Testsignale.

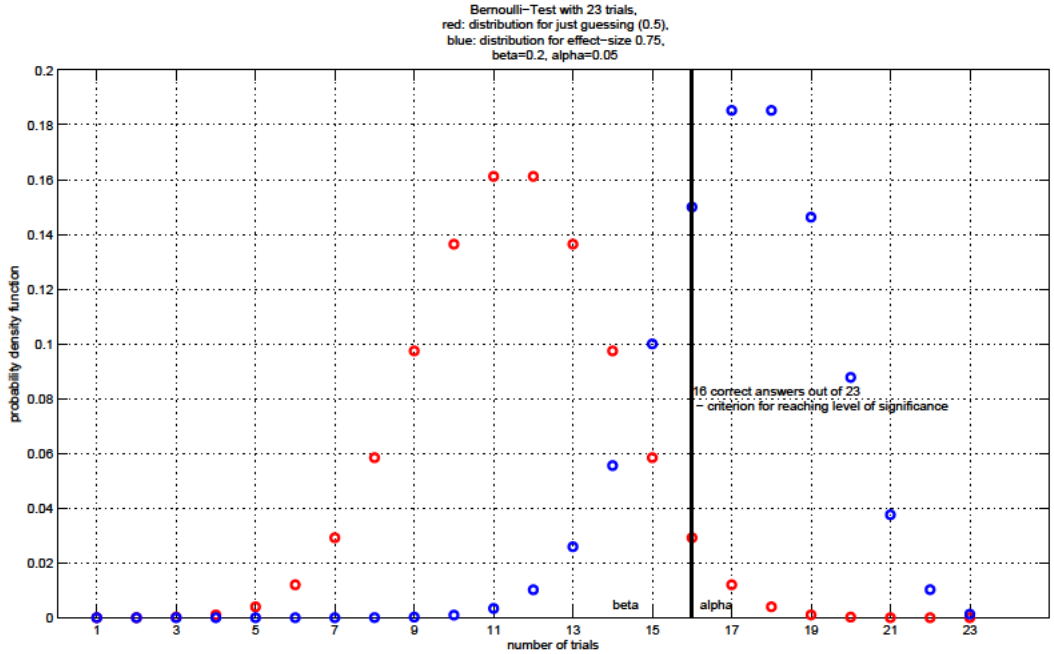


Abbildung 4.6: Bernoulli-Verteilungen für die Effektgrößen 50 und 75% bei 23 Trials ab einem kritischen Wert von 16 ist der α-Fehler kleiner 5% und der β-Fehler kleiner 20% (die Teststärke (1-β) damit größer 80%)

4.2.4 Probanden

Als Probanden wurden normalhörende Menschen eingeladen, die als Expertenhörer Hörerfahrung vorweisen konnten. Interessierte wurden gebeten, ihre Zeit des aktiven Hörens im Alltag (beruflich wie privat) quantitativ und qualitativ zu beschreiben. Dies und ihre aktuelle Lärmexposition, sowie der gesundheitliche Zustand wurde in einem Probandenfragebogen (siehe Anhang B und C) festgehalten. Der Median aus Altersangaben, der sieben männlichen, und vier weiblichen Personen betrug 29 Jahre. Die eigene gesundheitliche Verfassung beschrieben fast alle Probanden als mindestens gut. Darüber hinaus wurde unmittelbar vor dem Experiment die Ruhehörschwelle der Luftleitung eines jeden Teilnehmenden mittels Békèsy-Verfahren mit Festfrequenzen (125 Hz, 250 Hz, 500 Hz, 1 kHz, 1.5 kHz, 2 kHz, 4 kHz, 6 kHz, 8 kHz, nach E DIN EN ISO 8253-1:2008-09) ermittelt. Prüfton war stets ein Sinus-Puls. Gemessen wurde mit der Software „ear“ unter Windows und einem Audiometriekopfhörer Sennheiser HDA 200. Jede der elf Personen erwiesen sich nach

den Kriterien der Weltgesundheitsorganisation[12] als normalhörend (Tonaudiogramme im Tabellenanhang C.1).

Bei den Teilnehmenden handelte es sich um Mitarbeiter im Fachgebiet Audiokommunikation der TU Berlin, darunter die Versuchsleitung, sowie andere Hörexperten.

4.2.5 Geräte, Signalpfad

Verwendete Geräte:

- PC-Anwendungen unter Linux (Ubuntu):
  - fWonder (Software zur Faltung in (Echtzeit) + ITD-Stretcher)
  - pure data pd
  - Jack-Connect
- Tablet-PC-Anwendung unter Windows:
  - Matlab 2009a
  - WHISPER v1.06 (in Matlab programmierte Umgebung zur Durchführung psychoakustischer Tests)
- Head-Tracker POLHEMUS Patriot
- extraauraler Kopfhörer BK211 + BKamp [ESLW12]

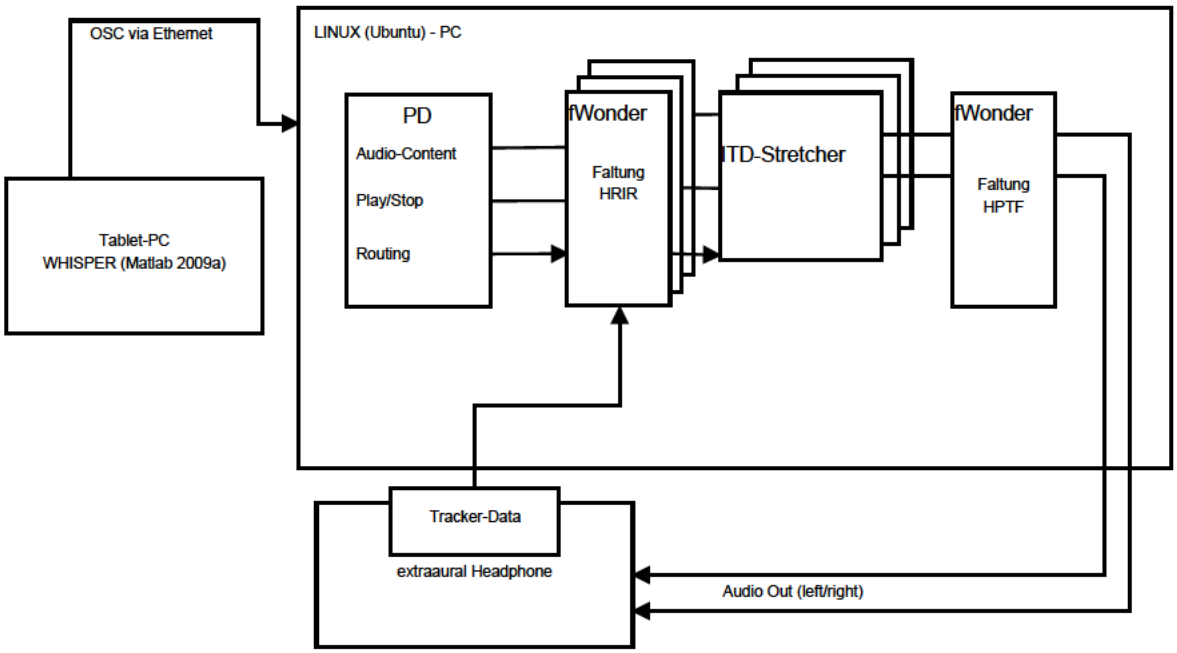


Abbildung 4.7: im ABX-Vergleich verwendeter Signalpfad

[12]normalhörend: Mittelwert der Hörschwelle in dB HL bei 500, 1000, 2000 und 4000 Hz < 25 dB HL auf dem besseren Ohr; http://www.who.int/pbd/deafness/hearing_impairment_grades/en/index.html, 20.07.2013 10:05

In WHISPER, einer Versuchsumgebung in Matlab für psychoakustische Experimente, wurden durch die vom Probanden angewählten Button einer Oberfläche (Abb. 4.8) hinterlegte OSC-Befehle[13] via Netzwerkkabel zunächst an pd gesendet (Abb. 4.7). Für sechs Quellen, für die jeweils zwei Bewegungen auralisiert werden sollten, wurden im Vorfeld stets zwölf fWonder-Konfigurationen geöffnet, die so zum unverzögerten Falten und Ausgeben der Daten bereit waren. Letztlich wurden im Experiment davon sechs genutzt, da nur HRTFs für drei Quellpositionen für die Stimuli verwendet wurden. Mittels „Jack Connect“ und zugewiesenen Ports erhält die jeweilige Konfiguration der in C++ programmierten Software fWonder diese Befehle und greift auf die anzusprechenden HRIR-Daten zu, die für die Quelle und Bewegung je in einem Ordner hinterlegt waren. Tracker-Informationen gelangten per serielle Schnittstelle (9-Pin-zu-USB-Adapter) in ein fortlaufendes txt-File, welches stets direkt „hard gecoded“ von fWonder ausgewertet wurde, sodass die horizontale Position des Kopfes bei z.B. 20° Azimuth zum Falten des im reflexionsarmen Raum aufgenommenen Sprachsignals oder des rosa Rauschens mit der HRTF bei 20° führt. Vorheriges kalibrieren der neutralen Kopfposition ist vorausgesetzt. Da die Faltungssoftware zunächst keine 0.5°-Schritte lesen konnte, war es notwendig die Daten im txt-File so zu manipulieren, dass dieses Problem umgangen werden konnte. Im PD-File wurde dafür eine Unterfunktion tracker_pan_converter aufgerufen (Abb. 4.9).

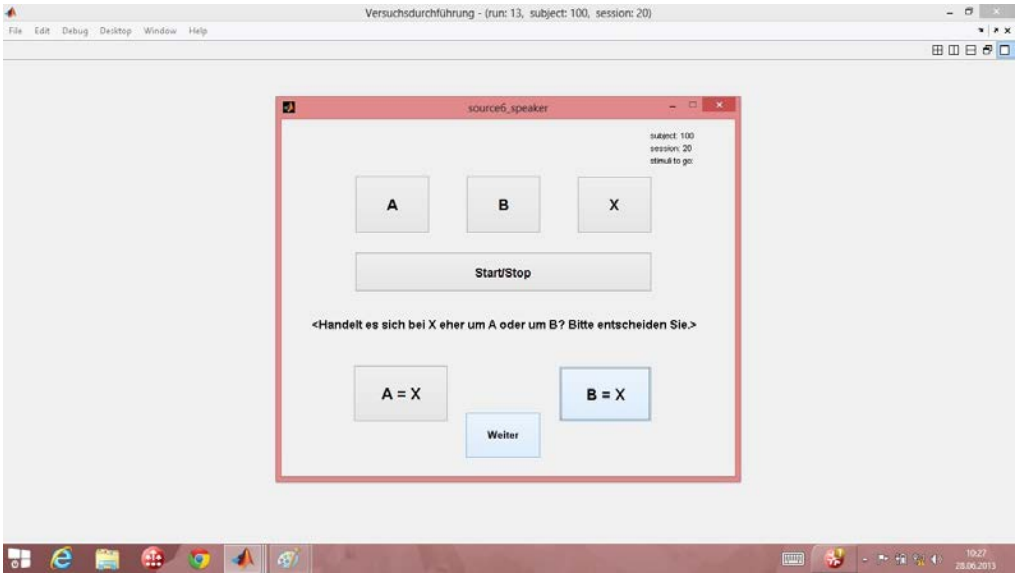


Abbildung 4.8: Benutzeroberfläche in WHISPER

Da minimalphasige HRTFs bzw. HRIRs zum Auralisieren verwendet wurden[14], deren ITDs nicht mehr im zeitlichen Signal selbst enthalten sind, fügte der mit fWonder gelieferte Stretcher die jeweilige, im Processing bestimmten ITDs zu den Signalen hinzu, sodass die Quelle in ihrer ungefähren Richtung vom Probanden wahrgenommen werden konnte. (Da es sich nicht um individuelle HRTFs handelte, sind Lokalisationsabweichung, Internalisierung und Vorne-Hinten-Vertauschungen

[13] OSC Open Sound Control

[14] Die Originalphase wurde auf Grund von Artefakten bei vor allem schnelleren Kopfbewegungen nicht verwendet.

wahrscheinlich gewesen. Jedoch waren diese Effekte im Vorfeld für dieses Experiment als irrelevant einzuschätzen, da sie für jede Bewegung einer Quelle theoretisch und praktisch gleichermaßen, systematisch auftreten würden.) Die ITDs sind in der Vorbereitung der HRIR-wav-Files in ein txt-File geschrieben und im gemeinsamen Ordner abgespeichert worden. Nach dem Stretchen fügte sich eine weitere fWonder-Konfiguration zum Kompensieren des Kopfhörers an. Die Filterfunktion (invertierte Übertragungsfunktion mit regularisiertem Notch) lag ebenfalls als Zeitsignal vor. So bearbeitet geht das zweikanalige Signal letztlich über den BKamp an den extraauralen Kopfhörer BK211 [ESLW12]. Bei Kopfbewegung auftretende Latenzen sind dabei nicht wahrzunehmen.

Der Pegel der Signale musste nicht auf gleiche Lautheit angepasst werden, da HRIRs alle unter gleichen Bedingungen entstanden, dennoch wurde die Lautheit noch einmal subjektiv, informell überprüft.

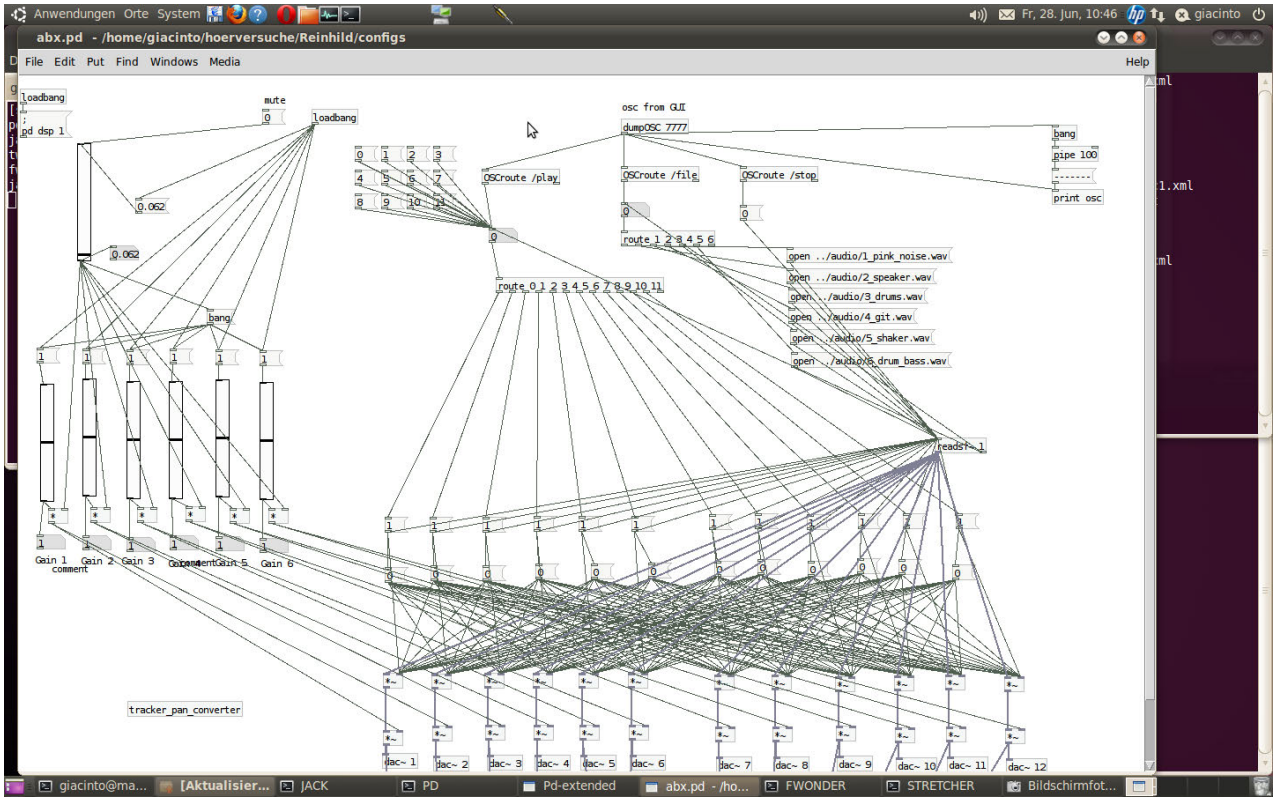


Abbildung 4.9: Signalwege in pure data pd
Einzelne Ports der zwölf fWonder-Konfigurationen konnten nicht nur über OSC, sondern zwecks vorherigen Testens auch direkt aus pd angesprochen werden, links unten findet sich der Aufruf des Skripts tracker_pan_converter, weitere Funktionen sind die verstellbare Lautstärke (da das Experiment bei individuell angenehmer Lautheit durchgeführt wurde, mittig links), die fortlaufende Ausgabe der OSC-Befehle (rechts oben)

4.2.6 Ablauf und Instruktion des Probanden

Nach Beantwortung des Fragebogens und der Tonaudiometrie setzte der Proband den extraauralen Kopfhörer auf, an dem der Head-Tracker (oben, Abb. 4.10) befestigt wurde, und wurde in einer Übungsphase instruiert. Es handelte sich bei dem Testdurchlauf um zwei auf fünf Trials gekürzte Testeinheiten für Sprache und Rauschen bei Quelle 4. Diese Quelle wurde gewählt, da bei ihr Unterschiede gut gehört werden und Probanden schnell das Prinzip des Verfahrens erkennen konnten. Es wurde deutlich darauf hingewiesen, dass das „Probieren“ verschiedener Kopfpositionen im Bereich von ca. ±80° bei der Zuordnung des Signals X zu A oder B hilft. Um ein Bewegen zu erleichtern saß der Proband auf einem Drehstuhl und bediente die Oberfläche (Abb. 4.8) auf einem Tablet-PC. Die Lautheit wurde nach Belieben des Probanden von der Versuchsleitung eingestellt[15]. Pausen konnten jeder Zeit gemacht werden, um die Konzentration des Probanden zu erhalten; ohne Pause brauchten Probanden 60 bis 90 min um das gesamte Hörexperiment abzuschließen.

Jeder Proband erhielt eine zufällige Probandennummer, der eine randomisierte Reihenfolge der Testeinheiten zugewiesen wurde. Die drei Testeinheiten mit Sprache und Rauschen folgten dabei aber stets aufeinander.

Abbildung 4.10: Proband auf Drehstuhl mit Tablet und BK211 auf dem Kopfhörer Teil des Headtrackers (grau)

[15] Versuchsumgebung war ein ruhiger Raum im Hauptgebäude der TU-Berlin, Schallpegelmessung ergaben $L_{eq,A}$=38 dB

4.2.7 Erhobene Daten

Abbildung 4.11 zeigt Ergebnisse aller Probanden je Testeinheit. Für das Testpaar rosa Rauschen und Quelle 6 ordneten alle Probanden das Signal X in jedem der 23 Trials richtig zu. Insgesamt liegt die Verteilung für Rauschen qualitativ bei besseren Werten, während sich eine breitere Streuung für Sprache ergibt. Betrachtet man die Quellpositionen, scheint sich hier bereits zu bestätigen, dass die Zuordnung der HRTF-Daten für Bewegung a) und b) für Quelle 3 am schwersten fiel.

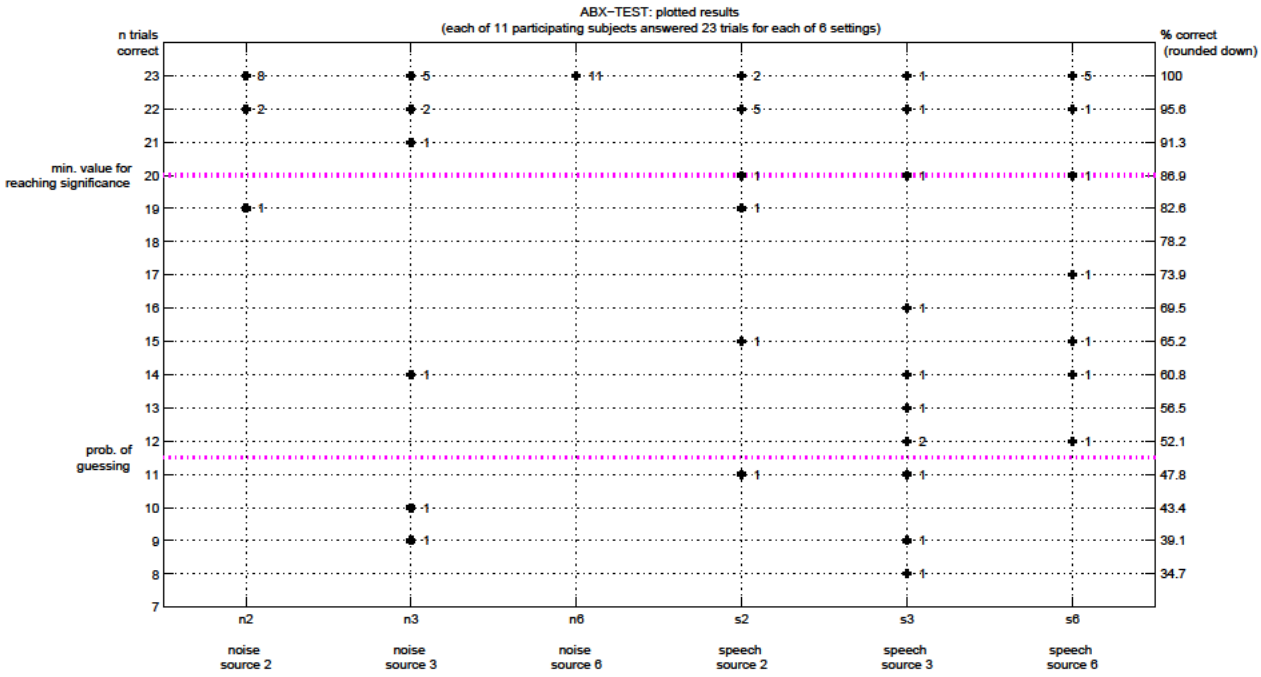


Abbildung 4.11: Rohdaten aller Probanden
Eine Zahl neben '+' gibt die Anzahl der Probanden, die mit gleicher Genauigkeit für diese Testeinheit antworteten

Probanden gaben an (siehe auch Tabelle C letzte Frage im Erhebungsfragebogen) bei Rauschen vornehmlich auf spektrale Verfärbungen gehört zu haben. Bei Sprache fiel es einem Teil der Teilnehmenden schwer, Merkmale direkt zu benennen; es überwog allerdings der Eindruck Lokalisationsunterschiede zwischen Quellpositionen der Signale A und B wahrgenommen zu haben.

4.2.8 Statistische Methode und Auswertung

4.2.8.1 Hörbarkeitshypothese - individuell betrachtet

Mit dem Ziel zwar möglichst effiziente aber auch artefaktfreie Anwendungen für akustische Simulationen zu entwickeln, legt man die Schwelle für ein willkürliches Hörbarkeitskriterium gering. Würde man in diesem Hintergrund dieses Maß festsetzen wäre die Hörbarkeit belegt, könnte man schon für einen Probanden bei einem Testpaar zeigen, dass dieser nicht zufällig richtig antwortete.

Um eine Hörbarkeit des Torsoeinflusses auf das Schallfeld am Trommelfell zu

belegen oder aber die Nullhypothese beizubehalten wurde daher individuell für jede Testeinheit die Abweichung von der Ratewahrscheinlichkeit auf Signifikanz überprüft. Dem entsprechen 66 auszuwertende Bernoulli-Tests innerhalb einer Stichprobe. Die Nullhypothese könnte also 66 mal zufällig fälschlicherweise verworfen werden. Bortz [Bor05, S. 129 f.], der die Theorie zum sogenannten multiplen Testen erläutert, schlägt gegen die Alphafehler-Inflation die Bonferoni-Korrektur als konservatives Verfahren vor. Das Signifikanzniveau verringert sich nach Bonferoni um den Faktor $\frac{1}{AnzahlTests}$. Das neue Niveau beträgt demnach $\alpha' = \frac{\alpha}{66} = 0.07576\%$[16] mit $\alpha = 5\%$. Dadurch verschiebt sich der Wert des Signifikanzkriteriums von 16 auf 20 richtige Entscheidungen aus 23. Die Teststärke reduziert sich durch die Korrektur der Alpha-Kumulierung enorm. Allerdings mit dem Argument bereits im Vorfeld große Effekte physikalisch und in einem perzeptiven Vortest ermittelt zu haben, kann ein großer β-Fehler in Kauf genommen werden. Behält man die Teststärke von 80% bei, können bei gleicher Stichprobengröße von 23 Trials je Proband und Test-Sektion Effektgrößen von 90% (23*0.9=20.7) und größer gesichert werden (Berechnungen mit GPower oder auch Burstein [Bur88, Abs. 4]).

Führt man die Bernoulli-Tests (siehe auch den Matlabcode zur Analyse `abx_analyse` Diiii) über den Zusammenhang der kumulierten Wahrscheinlichkeit (nach Beschreibungen in [Lev86], Glg. 4.4 mit x als jeweilige Anzahl korrekt gegebener Antworten) durch, erhält man je Testeinheit die Angaben für wieviele Probanden die Alternativhypothese, also die Hörbarkeit des Torsoeinflusses auf das Schallfeld am Trommelfell angenommen werden darf. Es ist immer dann der Fall wenn $\alpha_{Fehler} \leq \alpha$'. Die horizontale Linie in Abbildung 4.6 bei 20, dem kritischen Wert, lässt leicht erkennen wieviele Probanden auf und oberhalb dieser Linie liegen und damit den Bernoulli-Test „bestanden“ haben. In Tabelle 4.1 ist dies ebenso in der Summenzeile für jedes Testpaar ablesbar.

$$\alpha_{Fehler} = 1 - \sum_{i=0}^{x-1} \binom{n}{i} p_1^i (1-p_1)^{n-i} \tag{4.4}$$

[16] Durch die diskrete Binomialverteilung ist das nächst kleinere α' zum gewünschten Signifikanzniveau zu wählen, das mit dem kritischen Wert hier 20 richtige aus 23 Antworten korrespondiert.

Tabelle 4.1: Ergebnisse des Bernoulli-Tests aufgeschlüsselt nach Probanden und Testpaaren

	Rauschen			Sprache		
	Quelle2	Quelle3	Quelle6	Quelle2	Quelle3	Quelle6
VP 1	1	1	1	1	0	1
VP 2	1	1	1	1	0	1
VP 3	1	0	1	1	0	1
VP 4	1	1	1	1	0	1
VP 5	1	0	1	0	0	0
VP 6	1	1	1	1	1	1
VP 7	1	1	1	0	0	0
VP 8	1	0	1	0	0	0
VP 9	1	1	1	1	1	1
VP 10	1	1	1	1	0	1
VP 11	0	1	1	1	1	0
Summe	*10*	*8*	*11*	*8*	*3*	*7*

1 H_1 annehmen
0 H_0 beibehalten

Wie erwartet kann trotz der Bonferoni-Korrektur oft die Nullhypothese zugunsten der Alternativhypothese verworfen werden. Ein besonders kritisches Hörbarkeitskriterium ist damit für jede der untersuchten Quellpositionen bei Rauschen und auch Sprache übererfüllt, wenn es auch „nur“ drei Versuchspersonen waren, die für die scheinbar schwierigste aller Testbedingung Quelle 3 kombiniert mit Sprache nachweislich nicht geraten haben.

4.2.8.2 Hörbarkeitshypothese - über die Summe aller Ergebnisse betrachtet

Betrachtet man Messergebnisse nicht individuell um die Teststärke zu erhalten, sondern addiert Gesamtanzahl aller und richtiger Antworten je Testpaar zunächst auf, ändert sich an der Rechenweise nichts. Es würde lediglich für die Bonferoni-Korrektur der Faktor $\frac{1}{6}$ für sechs durchgeführte Signifikanzprüfungen gelten und die Stichprobenanzahl nicht 23 sondern 253 betragen. Damit können bei einer Teststärke von 80% Effekte größer als 0.625 bzw. 62.5% gesichert werden. Der kritische Wert zum Erreichen des Signifikanzniveaus von α'=0.08334% liegt dabei bei 152 Richtigen aus 253. Es kann bei dieser Methode ebenfalls für jedes Testpaar die Hörbarkeit bestätigt werden (siehe 4.2). Prozentangaben für richtige Antworten der gesamten Stichprobe sind ebenfalls aufgeführt. Diese Angabe beträgt für die schwerste Testbedingung (Sprache/Quelle3) immerhin 63.2 % (Werte sind abgerundet).

Tabelle 4.2: Ergebnisse des Bernoulli-Tests durchgeführt für alle im Experiment gegebenen Antworten je Testpaar

	Rauschen			Sprache		
	Quelle2	Quelle3	Quelle6	Quelle2	Quelle3	Quelle6
VP 1 bis 11	1	1	1	1	1	1
Richtige Antworten aus 253	247	213	253	221	160	215
Richtige Antworten %	97.6	84.1	100	87.3	63.2	84.9

1 H_1 annehmen
0 H_0 beibehalten

Die Hörbarkeitshypothese über die Summe aller Ergebnisse zu betrachten ist damit statistisch sauberer und daher vorzuziehen.

4.2.8.3 Ablehnung einer Gleichverteilungsannahme für Sprache und Rauschen

Es wäre möglich anhand der gemessenen Daten qualitativ abzulesen, dass wie im Vorfeld angenommen die Zuordnung der Signale zu X für Rauschen leichter fiel als für Sprache. Diese Hypothese soll quantitativ mit Hilfe eines χ^2-Tests mit einer Variable [SR13, 542 ff.] geprüft werden. Dieser ist üblich für den Vergleich von Häufigkeiten.

Es gilt die Gleichverteilungsannahme als Nullhypothese, d.h. die Anzahl richtiger Antworten für Sprache weicht nicht wesentlich von der Anzahl richtiger Angaben bei Rauschen ab. Da k=2 Merkmale miteinander verglichen werden ist die χ^2-Verteilung bei einem Freiheitsgrad df=(k-1)=1 heranzuziehen. Innerhalb der Stichprobe liegen $f_{b,1}$=713 bei Rauschen und $f_{b,2}$=596 richtige Antworten bei Sprache vor. Der gemeinsame Durchschnitt bzw. die erwartete Häufigkeit $f_{e,i}$ beträgt $\frac{713+596}{2}$.

Die Formel zur Berechnung des χ^2-Wertes lautet:

$$\chi^2 = \sum_{i=1}^{k} \frac{(f_{b,i} - f_{e,i})^2}{f_{e,i}} = 10.4576 \tag{4.5}$$

Ähnlich wie für die Binomialverteilung wird auch hier die kumulierte Wahrscheinlichkeit über dem Erwartungswert bis zum errechneten χ^2-Wert gebildet und von der Gesamtwahrscheinlichkeit abgezogen. Gemessen an einem α von 5%[17] kann behauptet werden, dass die Ergebnisteilmenge für Rauschen statistisch gesichert nicht der gleichen Verteilung für Sprache entspringt. Hinsichtlich der gemachten qualitativen Betrachtung in vorangegangenen Abschnitten, sind die Bewegungen a) und b) für Sprachsignale schwieriger zu erkennen.

4.2.8.4 Ablehnung einer Gleichverteilungsannahme für Quellpositionen

Bereits in der Einleitung wird behauptet, dass für den Unterschied der Torsoposition in Bewegungen a) und b) kritische Quellpositionen, mit gut hörbaren, und unkritische

[17] Innerhalb der Stichprobe wird eine neue Hypothese überprüft ohne multiples Testen, daher keine α-Fehler-Inflation und keine Korrektur.

Quellpositionen, mit weniger gut hörbaren Unterschieden, existieren. Auch hier ergaben bereits die qualitativen Betrachtungen der Messdaten des ABX-Tests und sogar vorher die physikalische Evaluation, dass vor allem Quelle 3 oberhalb des Kopfes kleine, schwer zu hörende Unterschiede hinsichtlich verschiedener Kopf-zu-Torso-Orientierungen mit sich bringt. Mit dem vorgestellten χ^2-Test einer Variable kann dies statistisch gesichert werden. Es handelt sich in diesem Fall um k=3 Merkmalsausprägungen und damit um einen Freiheitsgrad df von 2. Es werden zur Überprüfung dieser Hypothese insgesamt vier Tests durchgeführt mit α'=$\frac{5}{4}$%. In einem ersten Schritt wird geprüft, ob die Gleichverteilungsannahme gilt oder aber zurückgewiesen werden kann und in drei Weiteren ob die Anzahl richtiger Antworten für eine Quelle mit dem Durchschnitt korrekter Antworten der jeweils Anderen einer Verteilung entstammt.

Es zeigte sich, dass die Quellpositionen tatsächlich unterschiedliche Detektierbarkeit verursachen und zudem dass es hier keinen statistisch nachzuweisenden Unterschied zwischen den Positionen der Quellen 2 und 6 gibt, aber ein signifikanter Unterschied dieser beiden zur Quelle 3 existiert.

4.3 Zusammenfassung der Evaluation des Torsoeinflusses

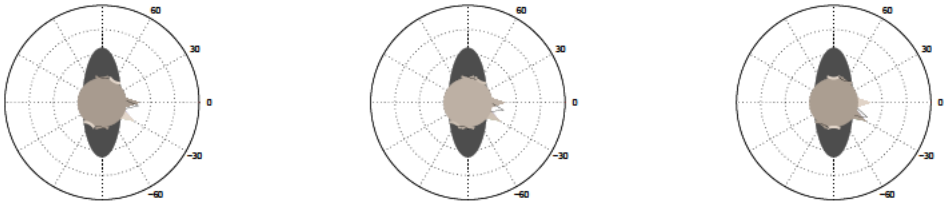
Die Vermutung, dass der Torso in einer natürlichen Kopfbewegung ähnlichen Einfluss zeigt, wie in der Literatur beschrieben für statische HATO bei bewegter Quelle (siehe Abs. 2.3), wurde für die Kopfbewegung in der Horizontalebene untersucht und bestätigt. Kammfilter des Torsos waren in betrachteten Spektren und Differenzspektren deutlich erkennbar - bereits ab Frequenzen um 500 Hz. Kritische und unkritische Quellen zeigten sich im Sinne der Studie nach Minnaar et al. [MPC05]. So sind die schwächsten Veränderungen des Schallfelds durch den Torso für die 90° elevierte Quellposition (oberhalb des Kopfes) zu ermitteln. Stärkste Unterschiede traten für die Quelle 6 unterhalb der Horizontalen auf, die damit innerhalb des angelegten Datensatzes die kritischste Quellposition darstellt. Untersuchte Abweichungen der ILD und ITD hielten sich sehr gering und sollten an der Schwelle oder aber im unterschwelligen Bereich der Wahrnehmbarkeit liegen, sodass Ursache einer etwaigen Hörbarkeit eher spektralen Verfärbungen zuzuweisen wäre. Das erstellte Einzahlmaß gab, wie auch bei Minnaar et al. [MPC05], bereits vor dem eigentlichen Hörexperiment einen Hinweis auf die Hörbarkeit, die sich im durchgeführten ABX-Test mit elf normalhörenden Probanden für alle getesteten Quellen bei Rauschen und bei Sprache mit deutlichen Effekten als gegeben herausstellte. Die Erkennungsrate im ABX-Test für Sprache war dabei jedoch signifikant kleiner als für die Synthese mit einem rosa Rauschen. Ebenso zeigte sich Quelle 3 hinsichtlich einer Hörbarkeit auch perzeptiv unter drei verschiedenen Quellen als unkritischste.

Kurz, Ergebnisse sprechen deutlich für die Berücksichtigung der Kopf-über-Torso-Orientierung in zu auralisierenden HRTF-Daten und lehnen die Vereinfachung HRTFs der natürlichen Kopfbewegung mit HRTFs statischer HATO und korrespondierender Quellbewegung zu ersetzen ab.

Kapitel 5

HRTF Interpolation

Da sich in der physikalischen und perzeptiven Evaluation die Kopf-über-Torso-Orientierung als hörbarer Einfluss bei der Auralisation von HRTF-Daten erwies, ergibt sich die Notwendigkeit bei möglichst getreuer Reproduktion einer natürlichen akustischen Umgebung mit Kopfbewegung diesen Effekt zu berücksichtigen. Dadurch erhöht sich der Messaufwand von HRTFs für die dynamische Binauralsynthese. Eine Möglichkeit diesen jedoch zu minimieren ist die Interpolation zwischen benachbarten HATOs. Wie an manchen Stellen dieser Arbeit bereits vorgegriffen wurde, ist es prinzipiell über zwei Wege möglich die HRTF einer bestimmten HATO bei gegebener zu-Quell-Ausrichtung aus zwei Stützstellen zu interpolieren. Abbildungsgruppen 5.1 und 5.2 zeigen schematisch wie drei HATOs (je Bild 1, schwarz Umrissen dargestellt) bei einer Interpolationsauflösung von 30° aus zwei Positionen gleicher Kopf- bzw. gleicher Torsoausrichtung errechnet werden können. Farblich heller sind dabei entfernter liegende Stützstellen, was auf das noch zu klärende Verfahren der Distanzwichtung hindeutet. In diesem Kapitel werden Interpolationsalgorithmen und -ergebnisse für beide Ansätze, angewendet auf die Kopf-über-Torso-Bewegung zwischen ±82° bei 0.5° Auflösung in der Horizontalebene für eine statische, neutrale Torso-Ausrichtung für alle sechs Quellen, besprochen und anschließend physikalisch evaluiert. Des Weiteren konnte die „Kopf-Interpolation“ auf Daten der Kopfbewegung in Median- und Frontalebene angewendet werden. Ergebnisse dessen werde in diesem Kapitel angeschnitten.

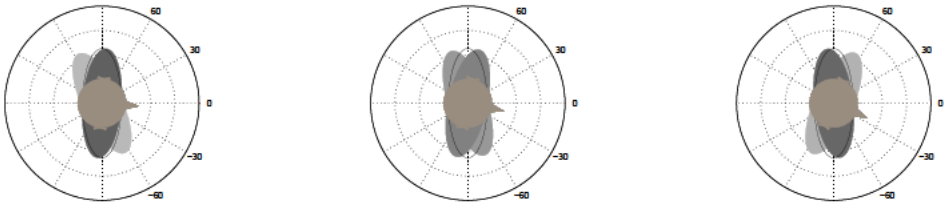


(a) interpoliert wird HATO von ca. -5°

(b) von ca. -12°

(c) von ca. -25°

Abbildung 5.1: Interpolation „via Kopf“ bei Interpolationsintervall von 30° in jeder Teilabbildung haben Stützstellen und die daraus Interpolierte die selbe neutrale Torso-Ausrichtung bei unterschiedlicher Kopfposition; in allen Unterabbildungen handelt es sich um die selben Stützstellen



(a) interpoliert wird HATO von ca. -5°

(b) von ca. -12°

(c) von ca. -25°

Abbildung 5.2: Interpolation „via Torso“ bei Interpolationsintervall von 30° in jeder Teilabbildung haben Stützstellen und die daraus Interpolierte die selbe Kopf-Ausrichtung bei unterschiedlicher Torsoposition; in allen Unterabbildungen handelt es sich um verschiedene Stützstellen

5.1 Stützstellen

HRTF gleicher Torso- und unterschiedlicher Kopfpositionen für die Interpolation zu nutzen, lag intuitiv nahe, da es sich bei den zu extrahierenden Stützstellen bereits um äquidistante, diskrete Punkte der gewünschten Bewegung handelte - der fein aufgelösten Referenzbewegung wurden lediglich Winkel zugeordnete HRTF-Daten entnommen. In Abschnitt 3.1.2 wurde bereits beispielhaft erläutert, welche Stützstellen gewählt werden müssen um die neutrale Torsohaltung von 0° bei einem Kopfazimuth von 7° im Intervall 30° aus dem Torso zu interpolieren[18]. Abbildung 5.2 enthält drei weitere Beispiele. Für die perzeptiv kontinuierliche Auralisation wird in dieser Arbeit begründet von einer Winkelauflösung von 0.5° ausgegangen. Damit sind für die

[18] 1. Stützstelle: Torso 7°/Kopf-Az. 7°; 2. Stützstelle: Torso -23°/Kopf-Az. 7°

gesamte Kopfdrehung in der Horizontalen von ±82° für 329 HRTFs diese Stützstellen zu ermitteln.
Das Skript `hrtf_interpolation.m` (Anhang Diiia) ruft die Funktion `catch2interpol` (Anhang Diiib) auf, die wiederum für sämtliche Interpolationsintervalle benötigte Daten extrahiert und speichert. Darunter ist stets der Winkel 0° ob als Azimuth in der Horizontalebene oder als Elevation in den anderen Ebenen. Vielfache des Interpolationsintervalls liegen demnach symmetrisch um diesen Winkel. In der Horizontalen ist es möglich gewesen, über ein Maximum von 82° hinaus zu messen, daher kann beispielsweise für das Interpolationsintervall von 30° zwischen 90 und 60° interpoliert werden. Stützstellen befinden sich für dieses Beispiel bei 90, 60, 30, 0, -60, -30, -60 und -90°. In Frontal- und Medianebene liegen keine HRTFs über die Elevation von 45° hinaus vor. Bei dem Interpolationsintervall von 30° wurde daher nur der Bewegungsbereich zwischen ±30° betrachtet. Es handelt sich jeweils um eine Einschränkung, wenn 45 kein ganzzahliges Vielfache des zu überbrückenden Winkels ist.

Tabelle 5.1 führt alle gewählten Interpolationsintervalle von Kopf-über-Torso-Orientierungen auf. Die Auswahl ist auf eine spätere perzeptive Evaluation des Interpolationsalgorithmuses bedacht und deckt einen weiten Bereich ab, in dem die Schwelle an der Hörbarkeit von Artefakten vermutet wird.

Tabelle 5.1: HATO-Interpolationsintervalle aller Ebenen in °

1	2	3	4	5
6	7	8	9	10
12	14	16	18	20
22	24	26	28	30
35	40	45		

5.2 Algorithmen

In der Literatur zu findende sphärische Interpolationsverfahren können auf den Datensatz, im Kapitel 3 vorgestellt, nicht angewendet werden. Bei Hartung et al. [HBS99] beschriebene Methoden kommen daher nicht ohne Weiteres in Betracht und werden in dieser Arbeit von der sphärischen Kugeloberfläche auf die Zweidimensionalität übertragen. Die Entscheidung fiel auch hinsichtlich der Praktikabilität auf eine lineare Funktion und im Weiteren auf die häufig verwendeten kubischen Splines. Für weitere Vergleiche sind Daten im Verfahren „nearest neighbour“ interpoliert worden. Eigentliche Referenz stellt die in einer feinen Auflösung gemessenen HRTFs dynamischer HATO dar. In Nähe zu Hartung wurden Algorithmen in selbiger Systematik aus Kombinationen von Interpolante und den Domänen der Zeit-Frequenz-Transformation entwickelt.

Verwendete Algorithmen setzen sich aus Bausteinen verschiedener Kategorien zusammen:

- **Domäne:** Zeit- oder Frequenzbereich
- **Interpolante:** linear, spline, nearest neighbour
- **Interpolationsvariante:** „via Torso“, „via Kopf“

- **Phase:** minimal- bzw. originalphasige HRTFs
- **Peakdetektion:** Onset-Detektion oder Kreuzkorrelation (nur für die Interpolation originalphasiger HRIRs im Zeitbereich)

Ergänzende Eigenschaften zur vollständigen Charakterisierung einzelner Datenreihen

- **Interpolationsintervall** (Tab. 5.1)
- **Ebene** (Hor., Med., Front.)
- **Quellposition** (siehe Abs. 3.1.3.5)

Teile dieser Kategorien sind nicht beliebig frei zu verknüpfen z.B. Splines sind für die „Torso-Interpolation“, wie noch erläutert wird, nicht geeignet. Tabelle 5.2 führt alle möglichen Kombinationen auf. Jede Zeile entspricht einem Algorithmus, der je Interpolationsintervall ausgeführt wurde. Theoretische Hintergründe zu einzelnen Kategorien werden direkt im Zusammenhang der verwendeten, bzw. möglichen Zusammenstellungen mit zugehörigem Quellcode in Matlab erläutert. Ausführungen folgen den Abschnitten 2) bis 8) in `hrtf_interpolation` (Anhang Diiia, beginnend nach 0) Konfigurationen und 1) Einlesen der Stützstellen).

Tabelle 5.2: Übersicht Interpolationsalgorithmen

Phase	**Interpolationsansatz (Ebenen)**	**Domäne**	**Interpolante**	**Peakdetektion**
minimal	via Kopf (Hor., Med., Front)	/	nearest	/
		zeitl.	linear	/
			spline	/
		spektral	linear	/
			spline	/
	via Torso (Hor.)	zeitl.	linear	/
		spektral	linear	/
original	via Kopf (Hor., Med., Front)	/	nearest	/
		zeitl.	linear	Onset-Detekt.
				Kreuzkorr.
			spline	Onset-Detekt.
				Kreuzkorr.
		spektral	linear	/
			spline	/
	via Torso (Hor.)	zeitl.	linear	Onset-Detekt.
				Kreuzkorr.
		spektral	linear	/

5.2.1 Nearest neighbour „via Kopf“

Die Methode des „nächsten Nachbarn“ ist im Grunde keine Interpolationsfunktion, sondern lediglich die Verwendung einer HRTF über das halbe Interpolationsintervall, das sie sich mit der benachbarten HRTF teilt. Dieses Verfahren lässt sich auf die Stützstellen-HRTFs der „via-Kopf“-Interpolation anwenden. Ob Zeit- oder

Frequenzbereich ist nicht relevant oder als identisch anzusehen. Da nichts berechnet, sondern lediglich aneinandergereiht wird, gibt es außer der Wahl der jeweiligen Daten keine Verfahrensabweichungen für minimal- oder originalphasige HRTFs. Der Aufruf `interpolate_head_time` (Anhang Diiie) führt zur Ausführung der Funktion `in_time` (Anhang Diiig), in der letztlich gleichzeitig alle Winkel der Stützstellen x und zugehörige HRIRs, das zu interpolierende Gitter xi und das Argument 'nearest' an die Matlab interne Funktion `interp1` für die linke, dann rechte Seite des Signalpaares übergeben wurden. Auch mitgeführte ITDs wurden phasenunabhängig auf die Weise interpoliert.

Es konnten alle Ebenen berücksichtigt werden, wie es im Folgenden für alle Versionen der Interpolation „via Kopf“ gilt.

5.2.2 Linear „via Kopf“ im Zeitbereich

Interpoliert man zwischen zwei Punkten der Werte y_1 und y_2 die Werte y_i linear, so werden diese gemäß ihres Abstandes zu y_1 und y_2 gewichtet. Die Abbildungen in Gruppen 5.1 und 5.2 deuten die Gewichtung farblich an. Zwei HRTFs werden gleich gewichtet und mit farblich gleicher Helligkeit kodiert, liegt die zu interpolierende HRTF exakt zwischen diesen beiden. Ist die Entfernung zu einer HRTF kleiner, wird diese stärker gewichtet und ist daher dunkler dargestellt, hingegen die andere heller.

Für Daten aller Ebenen jeder Phase wurde wie im Abschnitt zu „nearest neighbour“ `interpolate_head_time` mit der Interpolationsart als String 'linear' aufgerufen. Für minimal-HRTFs wird das Skript einmal durchlaufen. `in_time` interpoliert linear und speichert je 0.5°-Winkel zwei HRIRs mit ITD. Im Falle der Originalphase sind allerdings die ITDs noch in den Zeitsignalen enthalten. Es ist bei der sampleweisen Interpolation benachbarter HRIRs zu berücksichtigen, dass Peaks des eintreffenden Direktschalls durch die sich über dem Winkel der Bewegung ändernde ITDs nicht aufeinander fallen. Um diese Peaks nicht zu glätten, sondern um für alle HRIRs der Bewegung einen ähnlichen zeitlichen Verlauf zu gewährleisten wurden zum einen über die in Abschnitt 3.2.3 erklärte Onset-Detektion zunächst Peaks zeitlich ermittelt, dann vor der Interpolation aufeinander geschoben, um danach anhand interpolierter ITDs jeder HRIR ihre eigentliche Verzögerung des Direktschalls zuzuordnen. In einem zweiten Verfahren wurde das Aufeinanderlegen der Peaks mittels Kreuzkorrelation erreicht. Für Onset-Detektion und Kreuzkorrelation, wie sie auch in der Studie von Hartung et al. [HBS99] verwendet wird, ließen sich ähnliche Ergebnisse erwarten. Dennoch wurden aber wegen eines nur geringen Mehraufwandes beide Verfahren mitgeführt, um sie in ihrer Genauigkeiten zu vergleichen.

5.2.3 Spline „via Kopf“ im Zeitbereich

Ein Spline bzw. eine Splinefunktion ist ein Polynomenzug, also ein endliches Polynom, durch mindestens drei Punkte. Zwei Punkte ergäben in jedem Fall einen linearen Streckenzug. Da bei der Interpolation „via Torso“ nicht gleichzeitig alle Stützstellen der gesamten herzustellenden Bewegung als Punkte vorliegen, sondern die Betrachtung je Winkel mit nur zwei HRTFs gleicher Kopf-Position erfolgt, lieferten Splines redundante Ergebnisse zur linearen Methode. Daher sind Splines innerhalb eines Algorithmuses nur mit der Interpolation „via Kopf“ zu verbinden. Oft ist mit dem Wort „Spline“, wie auch hier, bereits das kubische Spline dritten Grades gemeint. Der Vorteil

einer stückweisen Definition eines Polynoms liegt in der Vermeidung oszillierenden Verhaltens von Polynomen bei wachsender Stützstellenanzahl (Plato [Pla10, Kap. 2]). Quadratische Splines haben laut Plato in der Praxis eine untergeordnete Rolle.

Da es sich auch um einen Algorithmus für den Zeitbereich mit den selben Stützstellen wie im vorangegangenem Abschnitt handelt, unterscheidet sich der Verlauf im Quellcode lediglich in dem Argument, das die Interpolante definiert - 'spline' statt 'linear'.

5.2.4 Linear „via Kopf“ im Frequenzbereich

Im Zeitbereich wurden Werte einzelner Samples bzw. Zeitpunkte über dem Winkel interpoliert. Im Frequenzbereich sind es die einzelnen Frequenzbins benachbarter HRTFs, die über dem Winkel interpoliert werden. Dazu wurden HRIRs zu HRTFs transformiert und Beträge sowie Phase bis zum Bin der halben Samplingfrequenz gesondert interpoliert, anschließend über den Eulerschen Zusammenhang ($|H|e^{j\varphi}$) des Betrags $|H|$ und der Phase φ zusammengesetzt und über die Symmetrieeigenschaft eines reellen Signals im Frequenzbereich bis zum Frequenzbin der Abtastfrequenz (44100 Hz) vervollständigt [19]. Die Werte für das Bin um die halbe Abtastfrequenz ist auf Null gesetzt worden, um numerische Ungenauigkeiten durch die digitale Signalverarbeitung zu mindern. Ungenauigkeiten drückten sich in sehr kleinen imaginären Anteilen gebildeter HRIRs aus, obwohl theoretisch stets ein reelles Signal hätte entstehen müssen. Dieser Effekt konnte dadurch gemindert werden. Anders als bei Hartung et al. wurden die Beträge vor der Interpolation nicht logarithmiert, um so eine tatsächlich lineare Interpolation der Schalldrücke zu erzielen. Die Motivation der Autoren geht darüber hinaus nicht aus dem Artikel hervor. Die Interpolation noch einmal mit vorheriger Logarithmierung durchzuführen, wäre eine weitere Option vor einem etwaigen Hörversuch, um diesen Parameter zu evaluieren.

Für die Datenreihen der Minimalphasen wurden Phasen anhand der interpolierten Beträge neu gebildet (auf gleiche Weise, wie es im Processing 3.2.4.4 beschrieben worden ist).

5.2.5 Spline „via Kopf“ im Frequenzbereich

Das Vorgehen bei Splineinterpolation „via Kopf“ im Frequenzbereich unterscheidet sich mit dem aus vorangegangenem Abschnitt nicht. Es ist lediglich statt der linearen Funktion das Argument 'spline' an `interp1` übergeben worden bei sonst gleicher binweisen Interpolation.

5.2.6 Linear „via Torso“ im Zeitbereich

Algorithmen „via Torso“ verlangen eine Interpolation je Winkel, also ein schrittweises Vorgehen in 0.5°. Die Wahl anderer Stützstellen und das Hinzufügen einer Schleife im Quellcode (Anhang D f) über dem Winkel waren daher notwendig. Es handelte sich aber dennoch um die selbe bereits vorgestellte sampleweise Interpolation mit `in_time` (Anhang Diiig). Für die originalphasigen HRTFs wurden auch hier Peaks

[19] Für reelle Signale gilt der Zusammenhang zwischen Werten negativer Frequenzen und positiver Frequenzen über die komplexe Konjugation: H(ω)=H(-ω)*

mittels Onset-Detektion und Kreuzkorrelation verschoben.

Da schlicht keine Daten als Stützstellen für diese Herangehensweise für Median- und Frontalebene gemessen werden konnten (siehe Kap.3.1.2), ist nur die horizontale Kopfbewegung auf diese Art interpoliert worden.

5.2.7 Linear „via Torso“ im Frequenzbereich

Dieser Algorithmus ist in den Skripten `interpolate_torso_freq` (Anhang Diiid) implementiert. Wie auch im Zeitbereich wurde schrittweise und nicht die ganze Bewegung zusammen interpoliert; die Interpolation an sich ist jedoch die selbe, die bereits für den Frequenzbereich für `in_freq` beschrieben wurde.

5.3 Physikalische Evaluation der Interpolationsalgorithmen

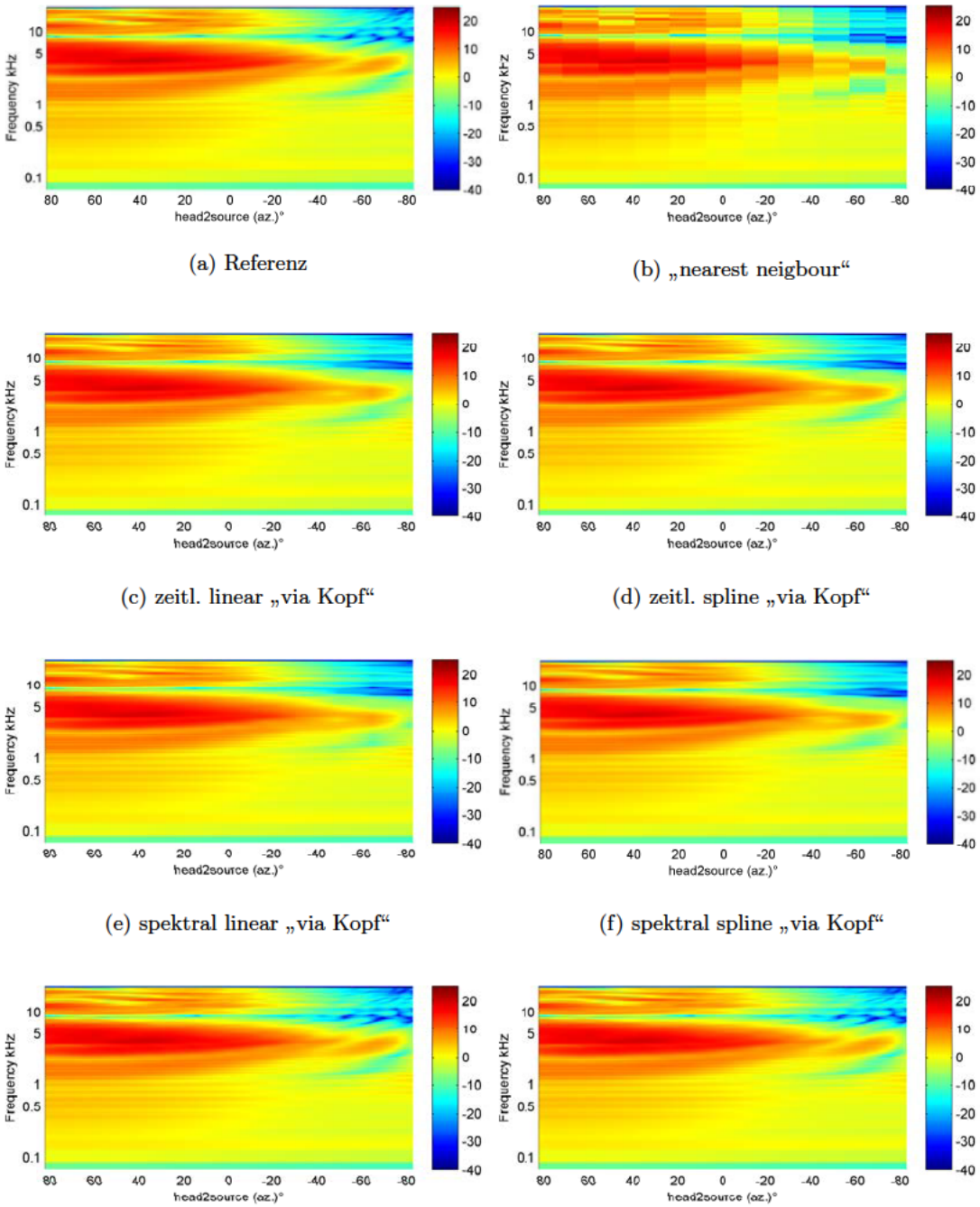
Die physikalische Evaluation der Interpolationsalgorithmen für verschiedene Interpolationsintervalle ist im Grunde sehr ähnlich zu gemachten Betrachtungen im Kapitel über den Einfluss der HATO auf das Schallfeld am Trommelfell bzw. geblockten Ohrkanal (Kap. 4). So wurde auch hier zunächst jede Datenreihe, die jeweils die interpolierte Bewegung des Kopfes bei statischem Torso enthält, mit einer Referenz verglichen. Die Referenz ist nun jedoch die selbige Bewegung, die mit einer Winkelauflösung von 0.5° gemessen wurde, statt der Bewegung bei statischer HATO. Jeder Algorithmus wurde für jedes Intervall angewendet und die meisten davon auch auf jede der drei Ebenen, sowie immer auch auf minimal- und originalphasige HRTFs, die als Stützstellen dienten. Es liegen damit eine Vielzahl von Daten vor, die mit genannter Referenz verglichen werden können. Im Weiteren werden das bereits vorgestellte perzeptiv angelehnte Fehlermaß und Abweichungen der ITDs sowie ILDs betrachtet. Um systematisch vorzugehen, werden zunächst die Ergebnisse für die minimalphasigen Datenreihen besprochen, um anschließend Unterschiede zu originalphasigen HRTFs zu klären - beides für die Bewegung in der Horizontalebene, weshalb in diesen Abschnitten der Unterschied von „via Kopf“- und „via Torso“-Interpolation zu finden ist. Zweitrangig und sehr knapp werden Ergebnisse in Median- und Frontalebene für das physikalisch-perzeptive Einzahlmaß gezeigt.

5.3.1 Minimalphasige HRTFs bei Kopfbewegung innerhalb der Horizontalebene

5.3.1.1 Spektraler Vergleich

Beispielhaft seien hier für die Bewegung in der Horizontalebene bei Quelle 1 Spektren für das Interpolationsintervall von 16° jeweils für das rechte Ohr dargestellt (Abb. 5.3). Es wird daran deutlich, wie es vorstellbar war, dass Spektren für „nearest neighbour“ mit dem Interpolationsintervall größer werdende Stufen enthalten. Hier im Beispiel ist zu sehen, dass interpolierte Spektren über dem Winkel bei „via Kopf“-Interpolation für sich betrachtet auch bei einem relativ großen Intervall von 16° glatt oder auch „smooth“ wirken und ein Fehler nicht direkt offensichtlich ist. Verglichen mit der Referenz werden die Glättungseffekte an der kaum noch vorhandenen Feinstruktur, verursacht durch die Pinna im höheren Frequenzbereich, sichtbar. Dieser Effekt nimmt mit großen Intervallen

zu. Entsprechend entwickelt sich die Differenz und das ERB-Skalen basierte Fehlermaß, wie noch zu sehen sein wird. Werden in selbiger Bildgruppe die letzten beiden Grafiken, „via Torso“, zur Referenz betrachtet, sind mit bloßem Auge die feineren Strukturen im Winkelbereich um -70° zu hohen Frequenzen hin noch erhalten und Abweichungen nur schwer zu erkennen. Ob zeitlich oder spektral interpoliert dabei bessere Resultate liefert, ist noch dahingestellt.



(a) Referenz

(b) „nearest neigbour“

(c) zeitl. linear „via Kopf“

(d) zeitl. spline „via Kopf“

(e) spektral linear „via Kopf“

(f) spektral spline „via Kopf“

(g) zeitl. linear „via Torso“

(h) spektral linear „via Torso“

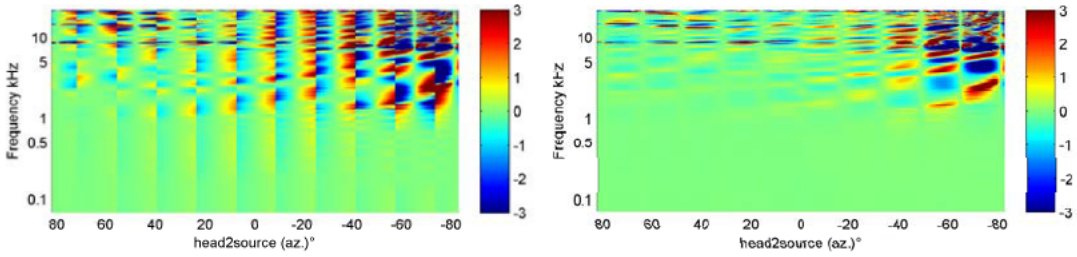
Abbildung 5.3: Bsp. Gegenüberstellung von interpolierten Daten und Referenz für minimalphasige HRTFs je rechtes Ohr
hier beispielhaft für das Interpolationsintervall 16°, Quelle 1

5.3.1.2 Differenzspektren

Wie in Abschnitt 4.1.1 wurden auch hier Differenzen gebildet, dabei wurden interpolierte Datenreihen und die jeweilige Referenz übergeben.

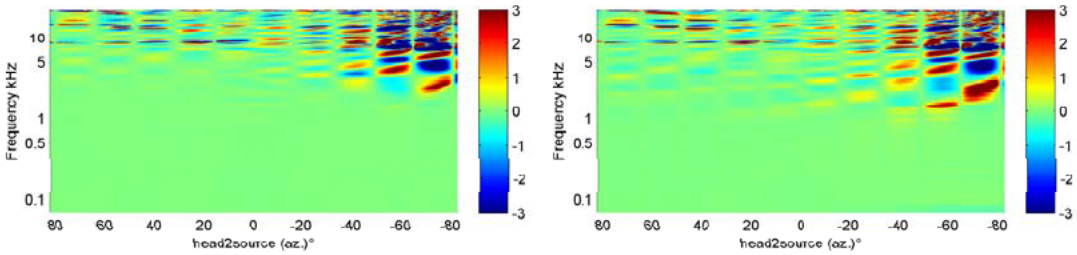
$$20 \cdot \log_{10} \frac{|HRTF_{interp.}|}{|HRTF_{ref}|} \tag{5.1}$$

Die zu Abbildungen 5.3 gehörenden Differenzplots in der Aufsicht, bei zu beachtenden Clipping bei ±3 dB, sind in der Gruppe 5.4 zu sehen. Wie im vorangegangenen Abschnitt schon vermutet wurde, sind Abweichungen für „via Torso“-Interpolation geringer. Um Stützstellen herum ist erwartungsgemäß der Fehler kleiner und zwischen ihnen am größten.



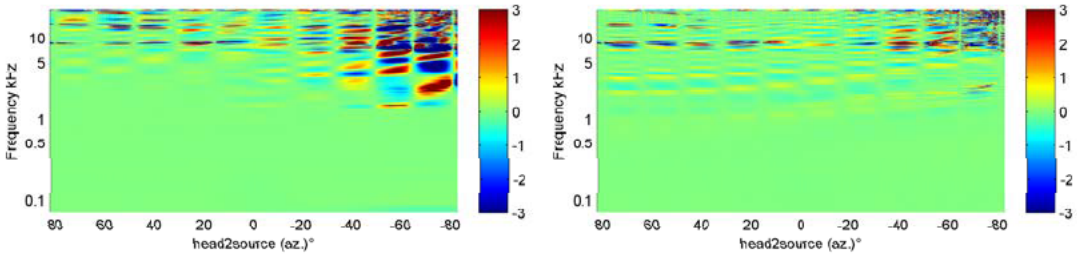
(a) „nearest neigbour“

(b) zeitl. linear „via Kopf“



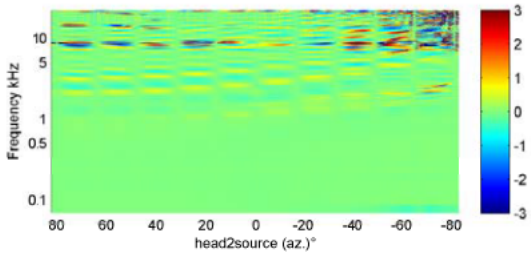
(c) zeitl. spline „via Kopf“

(d) spektral linear „via Kopf“



(e) spektral spline „via Kopf“

(f) zeitl. linear „via Torso“



(g) spektral linear „via Torso“

Abbildung 5.4: Differenzspektren aus interpolierten Daten und Referenz für minimalphasige HRTFs je rechtes Ohr
hier beispielhaft für das Interpolationsintervall 16°, Quelle 1

5.3.1.3 Physikalisch-perzeptives Einzahlmaß

In Abschnitt 4.1.2 wurde das physikalisch-perzeptive Einzahlmaß vorgestellt, das in Anwendung auf die Evaluation des Interpolationsalgorithmus so eingesetzt wurde:

$$E(f_c) = 10\log_{10}\left(\frac{\int_{-\infty}^{\infty} C(f,f_c)|HRTF_{interp.}|^2 df}{\int_{-\infty}^{\infty} C(f,f_c)|HRTF_{ref}|^2 df}\right) \tag{5.2}$$

Auch hier handelt es sich eigentlich um die Summe für n Mittenfrequenzen zwischen 70 Hz und 20 kHz. Beträge der E(f_c)-Werte rechter und linker Seite wurden je n (also je auditorischen Filter) addiert, anschließend gemittelt und über dem Winkel in Stufen aufgetragen. Da sich diese Berechnung für jedes Interpolationsintervall, sprich 23 mal, wiederholte, ist die von der Differenz gesonderte Darstellung sinnvoll, um die Entwicklung diesen Maßes mit dem Intervall zu zeigen.

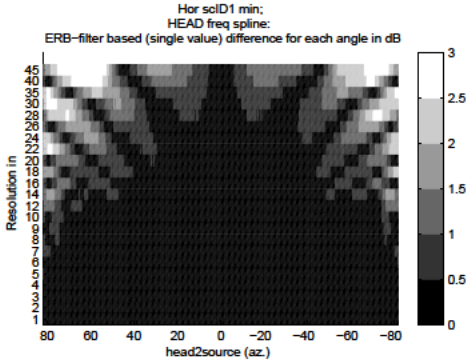


Abbildung 5.5: Bsp. ERB-Skalen basiertes Fehlermaß aller Interpolationsintervalle für spektrale Spline-Interpolation „via Kopf“, Kopfbewegung in Horizontalebene, Quelle 1

Im Beispiel (Abb. 5.5) sind Fehlerwerte über dem Winkel für kleine Intervalle im Bereich 0 bis 0.5 dB und damit in einem etwaigen ABX-Test mit dem Vergleich von interpolierten Datenreihen und Referenz eher unauffällig wenn nicht sogar unhörbar. Zwischen weit auseinanderliegenden Stützstellen werden weiße Flächen deutlich, die

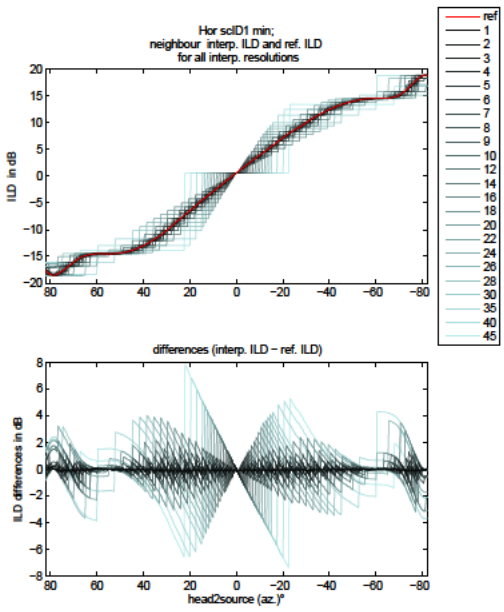
einen Fehler oberhalb von 3 dB kodieren und sicher als Artefakte in einem ABX-Test erkennbar sind.
Im Anhang sind für alle Interpolationsalgorithmen der minimalphasigen HRTFs der Horizontalebene für jede Quellposition diese Ergebnisse gezeigt (Abb.gruppen A.6 bis A.11).

Innerhalb einer Quelle betrachtet, sind Fehler für die „via Torso“-Interpolation erstaunlich gering, bei Quelle 1 übersteigt er nicht den Wert von 1 dB, selbst bei 45° Interpolationsintervall. Andere Algorithmen zeigen die abgeschwächte Fächerform der „nearest neighbour“-Methode. Die Tendenz ist für alle Quellpositionen zu beobachten, allerdings scheint es, wie schon bei der Evaluation des Torsoeinflusses, kritische und weniger kritische Quellen zu geben. Quelle 3 zeigt für alle Algorithmen durchweg Werte kleinster Stufe (bis 0.5 dB) für das physikalisch-perzeptive Fehlermaß. Betrachtet man für diese Quellposition einmal die Referenz, die innerhalb der Abbildungsgruppe A.2 unter (c) zu finden ist, sind darin im Vergleich zu anderen Quellen keine Verstimmungen mit sich änderndem Winkel erkennbar, weshalb sich selbst das größte Intervall gut interpolieren lässt. Auch zeitlich ist das der Fall. Für die Schallquelle bei 90° Elevation scheint damit selbst die Methode „nearest neighbour“, die für andere Quellpositionen fächerförmig, systematisch schlechter werdende Einzahlwerte je Winkel annimmt, ausreichend zu sein. Möglicherweise sind sogar noch weniger Stützstellen notwendig, um bei einer späteren Auralisation dieser HRTFs unterhalb der Hörschwelle von Artefakten zu bleiben.

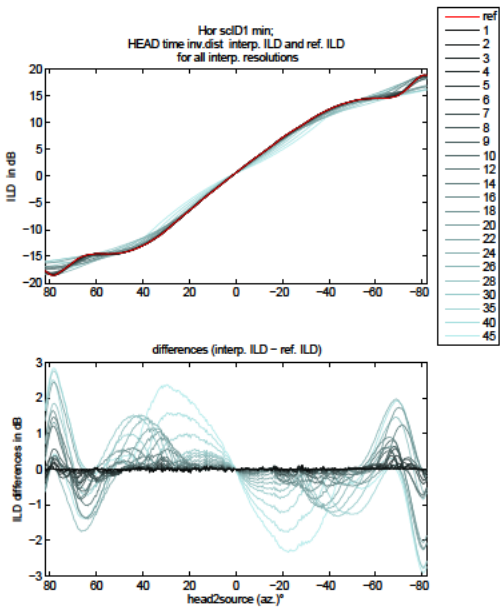
Generell scheinen Splines etwas bessere Werte als die lineare Interpolation zu liefern, während man für die Domäne, der zeitlichen oder spektralen Interpolation, qualitativ einen Unterschied zugunsten der Interpolation im Zeitbereich nur vermuten könnte.

5.3.1.4 ILD

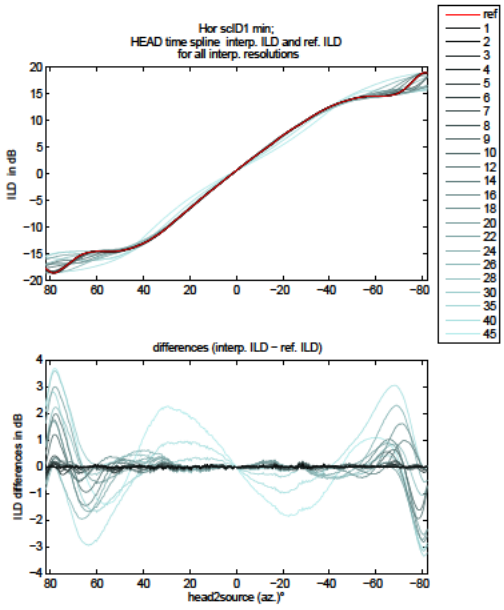
Nach Gleichung 4.3 wurde für interpolierte Daten je Winkel und auch für die Referenz die ILD gebildet. Um möglicherweise hörbare Abweichungen sichtbar zu machen, wurden Werte von einander abgezogen. Für die beispielhaft gewählte Quelle 1 sind Ergebnisse dieser Betrachtung in 5.6 zu sehen. Jeweils oben sind Einzelverläufe dargestellt, unten die Differenz. Farblich heller sind große Intervalle, die eine größere Auslenkung in der Abweichung aufzeigen. Für die Torso-Interpolation weicht kein Wert mehr als 0.3 dB von der Referenz ab. Hingegen ist die Interpolation „via Kopf“ für kleinere Intervalle bei Verwendung von Splines vermutlich gerade noch akzeptabel. Für die lineare Variante werden Unterschiede schon bei kleineren Intervallen groß.



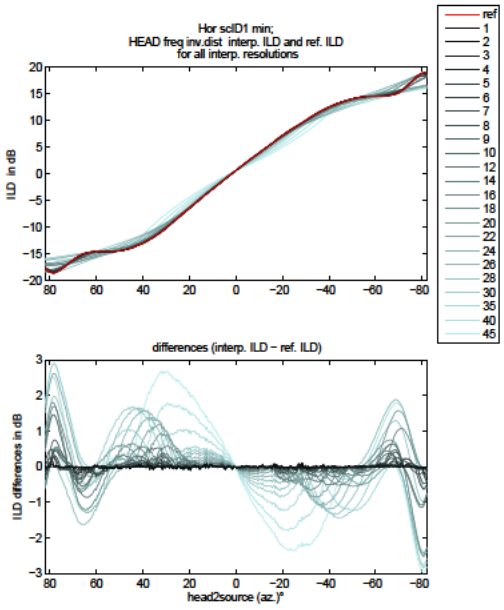
(a) „nearest neighbour“



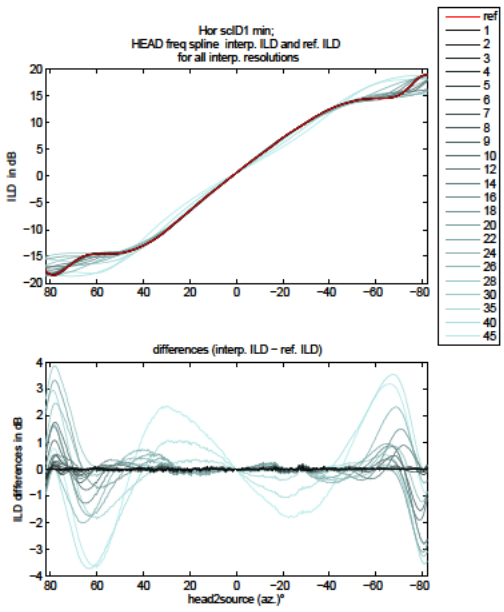
(b) zeitl. linear „via Kopf“



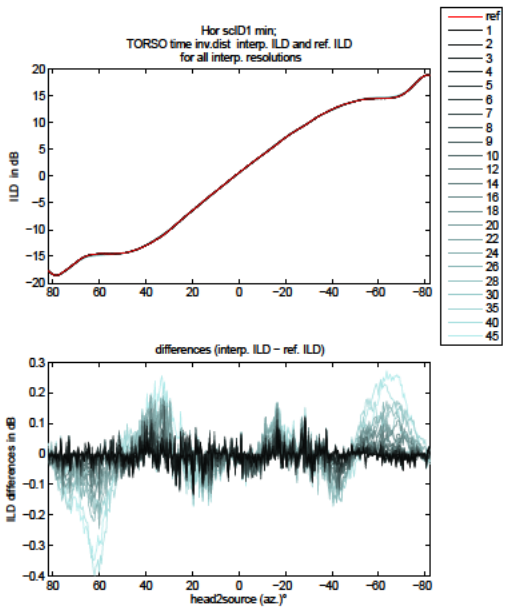
(c) zeitl. spline „via Kopf“



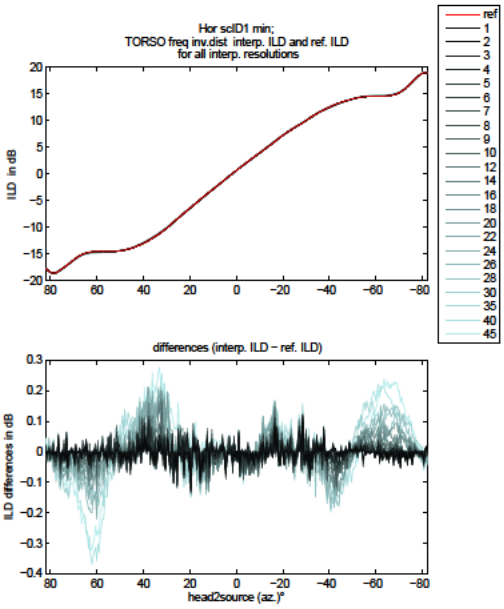
(d) spektral linear „via Kopf“



(e) spektral spline „via Kopf“



(f) zeitl. linear „via Torso“



(g) spektral linear „via Torso“

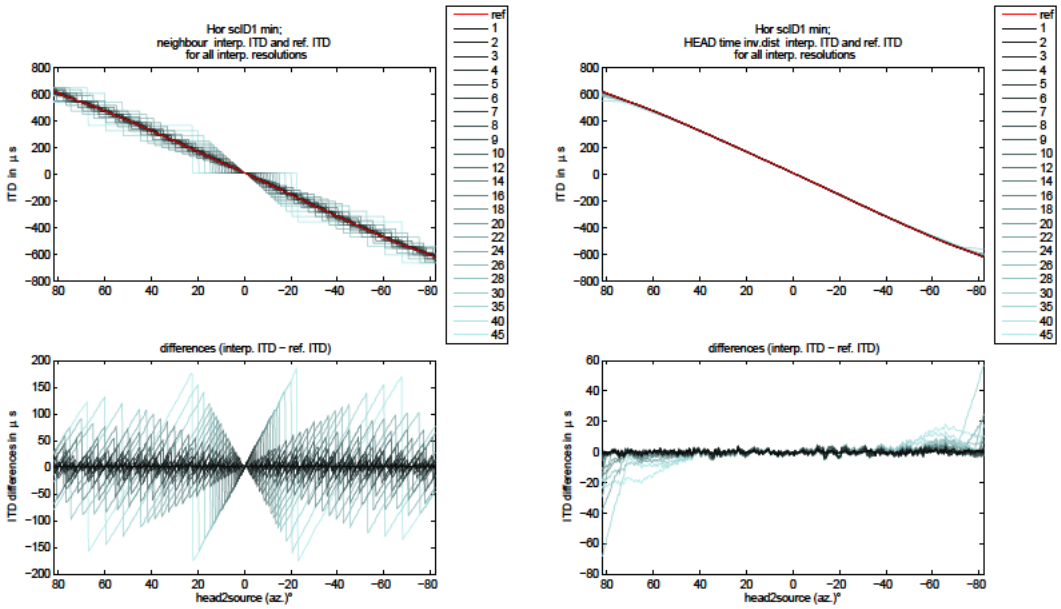
Abbildung 5.6: ILD-Abweichungen über dem Winkel je Algorithmus, Horizontalebene, Quelle 1

Abbildungen im Anhang A.14 zeigen sämtliche über dem Winkel gemittelten Abweichungsbeträge - aufgetragen über das Interpolationsintervall für jeden Algorithmus. Hinsichtlich dieser Verläufe sind ebenso mit der „via Torso“-Interpolation

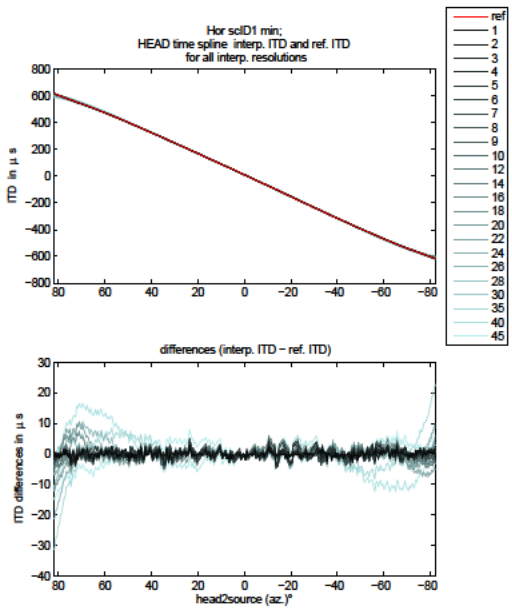
stets die besten Resultate zu gewinnen. Auch hier zeigt sich die für Artefakte eher unempfängliche Quelle 3, für die mittlere ILD-Werte für alle Algorithmen dicht beieinander liegen. Ob hörbare Artefakte jedoch durch Interpolation induziert werden, kann ein über dem Winkel gemitteltes Ergebnis schlecht vorhersagen. Auch wenn gemittelte Werte 3 dB nicht übersteigen, sind für einzelne HATOs die ILD-Abweichung zum Teil oberhalb von 5 dB. In einem weiteren ABX-Versuch würde der Hörende Artefakte winkelspezifisch wahrnehmen, weshalb die gemittelten Beträge lediglich einen Trend darlegen und für den Vergleich der Algorithmen untereinander herangezogen werden können, weshalb auf die Darstellungen 5.6, zumindest beispielhaft für eine Quelle, nicht verzichtet werden sollte.

5.3.1.5 ITD

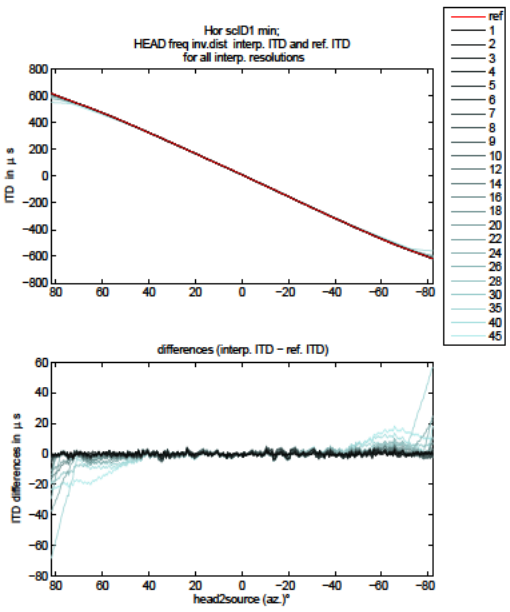
Die Darstellungsweise der über Onset-Detektierung (Abs. 3.2.4.1) ermittelten ITD-Werte ähnelt der für die ILDs. Beispielhaft finden sich für Quelle 1 in Abbildungen 5.7 Algorithmen spezifische ITD-Verläufe in μs über dem Winkel, einmal für die Referenz sowie für die interpolierten HRTFs. Unterhalb dessen sind jeweils die Differenzen aufgetragen. Es ist zu entnehmen, dass es sich auch für diesen Parameter anbietet mit HRTFs verschiedener Torso- und gleicher Kopfpositionen zu interpolieren, da Werte sogar kleiner als 5μs sind. Für andere Vorgehensweisen sind ITD-Abweichungen oft an den Rändern der Bewegung zu groß, da sich Lateralisationsänderungen, wie in Tabelle 2.1 aufgeführt, stimulusabhängig bereits zwischen 2 bis 10μs einstellen.



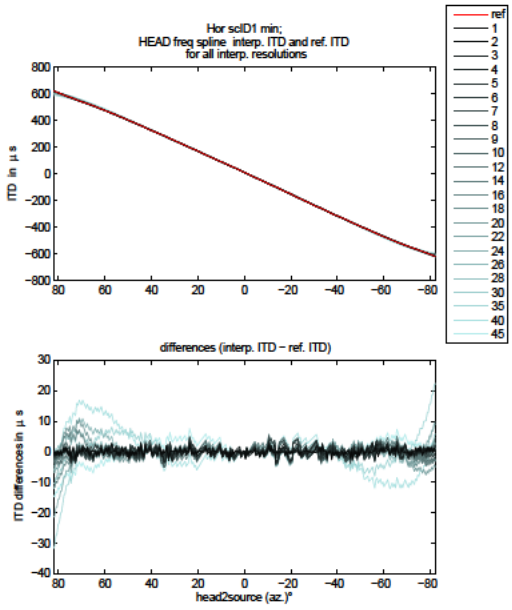
(a) „nearest neighbour“ (b) zeitl. linear „via Kopf“



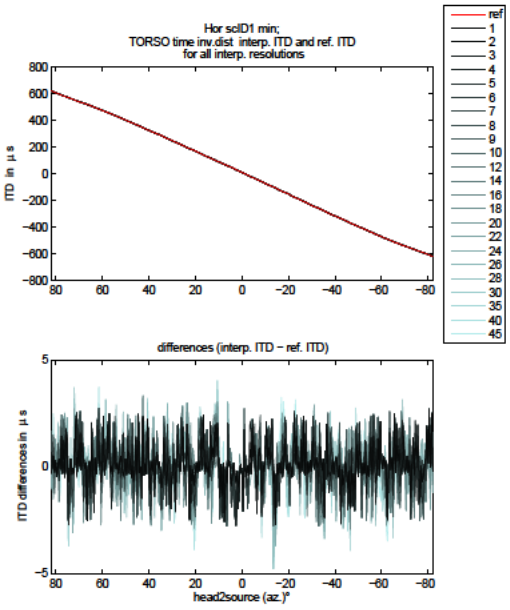
(c) zeitl. spline „via Kopf“



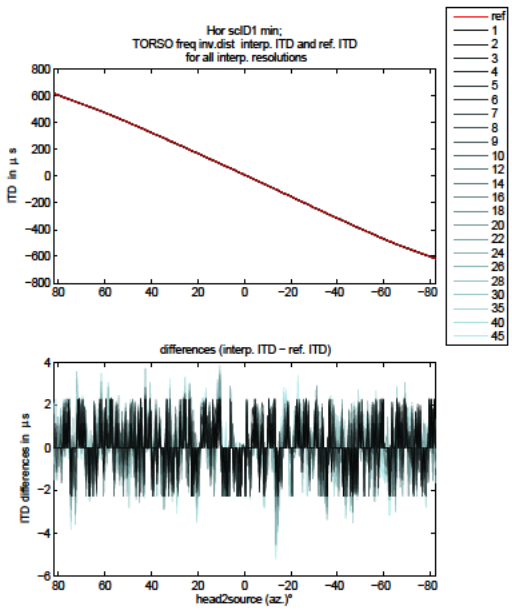
(d) spektral linear „via Kopf“



(e) spektral spline „via Kopf“



(f) zeitl. linear „via Torso“



(g) spektral linear „via Torso“

Abbildung 5.7: ITD-Abweichungen über dem Winkel je Algorithmus, Horizontalebene, Quelle 1

Über dem Winkel gemittelte und über das Interpolationsintervall aufgetragene ITD-Werte in Abbildungen A.15 halten sich außer für “nearest neighbour“ fast nur im Bereich unterhalb 10μs auf. Qualitativ scheint die Torsointerpolation auch hier besser geeignet bei kaum sichtbaren Unterschieden zwischen den Varianten im Frequenz- und Zeitbereich. Für Quelle 3 liegen Kurven einzelner Algorithmen sehr dicht.

5.3.2 Minimalphasige HRTFs bei Kopfbewegungen innerhalb der Median- und Frontalebene

Abbildungen der Gruppe A.13 im Anhang zeigen die Einzahlwerte beispielhaft für die Medianebene bei Quelle 1 und einem Kopfazimuth von 0°. Die Einschränkung des betrachteten Bewegungsbereich je Interpolationsintervall, wie in Absatz 5.1 geschildert wurde, wird daran deutlich. Der Datensatz ermöglicht darüber hinaus keinen Vergleich der Interpolationsansätze „via Kopf“ und „via Torso“ für eine Kopfbewegung in der Median- und Frontalebene. Ausführungen zu diesen beiden Ebenen sollen sich daher auf einzelne, knappe Aussagen beschränken: Die physikalische Evaluation hinsichtlich der Interpolante und hinsichtlich der Quellpositionen ergeben gleiche Tendenzen wie die Ergebnisse in der Horizontalebene. Allerdings nimmt Quelle 3 keine herausragende Sonderstellung als unkritischste Quellposition ein. Es bildet der Interpolationsfehler auch hier einen Fächer über das Interpolationsintervall, der besonders ausgeprägt ist für „nearest neighbour“ und sich abgeschwächt für die Interpolationsvarianten zeigt. Mehr Daten dazu sind auf der beiliegenden DVD-R zu finden.

In Hörexperimenten ist es denkbar Daten dieser Ebenen genauer zu untersuchen.

5.3.3 Originalphasige HRTF

Bei der Darstellung beschränken sich Abbildungen im Anhang auf die minimalphasigen Ergebnisse, die mit den originalphasigen in den Ergebnissen übereinstimmen. Lediglich ein Unterschied ist anzumerken. In der Zeit interpolierte originalphasige HRTFs zeigen einen Abfall zu hohen Frequenzen durch das zusätzliche up-und-down-Sampling der Kreuzkorrelation bzw. der Onset-Detektion für das Übereinanderlegen der Zeitpunkte des eintreffenden Direktschalls. Zum Vergleich dieser Daten wurde auch auf die originalphasige Referenz das gleiche Verfahren des up-down-Samplings angewendet. Zwischen Kreuzkorrelation und Onset-Detektion kann letztlich jedoch qualitativ kein Unterschied festgestellt werden. Auf weitere Ausführungen zu den originalphasigen HRTFs soll daher verzichtet werden. Für eine mögliche Auralisation lägen die Daten aber bereit.

5.4 Zusammenfassung der Interpolation

Betrachtet wurden maßgeblich die physikalischen Ergebnisse der interpolierten Kopfbewegung innerhalb der Horizontalebene. Bei Hartung et al. [HBS99] gefundene Herangehensweise der Kategorisierung in Zeit-Frequenz-Domäne und Interpolante (linear/inverse Distanzwichtung und kubischer Spline) wurde erweitert vor allem durch den Interpolationsansatz „via Kopf/Torso“, sowie durch die Art der Phase (org/min). Letztlich wurden Algorithmen im Zweidimensionalen angewendet. Die Interpolation von HRTFs bzw. HRIRs gleicher Kopf- und unterschiedlicher Torsoposition lieferte sehr vielversprechende Ergebnisse selbst noch für die größten Interpolationsintervalle um 35°. Diagramme des Einzahlmaßes zeigen flächig einen Fehler je Winkel kleiner als 0.5 dB und liegt damit möglicherweise in einem nicht zu hörenden Bereich. Der Ansatz „via Kopf“ zeigt für das Einzahlmaß einen sichtbaren Fortschritt im Vergleich zu „nearest neighbour“, allerdings tritt dennoch die zweite Stufe des Einzahlmaßes zwischen 0.5 und 1 dB für alle Quellen, mit Ausnahme der Dritten, schon für kleine Intervalle unterhalb 10° oder auch weniger auf. In der Evaluation des Torsoeinflusses zeigte sich, dass Daten, die sich in der Größenordnung diesen Fehlers unterschieden, im ABX-Vergleich detektiert werden konnten[20]. Das unterscheidet leicht das hier vorgestellte Fehlermaß von Minnaars et al., in deren Studie Artefakte in Datenreihen erst wahrgenommen werden konnten, für die das Maß mindestens bei 1 dB lag.

Der kubische Spline hat wie auch die zeitliche Interpolation einen vermutbaren Vorteil gegenüber dem jeweiligen Pendant der linearen Funktion bzw. der Interpolation im Frequenzbereich.
Abweichungen für ITD und ILD über dem Winkel wurden ebenfalls berechnet und sind in diesem Kapitel für die Kopfbewegung in der Horizontalen bei Quelle 1 gezeigt worden. Überall dort, wo sie im Bereich der jeweiligen (stimulusabhängigen) JNDs liegen, wird eine Lokalisationsänderung als Artefakt in einem möglichen ABX-Vergleich wahrscheinlich. Über dem Winkel gemittelte Werte zeigen den Vorteil der linearen Interpolation „via Torso“ und auch die Überlegenheit der Spline-Methode gegenüber der linearen Interpolation „via Kopf“, allerdings verschwindet für gemittelte Werte der Unterschied zwischen einer Interpolation im Frequenz- und Zeitbereich, der für das physikalisch-perzeptive Einzahlmaß qualitativ behauptet werden konnte.

[20] Auch für die unkritischste Quelle 3 waren Probanden in der Lage die Bewegungen in den Signalen A und B vor allem bei Rauschen dem Signal X zuzuordnen.

Kapitel 6

Fazit und Ausblick

In dieser Arbeit ist der physikalische Einfluss des Torsos auf das Schallfeld am geblockten Ohrkanal innerhalb einer horizontalen Kopf-über-Torso-Bewegung gezeigt worden indem für feste Quellen die Bewegungen des Kopfes bei einmal statischem und zum anderen mitgeführtem Torso verglichen wurden. Grundlage dessen bildete ein eigens angelegter HRTF-Datensatz, der mit dem Kopf-und-Torso-Simulator FABIAN gemessen wurde. Neben einfachen Betragsspektren, die die Frequenzänderungen der Bewegungen über dem Winkel zeigten, wurden Differenzen aus Betragsspektren der Bewegungen gebildet. Für verschiedene Quellpositionen waren Abweichungen voneinander unterschiedlich deutlich. Es wurde hinsichtlich einer perzeptiven Evaluation anhand dessen schon eine Hörbarkeit diesen Unterschieds vermutet. Die Differenzspektren bildeten die Datengrundlage für ein perzeptiv angelehntes Einzahlmaß über dem Winkel. Dieses war an Minnaar et al. [MPC05] angelehnt und sollte Tendenzen für eine Hörbarkeit des Torsoeinflusses in HRTFs grob vorhersagen und einem besseren Vergleich der Daten für verschiedene Quellpositionen dienen. Es konnten sich daran kritische und weniger kritische Quellen, wie sie ebenfalls bei Minnaar et al. beschrieben sind, herausstellen. Weitere physikalische Betrachtungen untersuchten die ILD und ITD beider Bewegungen je Winkel mit dem Ergebnis, nur geringe dem Torso zuzuschreibende Abweichungen gefunden zu haben. Für manche Winkelregionen bewegte sich die Differenz dieser Maße im Schwellbereich, sodass an diesen Stellen bei Auralisation Lokalisationsänderungen möglicherweise hervorrufbar wären. Wahrscheinlicher war aber, dass spektrale Verfärbungen als Unterschied zwischen beiden Bewegungen hörbar sein würden. Dies wurde in einem ABX-Test genauer untersucht. Die Quellen sind dazu anhand der physikalischen Evaluation möglichst unterschiedlich gewählt worden. Für alle sechs verschiedenen Testpaare, für die die elf normalhörenden Expertenhörer den ABX-Test zu absolvieren hatten, wurde die Hörbarkeit des Torsoeinflusses bei dynamischer Binauralsynthese statistisch gesichert. Für Sprache fielen Effekte etwas kleiner aus als für ein rosa Rauschen. Im Vergleich der Quellen waren Unterschiede der auralisierten Datensätze für die oberhalb des Kopfes positionierte Quelle am wenigsten offensichtlich. Die Vermutungen nach der physikalischen Evaluation bestätigten sich damit. Allgemein bleibt es fraglich, ob die Bewegung der statischen Kopf-über-Torso-Orientierung separat plausible Höreindrücke vermittelt.

Soll Binauralsynthese realistische Simulationen der akustischen Umgebung schaffen, ist die Darbietung exakter HRTFs auch hinsichtlich der Kopf-über-Torso-Orientierung

indiziert. Für eine Verringerung des dadurch aufkommenden Mehraufwands während der Messung von HRTFs wurden verschiedene Interpolationsalgorithmen angewendet und physikalisch evaluiert. Da der gezeigte Torsoeinfluss auf HRTF-Daten bzw. auf das räumliche Hören geringer ausfällt als die in der Literatur als maßgeblich eingeschätzte Bedeutung der Pinna und des Kopfes, war es wesentlicher Gegenstand dieser Arbeit zwei Interpolationsansätze mit einander zu vergleichen: Zum einen die Interpolation einer HRTF mit bestimmter HATO „via Kopf“ aus HRTFs unterschiedlicher Kopf- bei gleicher Torsoposition und zum anderen die Interpolation einer HRTF „via Torso“ bei unterschiedlicher Kopf- und gleicher Torsoposition. Nachdem verschiedene bei Hartung et al. [HBS99] gefundene Algorithmen angepasst und angewendet wurden, zeigten sich in Deutlichkeit die besseren Interpolationsergebnisse für die Variante „via Torso“, da durch die identische Kopforientierung der Stützstellen-HRTFs die komplizierte Pinnafeinstruktur der Ziel-HRTF bereits enthalten und so nicht fehleranfällig zu berechnen war. Diese Art der Interpolation benötigt für jede HRTF eines Winkels zwei Stützstellen unabhängig vom Interpolationsintervall. Da für vorhandene groß angelegte HRTF-Datensätze Messungen für viele Quellpositionen üblich sind, stehen benötigte HRTFs bereits als Stützstellen per se zur Verfügung, wodurch der Mehraufwand für die Messung verschiedener Kopf-über-Torso-Orientierungen verschwindet. Handelt es sich um sehr kleine Datensätze, empfiehlt sich schlussfolgernd die Interpolation der Daten für eine dynamische HATO „via Kopf“ in kleineren Intervallen - dafür wurden beste Ergebnisse mit nur kleinem Abstand zu anderen Verfahren für die Splinemethode im Zeitbereich erzielt.

Welches Intervall die Schwelle der Hörbarkeit von Artefakten darstellt ist im Ausblick in einer perzeptiven Evaluation zu hinterfragen. Dort könnten weitere Algorithmen beleuchtet und auch die Kopfbewegung innerhalb der Median- und Frontalebene näher betrachtet werden.

Literaturverzeichnis

[AAD01] V Ralph Algazi, Carlos Avendano, and Richard O Duda. Elevation localization and head-related transfer function analysis at low frequencies. *The Journal of the Acoustical Society of America*, 109(3):1110–1122, 2001. (Cited on pp. 6, 8, 9, 71.)

[ADD+02] V Ralph Algazi, Richard O Duda, Ramani Duraiswami, Nail A Gumerov, and Zhihui Tang. Approximating the head-related transfer function using simple geometric models of the head and torso. *The Journal of the Acoustical Society of America*, 112:2053, 2002. (Cited on p. 8.)

[BB08] Jens Blauert and Jonas Braasch. Räumliches Hören. In Stefan Weinzierl, editor, *Handbuch der Audiotechnik*, page 87–121. Springer, 2008. (Cited on p. 4.)

[BDVBVdO99] AJ Berkhout, D De Vries, J Baan, and BW Van den Oetelaar. A wave field extrapolation approach to acoustical modeling in enclosed spaces. *The Journal of the Acoustical Society of America*, 105:1725, 1999. (Cited on p. 6.)

[Bla97] Jens Blauert. *Spatial Hearing-Revised Edition: The Psychophysics of Human Sound Localization.* MIT press, 1997. (Cited on pp. 5, 8.)

[Bor05] Jürgen Bortz. *Statistik: Für Human-und Sozialwissenschaftler.* Springer DE, 2005. (Cited on p. 42.)

[BS75] MD Burkhard and RM Sachs. Anthropometric manikin for acoustic research. *The Journal of the Acoustical Society of America*, 58:214, 1975. (Cited on p. 10.)

[Bur88] Herman Burstein. Approximation formulas for error risk and sample size in ABX testing. *The Journal of the Audio Engineering Society*, 36(11):879–883, 1988. (Cited on pp. 35, 42.)

[BW97] Michael A. Blommer and Gregory H Wakefield. Pole-zero approximations for head-related transfer functions using a logarithmic error criterion. *Speech and Audio Processing, IEEE Transactions on*, 5(3):278–287, 1997. (Cited on p. 11.)

[CJM00] Flemming Christensen, Clemen Boje Jensen, and Henrik Møller. The design of VALDEMAR-An artificial head for binaural recording

purposes. In *109th AES-Convention*, Los Angeles, 2000. (Cited on p. 10.)

[EAT98] Michael J Evans, James AS Angus, and Anthony I Tew. Analyzing head-related transfer function measurements using surface spherical harmonics. *The Journal of the Acoustical Society of America*, 104:2400, 1998. (Cited on p. 11.)

[ESLW12] Vera Erbes, Frank Schultz, Alexander Lindau, and Stefan Weinzierl. An extraaural headphone system for optimized binaural reproduction. In *DAGA 2012, Darmstadt*, page 313–314, 2012. (Cited on pp. 37, 39.)

[FELB07] Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, 39(2):175–191, 2007. (Cited on p. 35.)

[For42] Adelbert Ford. Dynamic auditory localization. *The Journal of the Acoustical Society of America*, 13(4):367–372, 1942. (Cited on p. 5.)

[Gar73] Mark B Gardner. Some monaural and binaural facets of median plane localization. *The Journal of the Acoustical Society of America*, 54(6):1489–1495, 1973. (Cited on p. 8.)

[GSZ08] M Guldenschuh, A Sontacchi, and F Zotter. HRTF modeling in due consideration variable torso reflections. *The Journal of the Acoustical Society of America*, 123(5):3080–3085, 2008. (Cited on p. 9.)

[hb00] http://www.seacen.tu berlin.de/. Simulation und Evaluation akustischer Umgebungen (SEACEN), 20.08.13, 8:00. (Cited on pp. 3, 11.)

[HBS99] Klaus Hartung, Jonas Braasch, and Susanne Sterbing. Comparison of different methods for the interpolation of head-related transfer functions. In *16th AES-Convention*, pages 319–329, Rovaniemi, Finland, 1999. (Cited on pp. 11, 48, 50, 63, 65.)

[KIC95] A Kulkarni, SK Isabelle, and HS Colburn. On the minimum-phase approximation of head-related transfer functions. In *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, page 84–87, 1995. (Cited on p. 28.)

[Kle20] Otto Klemm. Untersuchungen über die Lokalisation von Schallreizen IV: Über den Einfluß des binauralen Zeitunterschiedes auf die Lokalisation. *Arch. ges. Psychol*, 40:117–145, 1920. (Cited on p. 5.)

[KW92] Doris J Kistler and Frederic L Wightman. A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction. *The Journal of the Acoustical Society of America*, 91:1637, 1992. (Cited on p. 11.)

[Lev86] Les Leventhal. Type 1 and type 2 errors in the statistical analysis of listening tests. *Journal of the Audio Engineering Society*, 34(6):437–453, 1986. (Cited on pp. 35, 36, 42.)

[LHW07] Alexander Lindau, Torben Hohn, and Stefan Weinzierl. Binaural resynthesis for comparative studies of acoustical environments. In *Audio Engineering Society Convention 122*, 2007. (Cited on p. 10.)

[LJGW12] Véronique Larcher, JM Jot, J Guyard, and O Warusfel. Study and comparison of efficient methods for 3d audio spatialization based on linear decomposition of HRTF data. In *108th AES-Convention*, 2012. (Cited on p. 11.)

[LW06] Alexander Lindau and Stefan Weinzierl. FABIAN-An instrument for software-based measurement of binaural room impulse responses in multiple degrees of freedom. In *24th TMT*, Leipzig, 2006. (Cited on pp. 11, 10.)

[LW09] Alexander Lindau and Stefan Weinzierl. On the spatial resolution of virtual acoustic environments for head movements on horizontal, vertical and lateral direction. In *Proc. of EAA Symposium on Auralization*, Espoo, Finland, 2009. (Cited on p. 11.)

[LW12] Alexander Lindau and Stefan Weinzierl. Assessing the plausibility of virtual acoustic environments. *Acta Acus. United Ac.*, 98:804–810, 2012. (Cited on p. 1.)

[Mø92] Henrik Møller. Fundamentals of binaural technology. *Apllied Acoustics*, 36:171–218, 1992. (Cited on pp. 1, 4, 6.)

[Mü08] Swen Müller. Messtechnik. In Stefan Weinzierl, editor, *Handbuch der Audiotechnik*, pages 1087–1168. Springer, 2008. (Cited on p. 6.)

[MG91] John C. Middlebrooks and David M. Green. Sound localization by human listeners. *Annual review of psychology*, 42(1):135–159, 1991. (Cited on p. 8.)

[Mil47] George A Miller. Sensitivity to changes in the intensity of white noise and its relation to masking and loudness. *The Journal of the Acoustical Society of America*, 19:609, 1947. (Cited on p. 31.)

[Mil58] Allen William Mills. On the minimum audible angle. *The Journal of the Acoustical Society of America*, 30:237, 1958. (Cited on p. 5.)

[Mol02] C Moldrzyk. Ein neuartiger Kunstkopf zur Verifikation einer akustischen Entwurfsmethodik für Architekten. In *22nd TMT*, Hannover, 2002. (Cited on p. 10.)

[Moo95a] Brian CJ Moore. Frequency analysis and masking. In Brian CJ Moore, editor, *Hearing*, page 161–205. Academic Press, San Diego/New York, 1995. (Cited on pp. 32, 33.)

[Moo95b] Brian CJ Moore. *Hearing*. Academic Press, San Diego/New York, 1995. (Cited on p. 32.)

[MPC05] Pauli Minnaar, Jan Plogsties, and Flemming Christensen. Directional resolution of head-related transfer functions required in binaural synthesis. *The Journal of the Audio Engineering Society*, 53(10):919–929, 2005. (Cited on pp. 8, 9, 11, 33, 45, 64.)

[Nic10] R Nicol. Binaural technology. In *Audio Engineering society*, 2010. (Cited on pp. 1, 4.)

[OSB+89] Alan V Oppenheim, Ronald W Schafer, John R Buck, et al. *Discrete-time signal processing*, volume 2. Prentice hall Englewood Cliffs, NJ:, 1989. (Cited on p. 28.)

[Pla10] Robert Plato. *Numerische Mathematik kompakt*. Springer, 2010. (Cited on p. 51.)

[RBW95] PR Runkle, MA Blommer, and GH Wakefield. A comparison of head related transfer function interpolation methods. In *Applications of Signal Processing to Audio and Acoustics, IEEE ASSP Workshop on*, page 88–91, 1995. (Cited on p. 11.)

[Rob97] Scott M. Robeson. Spherical methods for spatial interpolation: Review and evaluation. *Cartography and Geographic Information Science*, 24(1):3–20, 1997. (Cited on p. 11.)

[Sal95] Aurelia Maria Salomons. *Coloration and binaural decoloration of sound due to reflections*. Technische Universität Delft, 1995. (Cited on p. 32.)

[SBDC76] CL Searle, LD Braida, MF Davis, and HS Colburn. Model for auditory localization. *The Journal of the Acoustical Society of America*, 60:1164, 1976. (Cited on p. 8.)

[Sch95] A. Schmitz. Ein neues digitales Kunstkopfmeßsystem. *Acta Acustica united with Acustica*, 81(4):416–420, 1995. (Cited on p. 10.)

[SL09] Zora Schärer and Alexander Lindau. Evaluation of equalization methods for binaural signals. In *Audio Engineering Society Convention 126*, 2009. (Cited on pp. 7, 32.)

[Sla93] Malcolm Slaney. An efficient implementation of the patterson-holdsworth auditory filter bank. *Apple Computer, Perception Group, Technical Report*, 1993. (Cited on p. 32.)

[SR13] Peter Sedlmeier and Frank Renkewitz. *Forschungsmethoden und Statistik - Ein Lehrbuch für Psychologen und Sozialwissenschaftler*. Pearson, München, 2 edition, 2013. (Cited on p. 44.)

[SSS+97] P Schöps, N Seichert, M Schenk, U Petri, and E Senn. Alters- und geschlechtsspezifische Bewegungsausmaße der Halswirbelsäule. *Phys Rehab Kur Med*, 7(3):80–87, 1997. (Cited on p. 10.)

[Tau04] Jens Tausendfreund. *Dreidimensionale Bewegungsanalyse der Halswirbelsäule mit dem Zebris® Ultraschallsystem.* PhD thesis, Philips-Universität Marburg, 2004. (Cited on p. 10.)

[TMR67] WR Thurlow, JW Mangels, and PS Runge. Head movements during sound localization. *The Journal of the Acoustical society of America*, 42(2):489–493, 1967. (Cited on p. 10.)

[Vor08] Michael Vorländer. *Auralization: fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality.* Springer Publishing Company, Incorporated, 2008. (Cited on p. 6.)

[Wet70] R Wettschurek. Über Unterschiedsschwellen beim Richtungshören in der Medianebene. *Gemeinschaftstagung für Akustik und Schwingungstechnik, Berlin*, page 385–388, 1970. (Cited on p. 5.)

Abbildungsverzeichnis

2.1 Skizze des Schallwegs zum Trommelfell 5
2.2 Skizze zur datenbasierten Binauralsynthese 7
2.3 durch Torso verursachter Kammfilter (Bildquelle: [AAD01]) 9

3.1 Koordinatensystem und Quellpositionen in der Horizontalebene 15
3.2 Gerüst auf Stahlstangen im reflexionsarmen Raum 16
3.3 Signalpfad bei Messung 17
3.4 Signalpfad bei Referenzmessung 17
3.5 Setup 1a, 1b 19
3.6 DPA 4060 Miniaturmikrofon ohne Kapsel 20
3.7 Ergebnisse des vorab durchgeführten SNR- und Reflexionstests 21
3.8 Frequenzspezifisches SNR 22
3.9 mechanische Reliabilität 22
3.10 Ergebniss Lautsprecher-Belastungstest 23
3.11 Eigenschaften des Testsignals 24
3.12 Beispiel eines Kontrollplots: zeigt HRTFs der Quelle 1 [0°; 0°] in der Horizontalebene über alle Kopf-zu-Quellazimuthwinkel (da Quelle 1 gleich dem Kopfazimuth) bei steter HATO von 0° 26
3.13 Kontrollplot der HRTF-ITD und -ILD der Quelle 1 in der Horizontalebene über alle Kopf-zu-Quellazimuthwinkel bei stetem Torsoazimuth von 0° 27

4.1 statische HATO: Kopfbewegung bei Mitführung des Torsos 29
4.2 dynamische HATO: Kopfbewegung bei statischem Torso 30
4.3 dynam. HATO bei Torsoazimuth=0°, Quelle 1 30
4.4 Differenz (stat. - dynam.), Quelle 1 31
4.5 Differenz (stat. - dynam.), Quelle 1 33
4.6 Bernoulli-Verteilungen für die Effektgrößen 50 und 75% bei 23 Trials . . 36
4.7 im ABX-Vergleich verwendeter Signalpfad 37
4.8 Benutzeroberfläche in WHISPER 38
4.9 Signalwege in pure data pd 39
4.10 Proband auf Drehstuhl mit Tablet und BK211 40
4.11 Rohdaten aller Probanden 41

5.1 Interpolation „via Kopf“ bei Interpolationsintervall von 30° 47
5.2 Interpolation „via Torso“ bei Interpolationsintervall von 30° 47
5.3 Bsp. Gegenüberstellung von interpolierten Daten und Referenz für minimalphasige HRTFs je rechtes Ohr 54

5.4 Differenzspektren aus interpolierten Daten und Referenz für minimalphasige HRTFs je rechtes Ohr 56
5.5 Bsp. ERB-Skalen basiertes Fehlermaß aller Interpolationsintervalle . . . 56
5.6 ILD-Abweichungen über dem Winkel je Algorithmus, Horizontalebene, Quelle 1 59
5.7 ITD-Abweichungen über dem Winkel je Algorithmus, Horizontalebene, Quelle 1 62

A.1 Betragsspektren der Bewegung bei statischer HATO=0° 77
A.2 Betragsspektren der Bewegung bei HATO=± 82° und Torsoazimuth=0° 80
A.3 Differenzen der Bewegungen wie in Kapitel 4.1 beschrieben 83
A.4 Differenz der ILD-Werte beider Bewegungen 85
A.5 Differenz der ITD-Werte beider Bewegungen 86
A.6 ERB-Skalen basiertes Fehlermaß für Quelle 1, Horizontalebene 88
A.7 ERB-Skalen basiertes Fehlermaß für Quelle 2, Horizontalebene 89
A.8 ERB-Skalen basiertes Fehlermaß für Quelle 3, Horizontalebene 90
A.9 ERB-Skalen basiertes Fehlermaß für Quelle 4, Horizontalebene 92
A.10 ERB-Skalen basiertes Fehlermaß für Quelle 5, Horizontalebene 93
A.11 ERB-Skalen basiertes Fehlermaß für Quelle 6, Horizontalebene 94
A.12 ERB-Skalen basiertes Fehlermaß für Quelle 1, Medianebene bei Kopfazimuth 0° 95
A.13 ERB-Skalen basiertes Fehlermaß für Quelle 1, Frontalebene bei Kopfazimuth 0° 96
A.14 mittlerer ILD-Abweichung in dB in Abhängigkeit des Interpolationsintervalls je Algorithmus, Horizontalebene 97
A.15 mittlerer ITD-Abweichung in μs in Abhängigkeit des Interpolationsintervalls je Algorithmus, Horizontalebene 98

Tabellenverzeichnis

2.1 JNDs - Lateralisation/Lokalisation 5
2.2 hörtypischer und maximaler Bewegungsbereich 10

4.1 Ergebnisse des Bernoulli-Tests aufgeschlüsselt nach Probanden und Testpaaren 43
4.2 Ergebnisse des Bernoulli-Tests durchgeführt für alle im Experiment gegebenen Antworten je Testpaar 44

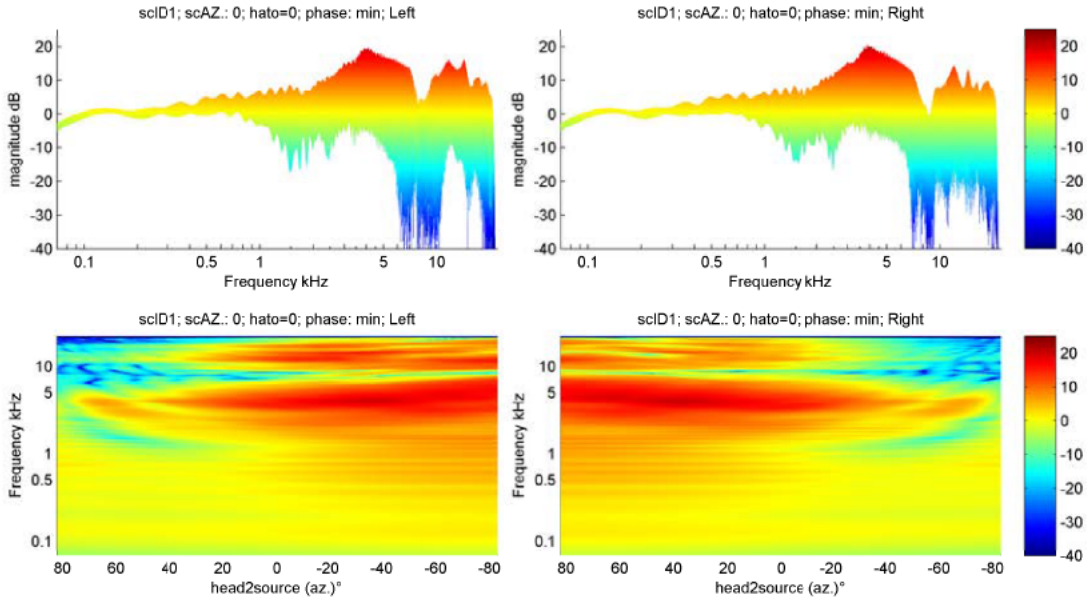
5.1 HATO-Interpolationsintervalle aller Ebenen in ° 48
5.2 Übersicht Interpolationsalgorithmen 49

C.1 Ruhehörschwellen der Luftleitung nach Békèsy-Festfrequenzaudiometrie 105
C.2 Antworten der Probanden 106

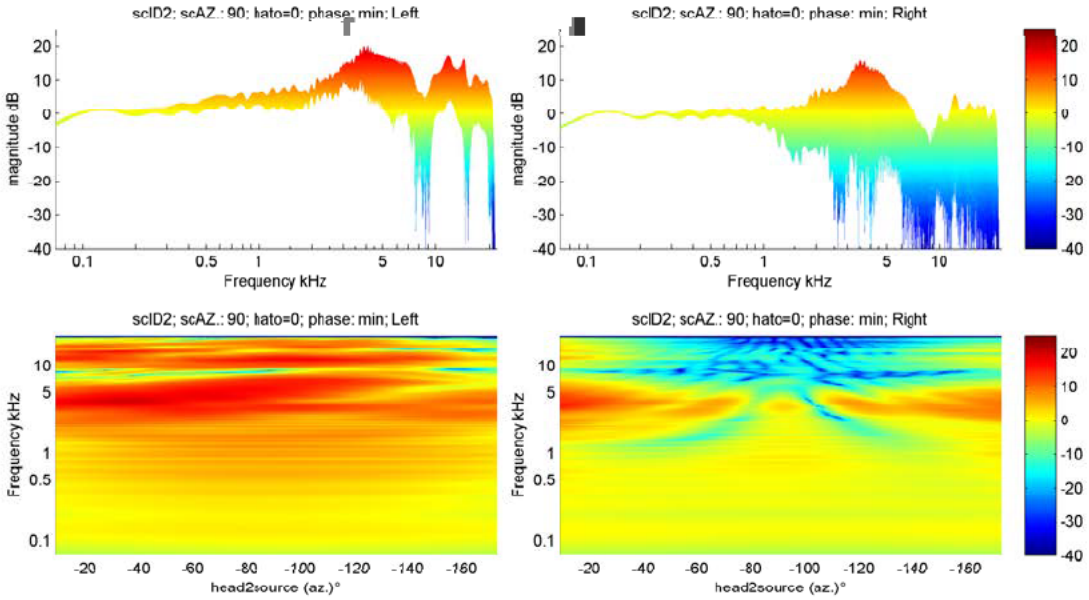
Anhang A

Abbildungen

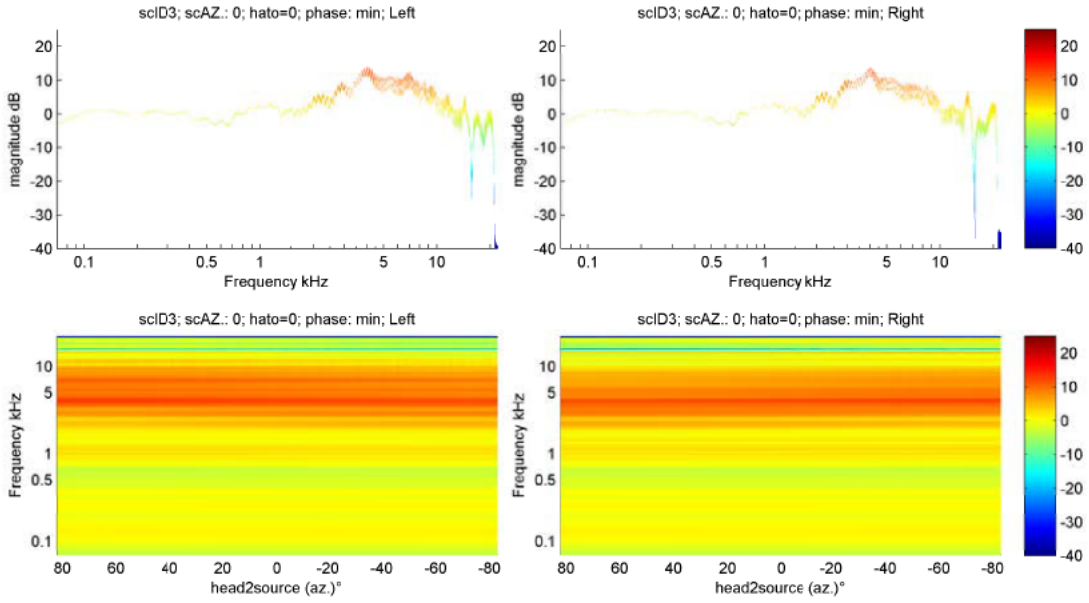
i zur Hörbarkeit



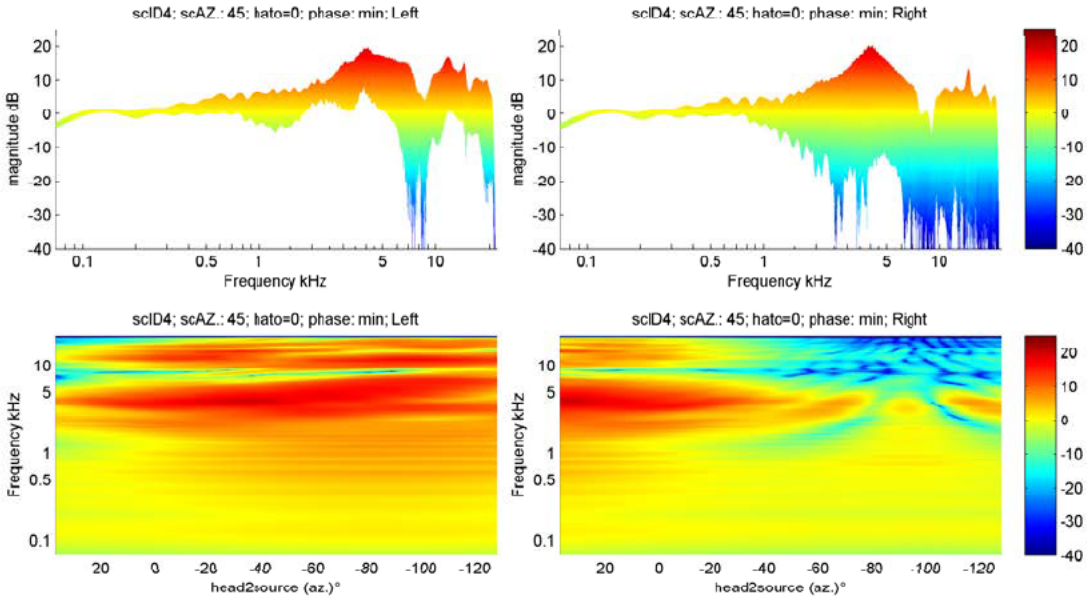
(a) stat. HATO, Quelle 1



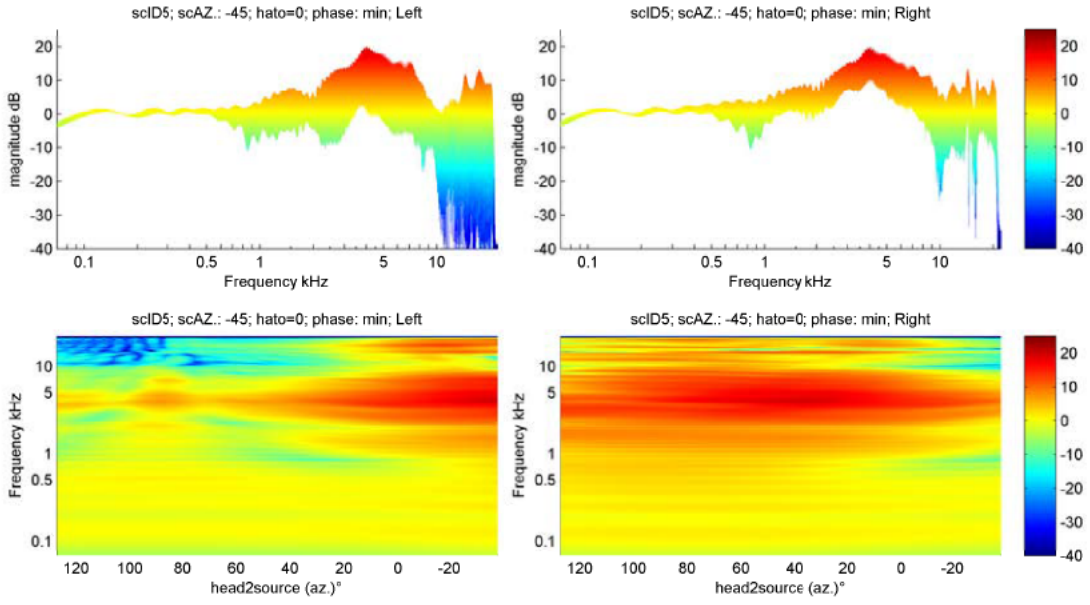
(b) stat. HATO, Quelle 2



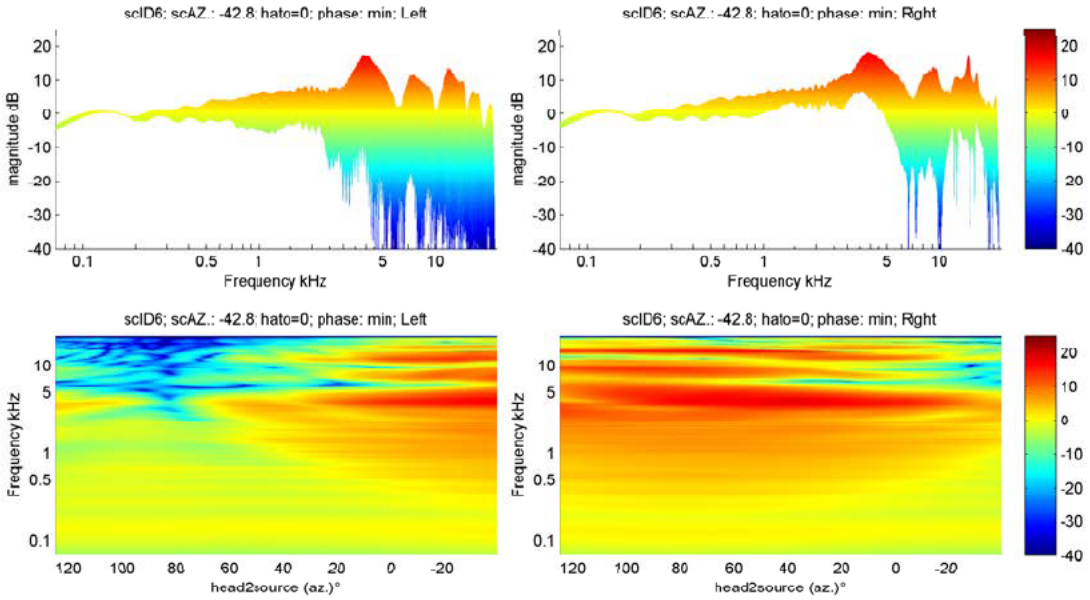
(c) stat. HATO, Quelle 3



(d) stat. HATO, Quelle 4

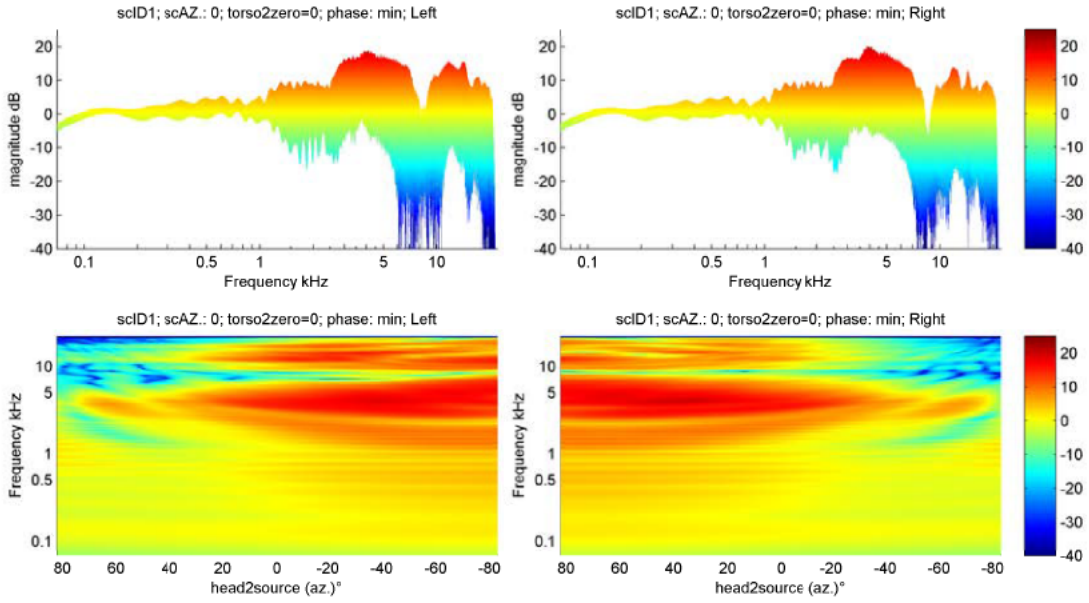


(e) stat. HATO, Quelle 5

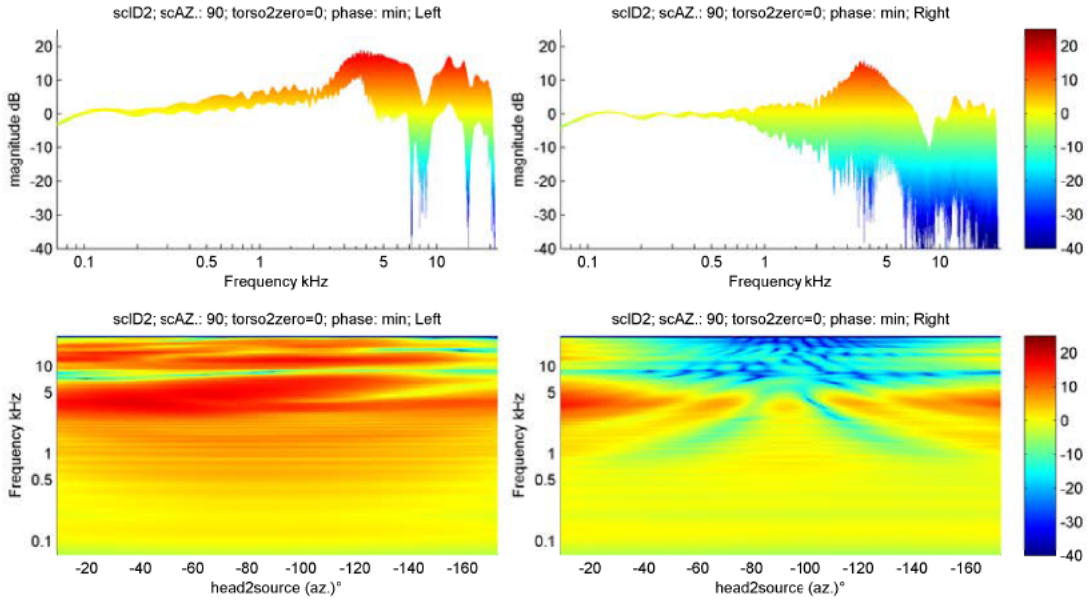


(f) stat. HATO, Quelle 6

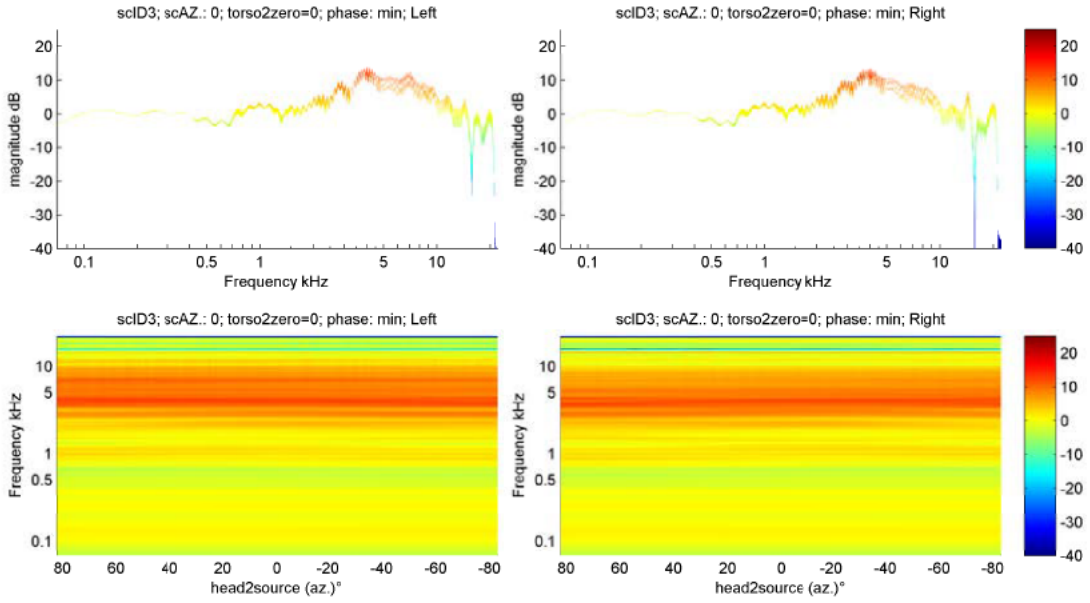
Abbildung A.1: Betragsspektren der Bewegung bei statischer HATO=0°
a) bis f) entsprechen den Quellpositionen 1 bis 6



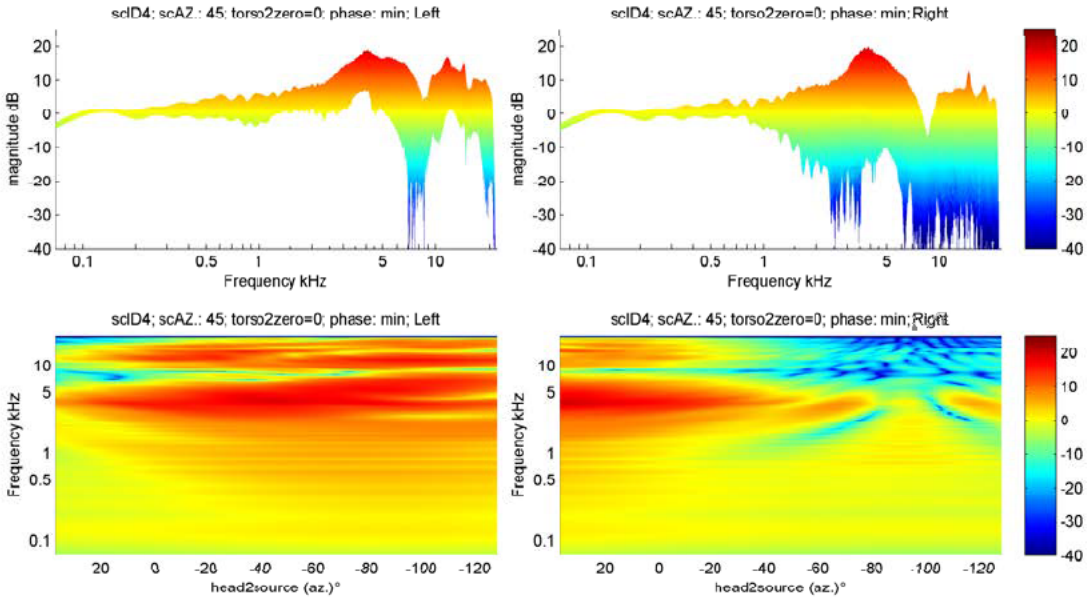
(a) dynam. HATO, Quelle 1



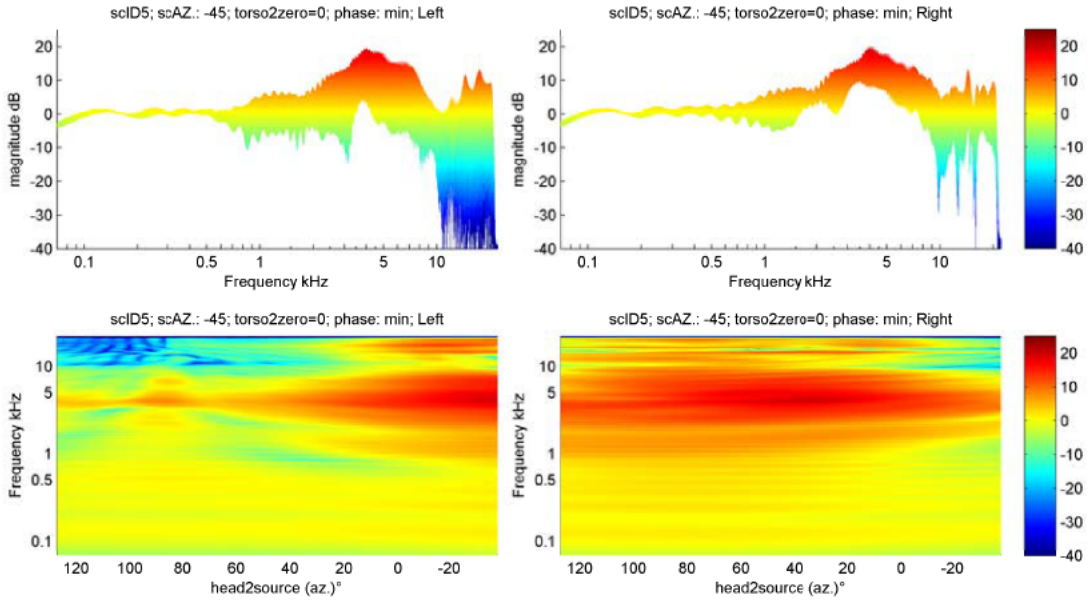
(b) dynam. HATO, Quelle 2



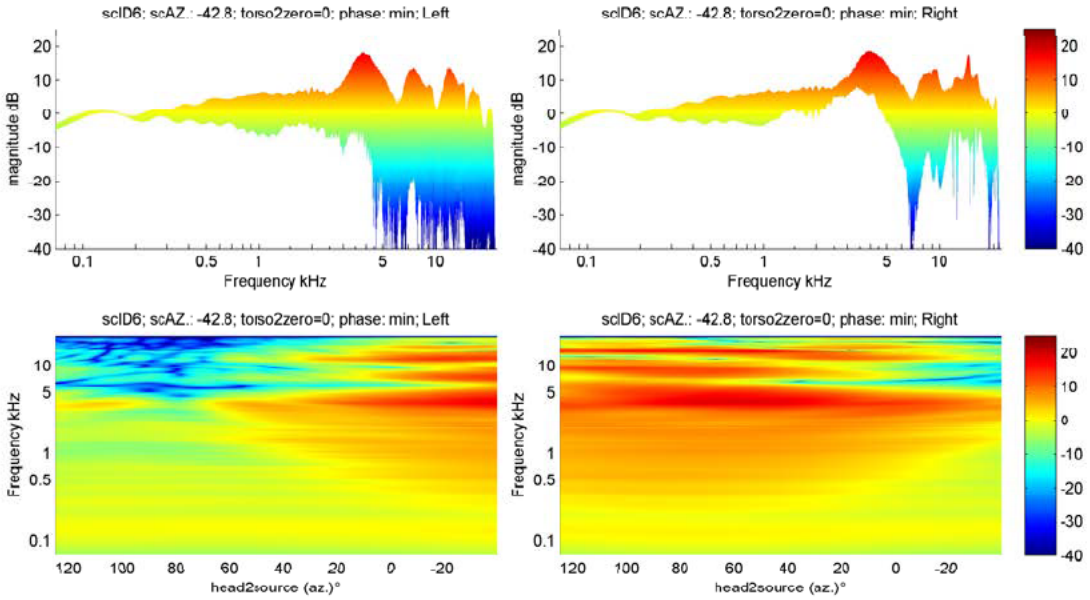
(c) dynam. HATO, Quelle 3



(d) dynam. HATO, Quelle 4



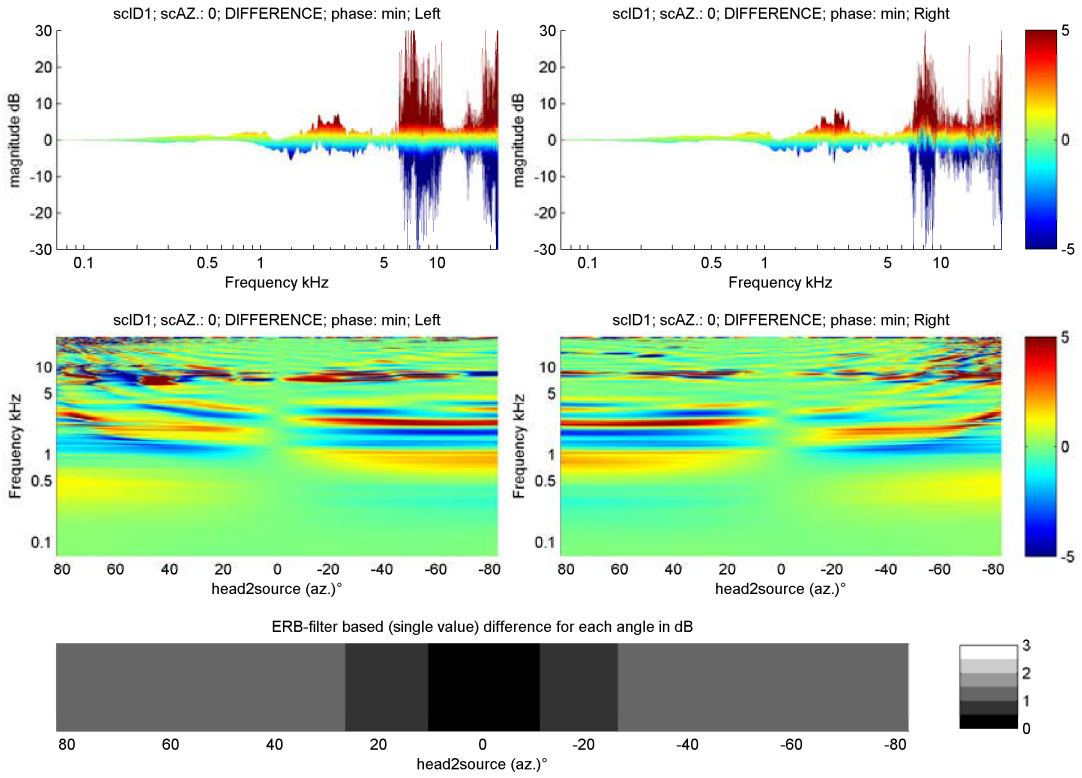
(e) dynam. HATO, Quelle 5



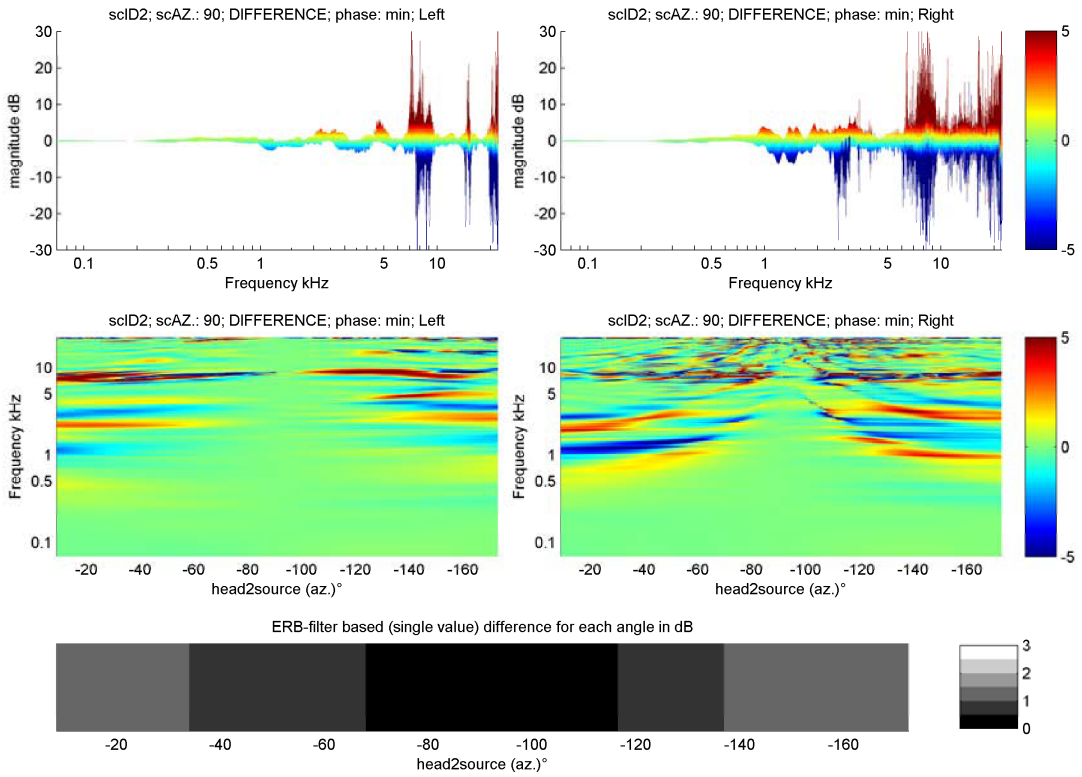
(f) dynam. HATO, Quelle 6

Abbildung A.2: Betragsspektren der Bewegung bei HATO=± 82° und Torsoazimuth=0°

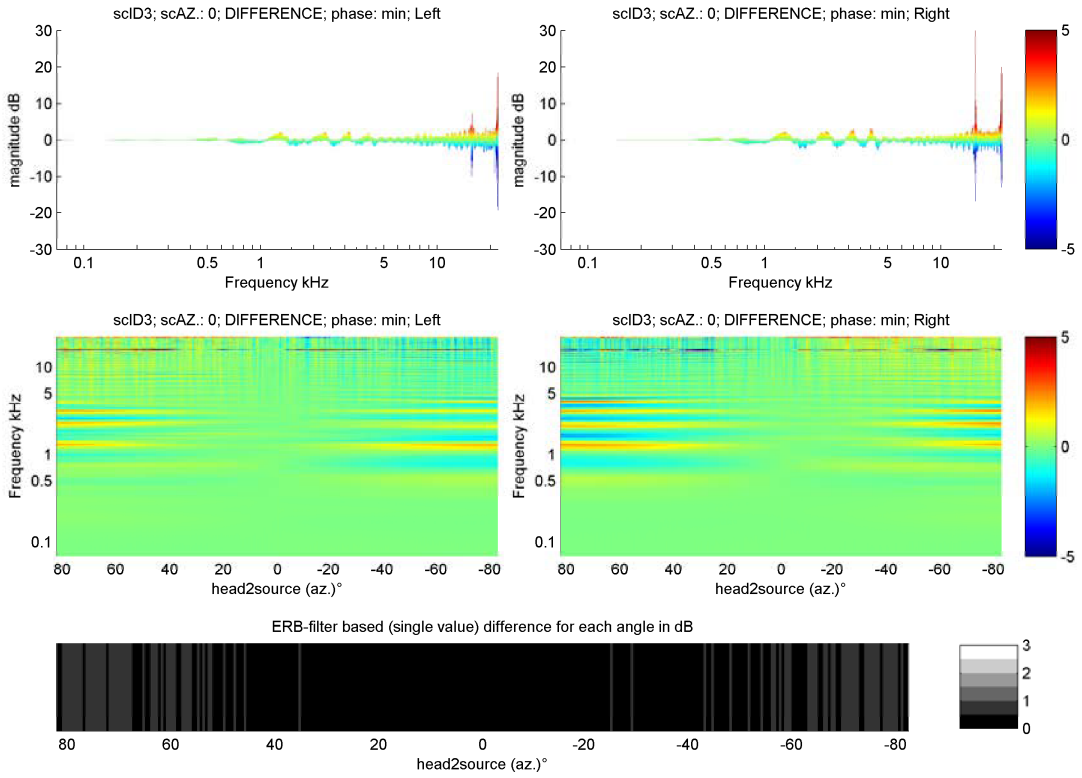
a) bis f) entsprechen den Quellpositionen 1 bis 6



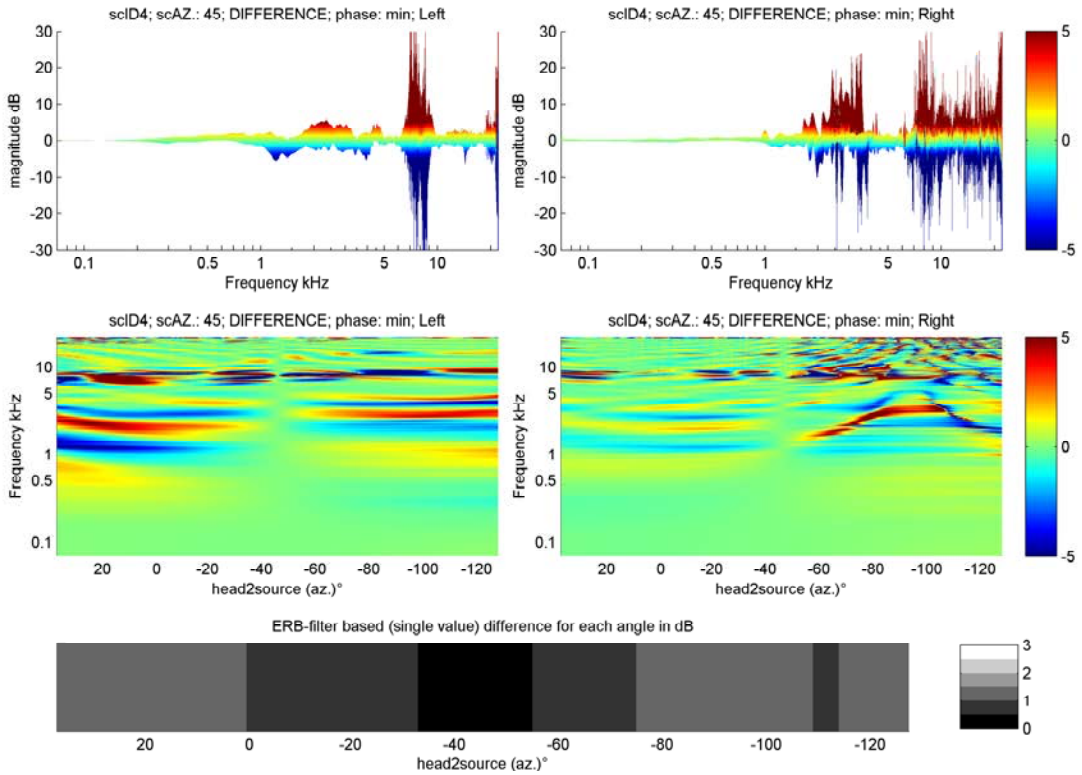
(a) Differenz (stat - dynam.), Quelle 1



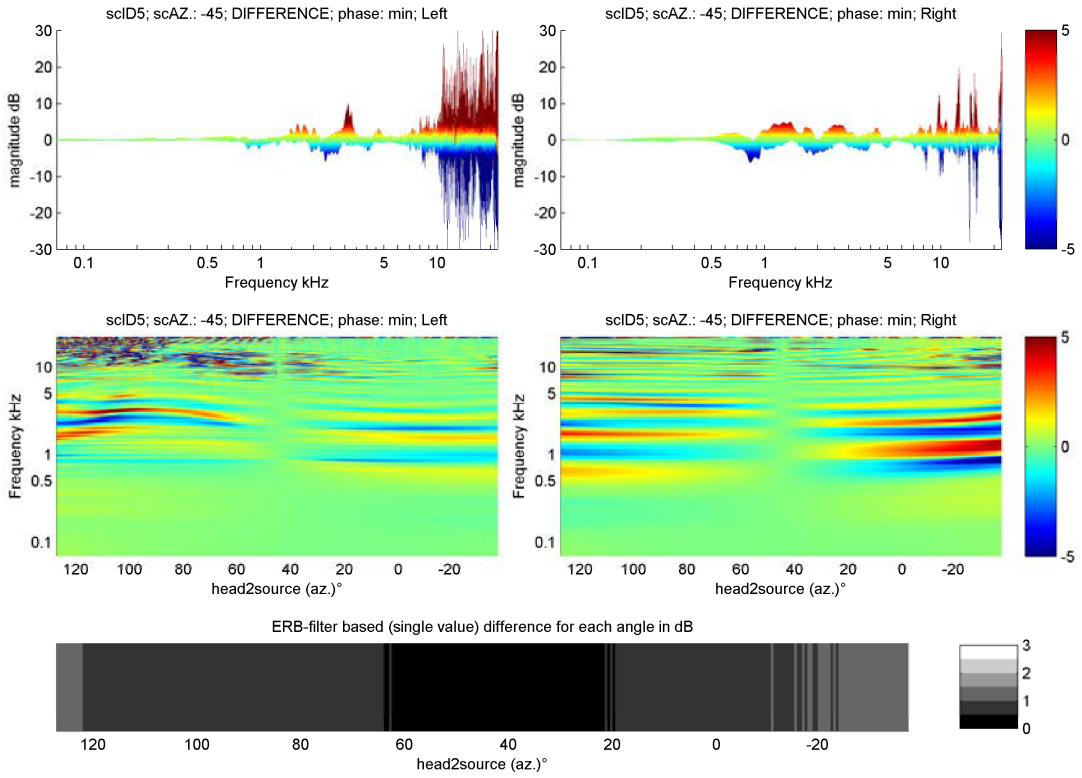
(b) Differenz (stat - dynam.), Quelle 2



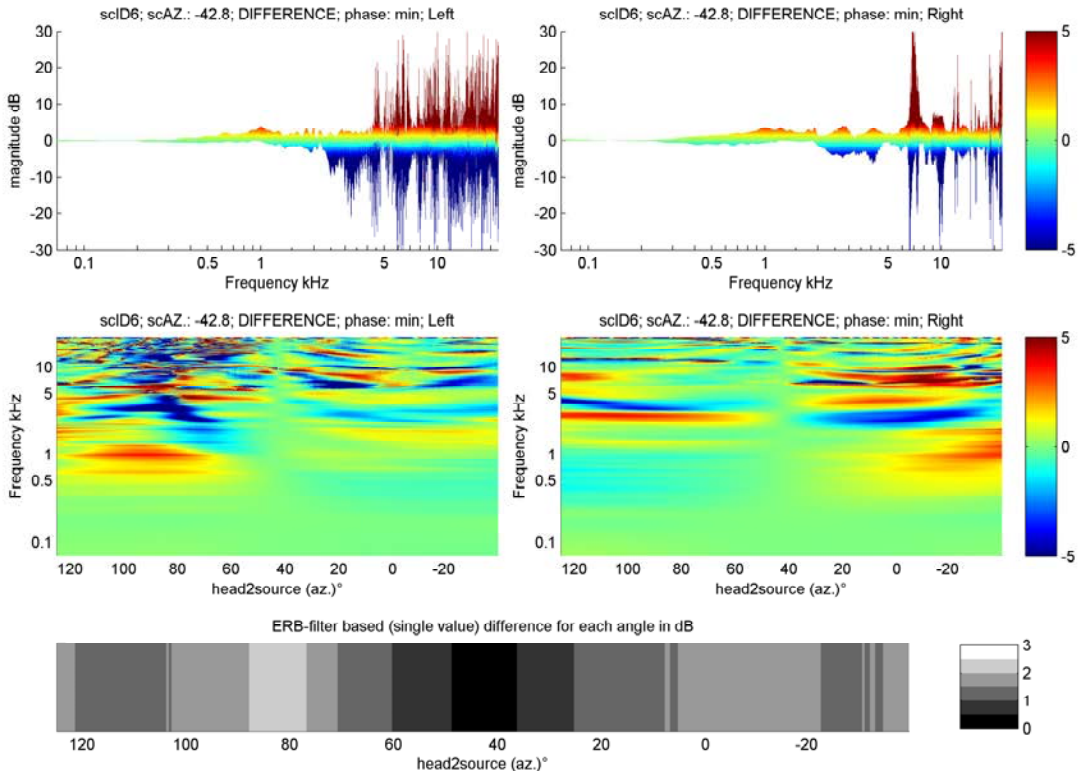
(c) Differenz (stat - dynam.), Quelle 3



(d) Differenz (stat - dynam.), Quelle 4

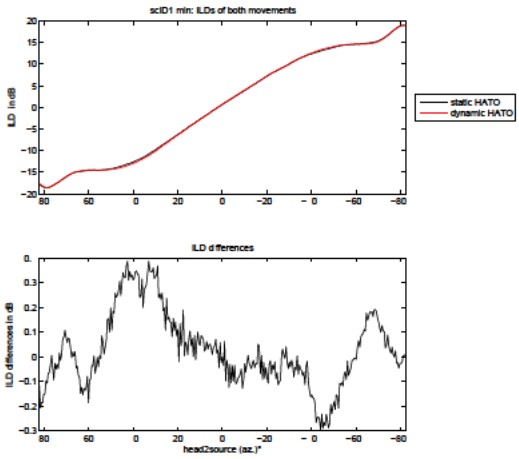


(e) Differenz (stat - dynam.), Quelle 5

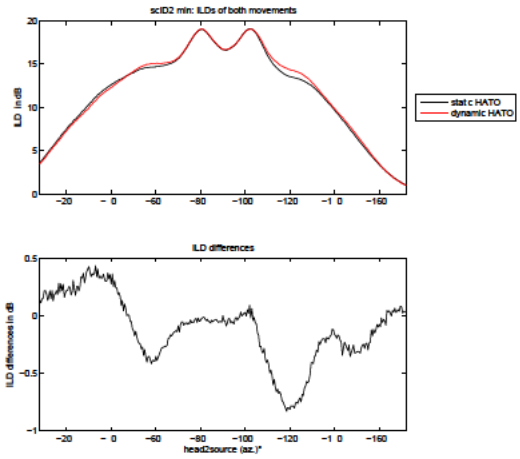


(f) Differenz (stat - dynam.), Quelle 6

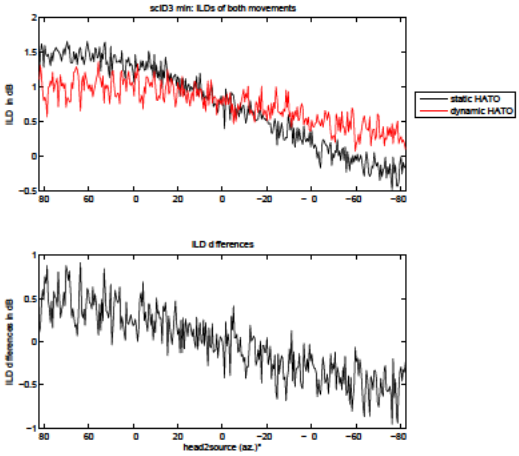
Abbildung A.3: Differenzen der Bewegungen wie in Kapitel 4.1 beschrieben a) bis f) entsprechen den Quellpositionen 1 bis 6



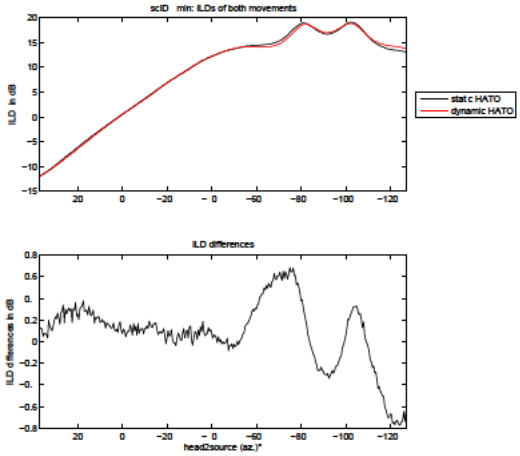
(a) Differenz der ILD, Quelle 1



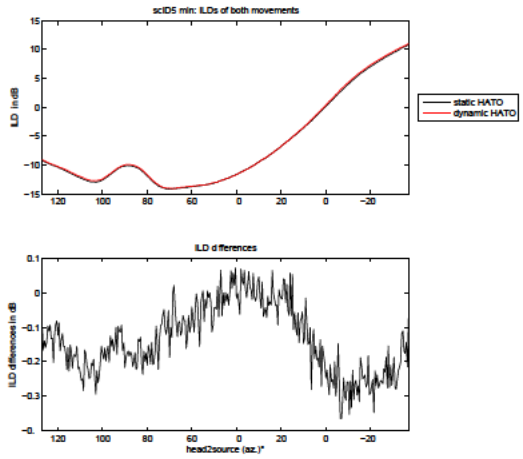
(b) Differenz der ILD, Quelle 2



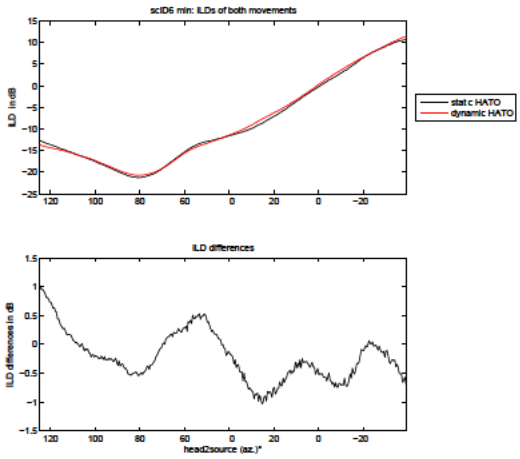
(c) Differenz der ILD, Quelle 3



(d) Differenz der ILD, Quelle 4

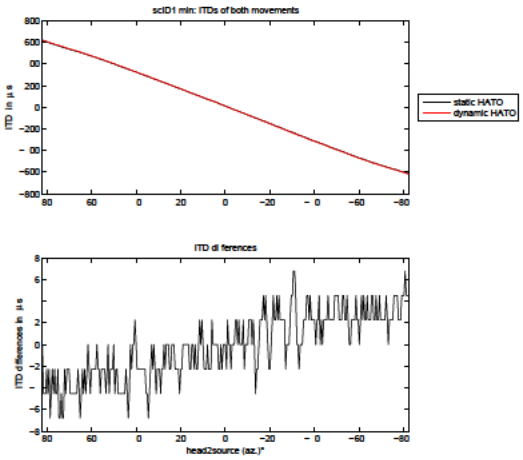


(e) Differenz der ILD, Quelle 5

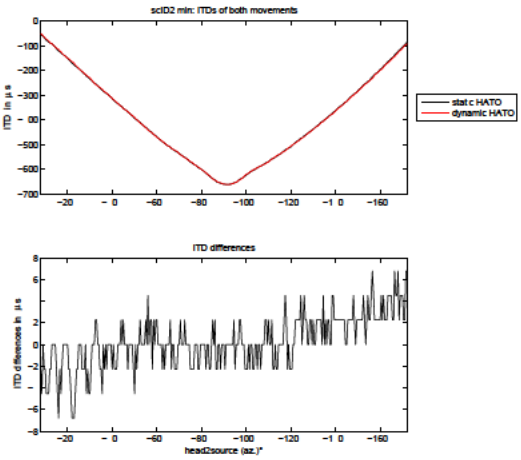


(f) Differenz der ILD, Quelle 6

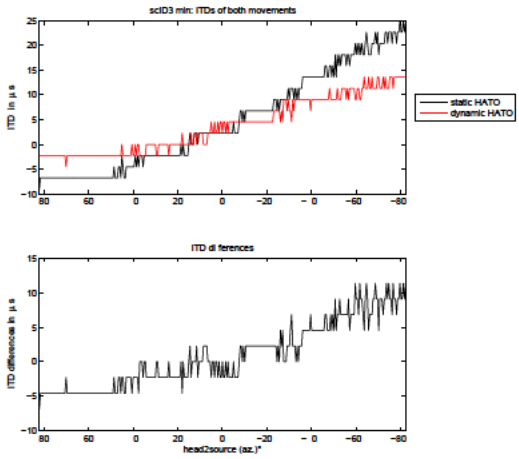
Abbildung A.4: Differenz der ILD-Werte beider Bewegungen
a) bis f) entsprechen den Quellpositionen 1 bis 6



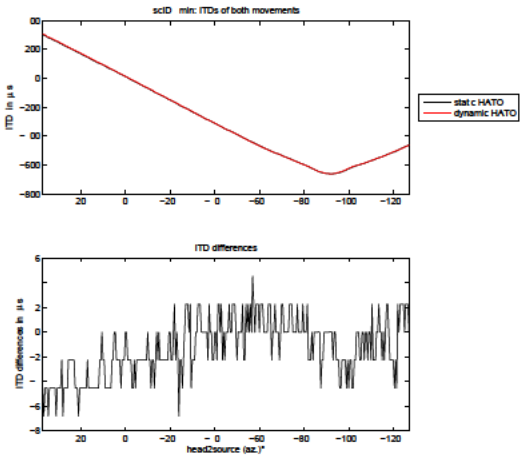
(a) Differenz der ITD, Quelle 1



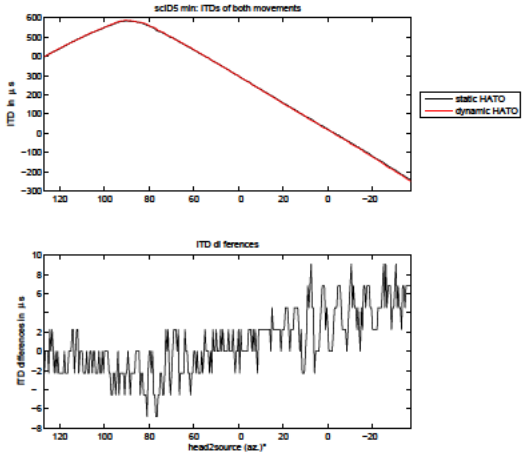
(b) Differenz der ITD, Quelle 2



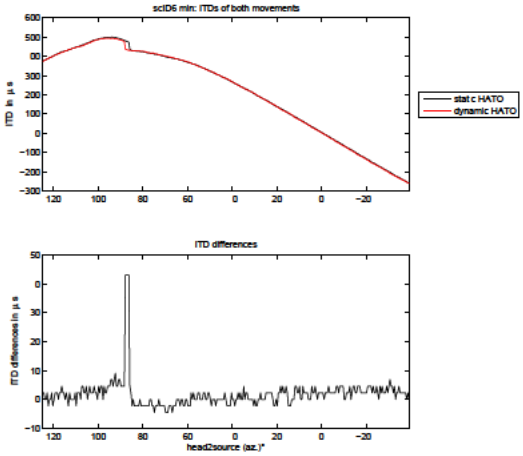
(c) Differenz der ITD, Quelle 3



(d) Differenz der ITD, Quelle 4



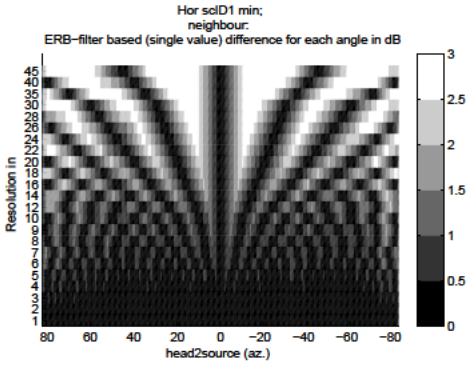
(e) Differenz der ITD, Quelle 5



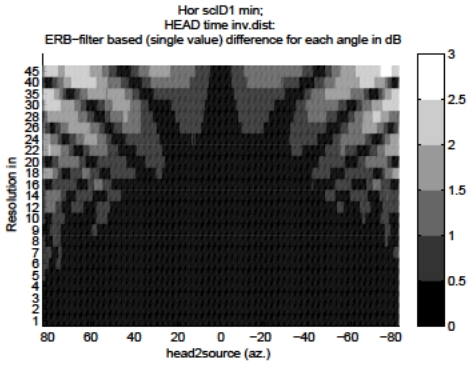
(f) Differenz der ITD, Quelle 6

Abbildung A.5: Differenz der ITD-Werte beider Bewegungen
a) bis f) entsprechen den Quellpositionen 1 bis 6

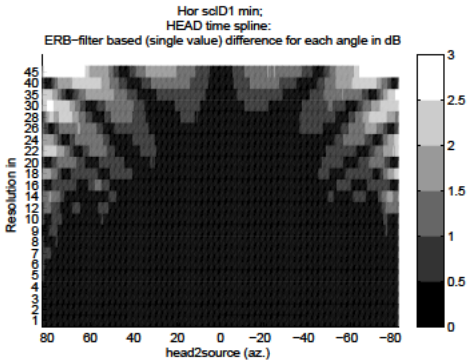
ii zur Interpolation



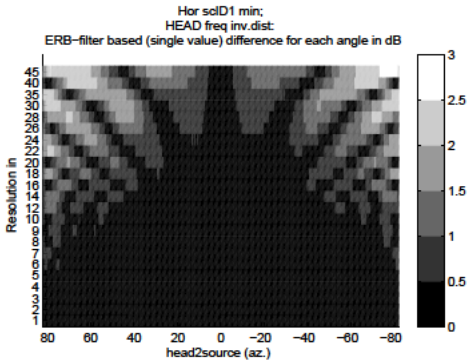
(a) „via Kopf“, nearest neighbour



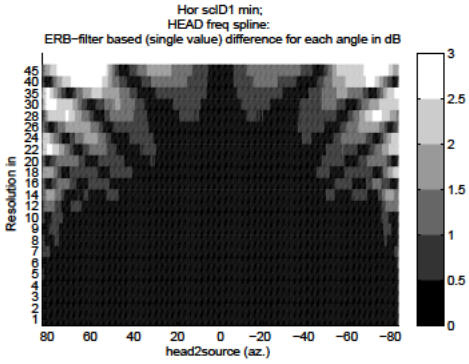
(b) zeitl. „via Kopf“, linear



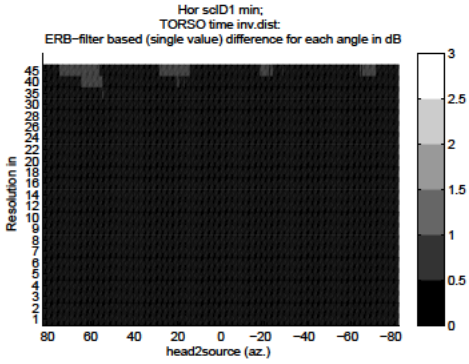
(c) zeitl. „via Kopf“, spline



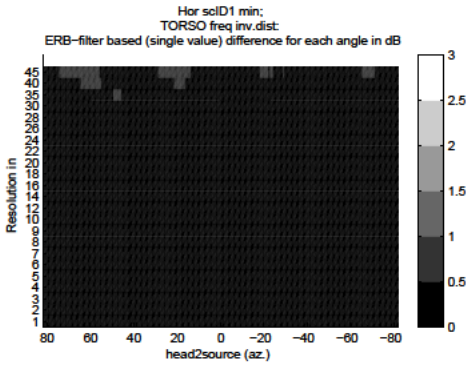
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

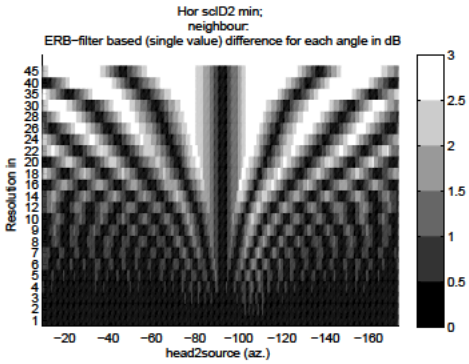


(f) zeitl. „via Torso“, linear

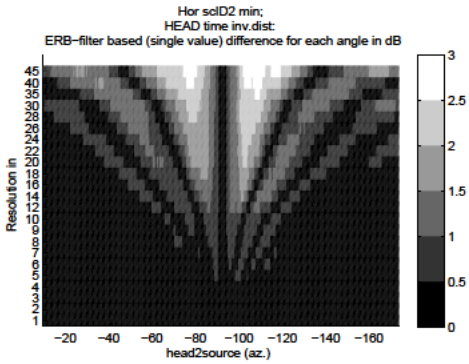


(g) spektral „via Torso“, linear

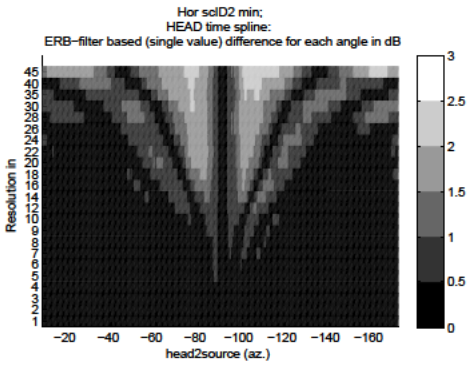
Abbildung A.6: ERB-Skalen basiertes Fehlermaß für Quelle 1, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



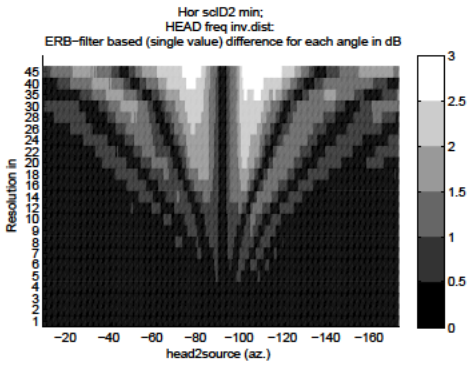
(a) „via Kopf“, nearest neighbour



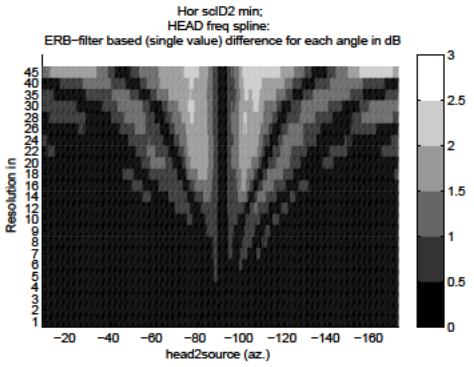
(b) zeitl. „via Kopf“, linear



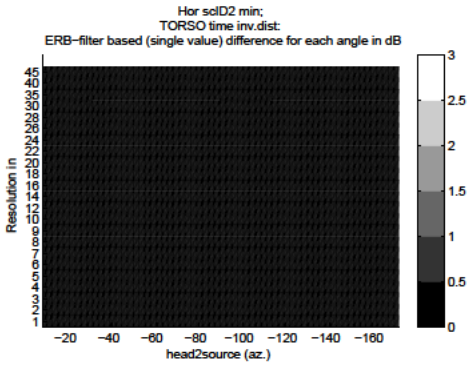
(c) zeitl. „via Kopf“, spline



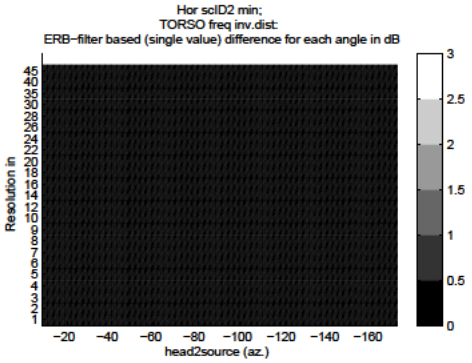
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

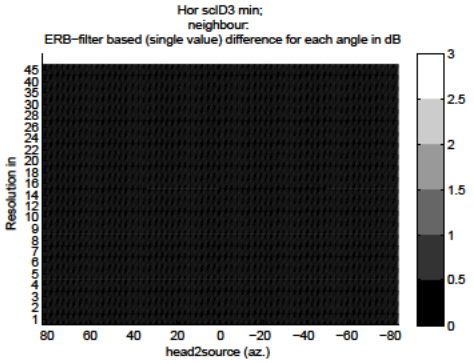


(f) zeitl. „via Torso“, linear

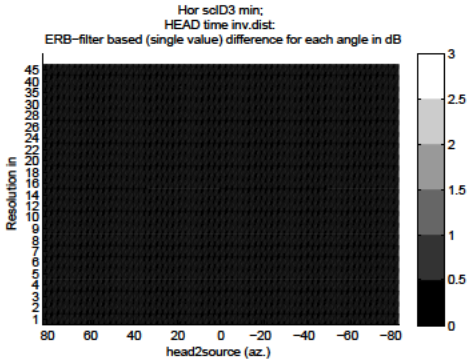


(g) spektral „via Torso“, linear

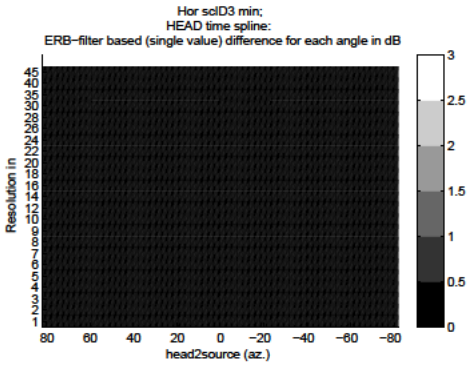
Abbildung A.7: ERB-Skalen basiertes Fehlermaß für Quelle 2, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



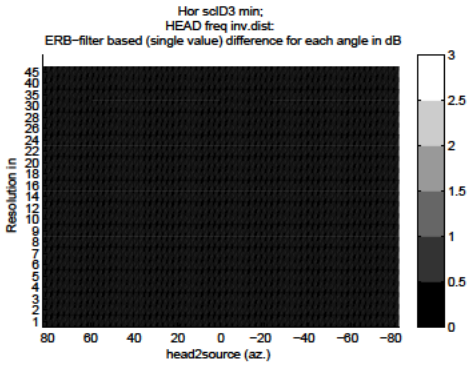
(a) „via Kopf“, nearest neighbour



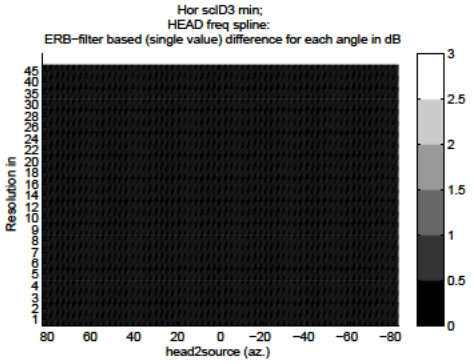
(b) zeitl. „via Kopf“, linear



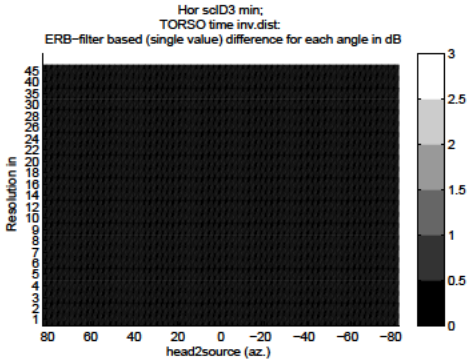
(c) zeitl. „via Kopf“, spline



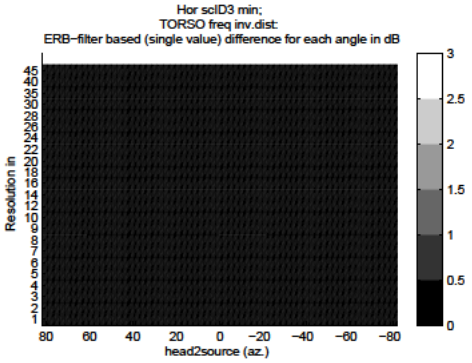
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

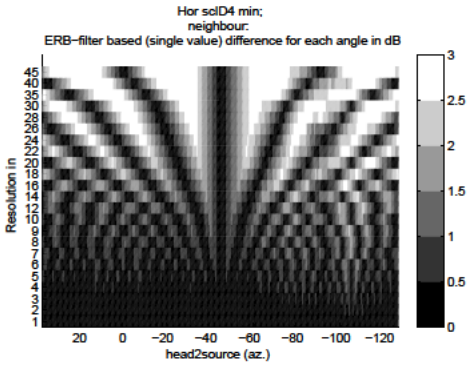


(f) zeitl. „via Torso“, linear

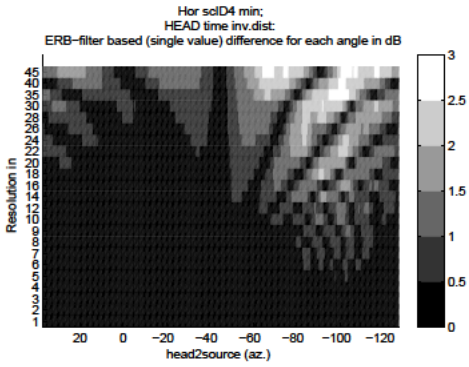


(g) spektral „via Torso“, linear

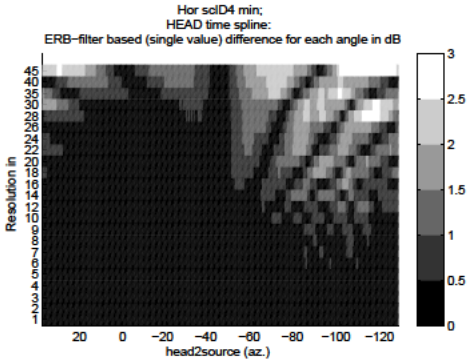
Abbildung A.8: ERB-Skalen basiertes Fehlermaß für Quelle 3, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



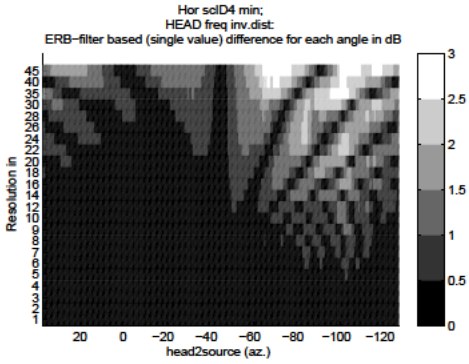
(a) „via Kopf“, nearest neighbour



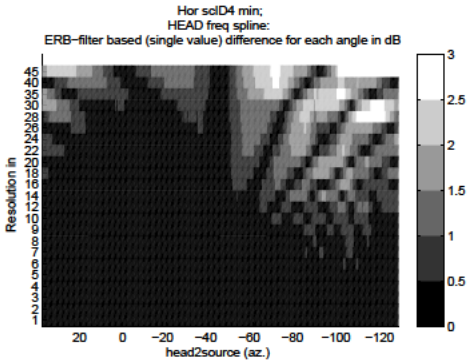
(b) zeitl. „via Kopf“, linear



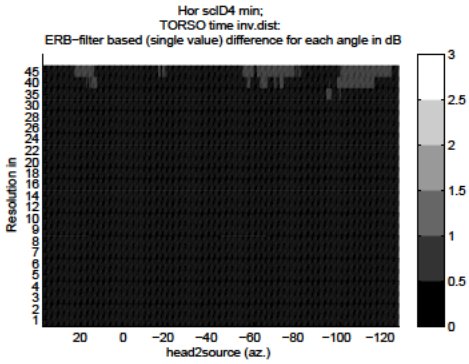
(c) zeitl. „via Kopf“, spline



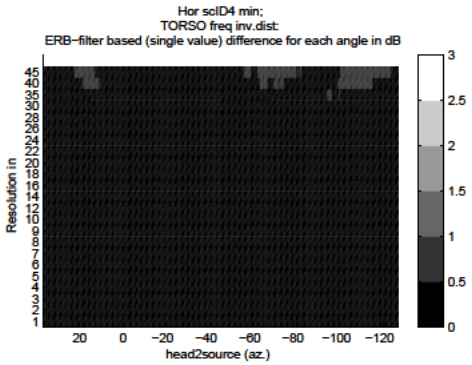
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

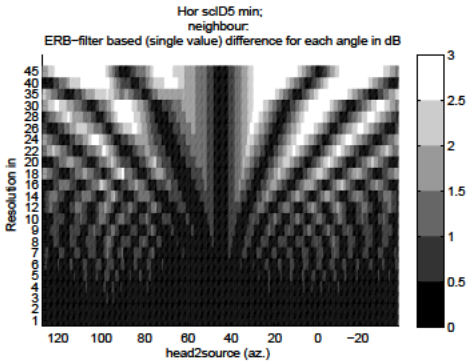


(f) zeitl. „via Torso“, linear

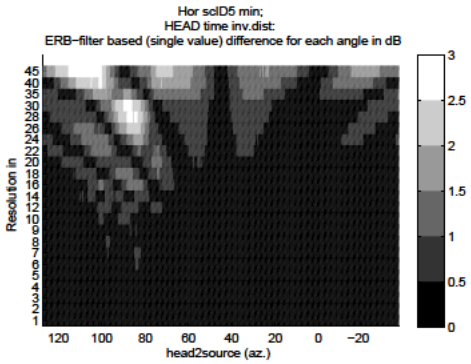


(g) spektral „via Torso“, linear

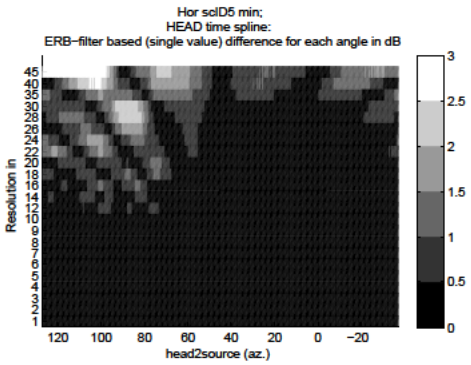
Abbildung A.9: ERB-Skalen basiertes Fehlermaß für Quelle 4, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



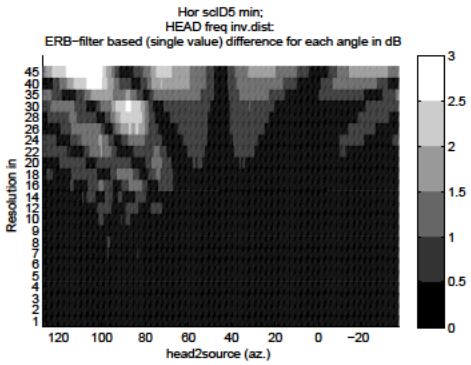
(a) „via Kopf“, nearest neighbour



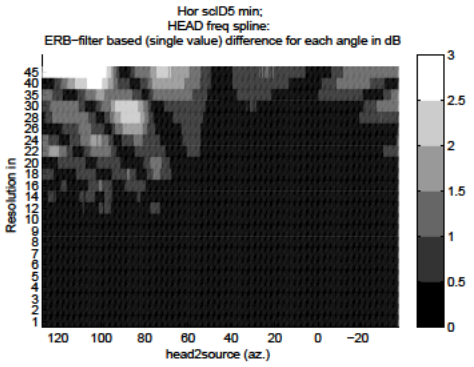
(b) zeitl. „via Kopf“, linear



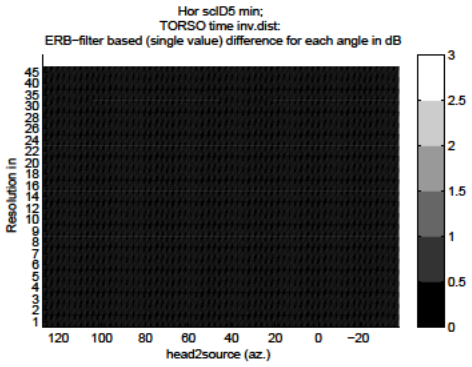
(c) zeitl. „via Kopf“, spline



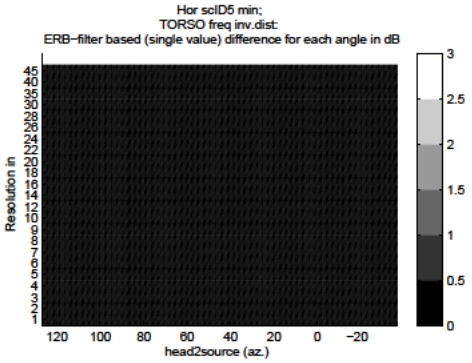
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

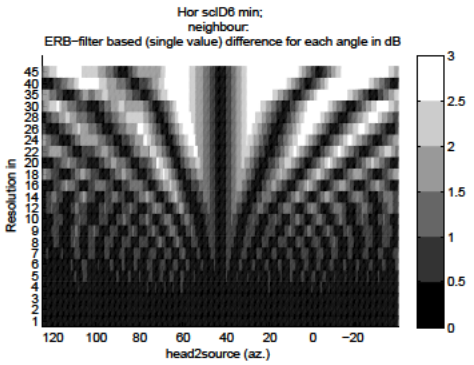


(f) zeitl. „via Torso“, linear

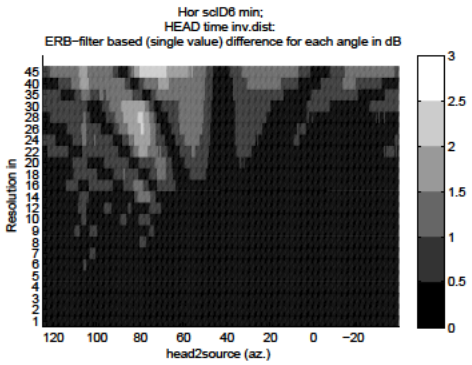


(g) spektral „via Torso“, linear

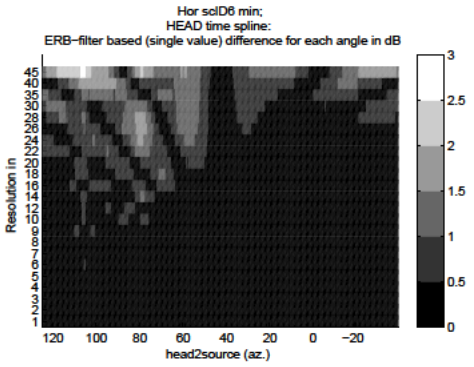
Abbildung A.10: ERB-Skalen basiertes Fehlermaß für Quelle 5, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



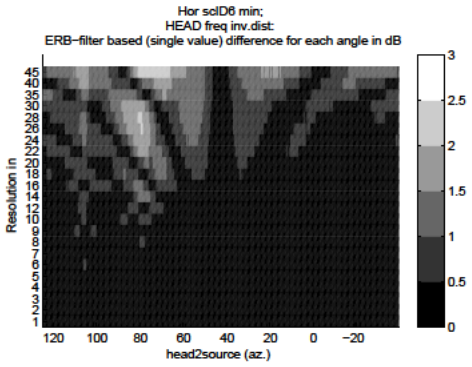
(a) „via Kopf“, nearest neighbour



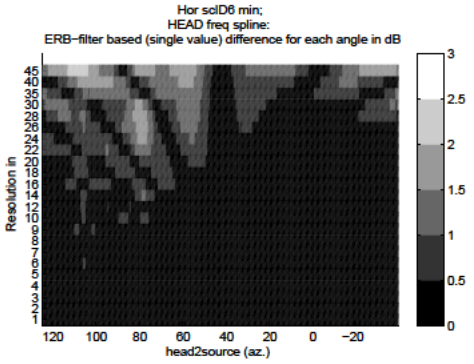
(b) zeitl. „via Kopf“, linear



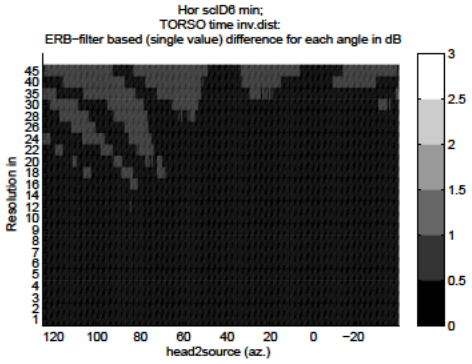
(c) zeitl. „via Kopf“, spline



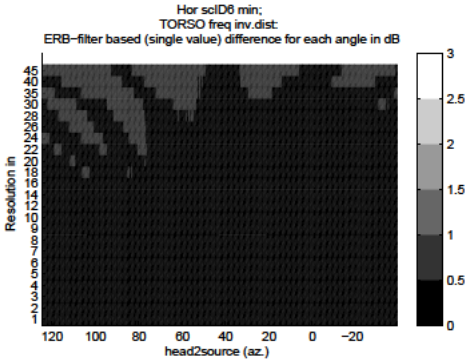
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

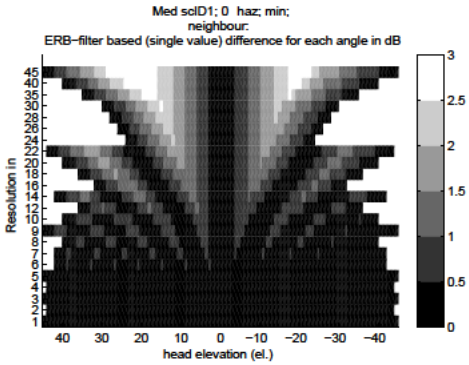


(f) zeitl. „via Torso“, linear

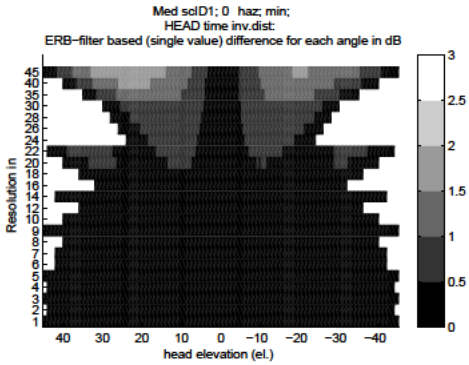


(g) spektral „via Torso“, linear

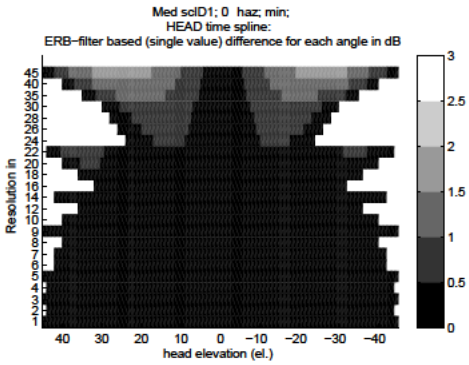
Abbildung A.11: ERB-Skalen basiertes Fehlermaß für Quelle 6, Horizontalebene a) bis g) zeigen Ergebnisse für die verschiedenen Algorithmen



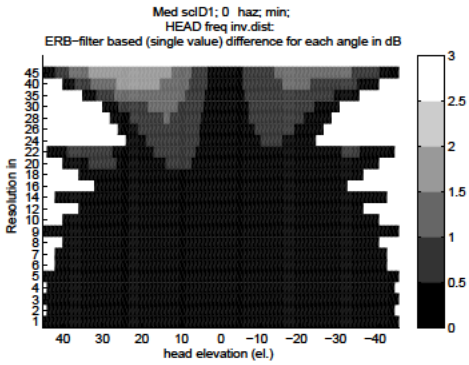
(a) „via Kopf“, nearest neighbour



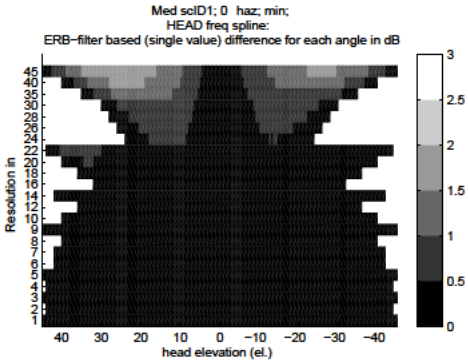
(b) zeitl. „via Kopf“, linear



(c) zeitl. „via Kopf“, spline



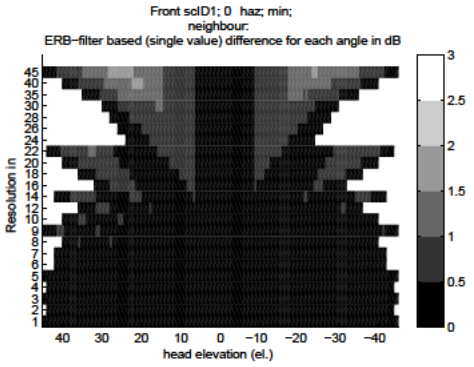
(d) spektral „via Kopf“, linear



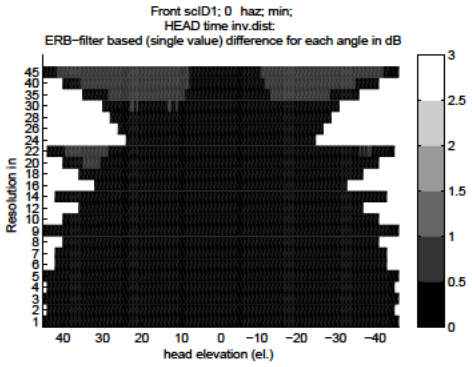
(e) spektral „via Kopf“, spline

Abbildung A.12: ERB-Skalen basiertes Fehlermaß für Quelle 1, Medianebene bei Kopfazimuth 0°

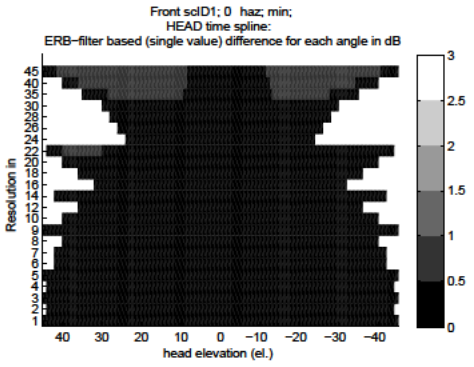
a) bis e) zeigen Ergebnisse für die verschiedenen Algorithmen



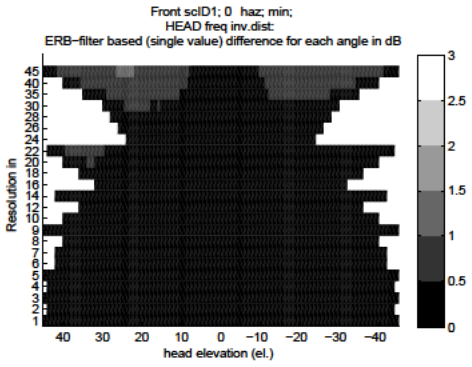
(a) „via Kopf“, nearest neighbour



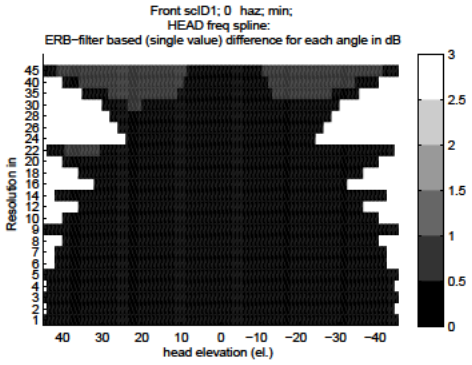
(b) zeitl. „via Kopf“, linear



(c) zeitl. „via Kopf“, spline



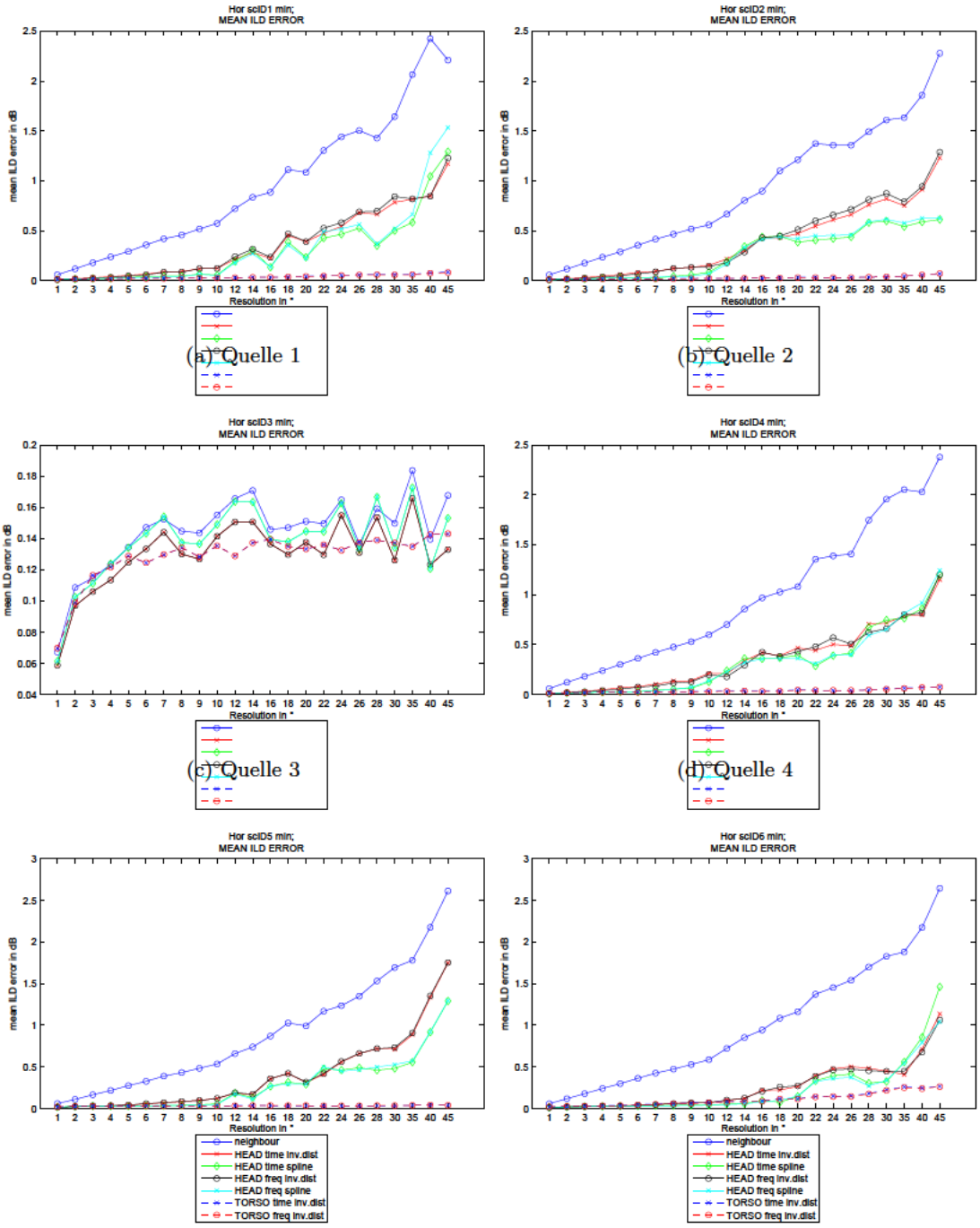
(d) spektral „via Kopf“, linear



(e) spektral „via Kopf“, spline

Abbildung A.13: ERB-Skalen basiertes Fehlermaß für Quelle 1, Frontalebene bei Kopfazimuth 0°

a) bis e) zeigen Ergebnisse für die verschiedenen Algorithmen



(a) Quelle 1

(b) Quelle 2

(c) Quelle 3

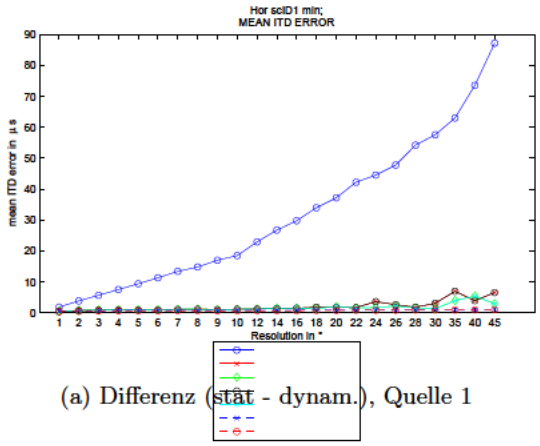
(d) Quelle 4

(e) Quelle 5

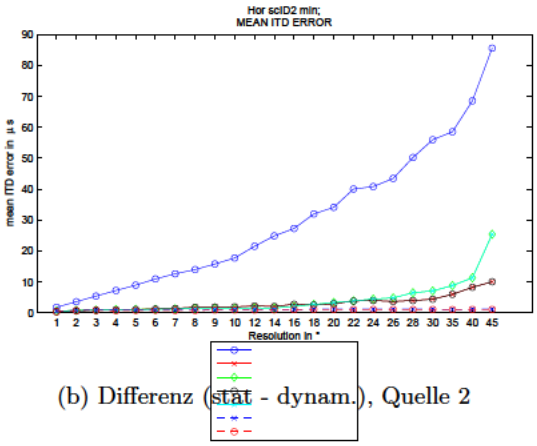
(f) Quelle 6

Abbildung A.14: mittlerer ILD-Abweichung in dB in Abhängigkeit des Interpolationsintervalls je Algorithmus, Horizontalebene

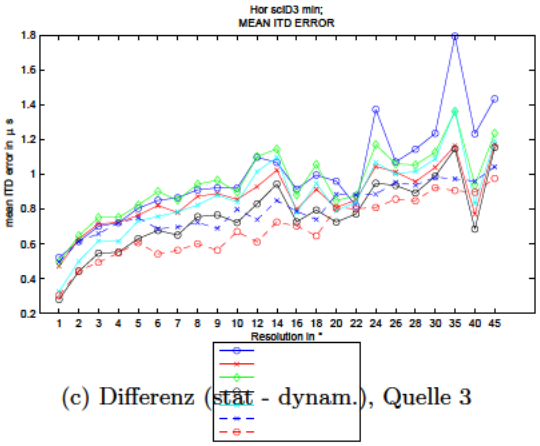
Legende in e) und f) gelten für die gesamte Gruppe



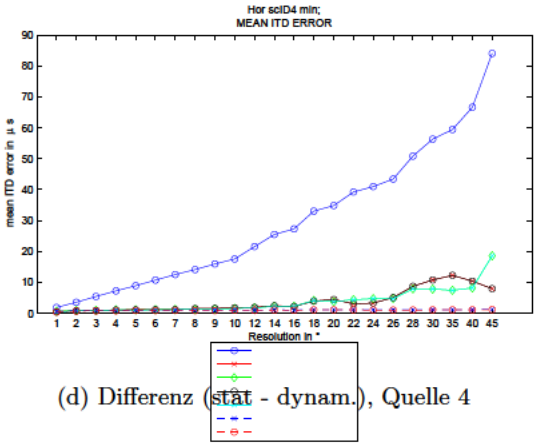
(a) Differenz (stat - dynam.), Quelle 1



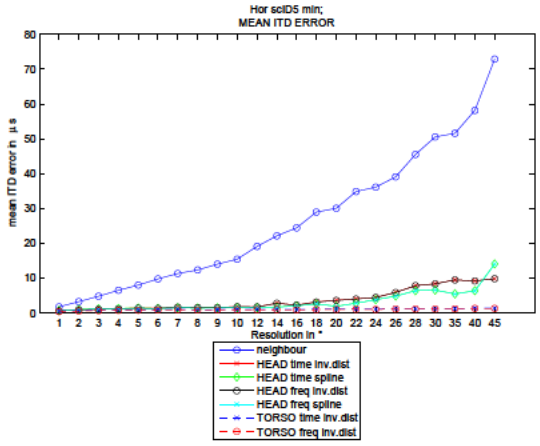
(b) Differenz (stat - dynam.), Quelle 2



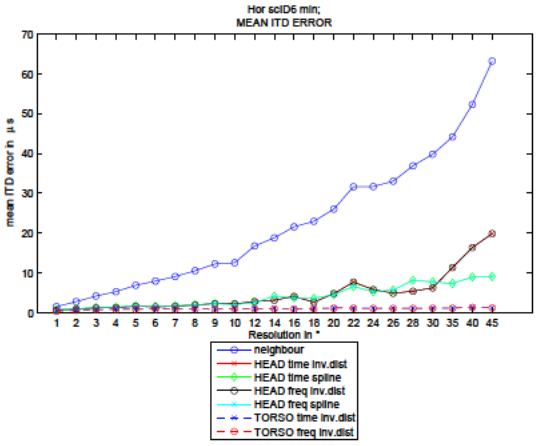
(c) Differenz (stat - dynam.), Quelle 3



(d) Differenz (stat - dynam.), Quelle 4



(e) Differenz (stat - dynam.), Quelle 5



(f) Differenz (stat - dynam.), Quelle 6

Abbildung A.15: mittlerer ITD-Abweichung in μs in Abhängigkeit des Interpolationsintervalls je Algorithmus, Horizontalebene

Legende in e) und f) gelten für die gesamte Gruppe

Anhang B

Unterlagen ABX-Test

Probandeninformation "Hörbarkeit des Torsoeinflusses in HRTF-Daten"

Die im April 2011 von der Deutschen Forschungsgemeinschaft eingerichtete Forschergruppe **„Simulation und Evaluation akustischer Umgebungen (SEACEN)"** befasst sich mit der Erforschung und Weiterentwicklung der gesamten Signalkette der virtuellen Akustik: Von der Simulation akustischer Umgebungen im Computermodell und der Aufnahme von realen Schallfeldern durch Mikrofonarrays über die Codierung räumlicher Schallsignale bis zur Wiedergabe durch Schallfeldsynthese oder Binauraltechnik. Insbesondere soll eine einheitliche Plattform und einheitliche Methoden der Qualitätsbeurteilung virtueller Umgebungen und der perzeptiven Validierung von Verbesserungen in der Signalverarbeitung entwickelt werden.
Die Forschergruppe wird koordiniert von Prof. Stefan Weinzierl (TU Berlin, Fachgebiet Audiokommunikation, Sprecher) und Prof. Michael Vorländer (RWTH Aachen, Institut für Technische Akustik).

Am Fachgebiet Audiokommunikation der Technischen Universität Berlin wird im Rahmen dessen ein Hörexperiment durchgeführt. Untersucht wird der Einfluss des Torsos auf den Schalldruck am Trommelfell und seine Hörbarkeit. Probanden werden dabei gebeten, bei einem einmaligen 1- bis 2-stündigen Termin zu versuchen, in vorgespielten Signalen einen Unterschied zu hören. Vor ab wird die Ruhehörschwelle des Teilnehmenden bestimmt, sowie ein kurzes Interview zur eigenen Einschätzung des Hörstatus geführt. Die Untersuchungen finden bei angenehmer Lautheit statt und sind gesundheitlich unbedenklich.

Bei wissenschaftlichen Untersuchungen werden persönliche Daten erhoben, die entsprechend strenger datenschutzrechtlicher Bestimmungen vertraulich behandelt werden - nur Versuchsleiter und Verantwortliche des Hörexperiments im Fachgebiet Audiokommunikation haben Zugang. Die erhobenen Daten werden in anonymisierter Form ausgewertet. Selbstverständlich ist die Teilnahme am Experiment freiwillig. Probanden können jederzeit Ihre Teilnahme ohne Angabe von Gründen abbrechen und die Einwilligung zurückziehen, ohne dass hierdurch Nachteile entstehen.

Jede Person, die keine Höreinschränkungen vorweist, sowie in Freizeit oder Beruf aktiv das Gehör einsetzt, ist zum Experiment eingeladen und willkommen.
Für eine Terminvereinbarung oder um weitere Informationen zum Projekt SEACEN und zum Ablauf des Hörexperiments zu erhalten, wenden Sie sich an uns:

TU Berlin * Fachgebiet Audiokommunikation * Reinhild Popko, Fabian Brinkmann, Alexander Lindau * Einsteinufer 17c * 10587 Berlin * Tel. 030 314 23137 * Fax 030 314 21143 * E-Mail: reinhild.popko@gmx.net * http://www.ak.tu-berlin.de

TU Berlin * Fachgebiet Audiokommunikation * Reinhild Popko, Fabian Brinkmann, Alexander Lindau * Einsteinufer 17c * 10587 Berlin * Tel. 030 314 23137 * Fax 030 314 21143 * E-Mail: reinhild.popko@gmx.net * http://www.ak.tu-berlin.de

Einverständniserklärung zur Teilnahme am Hörexperiment
Name des Probanden: _____
Die Probandeninformation zu der Untersuchung mit dem Titel „Hörbarkeit des Torsoeinflusses in HRTF-Daten“, die das Fachgebiet Audiokommunikation der TU-Berlin im Rahmen des Projekts SEACEN durchführt, habe ich gelesen.

Ich bin von Herrn / Frau _____ ausführlich und verständlich über Wesen und Bedeutung des Hörexperiments augeklärt worden und hatte ausreichend Zeit, Fragen zum Ablauf und zu möglichen Risiken zu stellen. Diese wurden zu meiner Zufriedenheit beantwortet. Mir ist bekannt, dass ich die Teilnahme jederzeit ohne Nachteile und ohne Angabe von Gründen abbrechen kann. Hiermit erkläre ich mein Einverständnis am beschriebenen Hörexperiment teilzunehmen.
Ich weiß, dass die bei den Untersuchungen mit mir gewonnenen Daten von Mitarbeitern des Hörexperiments im Fachgebiet Audiokommunikation mit Computern weiterverarbeitet und für wissenschaftliche, nicht-kommerzielle Zwecke verwendet werden sollen. Hiermit bin ich einverstanden, wenn die Auswertung und Veröffentlichung in einer Form erfolgt, die eine Zuordnung zu meiner Person ausschließt. Auch diese Einwilligung kann ich jederzeit ohne Angabe von Gründen widerrufen.
Eine Kopie der Einverständniserklärung habe ich erhalten.

Berlin, den _____ _____ _____
(Ort, Datum) (Unterschrift des Probanden) (Unterschrift des Versuchsleiters)

Einverständniserklärung zur Speicherung der Kontaktdaten
Ich weiß, dass meine persönlichen Daten und Erhebungsdaten zum Zweck eines wissenschaftlichen Folgeexperiments am Fachgebiet Audiokommunikation der TU-Berlin gespeichert werden. Ich bin damit einverstanden, dass meine Daten streng vertraulich aufbewahrt werden und ein Mitarbeiter des Fachgebiets Audiokommunikation zu einem späteren Zeitpunkt wieder Kontakt mit mir aufnimmt. Ich weiß ferner, dass ich diese Einwilligung zur Speicherung meiner Kontaktdaten jederzeit ohne Angabe von Gründen widerrufen kann.

Berlin, den _____ _____
(Ort, Datum) (Unterschrift des Probanden)

Probandenfragebogen

Teil A

Probandennummer: _____ Datum: _____
Muttersprache: ○ deutsch ○ andere
Alter:___
Geschlecht: ○ weiblich ○ männlich ○ anderes

Teil B

1. Waren Sie in den letzten 24 Stunden großem Lärm ausgesetzt?
○ Ja - Welcher Art? _____ Vor wievielen Stunden? _____
○ Nein

2. Haben Sie z. Zt. Schnupfen, der zu einer Hörminderung führt?
○ Ja, stark
○ Ja, etwas
○ Nein

3. Wie würden Sie Ihren Gesundheitszustand im Allgemeinen beschreiben?
○ ausgezeichnet
○ sehr gut
○ gut
○ weniger gut
○ schlecht

4. Haben Sie in einem oder in beiden Ohren Hörschwierigkeiten?
○ Nein
○ Rechts
○ Links
○ Weiß nicht

5. Haben Sie aktuell Ohrgeräusche wie Pfeifen, Zischen, Klingeln, Rauschen oder Ähnliches?
○ Ja
○ Nein
○ Weiß nicht

6. Haben Sie schon einmal an einem Hörexperiment teilgenommen?
○ Ja Wenn ja, wie oft: ____
○ Nein

7. Haben Sie beruflich oder privat aktiven Umgang mit Akustik und/oder Musik?
○Ja - Welcher Art?: _____ Wie häufig?: ____
○Nein

Teil C

8. Welche Merkmale haben Ihnen geholfen Unterschiede zu hören
bzw. worauf haben Sie besonders geachtet?
(Bitte ordnen Sie die Merkmale in ihrer Relevanz - von stark zu schwach)

Untersuchungsprotokoll HÖRBARKEIT - TORSOEINFLUSS

Probandennummer: _____

○ Einverständniserklärung an Studienteilnehmer ausgehändigt

○ subjektive Einschätzung:

○ sehr ruhig
○ mittel
○ starke Nebengeräusche wie:_____

○ Teil A, B des Interviews geführt

○ Tonaudiometrie LL

○ ABX-Vergleich ○ Tracker „Nullen“

○ Teil C des Interviews geführt

_____
Datum, Untersucher/in

Verfahrensabweichungen und Besonderheiten:

_____
_____
_____
_____
_____
_____
_____
_____

Anhang C

Tabellenanhang

Ergebnisse der Tonaudiometrie

Tabelle C.1: Ruhehörschwellen der Luftleitung nach Békèsy-Festfrequenzaudiometrie

	125 Hz	250 Hz	500 Hz	750 Hz	1 kHz	1.5 kHz	2 kHz	3 kHz	4 kHz	6 kHz	8 kHz
Horschwellen [db HL]											
VP 1											
rechts	-0.3	-3	-5.3	-4.5	-5.4	1.5	-2.4	-1.2	-6.4	8.5	-0.5
links	1.1	-0.2	-2.4	-5.7	-6.7	-1.2	-5.6	-3.0	5.5	4.8	0.3
Proband 2											
rechts	-1.2	-0.8	-2.2	-7.5	-9.2	-3.3	3.3	-7.9	-16.9	-4.3	1.5
links	3.5	1.5	0.6	-3.5	-7.9	-7.4	-3.9	-9.4	-11.4	-0.8	-5.8
VP 3											
rechts	1.9	0.8	2.7	7.4	10.8	7.2	4.2	8.1	4.6	3.5	4.2
links	12.3	6.5	3.0	2.0	1.8	12.6	12.9	3.7	0.8	10.7	8.5
VP 4											
rechts	0.7	-1.7	2.4	1.9	8.6	1.1	-1.4	-1.2	-2	16.9	7.9
links	1.8	0.3	2.6	-0.8	-1.5	1.9	2.8	5.4	1.4	-0.0	11.2
VP 5											
rechts	7.7	1.5	6.4	1.6	-3.7	-0.4	8.9	1.2	-8.4	1.7	-4.3
links	6.2	0.5	3.3	0.1	0.7	0.6	8.7	-0.7	-8.8	-5.9	1.4
VP 6											
rechts	1.7	4.6	1.9	-1.9	-6.6	-7.2	-4.5	0.9	-1.2	7.8	-2.7
links	4.5	5.4	3.2	-0.5	-5.0	-0.3	1.4	0.1	2.4	22.3	3.7
VP 7											
rechts	2.2	2.1	4.2	-1.3	-3.5	-6.5	5.4	-0.9	1.4	8.5	13.5
links	4.8	4.1	2.7	1.7	-2.3	-2.9	1.2	-5.0	1.2	15.4	13.0
VP 8											
rechts	-2.3	-3.9	3.5	-3.3	-6.5	8.1	1.9	3.1	12.1	24.7	31.3
links	2.2	-4.5	-3.1	-2.7	-4.2	-5.6	1.7	4.7	8.2	16.3	26.9
VP 9											
rechts	2.9	0.0	5.0	4.7	-0.5	-1.5	1.2	-6.2	-4.3	1.4	0.6
links	6.0	2.8	2.3	1.9	-1.0	-1.8	1.8	-2.0	2.4	0.3	-7.7
VP 10											
rechts	-1.0	-2.8	-0.2	-4.0	-7.9	-4.4	0.5	-0.9	-9.1	0.8	-1.8
links	3.1	-1.6	-0.7	-3.2	-4.9	-1.9	5.5	7.1	8.0	14.0	2.5
VP 11											
rechts	0.9	-0.8	2.0	-4.6	-7.1	-6.7	-2.3	-4.5	-4.3	2.2	-4.9
links	5.3	-0.9	-5.2	-3.9	-5.6	−7.2	-1.1	3.7	-2.7	6.9	2.7
VP=Versuchsperson											

Erhobene Angaben zum Probandenfragebogen

Tabelle C.2: Antworten der Probanden

Muttersprache	**deutsch** 9	**andere** 2			
Geschlecht	**männlich** 7	**weiblich** 4			
Alter	26; 27(3x); 28; 29(2x); 32(2x); 33; 36;				
1) Lärmexposition der letzten 24 h	**nein** 9	**ja** 2	**Art** Musik (in-ear-Kopfhorer)	**vor wie vielen stunden** 3h; 19h	
2) Schnupfen	**ja, stark** 0	**ja, etwas** 0	**nein** 11		
3) aktueller Gesundheitszustand	**ausgezeichnet** 2	**sehr gut** 6	**gut** 2	**weniger gut** 1	**schlecht** 0
4) Hörschwierigkeiten	**rechts** 0	**links** 0	**weiß nicht** 0	**nein** 11	
5) Ohrgeräusche	**ja** 1	**nein** 10	**weiß nicht** 0		
6) Teilnahme an Hörexperimenten	**nein** 2	**ja** 9	**Quantitätsangaben, wenn ja** 4; 5(2x); 8; 9; 10(2x); 12; >10		
7) aktiver Umgang mit Musik/Akustik	**nein** 0	**ja** 11	**Art des Umgangs** Student/in der Audiokommunikation; Chor-, Orchesterleiter/in; Musikpädagoge/in; semiprof. Kammermusiker/in; DJ; priv. Musikproduzent/in; Musikwissenschaftler/in; Audiotechniker/in; Akustikingenieur/in (5x), Mitarbeiter/in Audiokommunikation + -technologie(2x), Hobbymusiker/in (6x), begeisterter Konzertgänger (Klassik, Jazz);	**Quantitatsangaben, wenn ja** 1h/Wo; 0.5h/Tag; 4h/Wo; 6h/Wo; 1h/Tag(3x); 10h/Wo; 1.5h/Tag; 12h/Wo; 3h/Tag;	
8) genannte Merkmale, Adjektive	**bei Rauschen** spektrale Merkmale (2x), Kammfilter höherer Bereich, Betonung einzelner Frequenzbänder, Höhen-, Tiefenunterschiede; Klangfarbe(7x), Höhen(2x), Lokalisation, Amplitude bei Bewegung	**bei Sprache** Nasalität(2x), bandbegrenzt, nicht feststellbar(2x), spektrale Merkmale, „Raumklang“ (Lokalisation), Klangfarbe(4x), Phonemqualität, Lokalisation(5x), Quellbreite; Entfernung, bewegungsbasierte Quelleigenschaften, „voller vs. dünner Klang“			

Anhang D

Quellcode Matlab

i HRTF Processing

a path_definition_and_setup

```
% startup script that defines the paths for
% - raw data
% - saved data
% - needed m-files
% - hrtf grid
% - exitation signal
% - directory for plots
% modifiy processing steps and parameters
%
% and loads/initialises some data
% - some variables (fs, sweep)
%
% R. Popko, F. Brinkmann, TU Berlin, Audio Communication Group, 04/2013


% --- pathes ---
plane={'Hor'};% {'Front'} {'Med'}
planeID=1;
sources=[0 0; 90 0; 0 90; 45 0; -45 30; -42.8 -30];
%scID 5,6 for Med, Front:[-45.2 29.8;-41.5 -30] ;scID 5,6 for Hor: [-45 30; -42.8 -30]


sub = {'scID1','scID2','scID3','scID4','scID5','scID6'};


% for scID=1:6
%     s=sub{scID};
%     addpath(fullfile('C:\Dokumente und Einstellungen\ibm\Desktop\AP 2 HRTF↙
Datenbank\3 Raw data\Reference', s));
%     addpath(fullfile('C:\Dokumente und Einstellungen\ibm\Desktop\AP 2 HRTF↙
Datenbank\3 Raw data\data', plane{1}, s));
% end

% ---- root ----
dir.root = 'C:\Dokumente und Einstellungen\ibm\Desktop\AP 2 HRTF Datenbank';

% ---- raw hrtf data ----
dir.hrtf_raw = fullfile(dir.root, '3 Raw Data', 'data');
dir.plane=plane;
dir.planeID=planeID;
dir.sub =sub;
dir.N   = length(dir.sub);

% ---- reference  and sub measurements ----
dir.ref =  fullfile(dir.root,'3 Raw Data', 'Reference');

% ---- saving data ----
dir.hrtf_deconv = fullfile(dir.root, '5 Processed Data', '1 deconv raw');
% root dir for saving deconvolved and initially shorted HRIRs

dir.hrtf_final  = fullfile(dir.root, '5 Processed Data', '2 HRIR final');
% root dir for saving final HRIRs

dir.save_data   = fullfile(dir.root, '5 Processed Data', '0 Processing',...
    'Tools and Data', 'Data for processing');
```

```
% root dir for saving mat files that are needed for further processing

%%
% ---- sweep and grid ----
dir.sweep=fullfile(dir.root,'5 Processed Data','0 Processing',...
    'Tools and Data','Data for processing',...
    'SEACEN_HATO_interpolation_sweep.wav');
dir.grid =fullfile(dir.root,'5 Processed Data','0 Processing',...
    'Tools and Data', 'Data for processing', [dir.plane{1} '_grid.mat']);

load(dir.grid);
% ---- m-files needed for processing ----
dir.m_files=fullfile(dir.root,'5 Processed Data','0 Processing',...
    'Tools and Data');
addpath(genpath(dir.m_files), 0) % add to top of matlab search paths

% ---- include ita toolbox in matlab workspace ----
% (if you already have the toolbox in your workspace, comment next line)
ita_toolbox_setup
clc


% ---- plots for documenting the post-processing
dir.plots = fullfile(dir.root, '5 Processed Data', '0 Processing','Plots');

% ---- variables ----
setup.fs        = 44100;              % sampling frequency in Hz
setup.sweep     = wavread(dir.sweep); % sweep used for measurements
setup.NFFT      = 16;                 % sweep order used during measurement
setup.latency   = 1135;               % latency of soundcard in samples
setup.pre_shift = 48;   % circ shift applied to HRIRs after deconvolution
% to make sure that first peaks in HRIRs do not
% occur at negative times (at end of imp. response)
% About 8 samples = 6.2cm = half the distance
% between ears should be enough, but 20 samples are
% added for safety, further 20 samples for having
% time to fade-in in post_processing
setup.sc=sources;
setup.scID      = 1:6;
setup.plane     = dir.plane;
setup.grid=load(dir.grid); % grids used during measurements

if planeID==1;% {'Hor'}
    setup.grid_N    = length(Hor_grid(:,1)); % number of points in grids
elseif planeID==2;%{'Med'}
    setup.grid_N    = length(Med_grid(:,1));
elseif planeID==3;%{'Front'}
    setup.grid_N    = length(Front_grid(:,1));
else
    error('planeID could be 1 for Hor., 2 for Med. or 3 for Front. plane');
end
setup.N_raw     = 2^11;               % Length in samples HRIRs are shortened to after↙
deconvolution
setup.N         = 425;                % Final length of HRIRs in samples
setup.N_fade    = 15;                 % length of fade-in and fade-out in samples
```

```
% ---- processing ----
do.plot  = true;   % show and save plots

% select variable part of processing
do.itd                   = true;    % calculates itd and saves it as meta data entry
do.mic_setup_directivity = false;    % compensates the directivity of the microphone↙
setup
do.frac_delay            = true;    % applies fractional delay to impulse response to↙
compensate temperature fluctuations
do.low_freq_manipulation = false;    % First 2 frequency bins are manipulated to match↙
energy of neighboring bin
do.low_freq_mode         = 1;       % 1: Set to 0 dB 2: set according to different↙
distance of reference point and entrance to blocked ear canals
do.org_phase             = false;    % save HRIRs without phase manipulation
do.min_phase             = true;    % save min-phase HRIRs (ITD saved as metadata)

% specifications for saving
data.description = 'HRTF Dataset (Popko, Brinkmann (2013): to be used for interpolation↙
of HRTF-data)';
data.head        = 'FABIAN (Lindau et al. (2007): Binaural resynthesis for comparative↙
studies of acoustical environments. In: 122nd AES Convention, Conference Paper 7032.↙
Vienna, Austria.)';
data.microphone  = 'DPA 4060';
data.source      = 'GENELEC 8030 A';
data.room        = 'Anechoic chamber of Institute of Fluid Dynamics and Technical↙
Acustics (ISTA), Technische Universitaet Berlin, Germany';

clearvars p pl s scID Hor_grid Med_grid Front_grid
```

b pre_processing

```
% processing steps
% 1: calculate average temperature of all measurements (needed for
%    fractional delaying the impulse responses to compensate temperature
%    variance)
% 2: calculate high-pass and low-pass for deconvolution
%
% R. Popko, F. Brinkmann, TU Berlin, Audio Communication Group, 04/2013
close all; clear; clc

% run path_def script to obtain data paths
a_path_definition_and_setup

%% 1 ------------------------------------------------- average temperature
% read and average temperature values from all measurements
temp.data = [];
temp.time ={};
for scID=setup.scID
    % temp. while HRIR measurement

    load(fullfile(dir.hrtf_raw, dir.plane{1},dir.sub{scID}, 'Temperatur.mat'),↙
'TempVec', 'TimeVec');
    temp.data = [temp.data TempVec];
    temp.time = [temp.time TimeVec];

    % temp. while Ref measurement
    for n=1:2
        side={'left', 'right'};
        file=fullfile(dir.ref, dir.sub{scID}, [dir.plane{1} 'Ref_' side{n} '_scID'↙
num2str(scID) '.mat']);
        load(file, 'Temperature', 'rec_time');
        temp.data = [temp.data Temperature];
        temp.time = [temp.time rec_time];
    end
end
% delete doubled entries
[~,hold_ind]=unique(temp.time);
temp.time=temp.time(sort(hold_ind));
temp.data=temp.data(sort(hold_ind));

% sort temperature values for plotting purposes
tmp = sort(temp.data);
% mid-range temperature is defined as target
temp.target = median(tmp);%min(temp.data)+(max(temp.data)-min(temp.data))/2;
temp.target = floor(temp.target*10)/10;

% plot histogram
if do.plot
    myFigure(20,10)
    hist(tmp, min(tmp):.1:max(tmp))
    xlabel('Temperature in Celius')
    ylabel('Absolute frequency')
    title('Frequency distribution over all measurements')
    grid on
    print('-dpdf', '-r300', fullfile(dir.plots, ['b 1 ' dir.plane{1} ' Temperature↙
distribution']))
```

```
end

save(fullfile(dir.save_data,[dir.plane{1} 'temp']), 'temp');

clearvars p scID n file TempVec Temperature side


%% 2 ------------------------ calculate high and low pass for deconvolution
%   based on reference measurement for frontal sound incidence using deconv

% plot spectrum of reference measurement for bandwidth estimation
scID=1;
load(fullfile(dir.ref, dir.sub{scID}, [dir.plane{1} 'Ref_left_scID' num2str(scID) '.↙
mat']), 'rec_sweep')

myFigure(20,10)
p(rec_sweep(:,1), 's', setup.fs, 0,10,setup.fs/2,-40,40);
title([dir.plane{1} 'plane: speaker bandwith while considering one ref. measurement'])

if do.plot
    print('-dpdf', '-r300', fullfile(dir.plots, ['b 2a ' dir.plane{1} ' bandwidth↙
speaker']))
end

% filterdesign with sufficent low freq noise supresssion and little group
% delay distortion (if possible)
[~, h_hp, h_lp] = deconv(rec_sweep, setup.sweep, setup.fs, 'hp_param', [20 70 60 .1],↙
'plot_filter', 1);% hp_param: Stopband freq./Passband freq./Stopband attenuation in↙
dB/Passband tolerance in dB
if do.plot
    print('-dpdf', '-r300', fullfile(dir.plots, ['b 2b ' dir.plane{1} ' filter for↙
deconvolution']))
end
% save
save(fullfile(dir.save_data,[dir.plane{1} 'deconv_filter']), 'h_hp', 'h_lp')

clearvars rec_sweep ans
```

c deconvolution

```
% deconvolution and initial truncation to setup.N_raw samples
%
% R. Popko, F. Brinkmann, TU Berlin, Audio Communication Group, 04/2013
close all; clear; clc

%% run path_def script to obtain data paths
a_path_definition_and_setup

%% make directories for saving deconvolved HRTFs
for n = 1:dir.N
    dir_write = fullfile(dir.hrtf_deconv, dir.plane{1}, dir.sub{n});
    if ~exist(dir_write, 'dir')
        mkdir(dir_write)
    end
end

for n = 1:dir.N
    dir_write = fullfile(dir.hrtf_deconv, 'Reference');
    if ~exist(dir_write, 'dir')
        mkdir(dir_write)
    end
end

%% open txt file
fid = fopen(fullfile(dir.hrtf_deconv, ['1 ' dir.plane{1} ' log.txt']), 'w');
fprintf(fid, 'Log file for HRTF measurements\n');
fprintf(fid, '%s\n\n', datestr(clock));
fprintf(fid, 'Missing data:\n(dataset is complete if nothing is listed below)\n');

%% ------------------------------------------------------------ deconvolution
% load high and low pass
load(fullfile(dir.save_data, [dir.plane{1} 'deconv_filter']));

%% deconv data
ref_lr=[];
display(['starting deconv data of ' dir.plane{1} '. plane'])
load(dir.grid, [dir.plane{1} '_grid']);
if dir.planeID==1
    act_grid=Hor_grid;
elseif dir.planeID==2
    act_grid=Med_grid;
else dir.planeID==3
    act_grid=Front_grid;
end

act_plane=dir.plane{1};
for s=3:dir.N % for all sources
    display(['source ' num2str(s)])
    % load ref_meas. per source
    for n=1:2
        side={'left', 'right'};
        file=fullfile(dir.ref, dir.sub{s}, [dir.plane{1} 'Ref_' side{n} '_scID' num2str↙
(s) '.mat']);
        % check and load data
        if exist(file, 'file')
```

```
            load(file, 'rec_sweep');
        else
            fprintf(fid, '%s\tis missing.\n', filename);
        end
        ref_lr(:,n)=rec_sweep;
    end
    %deconv ref
    rir = deconv(ref_lr, setup.sweep, setup.fs, 'hp_type', h_hp, 'lp_type', h_lp,↙
'h_trunc', setup.N_raw, 'do_circshift', -setup.latency);
    % write deconvolved rir
    filename = [dir.plane{1} 'Ref_scID' num2str(s) '.mat'];
    save(fullfile(dir.hrtf_deconv, 'Reference', ['rir_' filename]), 'rir');


    for n=1:length(act_grid(:,1)) % for all files per source
        if ~mod(n, 100)
            disp([num2str(n) ' of ' num2str(length(act_grid(:,1)))]);
        end
        if dir.planeID==1 %read filename
            filename = [act_plane '_t' num2str(act_grid(n,1)) '_haz' num2str(act_grid↙
(n,2)) '_hel0_scID' num2str(s) '.mat'];
        else
            filename = [act_plane '_t0_haz' num2str(act_grid(n,2)) '_hel' num2str↙
(act_grid(n,3)) '_scID' num2str(s) '.mat'];
        end

        % check and load data
        if exist(fullfile(dir.hrtf_raw, dir.plane{1}, dir.sub{s}, filename), 'file')
            load(fullfile(dir.hrtf_raw, dir.plane{1}, dir.sub{s}, filename),↙
'rec_sweep');
            data_lr = rec_sweep;
        else
            fprintf(fid, '%s\tis missing.\n', filename);
        end
        % deconvolution
        hrir = deconv(data_lr, ref_lr, setup.fs, 'hp_type', h_hp, 'lp_type', h_lp,↙
'h_trunc', setup.N_raw, 'do_circshift', setup.pre_shift);
        % write deconvolved hrir
        save(fullfile(dir.hrtf_deconv, dir.plane{1}, dir.sub{s}, ['hrir_' filename]),↙
'hrir')

    end

    %%
    if do.plot
        run(fullfile(dir.plots, 'c1 deconvolution exemplary control plots',↙
'c1_plots'))
    end
end
clearvars -except setup do dir
```

d radius_estimation

```
% Estimation of radius by TOA detection (onset based)
%
% R. Popko, F.Brinkmann, Audio Communication Group, TU Berlin 04/2013,
close all; clear; clc

% run path_def script to obtain data paths
a_path_definition_and_setup

%% ------------------------------------------------ radius estimation
% allocate memory
radius = zeros(length(setup.scID), 2);

%load temperature values
temp.data=zeros(length(setup.scID),2);
for scID=setup.scID % source
    for n=1:2
        side={'left', 'right'};
        file=fullfile(dir.ref, dir.sub{scID}, [dir.plane{1} 'Ref_' side{n} '_scID'↙
num2str(scID) '.mat']);
        load(file, 'Temperature');
        temp.data(scID,n) = Temperature;
    end
end

for scID=setup.scID % source
    load(fullfile(dir.hrtf_deconv, 'Reference', ['rir_' dir.plane{1} 'Ref_scID' num2str↙
(scID) '.mat']), 'rir');

    % get temperature
    temperature = temp.data(scID,:);

    % initial average TOA in samples
    TOA = onset_detect(rir(1:400,:), 10, -3);
    % TOA = mean(TOA);

    % radius calculated based on correct temperature
    radius(scID,:) = TOA./setup.fs.*speed_of_sound(temperature);

end

save(fullfile(dir.save_data, [dir.plane{1} 'radius']), 'radius')
%%
if do.plot
    run(fullfile(dir.plots, 'c2_radius_plots'))
end

clearvars -except setup do dir
```

e post_processing

```
% Main script for post-processing the HRTFs
%
% PROCESSING STEPS (for detailed description and plots see documentation)
% 0.  Construct window based on setup.N and setup.N_fade
% 1.  Load radii and temperature data from reference measurement (Needed for frac.↙
delay)
% 2.  ITD estimation
% 3.  Fractional delay to compensate temperature fluctuation during
%     measurements
% 4.  Truncation to setup.N samples
% 5.  Window
% 6.  Min phase generation
% 7.  Collect data for saving
% 8.  Saving
%
% R. Popko, F.Brinkmann, Audio Communication Group, TU Berlin 04/2013

close all; clear; clc

% run path_def script to obtain data paths
a_path_definition_and_setup
%% ------------------------------------------------------------------- %%
% ---------------------------- 0. construnct window to be applied to HRIRs
hrir_window = ones(setup.N,1);
if setup.N_fade
    hrir_window(1:setup.N_fade+1)      = sin(linspace(0,pi/2, setup.N_fade+1)).^2;
    hrir_window(end-setup.N_fade:end) = cos(linspace(0,pi/2, setup.N_fade+1)).^2;
end

%% ---------------------- 1. read temperature values from reference data-set

if do.frac_delay
    %load temperature values
    temp_ref=zeros(length(setup.scID),2);
    for scID=setup.scID % source
        for n=1:2
            side={'left', 'right'};
            file=fullfile(dir.ref, dir.sub{scID}, [dir.plane{1} 'Ref_' side{n} '_scID'↙
num2str(scID) '.mat']);
            load(file, 'Temperature');
            temp_ref(scID,n) = Temperature;
        end
    end
    temp_ref= mean(temp_ref,2);

    % get target temperature and radii
    load(fullfile(dir.save_data, [dir.plane{1} 'temp']));
    load(fullfile(dir.save_data, [dir.plane{1} 'radius']));
    mean_radius=mean(radius,2);
    c_target = speed_of_sound(temp.target);
end
% loop over data directories
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
act_plane=dir.plane{1};
load(dir.grid, [dir.plane{1} '_grid']);
```

```
if dir.planeID==1
    act_grid=Hor_grid;
elseif dir.planeID==2
    act_grid=Med_grid;
else dir.planeID==3
    act_grid=Front_grid;
end

for scID=setup.scID
    display(['plane ' dir.plane{1} ' source ' num2str(scID)])

    if do.org_phase %prepare directories for saving
        save_in_org=fullfile(dir.hrtf_final, 'original phase', dir.plane{1}, dir.sub↙
{scID});
        if ~exist(fullfile(dir.hrtf_final, 'original phase', dir.plane{1}, dir.sub↙
{scID}), 'dir');
            mkdir(fullfile(dir.hrtf_final, 'original phase', dir.plane{1}, dir.sub↙
{scID}));
        end
    end
    if do.min_phase
        save_in_min=fullfile(dir.hrtf_final, 'minimum phase', dir.plane{1}, dir.sub↙
{scID});
        if ~exist(fullfile(dir.hrtf_final, 'minimum phase', dir.plane{1}, dir.sub↙
{scID}), 'dir');
            mkdir(fullfile(dir.hrtf_final, 'minimum phase', dir.plane{1}, dir.sub↙
{scID}));
        end
    end

    % loop over grid points
    for n = 1:length(act_grid(:,1)) % for all files per source
        display([num2str(n/length(act_grid)*100) '% processed of set: plane ' dir.plane↙
{1} ' source ' num2str(scID)] )

        % ---- process data ----
        % allocate meomory
        hrir_org_l = zeros(setup.N,1);
        hrir_org_r = zeros(setup.N,1);
        hrir_min_l = zeros(setup.N,1);
        hrir_min_r = zeros(setup.N,1);
        itd = zeros(1, 1);
        toa = zeros(1, 2);
        %low_freq_level = zeros(setup.grid_N, 2);

        % read deconvolved hrir
        if dir.planeID==1 %read filename
            filename = [act_plane '_t' num2str(act_grid(n,1)) '_haz' num2str(act_grid↙
(n,2)) '_hel0_scID' num2str(scID) '.mat'];
        else
            filename = [act_plane '_t0_haz' num2str(act_grid(n,2)) '_hel' num2str↙
(act_grid(n,3)) '_scID' num2str(scID) '.mat'];
        end
        load(fullfile(dir.hrtf_deconv, dir.plane{1}, dir.sub{scID}, [ 'hrir_'↙
filename]), 'hrir');
```

```
        % ---------------------------------------------- 2. itd estimation
        if do.min_phase || do.itd
            toa(1,:) = onset_detect(hrir(1:100,:), 10, -6);
            itd    = (toa(1,1)-toa(1,2))/setup.fs*10^6;
        end

        % ---------------------------------------------- 3. fractional delay
        if do.frac_delay
            % read coressponding timestamp
            load(fullfile(dir.hrtf_raw, dir.plane{1}, dir.sub{scID}, filename),↙
'rec_time');
            % get temperature closest to measurement time
            [~, id] = min(abs( datenum(rec_time)-datenum(temp.time) ));
            temp_hrir = temp.data(id);

            % get fractional delay in samples
            c_ref = speed_of_sound(temp_ref(scID));
            frac_ref = (1/c_target - 1/c_ref) * mean_radius(scID) * setup.fs;

            c_hrir = speed_of_sound(temp_hrir);
            frac_hrir = (1/c_target - 1/c_hrir) * mean_radius(scID) * setup.fs;

            frac = frac_ref - frac_hrir;

            % apply fractional delay
            if frac
                hrir = fractional_delay(hrir, frac, 'N', 70, 'kaiser_A', 80,↙
'keep_length', 1);
            end
        end

        % ------------------------------------------------------ 4. truncate
        hrir = hrir(1:setup.N,:);
        % ------------------------------------------------------- 5. window
        hrir = hrir .* [hrir_window hrir_window];
        % ---------------------------------------- 6. min phase generation
        if do.min_phase
            hrir_min = make_phase([hrir; zeros(4000,2)], 'min_phase');
            hrir_min = hrir_min(1:setup.N,:);
        end

        % ------------- 7. collect data
        if do.org_phase
            hrir_org_l(:,1) = hrir(:,1);
            hrir_org_r(:,1) = hrir(:,2);
        end
        if do.min_phase
            hrir_min_l(:,1) = hrir_min(:,1);
            hrir_min_r(:,1) = hrir_min(:,2);
        end

        % ------------------------------------------------------- 8. save data
        if do.org_phase || do.min_phase
            load(fullfile(dir.save_data, [dir.plane{1} 'radius']));
```

```
            % generate struct to be saved
            savename = [['final_' filename]];
            eval(['F' '.description = data.description;']);     % description
            eval(['F' '.head = data.head;']);                   % head
            eval(['F' '.microphone = data.microphone;']);       % microphone
            eval(['F' '.source = data.source;']);               % source
            eval(['F' '.room = data.room;']);                   % room
            eval(['F' '.fs = setup.fs;']);                      % sample rate
            eval(['F' '.torso = act_grid(n,1);']);              % torso orientation
            eval(['F' '.head_az = act_grid(n,2);']);            % head azimuth↙
orientation to ref. point
            eval(['F' '.head_el = act_grid(n,3);']);            % head elevation↙
orientation to ref. point
            eval(['F' '.sc_az = setup.sc(scID,1);']);           % source azimuth to↙
ref. point
            eval(['F' '.sc_el = setup.sc(scID,2);']);           % source elevation to↙
ref. point
            eval(['F' '.dist = mean_radius(scID);']);           % distance from↙
interaural center to speaker
            if exist('itd', 'var')
                eval(['F' '.itd = itd;']);                      % ITD
            end
        end
        if do.org_phase
            % add HRIRs to data
            eval(['F' '.leftIR = hrir_org_l;']);                  % left ear data
            eval(['F' '.rightIR = hrir_org_r;']);                 % right ear data

            save(fullfile(save_in_org, savename), '-struct', 'F');
        end
        if do.min_phase
            % add HRIRs to data
            eval(['F' '.leftIR = hrir_min_l;']);                  % left ear data
            eval(['F' '.rightIR = hrir_min_r;']);                 % right ear data

            save(fullfile(save_in_min, savename), '-struct', 'F');
        end

    end
end
```

ii abx_analyse

```
%% Script to analyse ABX-TEST
% given variables
% results from ABX-TEST
% 1. hypothesis: Audibility a) individual, b) over all subjects
% 2. hypothesis: it is harder to try for speech as for noise or vice versa
% 3. hypothesis: different difficulty to perform for given sources
% plots

clear; close all; clc

n_P=11; % current number of participants

% approx. of significance
m=0.5; % prob. of guessing
m_expected=0.75; % initial standard effectsize
alpha_bino=0.05; % initial level of significance (before bonferoni)
alpha_chi=0.05;
tail='right'; % one-sided bernoulli test

Pair=[{'n2' 'n3' 'n6' 's2' 's3' 's6'}]; % xlabels for plotting,  e.g. n2: noise/source2
PairID     = [12 13 16 22 23 26]; % coding test condition
n_trials=23*length(PairID); % whole number of trials for each subject
trials=23; % number of trial in one single section
num_chi2_test=4; % number of  chi-square-tests for 3rd hypothesis
alpha_cum_bonf= alpha_bino/(length(PairID)*n_P) % alpha correction by Bonferoni


% to be set CARFULLY!!!!!! Px_ord = [individ. section order; number of correct
% answers for each section, in its order]
% code for section order and PairID:
%           1st number: 1-noise, 2-speech,
%           2nd number: 2-source 2, 3-source 3, 6-source 6,
%           e.g. 13 means noise/source 3

P1_ord= sortrows([13 12 16 26 22 23; 22 23 23 22 22 12]')';
P1 = P1_ord(2,:);  sum(P1); % subject 1

P2_ord= sortrows([23 26 22 16 12 13; 14 23 22 23 23 23]')';
P2 = P2_ord(2,:);  sum(P2); % subject 2

P3_ord= sortrows([13 12 16 23 22 26; 9  23 23 8  20 20]')';
P3 = P3_ord(2,:);  sum(P3); % subject 3

P4_ord= sortrows([16 13 12 22 26 23; 23 21 23 22 23 13]')';
P4 = P4_ord(2,:);  sum(P4); % subject 4

P5_ord= sortrows([26 23 22 12 16 13; 12 11 15 22 23 14]')';
P5 = P5_ord(2,:);  sum(P5); % subject 5

P6_ord= sortrows([23 22 26 13 12 16; 20 23 23 23 23 23]')';
P6 = P6_ord(2,:);  sum(P6); % subject 6

P7_ord= sortrows([16 12 13 26 23 22; 23 22 23 15 9  11]')';
P7 = P7_ord(2,:);  sum(P7); % subject 7
```

```
P8_ord= sortrows([26 23 22 13 12 16; 14 12 19 10 23 23]')';
P8 = P8_ord(2,:);  sum(P8); % subject 8

P9_ord= sortrows([13 16 12 22 23 26; 23 23 23 23 23 23]')';
P9 = P9_ord(2,:);  sum(P9); % subject 9

P10_ord=sortrows([12 13 16 22 23 26; 23 23 23 22 16 23]')';
P10= P10_ord(2,:); sum(P10); % subject 10

P11_ord=sortrows([26 23 22 12 13 16; 17 22 22 19 22 23]')';
P11= P11_ord(2,:); sum(P11); % subject 11

x=[P1;P2;P3;P4;P5;P6;P7;P8;P9;P10;P11]
sum_data=sum(x,1)

%% 1. a) proof, if h0 can be rejected (proof for each subject and each
%test-section)
 for sec= 1:6
for pp= 1:n_P
        % Leventhal 1986
        percent_corr(pp,sec)=x(pp,sec)/trials*100; % percent correct
        prob_dens(pp,sec) = binopdf(x(pp,sec),trials,m); % value of probability density↙
function at x(pp,sec)
        p_sigtest(pp,sec) = 1 - binocdf(x(pp,sec)-1,trials,m); % 1-cumulated↙
probability gives alpha-risk

        if p_sigtest(pp,sec)<=alpha_cum_bonf
            h1(pp,sec)=1; % alpha-risk should be smaller than current level of signif.
        else
            h1(pp,sec)=0; % otherwise reject h1
        end

end
% 1. b)
    all_sigtest(1,sec)= 1 - binocdf(sum_data(1,sec)-1,trials*n_P,m); % do same for each↙
section not individ.
    if all_sigtest(1,sec)<=alpha_bino/6
            h1_all(1,sec)=1;
        else
            h1_all(1,sec)=0;
        end
end
prob_dens=prob_dens;
p_sigtest=p_sigtest
H1=h1
percent_corr=percent_corr

H1_all=[h1_all;sum_data/(trials*n_P)*100]


%% 2. proof, if it is harder to try for speech as for noise
% Chi-square-test with one treatment (content)
%df=(k-1)=1 with k=2

corr_n=sum(sum(x(:,1:3)));
```

```
corr_s=sum(sum(x(:,4:6)));
corr_p_cont=(corr_n+corr_s)/2;

chi2_cont=(corr_n-corr_p_cont).^2./corr_p_cont+(corr_s-corr_p_cont).^2./corr_p_cont;
chi_alpha_cont=1-chi2cdf(chi2_cont,1)
if chi_alpha_cont<=alpha_chi
    cont=1
else
    cont=0
end


%% 3. proof, if it is at a different difficulty to perform for given sources
% Chi-square-test with one treatment (sources)
%df=(k-1)=2 with k=3

clear alpha_cum_bonf
alpha_cum_bonf= alpha_chi/(num_chi2_test) % alpha correction by Bonferoni, new↙
hypothesis!

corr_2=sum(sum(x(:,[1 4])));
corr_3=sum(sum(x(:,[2 5])));
corr_6=sum(sum(x(:,[3 6])));
corr_p_sc=(corr_2+corr_3+corr_6)/3;

chi2_sc=(corr_2-corr_p_sc).^2./corr_p_sc+(corr_3-corr_p_sc).^2./corr_p_sc+(corr_6-↙
corr_p_sc).^2./corr_p_sc;
chi_alpha_sc=1-chi2cdf(chi2_sc,2)
if chi_alpha_sc<=alpha_cum_bonf
    source=1
else
    source=0
end
%% proof, if it is especially harder to try for one source (one vs. two)
% Chi-square-test with one treatment (sources)
% df=(k-1)=1 with k=2, source 2 vs. 3 - source 2 vs. 6 - source 3 vs. 6
corr_p236=(corr_2+(corr_3+corr_6)/2)/2;
corr_p326=(corr_3+(corr_2+corr_6)/2)/2;
corr_p623=(corr_6+(corr_2+corr_3)/2)/2;

vs236=(corr_2-corr_p236).^2./(corr_p236)+((corr_3+corr_6)/2-corr_p236).^2./(corr_p236);
chi_2vs36=1-chi2cdf(vs236,2)
if chi_2vs36<=alpha_cum_bonf
    chi236=1
else
    chi236=0
end

vs326=(corr_3-corr_p326).^2./(corr_p326)+((corr_2+corr_6)/2-corr_p326).^2./(corr_p326);
chi_3vs26=1-chi2cdf(vs326,2)
if chi_3vs26<=alpha_cum_bonf
    chi326=1
else
    chi326=0
end
```

```
vs623=(corr_6-corr_p623).^2./(corr_p623)+((corr_2+corr_3)/2-corr_p623).^2./(corr_p623);
chi_6vs23=1-chi2cdf(vs623,2)
if chi_6vs23<=alpha_cum_bonf
    chi623=1
else
    chi623=0
end



%% visualize sample-size analyse
t=1:23;
s1=binopdf(t, trials,m);
s2=binopdf(t, trials,m_expected);
a=plot(t, s1, 'ro','linewidth', 2); hold on;b=plot(t,s2, 'bo', 'linewidth', 2);
c=plot(16*ones(1,length([0:0.001:0.2])), [0:0.001:0.2],'k-','linewidth', 2);
xlabel('number of trials');
ylabel('probability density function');
text(16,.08,sprintf('16 correct answers out of 23 \n - criterion for reaching level of↙
significance'))
text(16.2,.005,'alpha')
text(14,.005,'beta')
grid on
set(gca, 'xtick', 1:2:23);
set(gca, 'xticklabel', 1:2:23);
title(sprintf('Bernoulli-Test with 23 trials,\n red: distribution for just guessing↙
(0.5),\n blue: distribution for effect-size 0.75,\n beta=0.2, alpha=0.05'))


%% plot results of subjects
figure(2)
line=2.2;

plot((0:8),trials/2*ones(1,9),'m:', 'linewidth', line);hold on;
text(0.3, trials/2, sprintf('prob. of\n guessing'),'HorizontalAlignment','right');
plot((0:8),20*ones(1,9),'m:', 'linewidth', line);
text(0.3, 20, sprintf('min. value for\n reaching↙
significance'),'HorizontalAlignment','right');
text(6.53, 24,sprintf('%% correct\n (rounded down)'),'HorizontalAlignment','left');
text(0.47, 24, sprintf('n trials\n correct'),'HorizontalAlignment','right');
plot(1.0,unique(x(:,1)), 'k+','linewidth', line); hold on;
text(1.07,23,'8');text(1.07,22,'2');text(1.07,19,'1');
plot(2.0,unique(x(:,2)),'k+', 'linewidth', line);
text(2.07,23,'5');text(2.07,22,'2');text(2.07,21,'1');text(2.07,14,'1');text↙
(2.07,10,'1');text(2.07,9,'1');
plot(3.0,unique(x(:,3)),'k+', 'linewidth', line);
text(3.07,23,'11');
plot(4.0,unique(x(:,4)),'k+', 'linewidth', line);
text(4.07,23,'2');text(4.07,22,'5');text(4.07,20,'1');text(4.07,19,'1');text↙
(4.07,15,'1');text(4.07,11,'1');
plot(5.0,unique(x(:,5)),'k+', 'linewidth', line);
text(5.07,23,'1');text(5.07,22,'1');text(5.07,20,'1');text(5.07,16,'1');text↙
(5.07,14,'1');text(5.07,13,'1');
text(5.07,12,'2');text(5.07,11,'1');text(5.07,9,'1');text(5.07,8,'1');
```

```
h1=plot(6.0,unique(x(:,6)),'k+', 'linewidth', line); plot([0 7],0);
text(6.07,23,'5');text(6.07,22,'1');text(6.07,20,'1');text(6.07,17,'1');text↙
(6.07,15,'1');text(6.07,14,'1');
text(6.07,12,'1');
set(gca, 'xtick', 0:8);
set(gca, 'xticklabel',{' ', 'n2', 'n3', 'n6', 's2', 's3', 's6', ' '});
set(gca, 'ytick', 7:23);
set(gca, 'ytickLabel',[7:23]);
grid on; %grid minor
pcent=floor((1:23)./23.*100*10)/10;
for k=8:23
    pc{k}=num2str(pcent(k));
    text(6.53,k,pc{k})
end



hold on
%y=[P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11]';
%z=repmat([1:6],[1 11])';
%h2=boxplot(y,z,'symbol', 'r+');%'plotstyle','compact');
set(gca, 'xtick', 0:8);
set(gca, 'xticklabel',{' ', 'n2', 'n3', 'n6', 's2', 's3', 's6', ' '});
set(gca, 'ytick', 7:23);
set(gca, 'ytickLabel',[7:23]);
axis([0.5 6.5 7 24]);
%set(h2,'LineWidth',line);
text(1,5.5,sprintf('noise \nsource 2'),'HorizontalAlignment','center');
text(2,5.5,sprintf('noise \nsource 3'),'HorizontalAlignment','center');
text(3,5.5,sprintf('noise \nsource 6'),'HorizontalAlignment','center');
text(4,5.5,sprintf('speech \nsource 2'),'HorizontalAlignment','center');
text(5,5.5,sprintf('speech \nsource 3'),'HorizontalAlignment','center');
text(6,5.5,sprintf('speech \nsource 6'),'HorizontalAlignment','center');
grid on; %grid minor
title(sprintf('ABX-TEST: plotted results\n (each of 11 participating subjects answered↙
23 trials for each of 6 settings)'))
```

iii Interpolation

a hrtf_interpolation

```
% interpolation of processed data of hrtf measurement series
% script can be evaluated blockwise. plots are saved. variables in
% workspace will not be saved.
%
% Processing order:
% 0) configs            - some given values
% 1) read               - reading/saving raw data of sampling points
% 2) nearest neighbour  - fills up matrix of sampling points to reach 329
%                         HRIRs/HRTFs for each interval of interval
%                         via nearest neighbour
% 3) time_inv_dist_head - interpolation of HEAD in time domain for all
%                         planes via linear interpolation
% 4) time_spline_head   - interpolation of HEAD in time domain for all
%                         planes via spline interpolation
% 5) freq_inv_dist_head - interpolation of HEAD in frequency domain for all
%                         planes via linear interpolation
% 6) freq_spline_head   - interpolation of HEAD in frequency domain for all
%                         planes via spline interpolation
% 7) time_inv_dist_torso- interpolation of TORSO in time domain for horiz.
%                         plane via linear interpolation
% 8) freq_inv_dist_torso- interpolation of TORSO in frequency domain for
%                         horiz. plane via linear interpolation
%
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

%% data handling and configuration
clc; clear; close all;

do_read                 =   0; %1 - do it, 0 skip it,already done
do_neighbour            =   0;
do_time_inv_dist_head   =   1;
do_time_spline_head     =   1;
do_freq_inv_dist_head   =   0;
do_freq_spline_head     =   0;
do_time_inv_dist_torso  =   0;
do_freq_inv_dist_torso  =   0;


path='D:\rpopko\AP 4 Interpolation';

plane={'Med' 'Front'}% 'Med' 'Front'};
method={'head' 'torso'};
hrtf_phase={'org'};%{'min' 'org'};
sourceID=[1 2 3 4 5 6];
hrir_length=425; % length of loaded HRIR
power=10;


%% read sampling points
if do_read

    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)
```

```
            for s=sourceID
                % make subdirectories if don't exsist

                if ~exist( fullfile(path, ['raw_' plane{pn}], hrtf_phase{ph}));
                    mkdir( fullfile(path,['raw_' plane{pn}], hrtf_phase{ph}));
                end
                save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});

                for m=1:length(method)
                    for intv=[1:10 12:2:30 35 40 45]
                        % read and save all HRIRs and parameters
                        if ~strcmp(method{m},'torso') || strcmp(plane{pn}, 'Hor')
                            data=catch2interpol(intv, plane{pn}, method{m}, hrtf_phase↙
{ph}, s);
                            save(fullfile(save_in_raw,[plane{pn} '_meth_' method{m}↙
'_scID' num2str(s) '_' hrtf_phase{ph} '_intv_' num2str(intv) '.mat']), '-struct',↙
'data');
                        end
                    end
                end

                if ~exist( fullfile(path, 'REF_fine', hrtf_phase{ph}));
                    mkdir( fullfile(path, 'REF_fine', hrtf_phase{ph}));
                end
                save_in_ref=fullfile(path, 'REF_fine', hrtf_phase{ph});

                data=catch2interpol(0.5, plane{pn}, method{1}, hrtf_phase{ph}, s);
                save(fullfile(save_in_ref,[plane{pn} '_REF_scID' num2str(s) '_'↙
hrtf_phase{ph} '.mat']), '-struct', 'data');
            end
        end
    end
else
    %nothing happens,

end

%% nearest neighbour1
if do_neighbour
    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)

            save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});
            interpolate_head_time('nearest', 'neighbour', sourceID, plane, pn,↙
hrtf_phase,ph, save_in_raw)
        end
    end
end

%% inverse distance - interpolation in time domain via head (all planes)
if do_time_inv_dist_head
    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)

            save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});
```

```
            interpolate_head_time('linear', 'time_inv_dist', sourceID, plane, pn,↙
hrtf_phase,ph, save_in_raw)
        end
    end
end

%% spline - interpolation in time domain via head (all planes)
if do_time_spline_head
    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)

            save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});
            interpolate_head_time('spline', 'time_spline', sourceID, plane, pn,↙
hrtf_phase,ph, save_in_raw)
        end
    end
end

%% inverse distance - interpolation in frequency domain via head (all planes)
if do_freq_inv_dist_head
    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)

            save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});
            interpolate_head_freq(power,'linear', 'freq_inv_dist', sourceID, plane, pn,↙
hrtf_phase,ph, save_in_raw)
        end
    end
end

%% spline - interpolation in frequency domain via head (all planes)
if do_freq_spline_head
    for pn=1:length(plane)
        for ph=1:length(hrtf_phase)

            save_in_raw=fullfile(path,['raw_' plane{pn}], hrtf_phase{ph});
            interpolate_head_freq(power, 'spline', 'freq_spline', sourceID, plane, pn,↙
hrtf_phase,ph, save_in_raw)
        end
    end
end

%% inverse distance - interpolation in time domain via torso (only HOR-plane)
if do_time_inv_dist_torso
    for ph=1:length(hrtf_phase)

        save_in_raw=fullfile(path,'raw_Hor', hrtf_phase{ph});
        interpolate_torso_time('time_inv_dist', sourceID,hrtf_phase,ph, save_in_raw)
    end
end

%% inverse distance - interpolation in frequency domain via torso (only HOR-plane)
if do_freq_inv_dist_torso
    for ph=1:length(hrtf_phase)
```

```
        save_in_raw=fullfile(path,'raw_Hor', hrtf_phase{ph});
        interpolate_torso_freq(power, 'freq_inv_dist', sourceID, hrtf_phase,ph,↙
save_in_raw)
    end
end
```

b catch2interpol

```
function [ data ] = catch2interpol(intv, act_plane, act_method, act_phase, scID)
%catch2interpol catches files in required order to interpolate the head-movment
%between +82° and -82° for given interval, act_method, act_phase, source
%Med./Front.plane: 5/3 matrices for each head_azimuth to interpol. via
%head, Hor. plane: interpol. via head 1 matrix, via torso 2 matrices
% input:
%   intv      - interval to be interpolated
%   act_plane     - used plane ('Hor' 'Med' 'Front')
%   act_method    - interpolation via 'torso' or 'head'
%   act_phase     - 'min' or 'org'
%   scID     - source ID (1, 2, 3, 4, 5 or 6)
%
% output:
%   data      - struct including hrir matrices and used parameter
% R. Popko, TU Berlin, Audio Communication Group, 07/2013
%%
%data=struct([]);

hrir_length=425; % length of loaded HRIR
switch act_phase
    case 'min'
        data_dir = fullfile('D:\rpopko\AP 2 HRTF Datenbank\5 Processed data\2 HRIR↙
final','minimum phase', act_plane, ['scID' num2str(scID)]);
    case 'org'
        data_dir = fullfile('D:\rpopko\AP 2 HRTF Datenbank\5 Processed data\2 HRIR↙
final','original phase', act_plane, ['scID' num2str(scID)]);
    otherwise
        disp('Unknown phase')
end
switch act_plane
    case 'Hor'
        max_range=82;

        samp_points=[];
        switch act_method
            case 'head'
                % extract sampling points for "head-interpolation"
                % samp_points= [head_az torso]
                tmp=mod(max_range,intv);
                if tmp==0
                    add_val=0;
                else
                    add_val=intv-mod(max_range,intv);
                end
                samp_points(:,1)= transpose(max_range+add_val):-intv:(-max_range-↙
add_val);
                samp_points(:,2)=zeros(length(samp_points(:,1)),1);
            case 'torso'
                % extract sampling points for "head-interpolation"
                % samp_points= [head_az torso1 torso2]
                samp_points(:,1)= (max_range:-0.5:-max_range)';
                samp_points(:,2)=(mod(samp_points(:,1),intv))';
                samp_points(:,3)=samp_points(:,2)-intv;
                ind= samp_points(:,3)==-intv;
                samp_points(ind,3)=0;
```

```
            otherwise
                disp('Unknown method')
        end

    case {'Med','Front'}
        max_range=45;

        samp_points=[];
        switch act_method
            case 'head'
                % extract sampling points for "head-interpolation"
                % samp_points= [head_az torso]
                samp_points(:,1)= transpose(max_range-mod(max_range,intv)):-intv:(-↙
max_range+mod(max_range,intv));
                samp_points(:,2)=zeros(length(samp_points(:,1)),1);
            case 'torso'
                disp('for Med./Front.-plane no possible interpolation via torso yet!!')

            otherwise
                disp('Unknown method')
        end
    otherwise
        disp('Unknown plane')
end


hrir_l_1=zeros(hrir_length,length(samp_points(:,1)));
hrir_r_1=zeros(hrir_length,length(samp_points(:,1)));
hrir_l_2=zeros(hrir_length,length(samp_points(:,1)));
hrir_r_2=zeros(hrir_length,length(samp_points(:,1)));
hrir_l_3=zeros(hrir_length,length(samp_points(:,1)));
hrir_r_3=zeros(hrir_length,length(samp_points(:,1)));
hrir_l_4=zeros(hrir_length,length(samp_points(:,1)));
hrir_r_4=zeros(hrir_length,length(samp_points(:,1)));
hrir_l_5=zeros(hrir_length,length(samp_points(:,1)));
hrir_r_5=zeros(hrir_length,length(samp_points(:,1)));

for kk=1:length(samp_points(:,1))
    if ~mod(kk, 50) || ~mod(kk, length(samp_points(:,1)))
        disp(['reading all hrir for interpolation: ' num2str(kk) ' of ' num2str(length↙
(samp_points(:,1)))]);
    end

    % get filename
    switch act_plane
        case 'Hor'
            filename1 = ['final_' act_plane '_t' num2str(samp_points(kk,2)) '_haz'↙
num2str(samp_points(kk,1)) '_hel0_scID' num2str(scID) '.mat'];
            load(fullfile(data_dir, filename1));
            data.hrir_l_1(:, kk) = leftIR(:, 1);
            data.hrir_r_1(:, kk) = rightIR(:, 1);
            data.itd_1(:, kk) = itd;
            clear leftIR rigthIR itd

            if strcmp(act_method,'torso')
```

```
                filename2 = ['final_' act_plane '_t' num2str(samp_points(kk,3)) '_haz'↙
num2str(samp_points(kk,1)) '_hel0_scID' num2str(scID) '.mat'];
                load(fullfile(data_dir, filename2));
                data.hrir_l_2(:, kk) = leftIR(:, 1);
                data.hrir_r_2(:, kk) = rightIR(:, 1);
                data.itd_2(:, kk) = itd;
                clear leftIR rigthIR itd
            else
            end

        case {'Med','Front'}
            filename1 = ['final_' act_plane '_t0_haz' num2str(0) '_hel' num2str↙
(samp_points(kk,1)) '_scID' num2str(scID) '.mat'];
            filename2 = ['final_' act_plane '_t0_haz' num2str(30) '_hel' num2str↙
(samp_points(kk,1)) '_scID' num2str(scID) '.mat'];
            filename3 = ['final_' act_plane '_t0_haz' num2str(-30) '_hel' num2str↙
(samp_points(kk,1)) '_scID' num2str(scID) '.mat'];

            load(fullfile(data_dir, filename1));
            data.hrir_l_1(:, kk) = leftIR(:, 1);
            data.hrir_r_1(:, kk) = rightIR(:, 1);
            data.itd_1(:, kk) = itd;
            clear leftIR rigthIR itd

            load(fullfile(data_dir, filename2));
            data.hrir_l_2(:, kk) = leftIR(:, 1);
            data.hrir_r_2(:, kk) = rightIR(:, 1);
            data.itd_2(:, kk) = itd;
            clear leftIR rigthIR itd

            load(fullfile(data_dir, filename3));
            data.hrir_l_3(:, kk) = leftIR(:, 1);
            data.hrir_r_3(:, kk) = rightIR(:, 1);
            data.itd_3(:, kk) = itd;
            clear leftIR rigthIR itd

            if strcmp(act_plane,'Med')
                filename4 = ['final_' act_plane '_t0_haz' num2str(60) '_hel' num2str↙
(samp_points(kk,1)) '_scID' num2str(scID) '.mat'];
                filename5 = ['final_' act_plane '_t0_haz' num2str(-60) '_hel' num2str↙
(samp_points(kk,1)) '_scID' num2str(scID) '.mat'];
                load(fullfile(data_dir, filename4));
                data.hrir_l_4(:, kk) = leftIR(:, 1);
                data.hrir_r_4(:, kk) = rightIR(:, 1);
                data.itd_4(:, kk) = itd;
                clear leftIR rigthIR itd

                load(fullfile(data_dir, filename5));
                data.hrir_l_5(:, kk) = leftIR(:, 1);
                data.hrir_r_5(:, kk) = rightIR(:, 1);
                data.itd_5(:, kk) = itd;
                clear leftIR rigthIR itd
            else
            end
        otherwise
```

```
            disp('Unknown plane')
    end
end
data.samp_points=samp_points;
data.max_range=max_range;
data.intv=intv;
data.act_plane=act_plane;
data.act_method=act_method;
data.act_phase=act_phase;
data.scID=scID;

end %end of function
```

c interpolate_head_freq

```
function [ ] = interpolate_head_freq(power, int_style, int_name, sourceID, plane,↙
planeID,hrtf_phase, phaseID, load_raw)
%interpolate_head_freq interpoaltes HRIRs and ITDs in all planes, relevant
%for outsourcing similar and frequently used code in script
%'hrtf_interpolation'
%
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

path='D:\rpopko\AP 4 Interpolation';
hrir_length=425;
fs=44100;
for s=sourceID
    for intv=[1:10 12:2:30 35 40 45]
        disp([int_name ': ' plane{planeID} ' ' hrtf_phase{phaseID} ' scID ' num2str(s)↙
' interval ' num2str(intv)]);
        % make subdirectories if don't exsist

        if ~exist( fullfile(path, ['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name));
            mkdir( fullfile(path,['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name));
        end

        save_in_int=fullfile(path,['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name);

        if strcmp(plane{planeID}, 'Hor')
            load(fullfile(load_raw,['Hor_meth_head_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
            x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
            xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

            [hrir_l_int,hrir_r_int,hrtf_l_int,hrtf_r_int,itd_int ] = in_freq(hrtf_phase↙
{phaseID},hrir_l_1,hrir_r_1,itd_1,x,xi,int_style,power);

            INPOL.itd=itd_int;
            INPOL.hrtf_l=hrtf_l_int;
            INPOL.hrtf_r=hrtf_r_int;
            INPOL.hrir_l=hrir_l_int;
            INPOL.hrir_r=hrir_r_int;

            INPOL.samp_points=samp_points;
            INPOL.intv=intv;
            INPOL.plane=plane{planeID};
            INPOL.int_method=int_name;
            INPOL.act_phase=hrtf_phase{phaseID};
            INPOL.sourceID=s;
            save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s' num2str(s)↙
'_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');

            clear INPOL
        elseif strcmp(plane{planeID}, 'Med')
            load(fullfile(load_raw,['Med_meth_head_scID' num2str(s) '_' hrtf_phase↙
```

```
{phaseID} '_intv_' num2str(intv) '.mat']));
            x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
            xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

            [hrir_l_int1,hrir_r_int1,hrtf_l_int1,hrtf_r_int1,itd_int1 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_1,hrir_r_1,itd_1,x,xi,int_style,power);
            [hrir_l_int2,hrir_r_int2,hrtf_l_int2,hrtf_r_int2,itd_int2 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_2,hrir_r_2,itd_2,x,xi,int_style,power);
            [hrir_l_int3,hrir_r_int3,hrtf_l_int3,hrtf_r_int3,itd_int3 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_3,hrir_r_3,itd_3,x,xi,int_style,power);
            [hrir_l_int4,hrir_r_int4,hrtf_l_int4,hrtf_r_int4,itd_int4 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_4,hrir_r_4,itd_4,x,xi,int_style,power);
            [hrir_l_int5,hrir_r_int5,hrtf_l_int5,hrtf_r_int5,itd_int5 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_5,hrir_r_5,itd_5,x,xi,int_style,power);

            INPOL.itd_haz_0=itd_int1;
            INPOL.hrtf_haz_0_l=hrtf_l_int1;
            INPOL.hrtf_haz_0_r=hrtf_r_int1;
            INPOL.hrir_haz_0_l=hrir_l_int1;
            INPOL.hrir_haz_0_r=hrir_r_int1;

            INPOL.itd_haz_p30=itd_int2;
            INPOL.hrtf_haz_p30_l=hrtf_l_int2;
            INPOL.hrtf_haz_p30_r=hrtf_r_int2;
            INPOL.hrir_haz_p30_l=hrir_l_int2;
            INPOL.hrir_haz_p30_r=hrir_r_int2;

            INPOL.itd_haz_m30=itd_int3;
            INPOL.hrtf_haz_m30_l=hrtf_l_int3;
            INPOL.hrtf_haz_m30_r=hrtf_r_int3;
            INPOL.hrir_haz_m30_l=hrir_l_int3;
            INPOL.hrir_haz_m30_r=hrir_r_int3;

            INPOL.itd_haz_p60=itd_int4;
            INPOL.hrtf_haz_p60_l=hrtf_l_int4;
            INPOL.hrtf_haz_p60_r=hrtf_r_int4;
            INPOL.hrir_haz_p60_l=hrir_l_int4;
            INPOL.hrir_haz_p60_r=hrir_r_int4;

            INPOL.itd_haz_m60=itd_int5;
            INPOL.hrtf_haz_m60_l=hrtf_l_int5;
            INPOL.hrtf_haz_m60_r=hrtf_r_int5;
            INPOL.hrir_haz_m60_l=hrir_l_int5;
            INPOL.hrir_haz_m60_r=hrir_r_int5;

            INPOL.samp_points=samp_points;
            INPOL.intv=intv;
            INPOL.plane=plane{planeID};
            INPOL.int_method=int_name;
            INPOL.act_phase=hrtf_phase{phaseID};
            INPOL.sourceID=s;
            save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s' num2str(s)↙
'_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');

            clear INPOL
```

```
        elseif strcmp(plane{planeID}, 'Front')
            load(fullfile(load_raw,['Front_meth_head_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
            x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
            xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

            [hrir_l_int1,hrir_r_int1,hrtf_l_int1,hrtf_r_int1,itd_int1 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_1,hrir_r_1,itd_1,x,xi,int_style,power);
            [hrir_l_int2,hrir_r_int2,hrtf_l_int2,hrtf_r_int2,itd_int2 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_2,hrir_r_2,itd_2,x,xi,int_style,power);
            [hrir_l_int3,hrir_r_int3,hrtf_l_int3,hrtf_r_int3,itd_int3 ] = in_freq↙
(hrtf_phase{phaseID},hrir_l_3,hrir_r_3,itd_3,x,xi,int_style,power);

            INPOL.itd_haz_0=itd_int1;
            INPOL.hrtf_haz_0_l=hrtf_l_int1;
            INPOL.hrtf_haz_0_r=hrtf_r_int1;
            INPOL.hrir_haz_0_l=hrir_l_int1;
            INPOL.hrir_haz_0_r=hrir_r_int1;

            INPOL.itd_haz_p30=itd_int2;
            INPOL.hrtf_haz_p30_l=hrtf_l_int2;
            INPOL.hrtf_haz_p30_r=hrtf_r_int2;
            INPOL.hrir_haz_p30_l=hrir_l_int2;
            INPOL.hrir_haz_p30_r=hrir_r_int2;

            INPOL.itd_haz_m30=itd_int3;
            INPOL.hrtf_haz_m30_l=hrtf_l_int3;
            INPOL.hrtf_haz_m30_r=hrtf_r_int3;
            INPOL.hrir_haz_m30_l=hrir_l_int3;
            INPOL.hrir_haz_m30_r=hrir_r_int3;

            INPOL.samp_points=samp_points;
            INPOL.intv=intv;
            INPOL.plane=plane{planeID};
            INPOL.int_method=int_name;
            INPOL.act_phase=hrtf_phase{phaseID};
            INPOL.sourceID=s;
            save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s' num2str(s)↙
'_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');

            clear INPOL
        else
            disp('Unknown plane')
        end
    end
end
end
```

d interpolate_torso_freq

```
function [ ] = interpolate_torso_freq(power, int_name, sourceID, hrtf_phase, phaseID,↙
load_raw)
%interpolate_torso - interpolates linear (inv. dist) via torso in time domain
%relevant for outsourcing similar and frequently used code in script
%'hrtf_interpolation'
%
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

path='D:\rpopko\AP 4 Interpolation';
hrir_length=425;
fs=44100;
for s=sourceID
    for intv=[1:10 12:2:30 35 40 45]
        disp([int_name ': ' 'Hor' ' ' hrtf_phase{phaseID} ' scID ' num2str(s) '↙
interval ' num2str(intv)]);
        % make subdirectories if don't exsist

        if ~exist( fullfile(path, ['int_' 'Hor'], hrtf_phase{phaseID}, int_name));
            mkdir( fullfile(path,['int_' 'Hor'], hrtf_phase{phaseID}, int_name));
        end

        save_in_int=fullfile(path,['int_' 'Hor'], hrtf_phase{phaseID}, int_name);


        % load data, including 2 HRIRs to be interpolated by distance
        % weighting
        load(fullfile(load_raw,['Hor_meth_torso_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
        ind=find(samp_points(:,3)==0);
        samp_points(ind,3)=-intv; %helpfull to give interp1 the right x and xi (for↙
"pseudo"interpolation at sampling points (2 identical hrirs XY are interpolated↙
linearly - results in XY))

        itd=zeros(1,329);
        abs_l=zeros(2^power/2,329);
        abs_r=zeros(2^power/2,329);
        ang_l=zeros(2^power/2,329);
        ang_r=zeros(2^power/2,329);
        for kk=1:329
            x=[(samp_points(kk,1)-samp_points(kk,2)) (samp_points(kk,1)-samp_points(kk,↙
3))] ;
            xi=samp_points(kk,1); %xi(2) closer to x(1) or x(3) - defines weigth

            itd_rough=[itd_1(:,kk) itd_2(:,kk)];
            hrir_l_rough=[hrir_l_1(:,kk) hrir_l_2(:,kk)];
            hrir_r_rough=[hrir_r_1(:,kk) hrir_r_2(:,kk)];


            [hrir_l_int,hrir_r_int,hrtf_l_int,hrtf_r_int,itd_int ] = in_freq(hrtf_phase↙
{phaseID},hrir_l_rough,hrir_r_rough,itd_rough,x,xi,'linear',power);

            itd(:,kk)=itd_int;
            hrir_l(:,kk)=hrir_l_int;
```

```
            hrir_r(:,kk)=hrir_r_int;
            hrtf_l(:,kk)=hrtf_l_int;
            hrtf_r(:,kk)=hrtf_r_int;

            clear hrir_l_int hrir_r_int hrtf_l_int hrtf_r_int itd_int

        end

        INPOL.itd=itd;
        INPOL.hrtf_l=hrtf_l;
        INPOL.hrtf_r=hrtf_r;
        INPOL.hrir_l=hrir_l;
        INPOL.hrir_r=hrir_r;

        INPOL.samp_points=samp_points;
        INPOL.intv=intv;
        INPOL.plane='Hor';
        INPOL.int_method=int_name;
        INPOL.act_phase=hrtf_phase{phaseID};
        INPOL.sourceID=s;
        save(fullfile(save_in_int,['Hor' '_TORSO_' int_name '_s' num2str(s) '_'↙
hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');

        clear INPOL

    end
end
end
```

e interpolate_head_time

```
function [ ] = interpolate_head_time(int_style, int_name, sourceID, plane, planeID,↙
hrtf_phase, phaseID, load_raw)
%interpolate_head_time interpoaltes HRIRs and ITDs in all planes, relevant
%for outsourcing similar and frequently used code in script
%'hrtf_interpolation'
%
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

path='D:\rpopko\AP 4 Interpolation';
hrir_length=425;
fs=44100;
up_samp=10;
onset_thresh=-6;
not_chop=100; %value for onset minus not_chop > preserves the transient (100 samples↙
after upsampling by 10 corresponds to 10 samples in original sampling rate
corr_or_ons={'cross' 'ons'};

if strcmp(hrtf_phase{phaseID},'min') || strcmp(int_style,'nearest')
    once_or_twice=1;
else
    once_or_twice=2;
end


for rr=1:once_or_twice
    CrossOns_case=corr_or_ons{rr};
    for s=sourceID

        for intv=[1:10 12:2:30 35 40 45]
            disp([int_name ': ' plane{planeID} ' ' hrtf_phase{phaseID} ' scID ' num2str↙
(s) ' interval ' num2str(intv)]);
            % make subdirectories if don't exsist



            if ~exist( fullfile(path, ['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name));
                mkdir( fullfile(path,['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name));
            end
            save_in_int=fullfile(path,['int_' plane{planeID}], hrtf_phase{phaseID},↙
int_name);

            if strcmp(plane{planeID}, 'Hor')
                load(fullfile(load_raw,['Hor_meth_head_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
                x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
                xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

                [yi_l, yi_r, itd]=in_time(hrir_l_1,hrir_r_1,itd_1,hrtf_phase{phaseID},↙
CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);

                INPOL.itd=itd;
```

```
                INPOL.hrir_l=yi_l;
                INPOL.hrir_r=yi_r;
                INPOL.samp_points=samp_points;
                INPOL.intv=intv;
                INPOL.plane=plane{planeID};
                INPOL.int_method=int_name;
                INPOL.act_phase=hrtf_phase{phaseID};
                INPOL.sourceID=s;

                if strcmp(hrtf_phase{phaseID},'min') || strcmp(int_style,'nearest')
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s'↙
num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');
                else
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_'↙
CrossOns_case '_s' num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-↙
struct','INPOL');
                end
                clear INPOL
            elseif strcmp(plane{planeID}, 'Med')
                load(fullfile(load_raw,['Med_meth_head_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
                x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
                xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

                [yi_l_1, yi_r_1, itd_1]=in_time(hrir_l_1,hrir_r_1,itd_1,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_2, yi_r_2, itd_2]=in_time(hrir_l_2,hrir_r_2,itd_2,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_3, yi_r_3, itd_3]=in_time(hrir_l_3,hrir_r_3,itd_3,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_4, yi_r_4, itd_4]=in_time(hrir_l_4,hrir_r_4,itd_4,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_5, yi_r_5, itd_5]=in_time(hrir_l_5,hrir_r_5,itd_5,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);


                INPOL.itd_haz_0=itd_1; %ITD in ys
                INPOL.itd_haz_p30=itd_2;
                INPOL.itd_haz_m30=itd_3;
                INPOL.itd_haz_p60=itd_4;
                INPOL.itd_haz_m60=itd_5;


                INPOL.hrir_haz_0_l=yi_l_1;
                INPOL.hrir_haz_0_r=yi_r_1;
                INPOL.hrir_haz_p30_l=yi_l_2;
                INPOL.hrir_haz_p30_r=yi_r_2;
                INPOL.hrir_haz_m30_l=yi_l_3;
                INPOL.hrir_haz_m30_r=yi_r_3;
                INPOL.hrir_haz_p60_l=yi_l_4;
                INPOL.hrir_haz_p60_r=yi_r_4;
                INPOL.hrir_haz_m60_l=yi_l_5;
                INPOL.hrir_haz_m60_r=yi_r_5;

                INPOL.samp_points=samp_points;
```

```
                INPOL.intv=intv;
                INPOL.plane=plane{planeID};
                INPOL.int_method=int_name;
                INPOL.act_phase=hrtf_phase{phaseID};
                INPOL.sourceID=s;
                if strcmp(hrtf_phase{phaseID},'min') || strcmp(int_style,'nearest')
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s'↙
num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');
                else
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_'↙
CrossOns_case '_s' num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-↙
struct','INPOL');
                end
                clear INPOL
            elseif strcmp(plane{planeID}, 'Front')
                load(fullfile(load_raw,['Front_meth_head_scID' num2str(s) '_'↙
hrtf_phase{phaseID} '_intv_' num2str(intv) '.mat']));
                x=max(samp_points(:,1)):-intv:min(samp_points(:,1));
                xi=max(samp_points(:,1)):-0.5:min(samp_points(:,1));

                [yi_l_1, yi_r_1, itd_1]=in_time(hrir_l_1,hrir_r_1,itd_1,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_2, yi_r_2, itd_2]=in_time(hrir_l_2,hrir_r_2,itd_2,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);
                [yi_l_3, yi_r_3, itd_3]=in_time(hrir_l_3,hrir_r_3,itd_3,hrtf_phase↙
{phaseID},CrossOns_case,x,xi, int_style,up_samp, onset_thresh, not_chop,fs);

                INPOL.itd_haz_0=itd_1; %ITD in ys
                INPOL.itd_haz_p30=itd_2;
                INPOL.itd_haz_m30=itd_3;

                INPOL.hrir_haz_0_l=yi_l_1;
                INPOL.hrir_haz_0_r=yi_r_1;
                INPOL.hrir_haz_p30_l=yi_l_2;
                INPOL.hrir_haz_p30_r=yi_r_2;
                INPOL.hrir_haz_m30_l=yi_l_3;
                INPOL.hrir_haz_m30_r=yi_r_3;

                INPOL.samp_points=samp_points;
                INPOL.intv=intv;
                INPOL.plane=plane{planeID};
                INPOL.int_method=int_name;
                INPOL.act_phase=hrtf_phase{phaseID};
                INPOL.sourceID=s;
                if strcmp(hrtf_phase{phaseID},'min') || strcmp(int_style,'nearest')
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_s'↙
num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');
                else
                    save(fullfile(save_in_int,[plane{planeID} '_HEAD_' int_name '_'↙
CrossOns_case '_s' num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-↙
struct','INPOL');
                end
                clear INPOL
            else
                disp('Unknown plane')
```

```
            end
        end
    end
end
end
```

f interpolate_torso_time

```
function [ ] = interpolate_torso_time(int_name, sourceID, hrtf_phase, phaseID,↙
load_raw)
%interpolate_torso - interpolates linear (inv. dist) via torso in time domain
%relevant for outsourcing similar and frequently used code in script
%'hrtf_interpolation'
%
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

path='D:\rpopko\AP 4 Interpolation';
hrir_length=425;
fs=44100;
hrir_length=425;
fs=44100;
up_samp=10;
onset_thresh=-6;
not_chop=100; %value for onset minus not_chop > preserves the transient (100 samples↙
after upsampling by 10 corresponds to 10 samples in original sampling rate
corr_or_ons={'cross' 'ons'};

if strcmp(hrtf_phase{phaseID},'min')
    once_or_twice=1;
else
    once_or_twice=2;
end


for rr=1:once_or_twice
    CrossOns_case=corr_or_ons{rr};
    for s=sourceID
        for intv=[1:10 12:2:30 35 40 45]
            disp([int_name ': ' 'Hor' ' ' hrtf_phase{phaseID} ' scID ' num2str(s) '↙
interval ' num2str(intv)]);
            % make subdirectories if don't exsist


            if ~exist( fullfile(path, ['int_' 'Hor'], hrtf_phase{phaseID},int_name));
                mkdir( fullfile(path,['int_' 'Hor'], hrtf_phase{phaseID}, int_name ));
            end

            save_in_int=fullfile(path,['int_' 'Hor'], hrtf_phase{phaseID},int_name);


            % load data, including 2 HRIRs to be interpolated by distance
            % weighting
            load(fullfile(load_raw,['Hor_meth_torso_scID' num2str(s) '_' hrtf_phase↙
{phaseID} '_intv_' num2str(intv) '.mat']));
            ind=find(samp_points(:,3)==0);
            samp_points(ind,3)=-intv;

            itd=zeros(1,329);
            hrir_l=zeros(hrir_length, 329);
            hrir_r=zeros(hrir_length, 329);
            for kk=1:329
```

```
                x=[(samp_points(kk,1)-samp_points(kk,2)) (samp_points(kk,1)-samp_points↙
(kk,3))] ;
                xi=samp_points(kk,1); %xi(2) closer to x(1) or x(3) - defines weigth
                itd_raw=[itd_1(:,kk) itd_2(:,kk)];
                hrir_l_rough=[hrir_l_1(:,kk) hrir_l_2(:,kk)];
                hrir_r_rough=[hrir_r_1(:,kk) hrir_r_2(:,kk)];

                [hrir_l_fine, hrir_r_fine, each_itd]=in_time(hrir_l_rough,hrir_r_rough,↙
itd_raw,hrtf_phase{phaseID},CrossOns_case,x,xi,'linear',up_samp, onset_thresh,↙
not_chop,fs);

                itd(kk)=each_itd;
                hrir_l(:,kk)=hrir_l_fine;
                hrir_r(:,kk)=hrir_r_fine;
            end
            INPOL.itd=itd;
            INPOL.hrir_l=hrir_l;
            INPOL.hrir_r=hrir_r;
            INPOL.samp_points=samp_points;
            INPOL.intv=intv;
            INPOL.plane='Hor';
            INPOL.int_method=int_name;
            INPOL.act_phase=hrtf_phase{phaseID};
            INPOL.sourceID=s;

            if strcmp(hrtf_phase{phaseID},'min')
                save(fullfile(save_in_int,['Hor' '_TORSO_' int_name '_s' num2str(s) '_'↙
hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-struct','INPOL');
            else
                save(fullfile(save_in_int,['Hor' '_TORSO_' int_name '_' CrossOns_case↙
'_s' num2str(s) '_' hrtf_phase{phaseID} '_int' num2str(intv) '.mat']), '-↙
struct','INPOL');
            end
            clear INPOL

        end
    end
end
end
```

g in_time

```
function [yi_l,yi_r,itd ] = in_time(hrir_l,hrir_r,itd_raw,phase, CrossOns_case,x,xi,↙
int_style,up_samp, onset_thresh, not_chop,fs)
% IN_TIME helpfull for interpolation in time and used in the further
% functions INTERPOLATE_HEAD_TIME and INTERPOLATE_TORSO_TIME
% takes hrirs to interpolate after switching the cases of phase
% min-phase requires no onset_detection and shifting befor
% time-interpolation
% input:
% hrir_l        left hrirs to be interpolated
% hrir_r        right hrirs
% phase         org or min phase
% CrossOns_case in case of min phase: crosscorr or onset detection
% x             grid before
% xi            grid after interpolation
% int_style     style of interpolation
% up_samp       factor of upsampling
% onset_thresh  threshold for onset_detection
% not_chop      onset-not_chop to preserve the transient behaviour
% output:
% yi_l          interpolated, (if org-phase downsampled) hrir (left)
% yi_r          -"- (right)
% itd           itd of interpolated hrirs
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

hrir_length=length(hrir_l(:,1));
range = 800;       % search range for cross corelation in mirco seconds

if strcmp(phase,'org') & ~(strcmp(int_style,'nearest'))

    % upsampling
    if up_samp>1
        hrir_l_us=zeros(hrir_length*up_samp,size(hrir_l,2));
        hrir_r_us=zeros(hrir_length*up_samp,size(hrir_l,2));
        parfor rr=1:size(hrir_l,2)
            hrir_l_us(:,rr)=interp(hrir_l(:,rr),up_samp,4);
            hrir_r_us(:,rr)=interp(hrir_r(:,rr),up_samp,4);
        end
    end

    switch CrossOns_case
        case 'cross'
            % onset-detect
            ons_l = onset_detect(hrir_l_us(1:100*up_samp,:),1, onset_thresh);
            ons_r = onset_detect(hrir_r_us(1:100*up_samp,:),1, onset_thresh);
            % interpolate onsets
            ons_l_int=round(interp1(x,ons_l,xi,int_style));
            ons_r_int=round(interp1(x,ons_r,xi,int_style));
            itd=(ons_l_int-ons_r_int)/up_samp/fs*10^6; %ITD in ys
            clear ons_l ons_r

            % cross-correlation
            % search range in samples
            N_range = ceil(range*1e-6 / (1/(fs*up_samp)));
```

```
            n_range = -N_range:N_range;

            cross_corr_l = zeros(N_range*2+1, size(hrir_l_us,2)-1);
            cross_corr_r = zeros(N_range*2+1, size(hrir_r_us,2)-1);
            % do cross-correlation
            parfor nn = 1:N_range*2+1
                cross_corr_l(nn,:) = mean(hrir_l_us(:,1:end-1) .* circshift(hrir_l_us↙
(:,2:end), [n_range(nn) 0]));
                cross_corr_r(nn,:) = mean(hrir_r_us(:,1:end-1) .* circshift(hrir_r_us↙
(:,2:end), [n_range(nn) 0]));
            end

            % detect maximum
            [~, id_l] = max(cross_corr_l);
            shift_l = n_range(id_l);
            shift_l = cumsum(shift_l);

            [~, id_r] = max(cross_corr_r);
            shift_r = n_range(id_r);
            shift_r = cumsum(shift_r);

            % shift
            hrir_l_2shift=hrir_l_us;
            hrir_r_2shift=hrir_r_us;
            parfor nn = 2:size(hrir_l_us,2)
                hrir_l_us(:,nn) = circshift(hrir_l_2shift(:,nn), [shift_l(nn-1),0]);
                hrir_r_us(:,nn) = circshift(hrir_r_2shift(:,nn), [shift_r(nn-1),0]);
            end
            clear shift_l shift_r
            % interpolate
            yi_l = transpose(interp1(x,transpose(hrir_l_us),xi,int_style));
            yi_r = transpose(interp1(x,transpose(hrir_r_us),xi,int_style));

            % shift to interpolated onset times
            % detect actual onset-times
            ons_l = onset_detect(yi_l(1:100*up_samp,:), 1, onset_thresh);
            ons_r = onset_detect(yi_r(1:100*up_samp,:), 1, onset_thresh);
            % get amount to shift
            shift_l = ons_l_int - ons_l;
            shift_r = ons_r_int - ons_r;
            yi_l_2shift=yi_l;
            yi_r_2shift=yi_r;
            parfor nn = 1:size(yi_l,2)
                yi_l(:,nn) = circshift(yi_l_2shift(:,nn), [shift_l(nn),0]);
                yi_r(:,nn) = circshift(yi_r_2shift(:,nn), [shift_r(nn),0]);
            end
            clear ons_l ons_r
        case 'ons'
            ons_l(1,:)=onset_detect(hrir_l(1:100,:), up_samp, onset_thresh)*up_samp; %↙
find the peak via onset detection
            ons_r(1,:)=onset_detect(hrir_r(1:100,:), up_samp, onset_thresh)*up_samp;
            ons_l_int=round(interp1(x,ons_l,xi,int_style));
            ons_r_int=round(interp1(x,ons_r,xi,int_style));
            itd=(ons_l_int-ons_r_int)/up_samp/fs*10^6; %ITD in ys
```

```
            hrir_l_shift=zeros(hrir_length*up_samp,length(x));
            hrir_r_shift=zeros(hrir_length*up_samp,length(x));
            parfor bb=1:length(x) %shift all peaks to the beginning of matrix
                hrir_l_shift(:,bb)=circshift(hrir_l_us(:,bb),-(ons_l(1,bb)-not_chop));
% dont chop the transient behaviour
                hrir_r_shift(:,bb)=circshift(hrir_r_us(:,bb),-(ons_r(1,bb)-not_chop));
% dont chop the transient behaviour
            end
            clear ons_l ons_r

            yi_l = transpose(interp1(x,transpose(hrir_l_shift),xi,int_style));
            yi_r = transpose(interp1(x,transpose(hrir_r_shift),xi,int_style));

            yi_l_2shift=yi_l;
            yi_r_2shift=yi_r;
            parfor cc=1:length(xi)% shift back with the interpolated onset
                yi_l(:,cc)=circshift(yi_l_2shift(:,cc),(ons_l_int(1,cc)-not_chop));
                yi_r(:,cc)=circshift(yi_r_2shift(:,cc),(ons_r_int(1,cc)-not_chop));
            end
    end
    % resample by taking just the 10th sample (up_samp th sample)
    if up_samp > 1
        tmp1 = yi_l;
        tmp2 = yi_r;
        yi_l = zeros(size(tmp1,1)/up_samp, size(tmp1,2));
        yi_r = zeros(size(tmp2,1)/up_samp, size(tmp2,2));
        parfor nn = 1:size(yi_l, 2)
            yi_l(:,nn) = decimate(tmp1(:,nn), up_samp, 8);
            yi_r(:,nn) = decimate(tmp2(:,nn), up_samp, 8);
        end
    end
elseif strcmp(phase,'min') || (strcmp(int_style,'nearest'))


    yi_l = transpose(interp1(x,transpose(hrir_l),xi,int_style));
    yi_r = transpose(interp1(x,transpose(hrir_r),xi,int_style));

    itd=round(interp1(x,itd_raw,xi,int_style));
             %ITD in ys

else
end


end
```

h in_freq

```
function [hrir_l_int,hrir_r_int,hrtf_l_int,hrtf_r_int,itd_int ] = in_freq(phase,hrir_l,↙
hrir_r,itd,x,xi,int_style, power)
% IN_TIME helpfull for interpolation in time and used in the further
% functions INTERPOLATE_HEAD_TIME and INTERPOLATE_TORSO_TIME
% takes hrirs to interpolate after switching the cases of phase
% min-phase requires no onset_detection and shifting befor
% time-interpolation
% input:
% phase         phase
% hrir_l        left hrirs to be interpolated
% hrir_r        right hrirs
% itd           itds to be interpolated
% x             grid before
% xi            grid after interpolation
% int_style     style of interpolation
% power         2^power - length of fft
% output:
% hrir_l_int          interpolated hrir (left)
% hrir_r_int         -"- (right)
% hrtf_l_int         interpolated hrtf (left)
% hrtf_r_int        -"- (right)
% itd           itd of interpolated hrirs
%
%
% R. Popko, TU Berlin, Audio Communication Group, 07/2013

hrir_length=length(hrir_l(:,1));

hrtf_l=fft(hrir_l,2^(power));
hrtf_r=fft(hrir_r,2^(power));
abs_l=abs(hrtf_l);
abs_r=abs(hrtf_r);
ang_l=unwrap(angle(hrtf_l));
ang_r=unwrap(angle(hrtf_r));

itd_int=interp1(x,itd,xi,int_style);

yi_abs_l=zeros(length(abs_l(:,1))/2, length(xi));
yi_abs_r=zeros(length(abs_r(:,1))/2, length(xi));
yi_ang_l=zeros(length(ang_l(:,1))/2, length(xi));
yi_ang_r=zeros(length(ang_r(:,1))/2, length(xi));

if strcmp(phase,'org')
% length(abs_l(:,1))/2 = length/2 of HRTF
yi_abs_l = transpose(interp1(x,transpose(abs_l(1:length(abs_l(:,1))/2,:)),xi,↙
int_style));
yi_abs_r = transpose(interp1(x,transpose(abs_r(1:length(abs_l(:,1))/2,:)),xi,↙
int_style));
yi_ang_l = transpose(interp1(x,transpose(ang_l(1:length(abs_l(:,1))/2,:)),xi,↙
int_style));
yi_ang_r = transpose(interp1(x,transpose(ang_r(1:length(abs_l(:,1))/2,:)),xi,↙
int_style));

hrtf_l_int=yi_abs_l.*exp(j.*yi_ang_l);
hrtf_l_int=[hrtf_l_int; zeros(1,length(hrtf_l_int(1,:))); conj(flipud(hrtf_l_int(2:↙
```

```
end,:)))];
hrtf_r_int=yi_abs_r.*exp(j.*yi_ang_r);
hrtf_r_int=[hrtf_r_int; zeros(1,length(hrtf_r_int(1,:))); conj(flipud(hrtf_r_int(2:↙
end,:)))];

hrir_l_int_long=real(ifft(hrtf_l_int)); hrir_l_int=hrir_l_int_long(1:hrir_length,:);
hrir_r_int_long=real(ifft(hrtf_r_int)); hrir_r_int=hrir_r_int_long(1:hrir_length,:);

elseif strcmp(phase,'min')
    %length(abs_l(:,1))/2 = length/2+1 of HRTF
yi_abs_l = transpose(interp1(x,transpose(abs_l(1:length(abs_l(:,1))/2+1,:)),xi,↙
int_style));
yi_abs_r = transpose(interp1(x,transpose(abs_r(1:length(abs_l(:,1))/2+1,:)),xi,↙
int_style));

hrtf_l_int=yi_abs_l;
hrtf_l_int=[hrtf_l_int; conj(flipud(hrtf_l_int(2:end-1,:)))];
hrtf_r_int=yi_abs_r;
hrtf_r_int=[hrtf_r_int; conj(flipud(hrtf_r_int(2:end-1,:)))];

hrir_l_int_long=ifft(hrtf_l_int);hrir_l_int_long=circshift(hrir_l_int_long, length↙
(hrir_l_int_long)/2);
hrir_l_int = make_phase([hrir_l_int_long; zeros(4000,size(hrir_l_int_long,2))],↙
'min_phase');
hrir_l_int=hrir_l_int(1:hrir_length,:);

hrir_r_int_long=ifft(hrtf_r_int);hrir_r_int_long=circshift(hrir_r_int_long, length↙
(hrir_r_int_long)/2);
hrir_r_int = make_phase([hrir_r_int_long; zeros(4000,size(hrir_r_int_long,2))],↙
'min_phase');
hrir_r_int=hrir_r_int(1:hrir_length,:);

hrtf_l_int=fft(hrir_l_int,2^power);
hrtf_r_int=fft(hrir_r_int,2^power);
else
    disp('unknown phase')
end

end
```

Anhang E

Datenträger

i Arbeit im PDF-Format

ii I_Problemvisualisierung

iii II_Messung_Processing

iv III_Torsoeinfluss

v IV_Interpolation