

AKtools – eine offene Software zur Erhebung, Verarbeitung und Inspektion akustischer Signale

Fabian Brinkmann, Stefan Weinzierl

fabian.brinkmann@tu-berlin.de, stefan.weinzierl@tu-berlin.de

Fachgebiet Audiokommunikation, TU Berlin, Einsteinufer 17c, 10587 Berlin.

AKtools sind verfügbar unter www.ak.tu-berlin.de/aktools

Einleitung

Die Erhebung, Verarbeitung und Inspektion von Audiodaten spielt eine zentrale Rolle in der täglichen akustischen Praxis. Oft sind diese Schritte jedoch auf mehrere, oftmals proprietäre Computer-Programme verteilt, wodurch eine Dokumentation dieser Arbeit erschwert wird. AKtools umfasst Matlab-Methoden zur Aufnahme und Wiedergabe sowie Funktionen zur Darstellung und Inspektion von ein- und mehrkanaligen Audiodaten die in beliebigen räumlichen Verteilungen gemessen wurden. Weitere Funktionen decken die Erstellung von Testsignalen (z.B. Impulse, Rauschen, Sweeps), die spektrale Entfaltung, die Invertierung von Übertragungsfunktionen und die sphärische Fourier-Transformation und Interpolation ab. Gut dokumentierte Demo-Skripte veranschaulichen die Benutzung von AKtools; detaillierte Information können der Beschreibung der jeweiligen Funktion entnommen werden. Im Sinne reproduzierbarer Forschung steht AKtools unter der European Union Public Licence (EURL) zur Verfügung, welche die Benutzung, Bearbeitung und Weiterverbreitung für beliebige Zwecke ermöglicht.

Reproduzierbare Forschung

Reproduzierbare Forschung hat das Ziel alle Daten, die für die Wiederholung einer Studie benötigt werden, frei zur Verfügung zu stellen. Dazu zählen neben den Forschungsdaten, wie z.B. Audioaufnahmen, Impulsantworten oder Bewertungen aus Hörversuchen, vor allem der für die Verarbeitung und Darstellung der Daten verwendete Quellcode [1]. Dies ist im Bereich der empirischen Wissenschaften besonders herausfordernd, da hier oftmals erhebliche Mengen an Daten erhoben werden und zudem eine dedizierte Vor- und Nachbereitung erfolgt. Darüber hinaus müssen neben der reinen Veröffentlichung der Daten auch Strategien für deren langfristige Verfügbarkeit (Datensicherheit) und Nachvollziehbarkeit von Änderungen (Versionierung) bedacht werden. In diesem Sinne sind die AKtools als eine Basis für die Reproduzierbarkeit von Forschungsergebnissen zu verstehen, denen die Erhebung, Verarbeitung und Darstellung akustischer Signale zugrunde liegt. Möglichkeiten für die Bereitstellung von akustischen Signalen selbst bestehen zum Beispiel durch Deposit Once [2] oder das im Aufbau befindliche Open Repository for Research Data in Acoustics (OPERA) [3].

AKtools

Aufbau

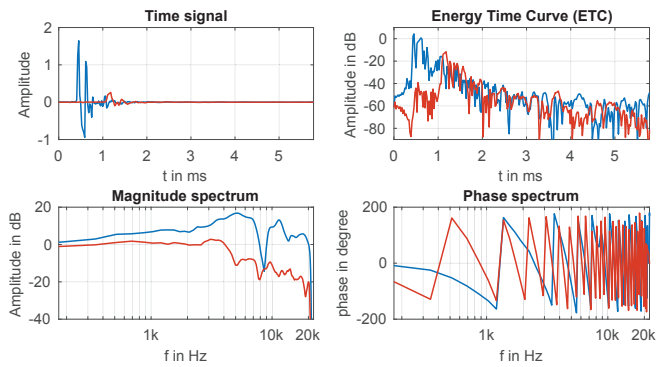
AKtools ist in drei Blöcke untergliedert, die (i) Demoskripte mit Beispielen für Funktionalität und Funktionsaufrufe, (ii) die eigentlichen Funktionen, sowie (iii) Demodaten zur Veranschaulichung bestimmter Anwendungen umfassen. Für einen schnellen und einfachen Einstieg in AKtools, sind die Installation, der Funktionsumfang und Verweise auf weiterführende Literatur in einer beigelegten Anleitung beschrieben. Zudem verfügen alle Funktionen über eine ausführliche Dokumentation der Eingabe- und Ausgabeparameter und beispielhafte Funktionsaufrufen, bzw. einem Verweis auf das entsprechende Demoskript. Um die Integration von AKtools in bestehende Projekte und Code zu erleichtern, ist der Funktionsumfang in inhaltlich sinnvolle, kleine Blöcke unterteilt, die bis auf wenige Ausnahmen mit Zeitsignalen arbeiten. Die Signale sind in Matrizen der Größe $[N \ M \ C]$ organisiert, wobei N der Anzahl der Samples, M der Anzahl der Messungen und C der Anzahl der Kanäle entspricht. Um effizient zwischen Zeit- und Frequenzdarstellung wechseln zu können, enthalten AKtools darüber hinaus Methoden zur Umwandlung zwischen beid- und einseitigen Spektren für den Fall von einer geraden, als auch ungeraden Anzahl an Frequenzbins, bzw. Stützpunkten des entsprechenden Zeitsignals.

Funktionsumfang

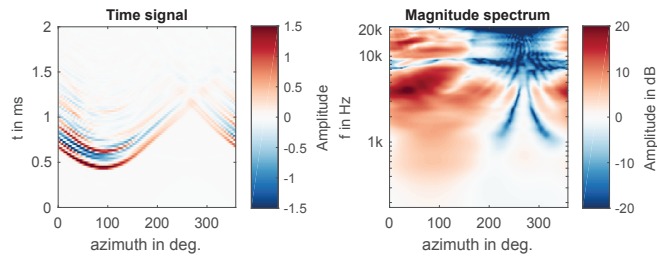
AKtools umfasst Methoden für die Aufnahme, Verarbeitung und Inspektion akustischer Signale, von denen einige im folgenden genauer erläutert werden. Für einen umfassenden Überblick sei auf die bereits erwähnten Demoskripte verwiesen, in denen die meisten Methoden Verwendung finden.

Die samplegenaue ein- und mehrkanalige Audio-Wiedergabe und Aufnahme realisiert `AKio.m` unter Verwendung von `playrec` [4] und `pa-wavplay` [5]. In Kombination mit der Sweepsynthese im Frequenz- oder Zeitbereich (`AKsweepFD.m`, `AKseepTD.m`) und der spektralen Entfaltung (`AKdeconv.m`) ermöglicht dies darüber hinaus eine vielseitig einsetzbare Messung vom Impulsantworten z.B. für die Identifikation von nicht-linearen Verzerrungsprodukten mit Hilfe exponentieller Sweeps [6], für die adaptive Schätzung von Impulsantworten mit Hilfe perfekter sweeps [7], oder eine Messung mit konstantem Signal-Rausch-Abstand unter Verwendung spektral beliebig gefärbter Sweeps [8].

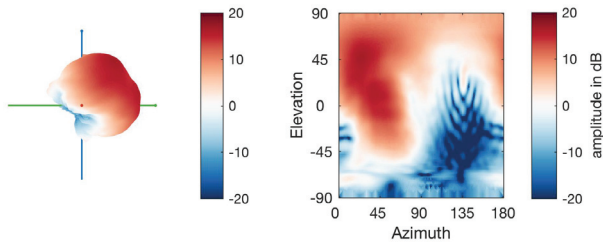
Neben der Aufnahme von Audiodaten spielt die Inspek-



(a)



(b)



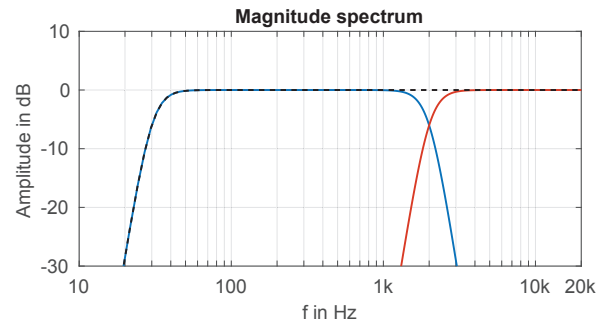
(c)

Abbildung 1: Beispiele für mit `AKp.m` erzeugte Grafiken unter Verwendung kopfbezogener Impulsantworten (HRIRs) und Übertragungsfunktionen (HRTFs) des FABIAN Kopf und Torso-Simulators [9]. (a) HRIRs und HRTFs für das linke (blau) und rechte (rot) Ohr und eine Quelle links des Hörers. (b) HRIRs und HRTFs des linken Ohres in der Horizontalebene (0° Azimuth entspricht einer Quelle vor dem Hörer, 90° einer Quelle links des Hörers). (c) Sphärische Plots des logarithmierten HRTF Betragsspektrums bei 6 kHz (90° Elevation entsprechen einer Quelle über dem Hörer, -90° einer Quelle unterhalb des Hörers).

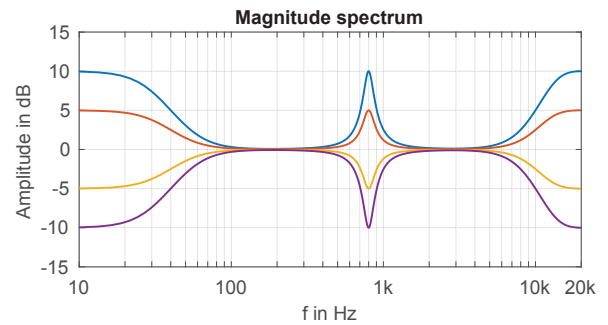
tion, bzw. Verifikation eine zentrale Rolle. Hierfür bieten `AKf.m` und `AKp.m` eine Vielzahl möglicher Darstellungen an, die neben einfachen Plots von Signalen im Zeit- und Frequenzbereich (Abb. 1a) auch Abbildungen zweidimensionaler (Abb. 1b) und dreidimensionaler Datensätze (Abb. 1c) enthalten. Der Matlab-Code zur Erzeugung aller in diesem Manuskript verwendeten Grafiken befindet sich im Anhang.

Eine einfache Form der Weiterverarbeitung akustischer Signale ist die Filterung. Hierfür stellt `AKfilter.m` Hoch- und Tiefpässe, Bandpässe und Bandsperren, Cross-Over Filter, Oktavfilter, sowie parametrische Equalizer und High- und Low-Shelves zur Verfügung (siehe Abb. 2).

In der akustischen Praxis wird zudem oftmals die Invertierung von Übertragungsfunktionen benötigt, sei es für die Entzerrung von Kopfhörern in der



(a)



(b)

Abbildung 2: Beispiele für mit `AKfilter.m` erzeugte Filter. (a) Linkwitz-Riley Hoch- und Tiefpässe achter Ordnung (blau, rot) sowie deren Addition (schwarz, $f_c = \{40, 2000\}$ Hz). (b) Low- und High-Shelves zweiter Ordnung ($f_c = \{40, 10.000\}$ Hz, Gain $\{-10, -5, 5, 10\}$ dB), sowie parametrische Equalizer mit identischen Gains ($f_c = 800$ Hz, $Q = 4$).

Binauraltechnik oder Lautsprechern in der Raumakustik. `AKtools` bietet hierfür mit der Funktion `AKregulatedInversion.m` die regularisierte Invertierung an, mit deren Hilfe Übertragungsfunktionen auf eine minimal- oder linearphasiges Zielfunktion hin entzerrt werden können. Zudem kann mittels frequenzabhängiger Regularisierung die Genauigkeit der Invertierung bei Bedarf beschränkt werden, beispielsweise um große und steilflankige Anhebungen in der invertierten Übertragungsfunktion zu vermeiden [10]. Abb. 3 zeigt die in der Invertierung involvierten Übertragungsfunktionen und das Entzerrungsergebnis für eine gemessene Kopfhörerübertragungsfunktion (`HpTF`). In diesem Fall wurde die exakte Kompensation tiefer, aber potentiell unhörbarer Einbrüche in der `HpTF` bei ca. 9 kHz und 13 kHz durch entsprechende Regularisierung verhindert. Ebenfalls häufige Anwendung in der akustischen Signalverarbeitung findet die Spherical Harmonics Transformation (SHT), die sphärische Datensätze in eine gewichtete Überlagerung orthogonaler Kugel-Basisfunktionen überführt. Neben einer erheblichen Datenreduktion bietet dies den Vorteil einer winkelkontinuierlichen Darstellung. Mit den `AKtools` Funktionen `AKsht.m` und `AKisht.m` lässt sich die diskrete SHT (Gl. (3.34) in [11]) auf komplexe Spektren, sowie getrennt auf Betrags- und Phasenspektren anwenden. Ein Beispiel einer kopfbezogenen Übertragungsfunktion vor und nach SHT ist in Abb. 4 dargestellt.

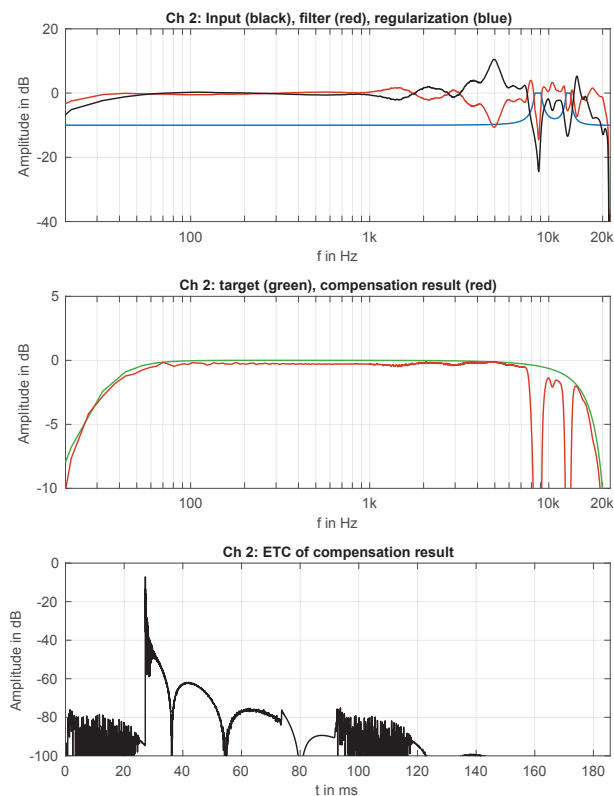


Abbildung 3: Regularisierte Invertierung am Beispiel einer Kopfhörerübertragungsfunktion (HpTF). Oben: HpTF (schwarz), invertierte HpTF (rot) und Regularisierung (blau). Mitte: Zielbandpass (grün) und Ergebnis (Faltung von HpTF und invertierter HpTF, rot). Unten: Zeitbereichsdarstellung des Ergebnisses ($x[n]^2$).

Verfügbarkeit

Aktools stehen unter der European Public Licence (EURL) zur Verfügung, die eine uneingeschränkte Verwendung ermöglicht, und können unter www.ak.tu-berlin.de/aktools heruntergeladen werden. Die Versionierung erfolgt mittels eines ebenfalls öffentlich zugänglichen SVN repositories, das über den zentralen IT-Dienst der TU Berlin gesichert wird (Backup). Aktools benötigen Matlab 2013 oder höher inklusive der Signal Processing und Statistics and Machine Learning Toolbox, wobei ein Großteil der Funktionalität auch ohne zusätzliche Toolboxes gewährleistet ist. Für einige Demos, sowie die Interpolation von HRIRs wird der FABIAN Datensatz benötigt [9].

Danksagung

Aktools enthält Beiträge von: Alexander Lindau, Andreas Rotter, André Giese, David Ackermann, Daniel Kahlhöfer, Dmitry Grigoriev, Fabian Brinkmann, Frank Schultz, Hannes Helmholz, Marc Voigt, Omid Kokabi, Zora Schärer Kalkandjiev. Aktools verwendet Code von: Matt Frear und Josphe Desloge (pa_wavplay), Robert Humphrey (playrec), Piotr Majdak *et al.* (SOFA Matlab API), Malcolm Slaney (auditory toolbox), Carlos Adrian and Vargas Aguilera (cbfreeze, cbhandle), Doug Hall (makeColormap), Duane Hanselman (mmpolar), Fe-

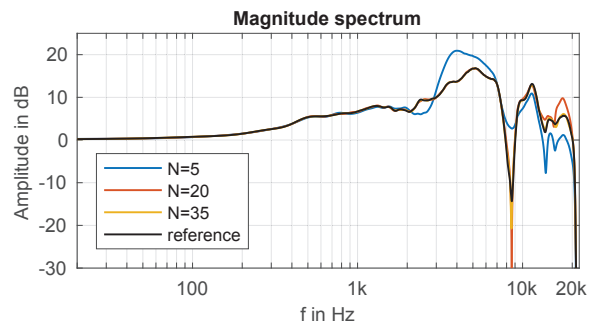


Abbildung 4: Kopfbezogene Übertragungsfunktion für eine Quelle in der Horizontalebene und das linke Ohr ohne und mit Ordnungsbegrenzung nach Spherical Harmonics Hin- und Rücktransformation.

lipe G. Nievinski and Pekka Kumpulainen (subtightplot), John Burkhardt (j_polynomial), color specifications and designs by Cynthia Brewer (colorbrewer.org). Aktools wurde inspiriert von der ITA-Toolbox [12].

Anhang

```
% This appendix demonstrates some AKtools basics.
% To use AKtools download it from:
% ak.tu-berlin.de/aktools, navigate to the AKtools
% folder in Matlab, and run AKtoolsStart.m
% This code was tested with AKtools (revision 25),
% and Matlab R2016a

% ----- PLOTTING
% get HRIRs on an evenly sampled grid
g = AKgreatCircleGrid(-90:2:90, 2, 90);
[1, r] = AKhrirInterpolation(g(:,1), g(:,2), 0, ...
    'measured_ir');

% get and plot HRIR to the left of the listener
id = AKsubGrid(g, 'any', [90 0]);

AKf(18,9)
AKp([1(:,id) r(:,id)])
print('-dpdf', 'figure1a')

% get and plot the horizontal plane
id = AKsubGrid(g, 'transverse', 0);

AKf(18,6)
subplot(1,2,1)
AKp(1(:,id), 't3d', 'y', g(id,1), 'dr', ...
    [-1.5 1.5], 'x', [0 2])
xlabel 'azimuth in deg.'
subplot(1,2,2)
AKp(1(:,id), 'm3d', 'y', g(id,1), 'dr', ...
    [-20 20])
xlabel 'azimuth in deg.'

print('-dpdf', 'figure1b')

% plot spherical HRTF dataset at 6 kHz
AKf(18,6)
subplot(1,2,1)
AKp(1, 's2', 'az', g(:,1), 'el', g(:,2), ...
    'dr', [-20 20], 'sph-f', 6000, ...
    'labeling', 'off', 'hp-view', [90 0])
subplot(1,2,2)
AKp(1, 's5', 'az', g(:,1), 'el', g(:,2), ...
    'dr', [-20 20], 'sph-f', 6000)
set(gca, 'xTick', 1:90:361)
```

```

print('-djpeg', '-r300', 'figure1c')

%% ----- FILTERING
% generate dirac signals for filtering
x = AKdirac(2^14, 1);

%% apply high-pass, and cross-over
y = AKfilter(x, 'hp', 30, 0, 44100, 8, 'LR');
y = AKfilter(y, 'xover', 2000, 0, 44100, 8, 'LR');

AKf(12,6)
AKp(squeeze(y), 'm2d', 'x', [10 20000])
AKp(sum(y,3), 'm2d', 'x', [10 20000], 'dr', ...
    [10 -30], 'dash', 1)
print('-dpdf', 'figure2a')

%% apply low/high-shelves, and parametric EQs
y = AKfilter(x, 'ls', 40, [10 5 -5 -10], ...
    44100, 2, 'mid');
y = AKfilter(y, 'hs', 10000, [10 5 -5 -10], ...
    44100, 2, 'mid');
y = AKfilter(y, 'peq', 800, [10 5 -5 -10], ...
    44100, 4, 'hpl', 'cos');

AKf(12,6)
AKp(y, 'm2d', 'x', [10 20000], 'dr', [-15 15])
print('-dpdf', 'figure2b')

%% ----- SPHERICAL HARMONICS TRANSFORM
% change coordinate convention for elevation
g2 = g;
g2(:,2) = 90 - g2(:,2);

% spherical harmonics transform
% (left ear HRIRs, SH order N=35)
f_nm = AKsht(1, true, g2, 35, 'complex');

%% inverse transform, different truncation orders
% (source on horizontal plane, 90 deg. to the left)
N = [5 20 35];
for nn = 1:numel(N);
    l_sh(:,nn) = AKisht(f_nm(1:(N(nn)+1)^2,:), ...
        true, [90 90], 'complex');
end

%% plot reference and order truncated HRTFs
id = AKsubGrid(g, 'any', [90 0]);

AKf(12,6)
AKp(l_sh, 'm2d', 'N', 4410)
AKp(1(:,id), 'm2d', 'N', 4410, 'dr', [-30 25])
legend('N=5', 'N=20', 'N=35', 'reference', ...
    'location', 'SouthWest')
print('-dpdf', 'figure4')

%% ----- REGULATED INVERSION
% run demo script - figure 3 is saved automatically
AKregulatedInversionDemo

```

- [4] R. Humphrey, “Playrec – Multichannel Matlab audio”, (2008), URL <http://www.playrec.co.uk>.
- [5] J. G. Desloge, “pa-wavplay for 32-bit and 64-bit”, (2014), URL <https://de.mathworks.com/matlabcentral/fileexchange/47336-pa-wavplay-for-32-bit-and-64-bit>.
- [6] A. Novák, L. Simon, F. Kadlec, and P. Lotton, “Nonlinear system identification using exponential swept-sine signal”, *IEEE Trans. on Instrumentation and Measurement* **59**, 2220–2229 (2010).
- [7] C. Antweiler, A. Telle, P. Vary, and G. Enzner, “Perfect-sweep NLMS for time-variant acoustic system identification”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Prague, Czech Republic) (2011).
- [8] S. Müller and P. Massarani, “Transfer function measurement with sweeps. directors cut including previously unreleased material and some corrections”, *J. Audio Eng. Soc. (Original release)* **49**, 443–471 (2001).
- [9] F. Brinkmann, A. Lindau, S. Weinzierl, G. Geissler, S. van de Par, M. Müller-Trapet, R. Opdam, and M. Vorländer, “The FABI-AN head-related transfer function data base”, <http://dx.doi.org/10.14279/depositonce-5718> (2016).
- [10] S. G. Norcross, M. Bouchard, and G. A. Soulodre, “Inverse filtering design using a minimal phase target function from regularization”, in *121th AES Convention, Convention Paper 6929* (San Francisco, USA) (2006).
- [11] B. Rafaely, *Fundamentals of spherical array processing*, 1st edition (Springer, Berlin, Heidelberg, Germany) (2015).
- [12] P. Dietrich, M. Guski, M. Pollow, M. Müller-Trapet, B. Masiero, R. Scharrer, and M. Vorländer, “ITA-Toolbox – An open source Matlab toolbox for acousticians”, in *Fortschritte der Akustik – DAGA 2012*, 151–152 (Darmstadt, Germany) (2012).

Literatur

- [1] P. Vandewalle, J. Kovačević, and M. Vetterli, “Reproducible research in signal processing – what, why, and how”, *IEEE Signal Processing Magazine* **26**, 37–46 (2009).
- [2] Technical University Berlin, “Deposit Once – repository for research data and publications”, (2016), URL <https://depositonce.tu-berlin.de>.
- [3] S. Weinzierl, “An open repository for research data in acoustics (OPERA)”, in *Fortschritte der Akustik – DAGA 2017* (Kiel, Germany) (2017).