

Praktikumsbericht:

Implementing Methods for Automatic Volume Equalization

Projektpraktikum (10h)

Institut: Department of Computational Perception, <http://www.cp.jku.at/>

Betreuer: Dipl. Ing. Klaus Seyerlehner, <http://www.cp.jku.at/people/seyerlehner/>

Durchgeführt von: Ulrich Brandstätter, 0055055, 921

Inhaltsverzeichnis

Inhalt

| | | |
|-------------------------------|-------|-----------|
| Kurzfassung | | 1 |
| Begriffsklärung | | 2 |
| Teilprobleme | | 4 |
| Filter Entwurf | | 7 |
| Realisierung | | 13 |
| Vergleich der Methoden | | 20 |
| Validierung | | 24 |

Anhang

| | | |
|---|-------|-----------|
| Filter Koeffizienten und Amplitudenantworten | | 26 |
| Code Auszug | | 31 |
| Referenzen | | 33 |

Inhalt

Kurzfassung

Gesetztes Ziel dieses Projektpraktikums ist die Entwicklung eines Programms / Plugins, welches für einen gegebenen Audioclip abhängig vom gewählten Lautheitsmodell (Frequenzgewichtung) einen Wert für die empfohlenen Lautstärkenanpassung errechnet. Darüberhinaus sollte das Programm / Plugin in der Lage sein, einerseits den gefilterten Audiostream für etwaige Weiterverarbeitung abzuspeichern, andererseits die aktuelle Lautstärke / empfohlene Lautstärkenanpassung zu visualisieren.

Realisiert wurde diese Zielsetzung durch die Implementierung zweier Komponenten:

- Plugin: Die Plugin-Komponente enthält die Berechnungsvorschriften mehrerer gängiger Lautheitsmodelle sowie Verfahren für deren Anwendung. Diese Komponente wurde gemäß dem VAMP-Standard [1] realisiert. SonicVisualizer [3] kann das Plugin laden und die Analyse-Ergebnisse darstellen.
- Plugin-Host: Ausführbares Programm (Command-Line basiert), welches die Plugin-Komponente lädt und auf Audio-Dateien anwendet.

Es wurde darauf geachtet, daß eine Vielzahl an unterstützten Lautheitsmodellen bereitgestellt werden, die wiederum mit unterschiedlichen Samplingfrequenzen umgehen können. Darüberhinaus können mehrere Merkmale für jeden Clip berechnet werden, unter anderen die empfohlene Lautheitsanpassung oder der Lautheits-Spitzenwert.

Begriffsklärung

Lautheit

Als Lautheit versteht man einen subjektiv empfundenen Eindruck der Lautstärke, verursacht durch Veränderungen des Schalldruckpegels. Dabei ist die empfundene Lautstärke frequenzabhängig. Gleiche Schalldruckpegel werden vom menschlichen Ohr bei unterschiedlichen Frequenzen mit unterschiedlicher Intensität wahrgenommen (Abbildung 1).

Auf Töne im Frequenzbereich von etwa 2700 – 3200 Hertz reagiert das Ohr besonders sensibel.

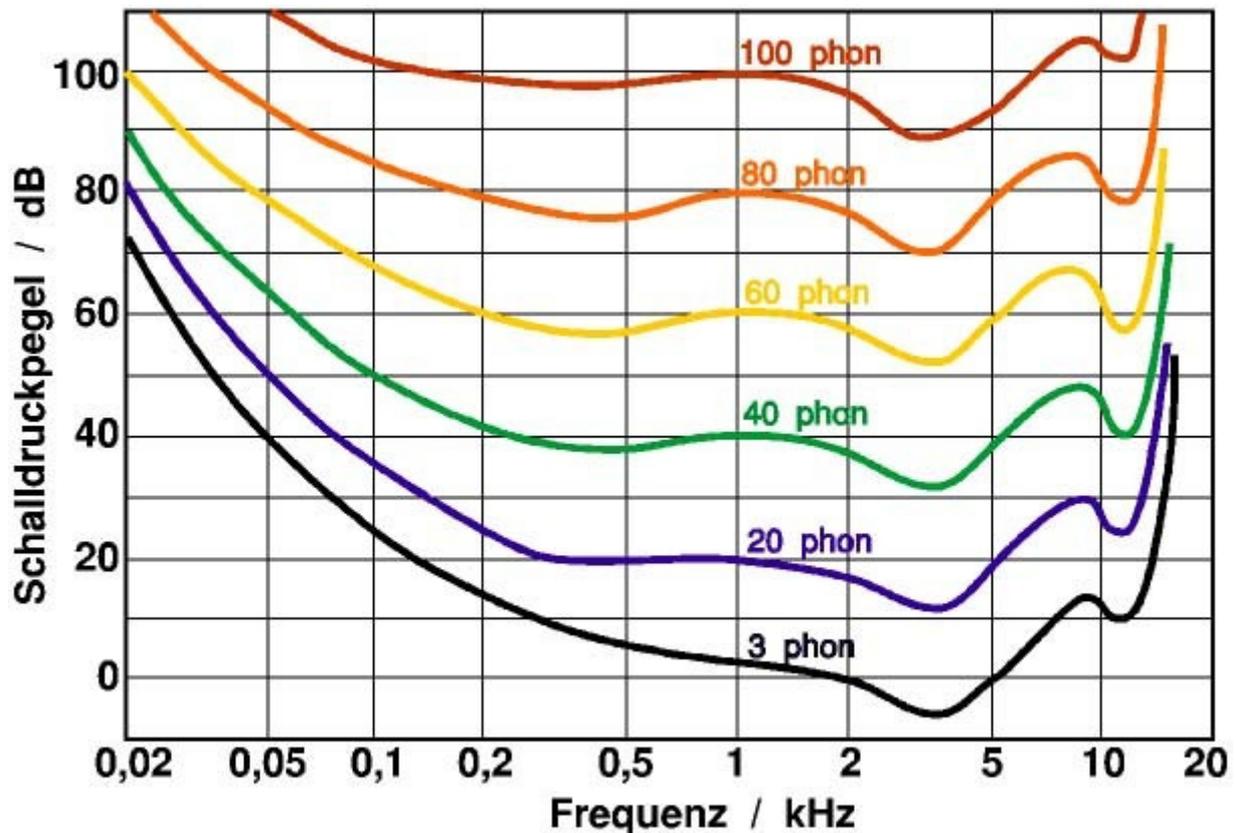


Abbildung 1: Phon Konturen

Einheiten der Lautheit

- **Schalldruckpegel**

Das menschliche Ohr ist speziell auf Veränderungen im Schalldruck besonders sensibel und verhält sich proportional zur Schallintensität. Bei Lautheits-Angaben wird im Allgemeinen der Logarithmus verwendet.

Der Schalldruckpegel errechnet sich folgendermassen:

$$\text{SPL} = 20 \times \log_{10} (P / P_0) \text{ dB}$$

wobei:

SPL = Sound pressure level

P_0 = Referenzschalldruck =
= $0.00005 \times (2 \times 10^{-5})$ newtons / m² =
= Hörschwelle bei 1000Hz

Die Einheit des Schalldruckpegels ist Dezibel (dB).

- **Phon**

Die Einheit Phon leitet sich direkt von den Fletcher / Munson Kurven (= Phon Konturen, siehe Abbildung 1) ab. Normiert ist die Einheit Phon auf den Dezibel-Wert, der den Phon-Konturen bei einem Kilohertz entnommen werden kann. Phon Konturen gestatten den frequenzabhängigen Vergleich der Lautstärken von Tönen (z.B. sei ein Sinus-Ton mit 1 kHz Frequenz und Intensität von 40 dB Schalldruckpegel gegeben – die Lautheit beträgt 40 Phon. Gesucht sei der Schalldruckpegel eines Sinus mit 100 Hertz, der gleich laut ist. Dazu muss lediglich die Phon-Kontur bei dem Schalldruckpegel von 40 dB nach links verfolgt werden, bis sie 100 Hertz schneidet – der gesuchte SPL ist in diesem Fall etwa 62 dB – siehe Abbildung 1).

- **Sone**

Ein Sone entspricht der Lautstärke von 40 Phon – einem 1 kHz Sinus Signal mit einem SPL von 40 dB. Die Besonderheit dieser Einheit ist, daß eine Verdopplung der Sone einer Verdopplung der Lautstärke entspricht.

- **dB(A), dB(C), dB(RLB), ...**

Für viele Anwendungen stellen weder Phon noch Sone ein geeignetes Maß dar: Einerseits sind beide Maße auf Sinus-Töne geeicht, die komplexeren Klängen nur ungenügend gerecht werden. Andererseits sind auch die Längen der Klänge unreflektiert: Lang anhaltende, konstante Töne mit gleich bleibender Intensität können vom menschlichen Ohr mit abnehmender Lautstärke wahrgenommen werden (speziell dann, wenn sie dem Hörer vertraut sind). Sehr kurze Töne können vom menschlichen Ohr unter Umständen auch dann wahrgenommen werden, wenn sie unterhalb der Gehörschwelle liegen. Deshalb wurden Gewichtungs-Verfahren entwickelt, die anwendungsspezifisch gemäß der Beschaffenheit des menschlichen Ohrs ein zu messendes Geräusch frequenzabhängig verstärken oder abschwächen, sprich gewichten. Mehrerer dieser Verfahren (A-, C-, RLB- und die Gewichtung nach ReplayGain) wurden im Zuge dieses Praktikums implementiert.

Teilprobleme

Für die Umsetzung dieses Praktikums ist die Lösung dreier Teilprobleme erforderlich:

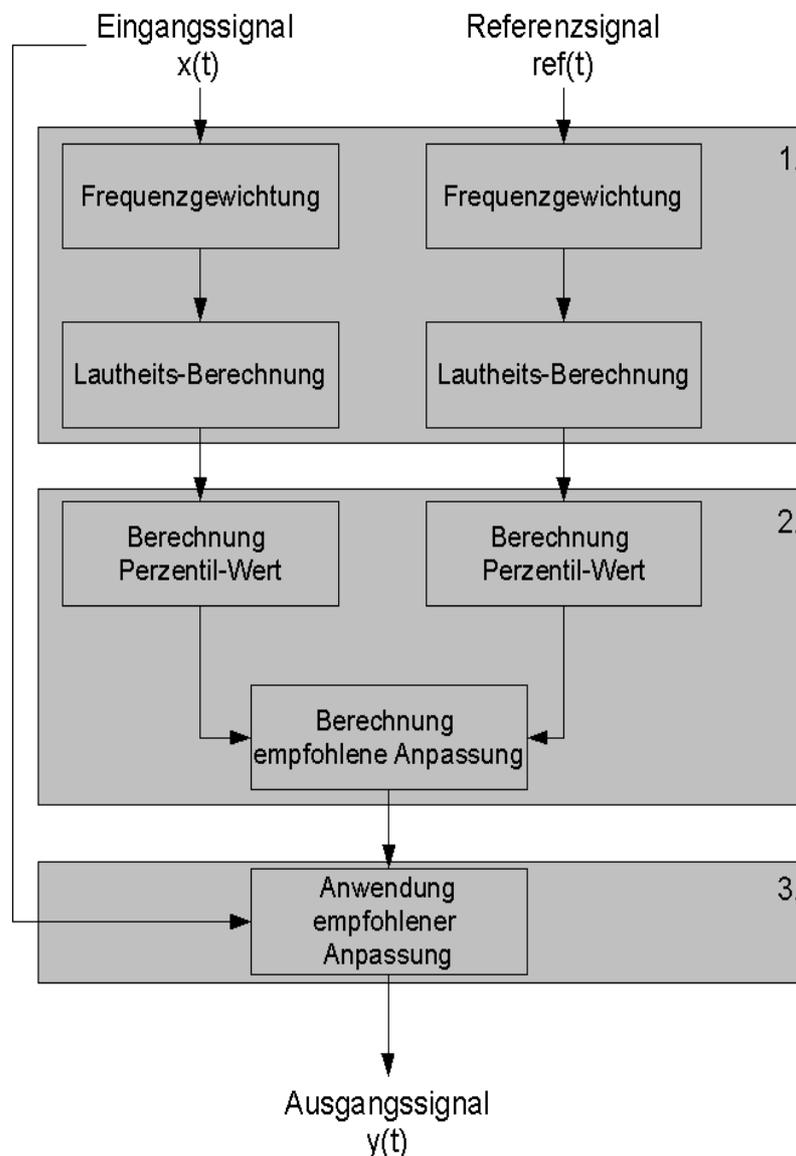


Abbildung 2: Die drei Teilprobleme als Blockdiagramm

1. Berechnung und Anwendung der Frequenzgewichtung eines Clips nach verschiedenen Modellen und anschließende Extraktion des Lautheits-Wertes

Zur Gewinnung der gewichteten Lautheit wird folgendermaßen vorgegangen (Skovenberg [5]):

- Frequenzgewichtung des Eingangssignals

Die Frequenzgewichtung wird durch Anwendung der gewählten Gewichtung entsprechender IIR-Filter realisiert (siehe Kapitel Filter-Entwurf).

- Berechnung der Lautheit des gewichteten Eingangssignals nach folgender Formel:

$$L_{eq} = 1_khz_add_konstant + 10 * \log_{10} ((\text{sum}(y(t)) / T) / 1_khz_ref_konstant)$$

wobei:

| | |
|-----------------------------|---|
| $y(t)$... | Frequenzgewichtetes Eingangssignal |
| T ... | Anzahl der Samples des Eingangssignals |
| $\text{sum}(y(t)) / T$... | RMS (= root mean square) Leistung des gewichteten Eingangssignals |
| $1_khz_ref_konstant$... | RMS Leistung eines Referenzsignals |
| $1_khz_add_konstant$... | Normierungskonstante |

- Normalisierung der errechneten Lautheit

Jedes Verfahren zur Frequenzgewichtung sollte bei 1 kHz denselben Pegel aufweisen. Hierfür wird üblicherweise ein 1-Kilohertz Testsignal eingespielt, wodurch etwa die Normierungskonstante ($1_khz_add_konstant$) zum Referenzpegel ermittelt werden kann oder aber die Stärke des benötigten Referenzsignals ($1_khz_ref_konstant$), oder beide in Kombination. Zum Zwecke der Validierung der bei den Plugins verwendeten Referenzleistungen und Normierungskonstanten wurde ein Testsample mit einer Sinusschwingung von 1 kHz und stufenweise absteigendem Pegel erzeugt und analysiert (Abbildung 3): Wie gewünscht liefern sämtliche Gewichtungsverfahren dieselben Lautstärken.

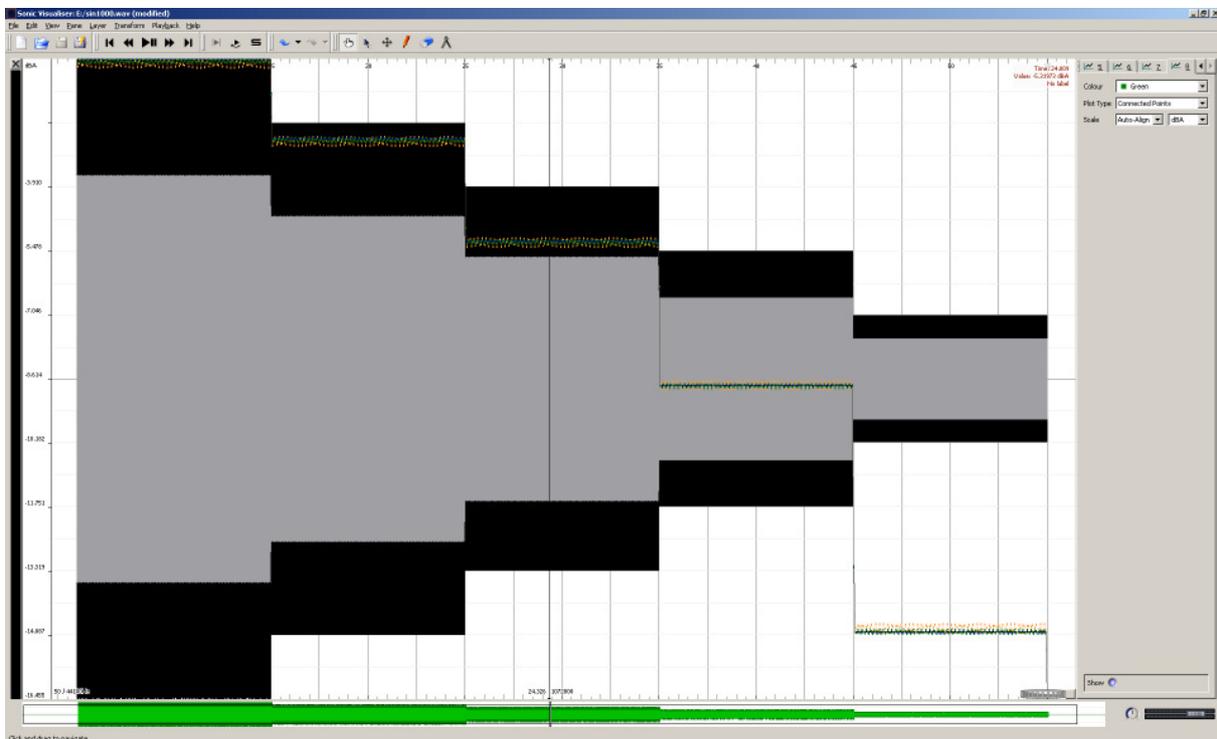


Abbildung 3: Sämtliche Filter, ausgeführt mit SonicVisualizer, angewandt auf 1kHz Sinus mit absteigendem Pegel

2. Gewinnung einer Maßzahl für die empfohlene Lautstärkenanpassung

Die empfohlene Lautstärkenanpassung wird durch den Vergleich der Lautheit des Eingangssignals mit der Lautheit eines Referenzsignals ermittelt:

- Für die Kalibrierung sämtlicher bereitgestellter Gewichtungsmethoden wurde ein Signal mit sog. Rosa Rauschen verwendet, welches etwa -20 dB laut ist, also einem im Allgemeinen angenehmen Pegel entspricht. Rosa Rauschen hat die Eigenschaft, daß alle Frequenzen des Schallspektrums gleich laut empfunden werden, was es zum idealen generierbaren Referenzsignal macht. Zwar ist diese Kalibrierung sozusagen als Preset fix implementiert, kann aber nach Bedarf verändert werden.
- Für den Vergleich der Lautstärken werden Perzentil-Werte verwendet: Sämtliche errechnete Lautstärken werden sortiert, jener, der dem einstellbaren Perzentil-Wert (Standard-Einstellung ist 95 Prozent) entspricht wird herangezogen. Alternativ könnten auch Spitzen- oder Durchschnittswerte verwendet werden, allerdings wird die empfundene Lautheit durch den Perzentil-Wert am besten wiedergegeben (nach Zwicker [6] ist der 95 Prozent Perzentil-Wert besonders geeignet).

3. Anwendung der Maßzahl zur tatsächlichen Lautstärkenanpassung

Die berechnete empfohlene Lautstärkenanpassung wird schließlich über eine einfache Signalverstärkung angewandt: Jedes Sample wird gemäß folgender Formel skaliert:

$$y(t) = x(t) * 10^{\text{recommendation} / 20}$$

wobei:

| | |
|-------------------|---|
| $x(t)$.. | Eingangssignal |
| recommendation .. | Vom Plugin errechnete empfohlene Lautheitsanpassung |
| $y(t)$.. | Ausgabesignal |

Abhängig von den analysierten Samples kann es hier zu Clipping kommen: Ist ein Sample durchwegs leise (etwa 1000 Sekunden Stille), aber an wenigen Stellen recht laut (etwa 5 Sekunden mit 98% Aussteuerung), so können diese wenigen lauten Stellen unter Umständen über die Vollaussteuerung hinaus verstärkt werden.

Filter Entwurf

Die im Zuge dieses Praktikums entwickelte Software stellt folgende Gewichtungsmethoden bereit:

- A Gewichtung
- C Gewichtung
- RLB Gewichtung
- Gewichtung wie im ReplayGain Projekt [8] vorgeschlagen
- Ungewichtet

Sämtliche Gewichtung wurden als IIR Filter (infinite impulse response filter – Filter unendlicher Impulsantwort) realisiert. Die im VAMP-Plugin verwendeten Koeffizienten wurden zum Teil den Standards entnommen ([9], [10], [13]), zum Teil mit Octave [15] berechnet, zum Teil selbst entwickelten Filtern (RLB) entnommen. A und C Gewichtung wurden über einen einzelnen IIR Filter realisiert, RLB und ReplayGain kaskadieren zwei. Generell haben die verwendeten IIR Filter unterschiedliche Ordnung – die Ordnung eines IIR Filters bestimmt die Anzahl der Filterkoeffizienten.

Allgemein läßt sich ein IIR Filter der Ordnung n wie folgt berechnen [7]:

$$y[n] = (a_0 * x[n]) + \dots + (a_m * x[n - M]) - (b_1 * y[n]) - \dots - (b_n * y[n - N])$$

wobei

| | |
|-----------------------|---------------------------------|
| $a_0 \dots a_m \dots$ | A-Koeffizienten des IIR Filters |
| $b_1 \dots b_n \dots$ | B-Koeffizienten des IIR Filters |
| $x[n] \dots$ | Signaleingang bei Sample n |
| $y[n] \dots$ | Signalausgang bei Sample n |

Teil der Problemstellung dieses Praktikums ist die Bereitstellung der Gewichtungsmethoden für unterschiedliche Samplingfrequenzen. Folglich müssen entsprechende Filterkoeffizienten für die unterstützten Samplingfrequenzen (32000, 44100 und 48000 Hertz) gefunden werden. Bei der ReplayGain Gewichtung [8] wurden diese der Homepage entnommen. Für die Berechnung der Koeffizienten der A und C Gewichtung wurden Matlab-Skripte [14] für die unterstützten Samplingfrequenzen ausgeführt und die gesuchten Koeffizienten ausgelesen. Unglücklicherweise enthält der ITU-Standard für die RLB Gewichtung keine Berechnungsvorschrift zur Koeffizientenberechnung, lediglich die gewünschten Amplitudenantworten und die Koeffizienten für die Samplingfrequenz 48000 Hertz sind darin enthalten. Deswegen wurde die RLB Gewichtung mit eigenen Filtern realisiert (in einer früheren Version mit einem Online-Tool zur Filterentwicklung [12]):

RLB Filter

Der ITU-Vorschlag (International Telecommunication Union, weltweite Organisation, die sich mit den technischen Aspekten der Telekommunikation beschäftigt) für den RLB Filter, ITU-R BS.1770-1 [9], beinhaltet einen Implementierungsvorschlag, der zwei IIR Filter kaskadiert. Neben den im Standard definierten Amplitudenantworten (für jeweils einen der IIR Filter, die kombinierte ist leider nicht enthalten) sind außerdem die Filterkoeffizienten für die Samplingrate von 48 kHz gegeben. Die gegebenen Koeffizienten wurden für 48 kHz in die Software übernommen, für die

anderen Samplingraten (32000 und 44100 Hertz) mußten eigene IIR-Filter entwickelt werden, die die Amplitudenantworten [11] des Verfahrens bestmöglich nachbilden.

Eine der zwei gegebenen Amplitudenantworten stellt einen Hochpaß-Filter dar:

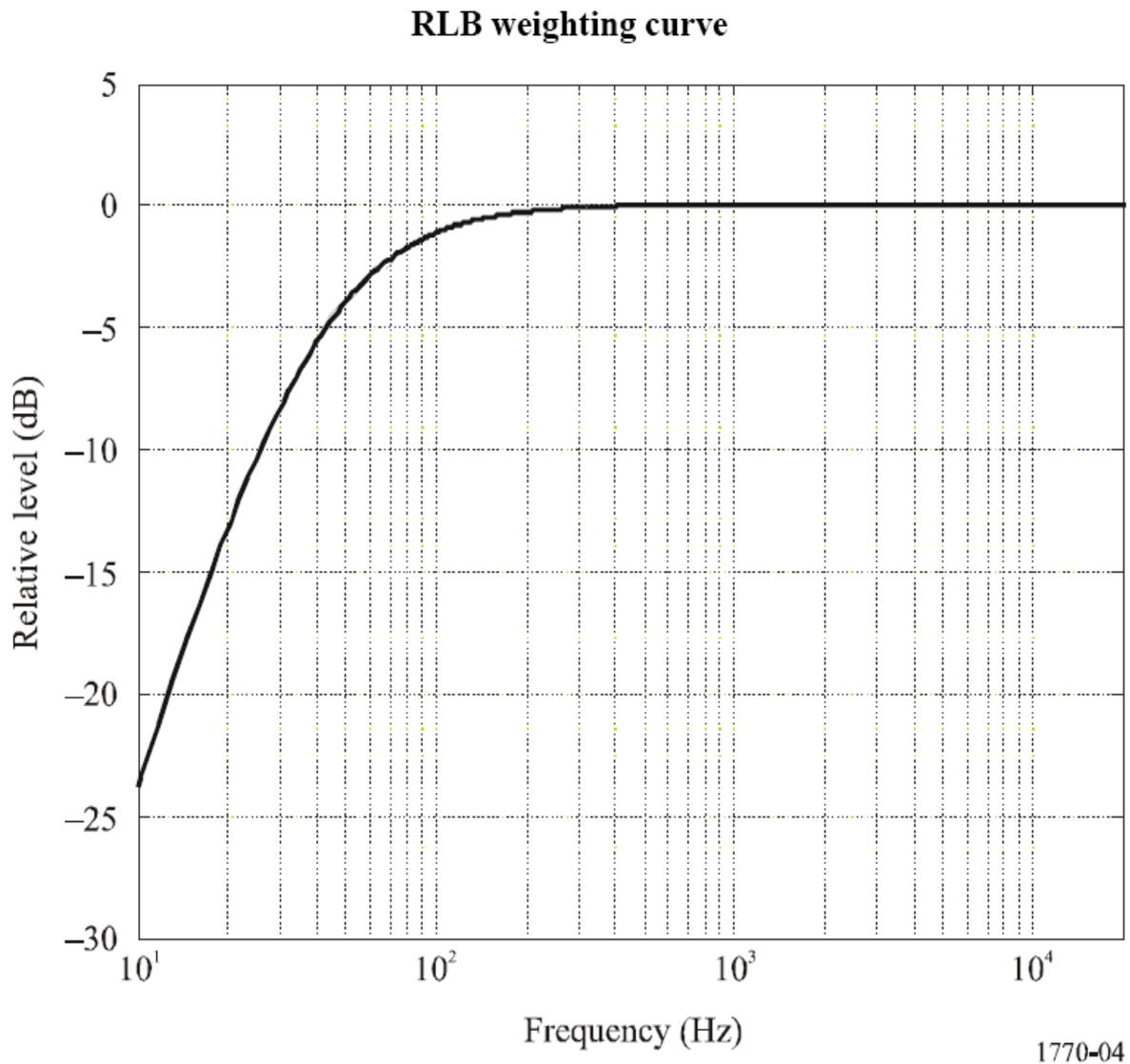


Abbildung 4: Hochpaß Filter des RLB Verfahrens (Abbildung entnommen aus [9])

Dieser Amplitudenantwort kann man drei wesentliche Charakteristika für den Filterentwurf entnehmen (Abbildung 4 und 5):

- Der Hochpaß-Bereich verläuft quasi linear
- Die Kennfrequenz (cut-off frequency) liegt bei 46 Hertz (Schnittpunkt der Verlängerungen beider „Geraden“ rot und grün, violett abgelesen; die Differenz in der Amplitude von Schnittpunkt und Kurve beträgt 3.01 dB)
- Die Verstärkungsabnahme beträgt 20 dB pro Dekade (10 – 20 Hz, türkis abgelesen)

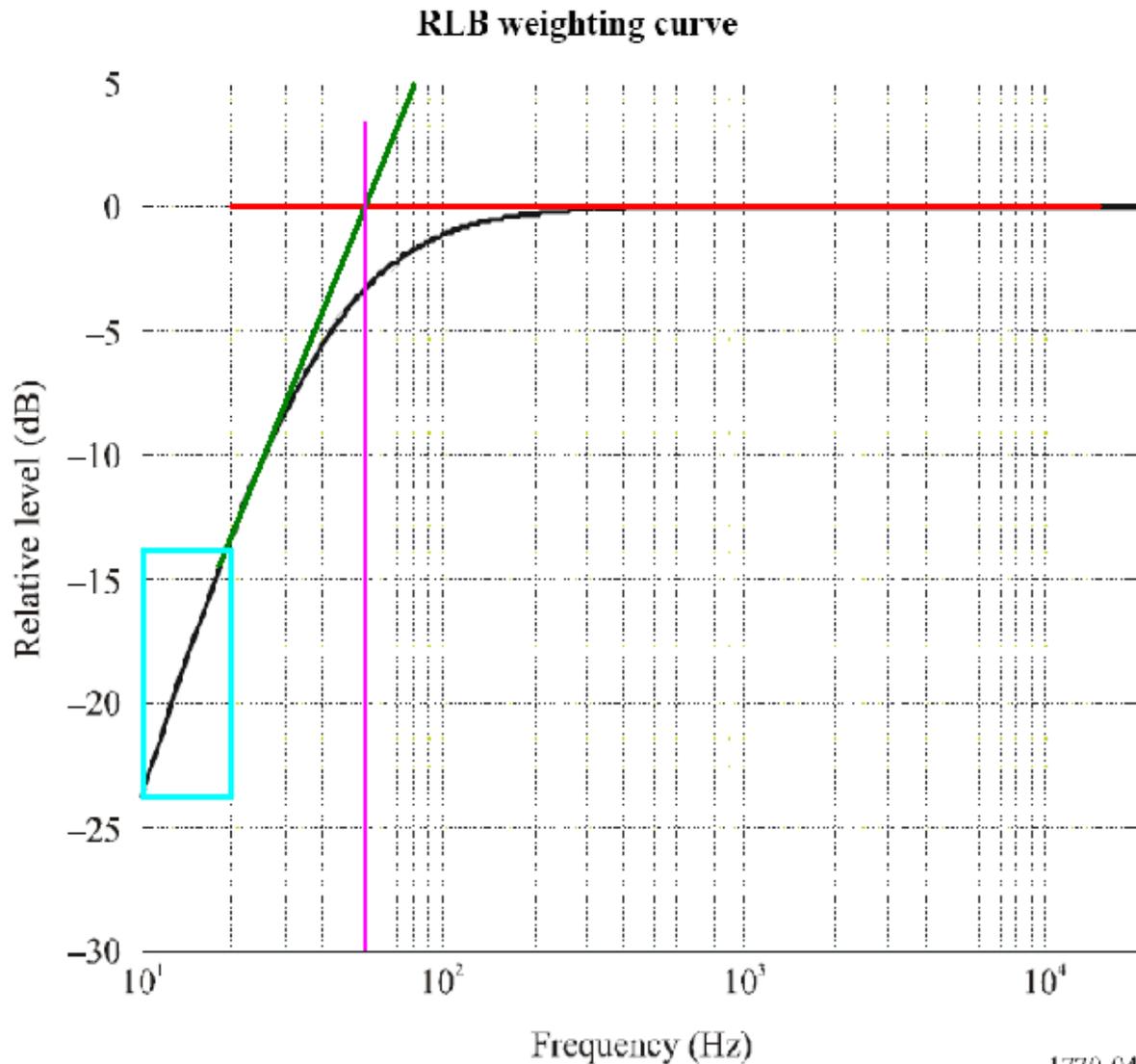


Abbildung 5: Hochpaß Filter des RLB Verfahrens (Abbildung entnommen aus [9]) mit Heraushebungen

Diese Charakteristika lassen sich perfekt auf einen Butterworth-Filter übertragen. Die Octave-Funktion zur Erstellung eines solchen Filters, „butter“, benötigt drei Parameter:

- Ordnung des Filters
- Kennfrequenz
- Art des Filters (Hochpaß / Tiefpaß / Bandpaß / Bandsperre)

Die Art des Filters wurde bereits als Hochpaß-Filter identifiziert. Die Kennfrequenz wurde mit 46 Hertz abgelesen. Der letzte Parameter, die Filterordnung, läßt sich aufgrund der Verstärkungsabnahme pro Dekade bestimmen: Beim Butterworth-Filter bestimmt die Ordnung den Faktor für die Verstärkungsabnahme:

| |
|--|
| $\text{Ordnung} * 20 \text{ dB} = \text{Verstärkungsabnahme pro Dekade}$ |
|--|

Nachdem die Abnahme in der vorliegenden Amplitudenantwort exakt 20 dB beträgt, ist die zu wählende Filterordnung eins.

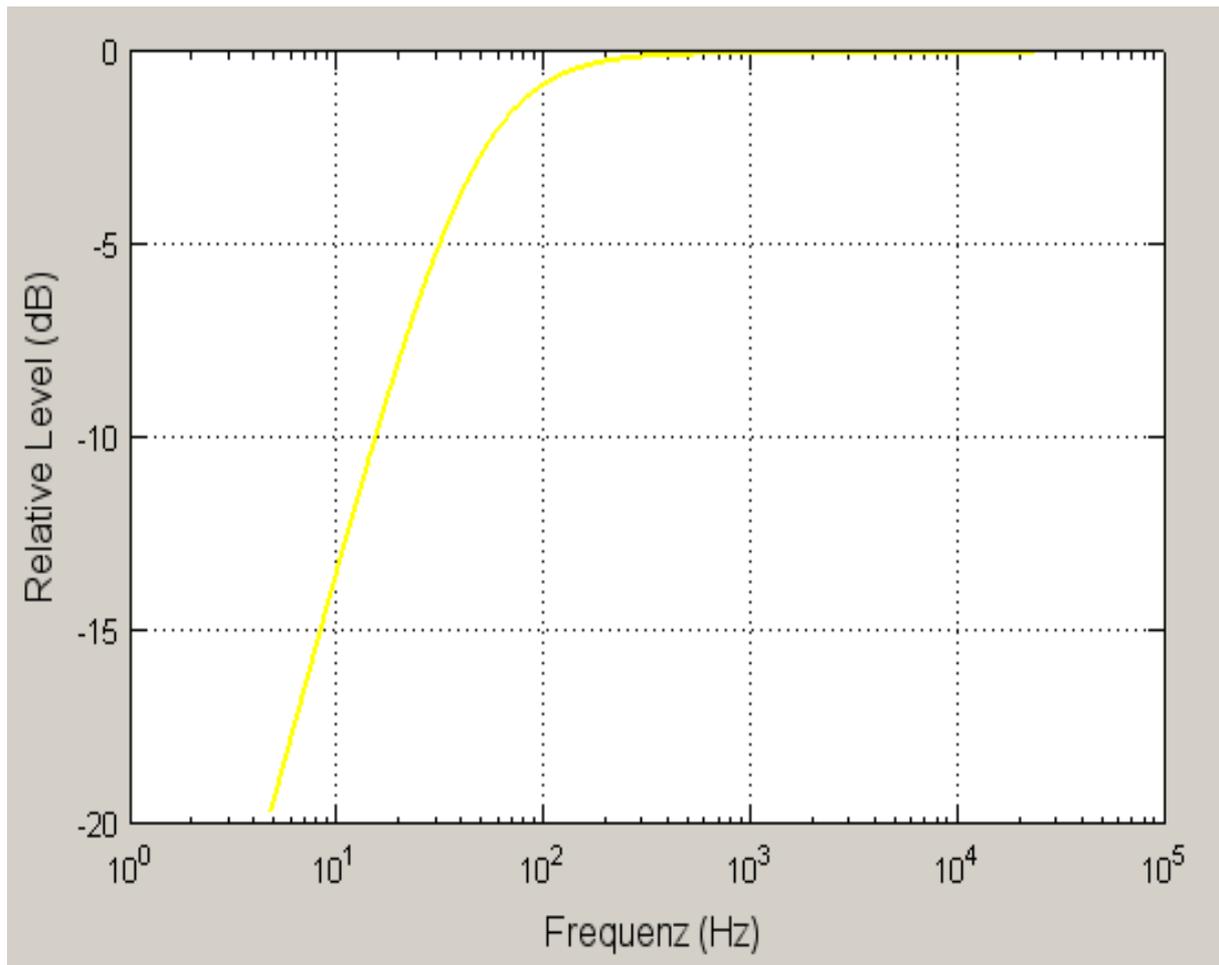
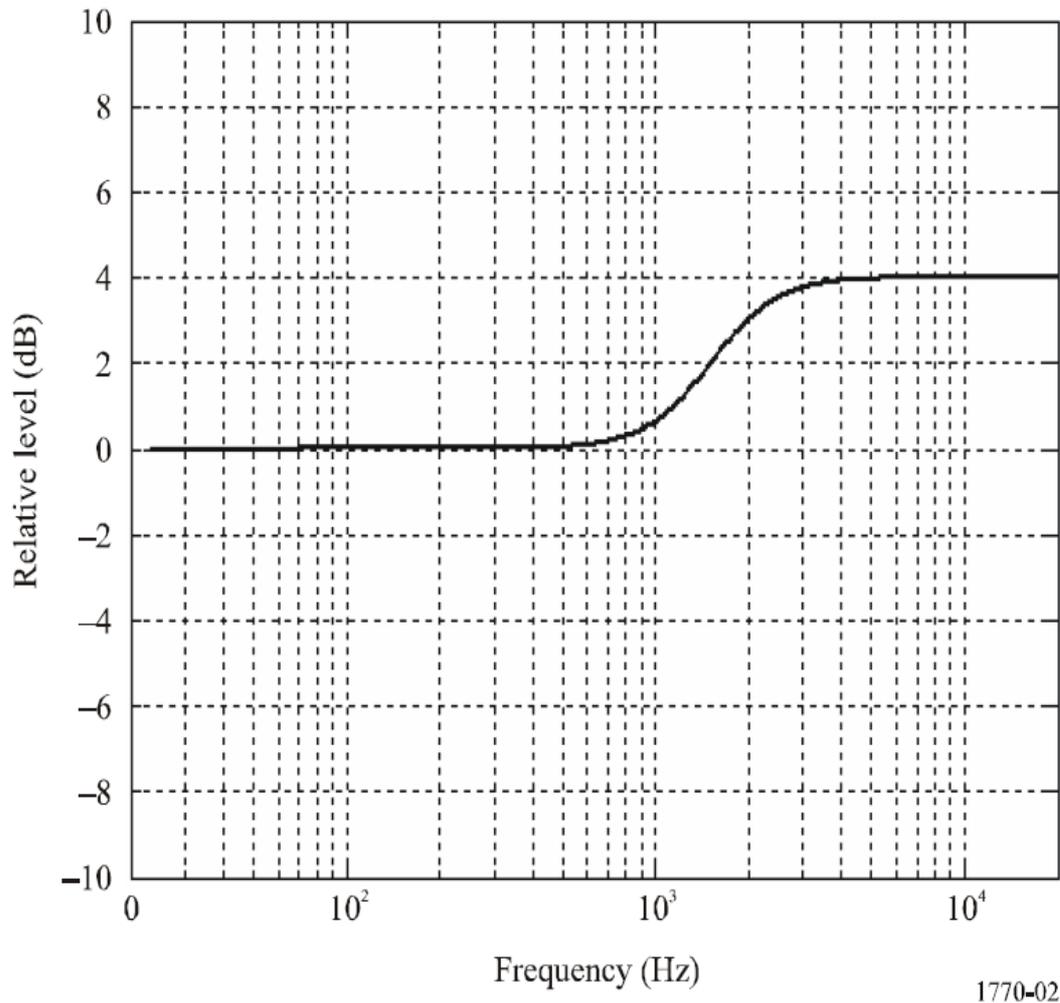


Abbildung 6: Amplitudenantwort des konstruierten Butterworth-Filters bei 48 kHz

Anmerkung: Die vorliegende Amplitudenantwort des Hochpaß-RLB Filters paßt perfekt auf den konstruierten Butterworth-Filter erster Ordnung – so gut, daß die Frage, ob sie nicht sogar mit einem solchen Filter generiert worden ist, durchaus ihre Berechtigung hat. Andererseits werden aber im ITU-Standard für den Hochpaß Koeffizienten für einen Filter zweiter Ordnung angegeben – der konstruierte Butterworth-Hochpaß ist als Filter erster Ordnung und identer Amplitudenantwort besser (weil performanter).

Die zweite Amplitudenantwort beschreibt einen Filter, der hohe Frequenzen verstärkt:

Response of the pre-filter used to account for the acoustic effects of the head



1770-02

Abbildung 7: Verstärkender Filter des RLB Verfahrens (ITU-R BS.1770-1)

Die Charakteristika dieser Amplitudenantwort:

- Die Bereiche von 0 – 500 Hertz und ab 4 kHz verlaufen quasi linear
- Die Verstärkung ab 4 kHz beträgt 4 dB
- Das Zentrum des Anstiegs selbst liegt bei 1550 Hertz

Verstärkende Filter werden als Shelving Filters (oder auch Kuhschwanz-Filter) bezeichnet. Für die Realisierung kommen zwei Verfahren in Frage:

Zum einen ist es möglich, den Filter in der z-Darstellung zu konstruieren, wie in [18] beschrieben.

Zum anderen kann der Filter auch direkt nach einem Verfahren von Zölzer [16] realisiert werden.

Auf Matlab Central [17] findet sich ein entsprechendes Programm, das folgende Parameter erfordert:

- G .. Verstärkung in dB
- FC .. Frequenz des Anstieg-Zentrums
- Fs .. Sampling Frequenz
- Q .. Form Parameter
- type .. Art der Verstärkung – hochfrequent oder niederfrequent

G wurde mit 4 dB abgelesen, FC mit 1550 Hertz, der Typ wurde als hochfrequenter Filter angenommen. Parameter Q andererseits wurde schließlich durch visuelle Analyse bei 0.74 gefunden.

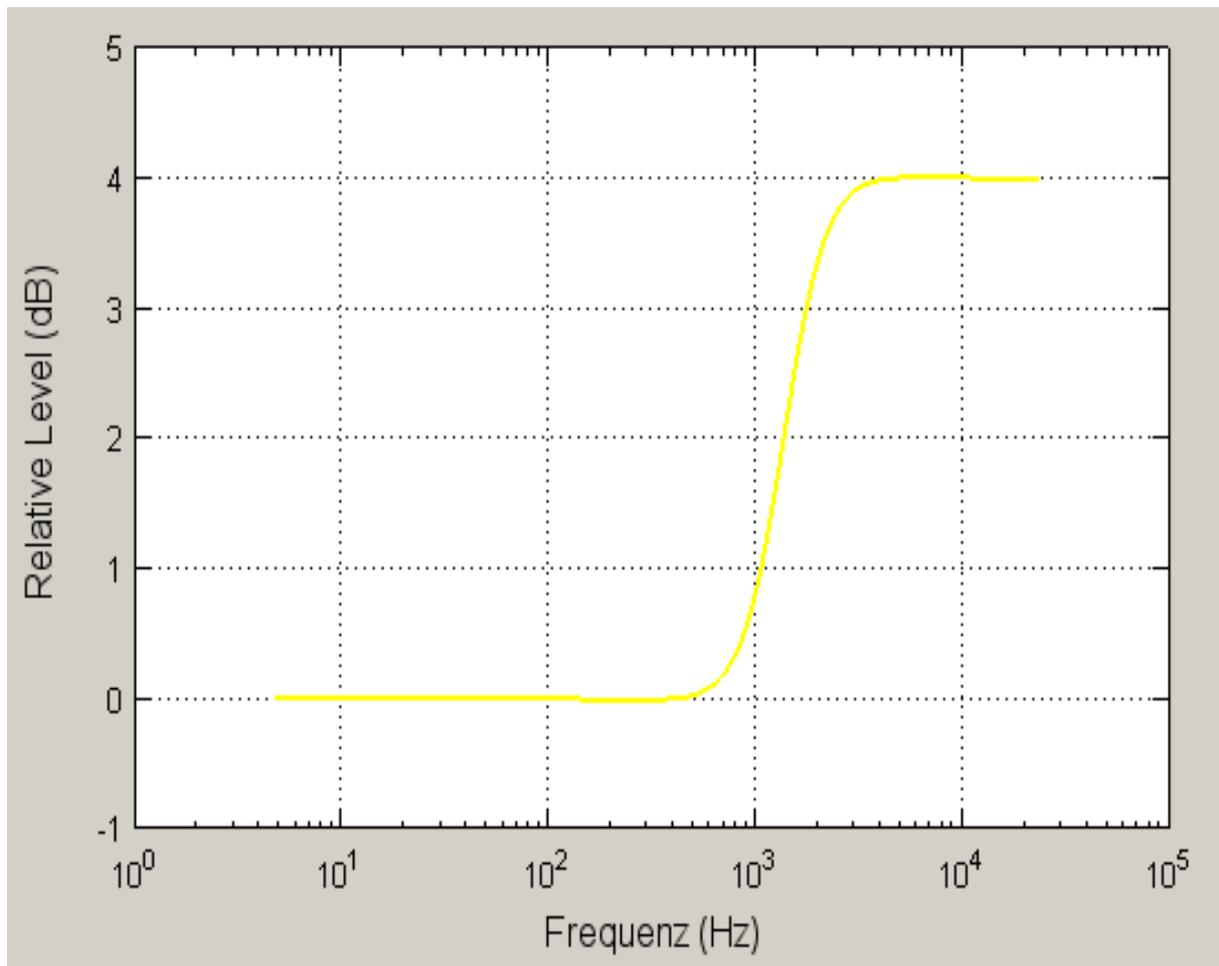


Abbildung 8: Amplitudenantwort des konstruierten Shelving Filter bei 48 kHz

Der ITU Standard für RLB sieht vor, daß beide Filter kaskadiert werden (die kaskadierte Amplitudenantwort ist auf Seite 30 im Anhang abgebildet).

Realisierung

Die im Zuge dieses Praktikums entwickelte Lösung besteht aus zwei Komponenten:

- Algorithmen und Filter als VAMP-Plugin
- Eigener Plugin-Host, der die Lautheitsanpassung im Batch-Modus ermöglicht.

VAMP-Plugin

Die Implementierung der Algorithmen zur dynamischen Lautheitsanpassung via VAMP [1] (Version 1.2) hat mehrere Vorzüge:

Zum einen steht mit SonicVisualizer [3], einem Programm zur Visualisierung von Audio-Dateien, das VAMP-Plugins laden und ausführen kann, ein mächtiges Werkzeug zur Visualisierung zur Verfügung: Sämtliche Wave-Formen können offline betrachtet und miteinander verglichen werden. Zum anderen ist VAMP ein etablierter Standard – nebst weiteren Programmen, welche den VAMP-Standard unterstützen (z.B. Audacity in Beta-Versionen) können Softwareentwickler mit relativ geringem Aufwand das Plugin nutzen. Ein weiterer Vorteil liegt schließlich in der einfachen Erweiterbarkeit: Die Entwicklung weiterer Lautheitsfilter bedeutet kaum zusätzlichen Aufwand (zumal auch eine Lautheit-Basisklasse implementiert wurde, deren Beerbung den Implementierungsaufwand minimiert).

Das Plugin stellt für jedes Gewichtungs-Verfahren folgende Funktionalitäten bereit:

- **Loudness $Leq(x)$** : Liefert Segment-basiert die **gewichtete Lautstärke** in $dB(x)$.

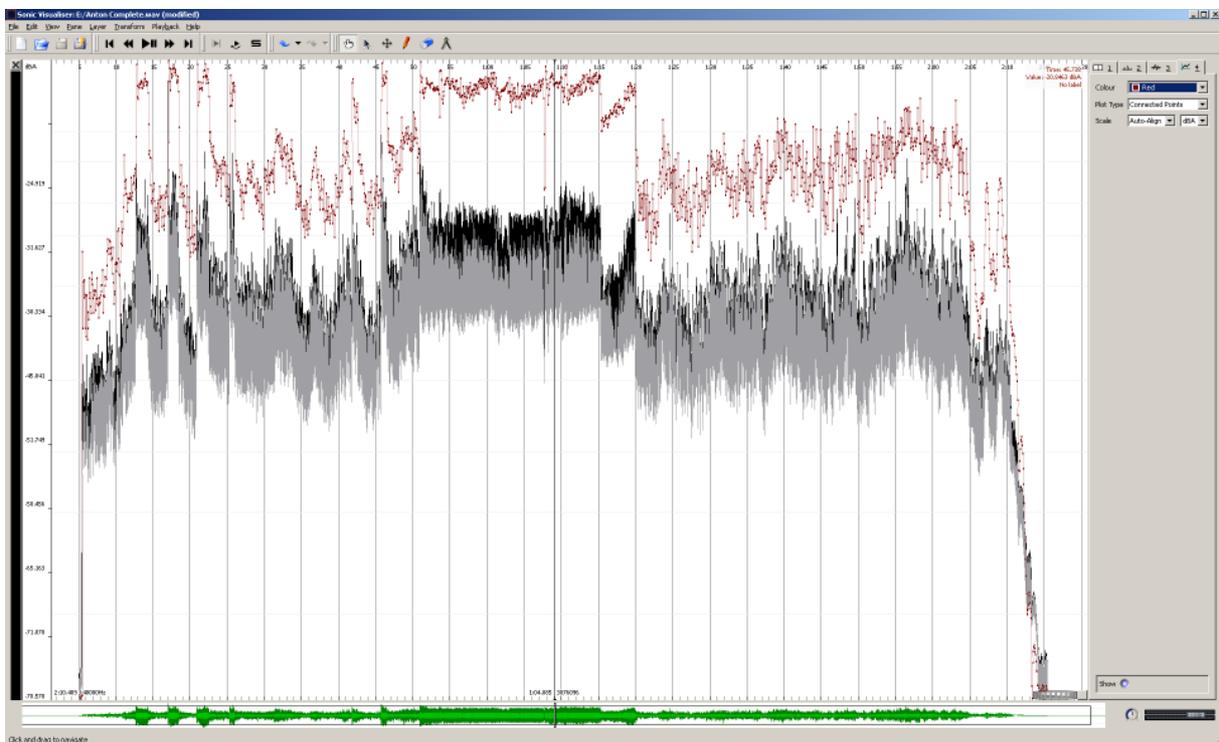


Abbildung 9: Errechnete gewichtete Lautstärke ($dB(A)$) in Rot, SonicVisualizer Ausgabe

- **Loudness equalization** $Leq(x)$: Führt die **Lautstärken-Gewichtung** tatsächlich durch. SonicVisualizer stellt die Hüllkurve nach der Gewichtung dar.

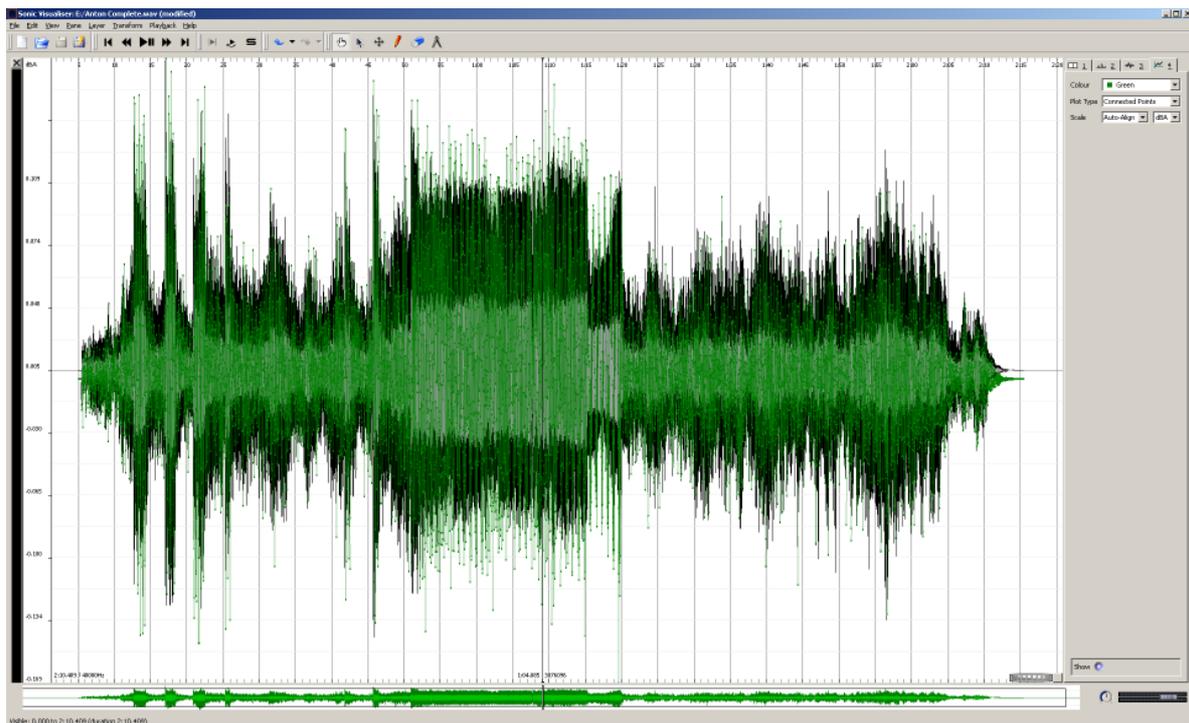


Abbildung 10: Hüllkurve bei angewandter Lautstärken-Gewichtung (dB(A)) in Grün, SonicVisualizer Ausgabe

- **Mean Loudness** $Leq(x)$: Liefert den **gemittelten Lautheitswert** (bis zum aktuellen Sample) über die gesamte Analyse.

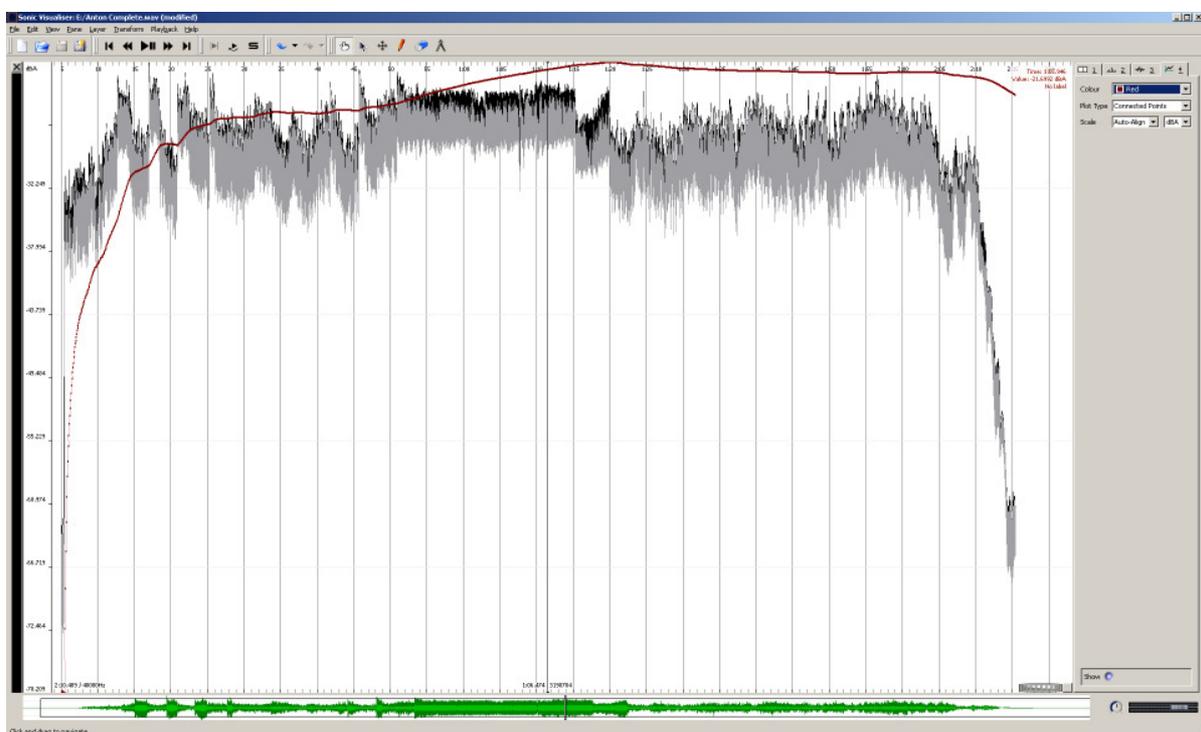


Abbildung 11: Errechnete durchschnittliche Lautstärke (dB(A)) in Rot, SonicVisualizer Ausgabe

- **Peak Loudness $Leq(x)$** : Liefert den (bisher) **maximalen Lautheitswert**.

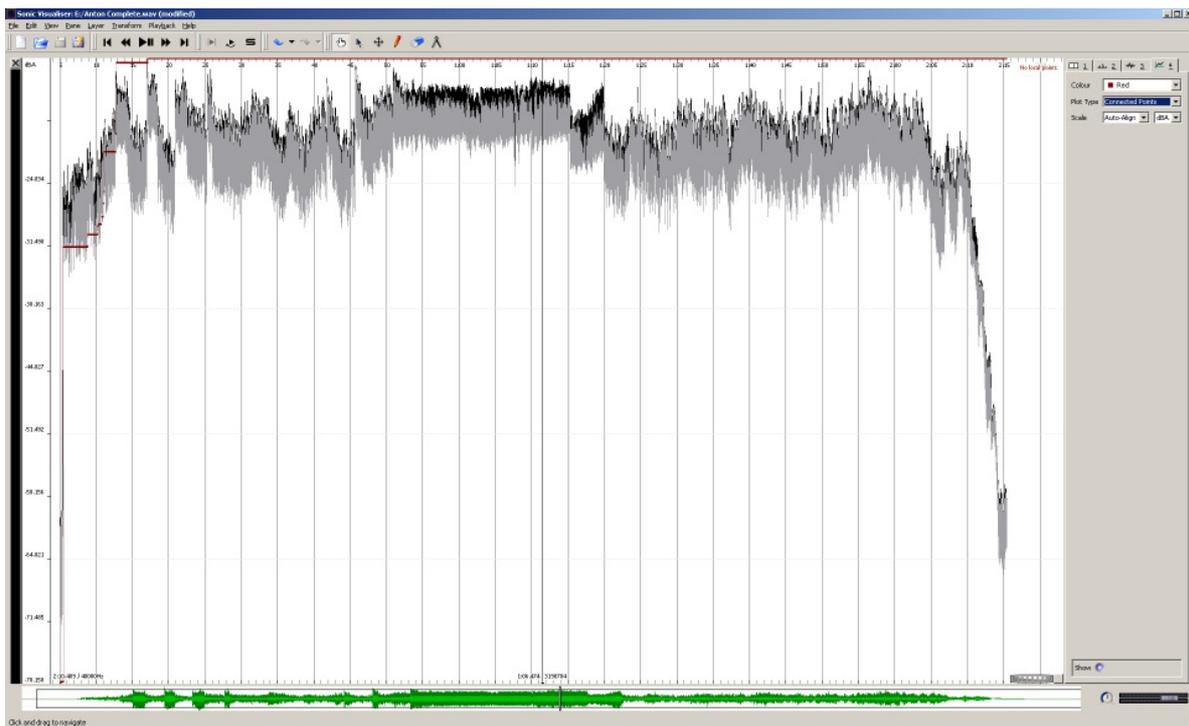


Abbildung 12: Errechnete maximale Lautstärke (dB(A)) in Rot, SonicVisualizer Ausgabe

- **Recommended loudness adaption $Leq(x)$** : Kernfunktionalität des Praktikums – liefert eine **Empfehlung für die vorzunehmende Lautheitsanpassung**. Dieser Wert ändert sich im Zuge der Analyse, weil fortlaufend mehr Samples in Betracht gezogen werden können. Ist ein Einzelwert gesucht, so sollte der letzte verwendet werden.

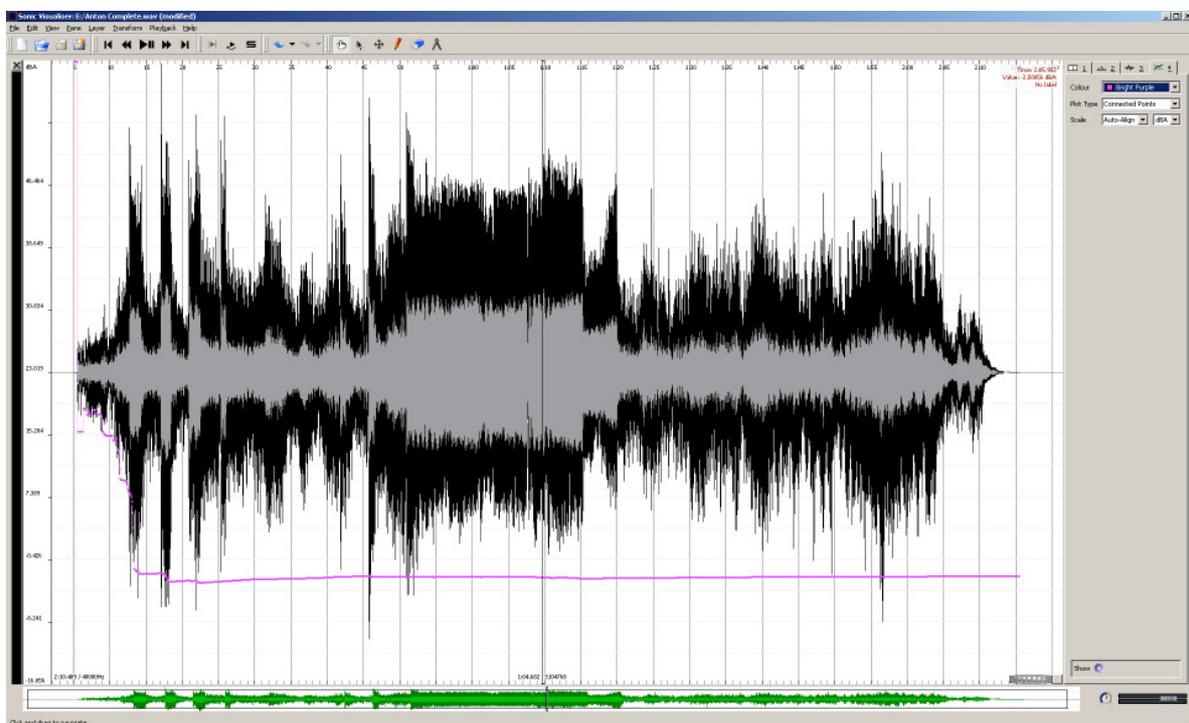


Abbildung 13: Errechnete empfohlene Lautstärken-Anpassung (dB(A)) in Rot, SonicVisualizer Ausgabe

- **Representative loudness** $Leq(x)$: Liefert (bis zum aktuellen Sample) jenen **Lautheitswert**, der zur Berechnung der empfohlenen Lautheitsanpassung herangezogen wird (Perzentil).

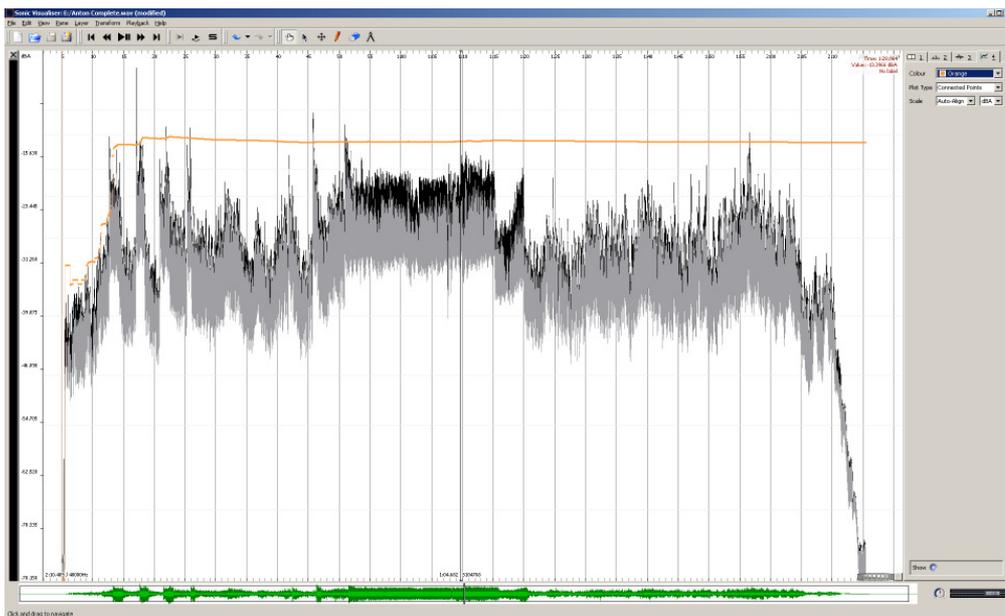


Abbildung 14: Errechneter repräsentativer Lautstärken-Wert (dB(A)) in Orange, SonicVisualizer Ausgabe

Jedes Plugin bietet auch zahlreiche Einstellungsmöglichkeiten:

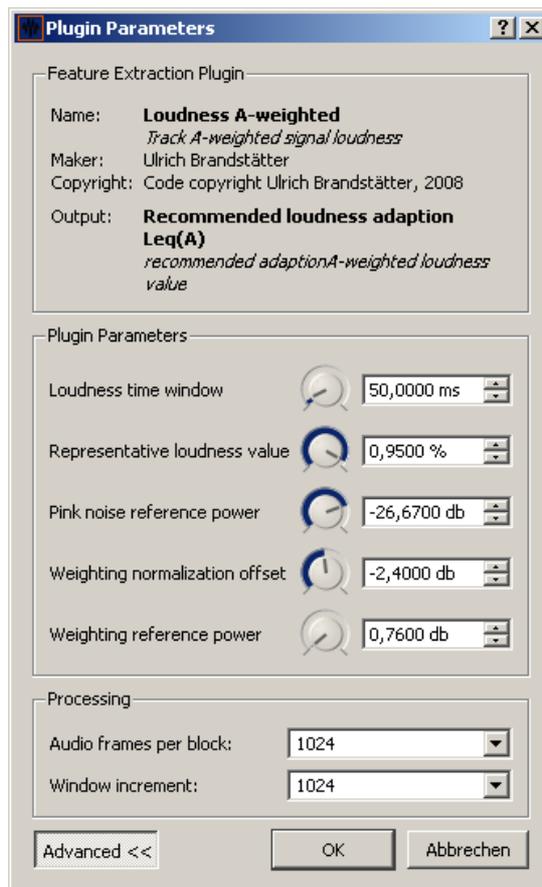


Abbildung 15: Einstellungsmöglichkeiten der Plugins

- Loudness time window

Gibt die Dauer der Zeitfenster vor, über die die Lautstärke berechnet wird.

- Representative loudness value

Bestimmt den Perzentil-Wert, der für die Wahl des tatsächlich verwendeten aus den bereits berechneten Lautheits-Werten verantwortlich ist.

- Pink noise reference power

Referenz-Lautstärke, die zur Bestimmung der empfohlenen Lautheits-Anpassung verwendet wird. Üblicherweise wird der repräsentative Lautheitswert eines generierten Rosa Rauschen Signals mit dem gewünschten Gewichtungsverfahren berechnet und an dieser Stelle eingetragen. Es ist aber auch möglich, anstatt des Rosa Rauschen ein als angenehm empfundenes Stück zu verwenden, oder den Wert direkt nach subjektiven Maßstäben einzutragen.

- Weighting normalization offset

Additive Konstante, um die Lautheit auf einfache Art und Weise auf einen gewünschten Referenz-Pegel zu normieren. Jeder bestehende Filter wurde auf ein 1-KiloHertz Sinus-Signal normalisiert (Standard-Einstellung).

- Weighting reference power

Konstante, die den Schalldruckpegel eines Referenzsignals beschreibt. Die Veränderung dieses Wertes ermöglicht das Normieren der berechneten Lautheit auf eine Referenzlautstärke. Die Standard-Einstellung ist die Amplitudenantwort der jeweiligen Plugins bei 1 kHz.

- Audio frames per block

Internes Sample-Fenster. Veränderungen an diesem Wert können die Performanz der Plugins beeinflussen.

- Window increment

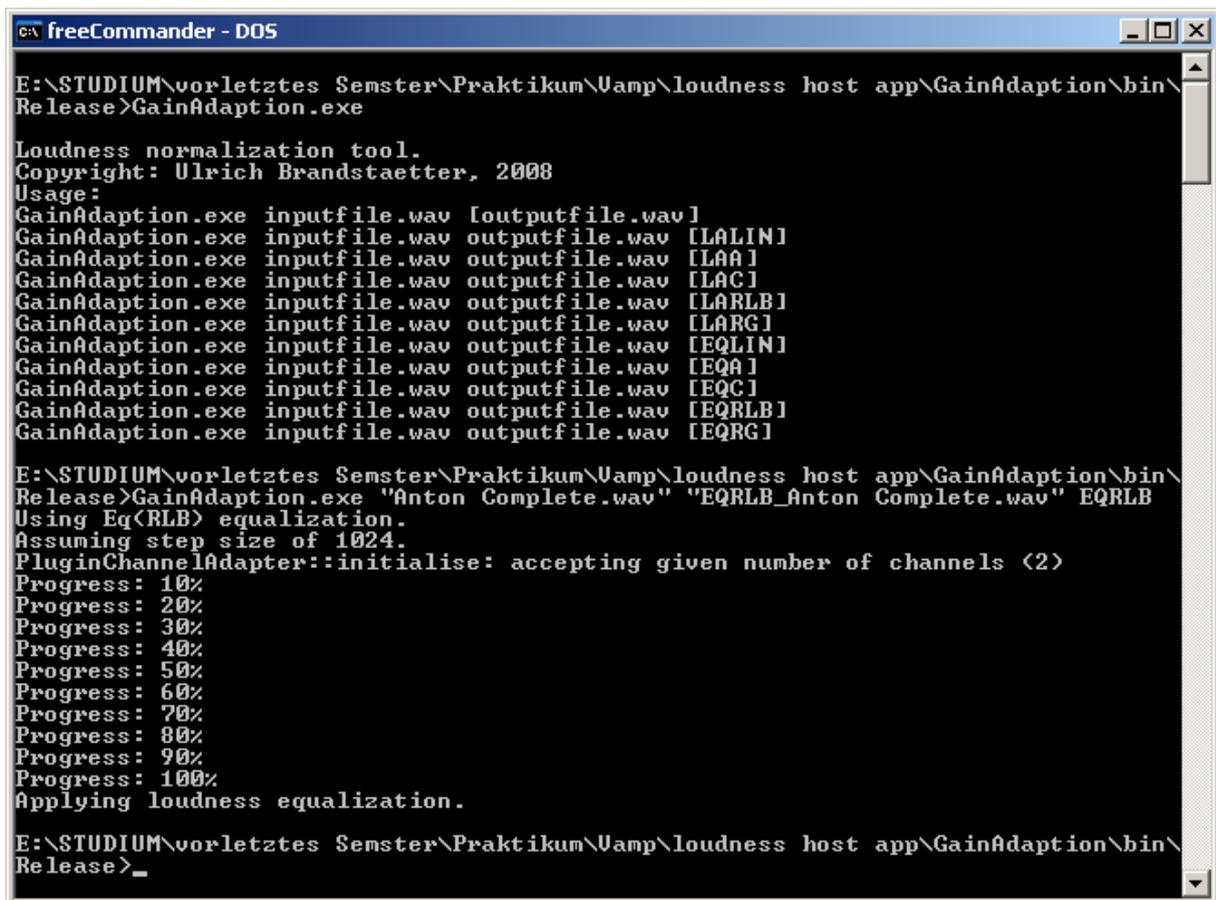
Interner Offset zum nächsten Sample-Fenster. Veränderungen an diesem Parameter können überlappende Analysen erwirken, aber auch ein Überspringen von Samples (kann speziell bei langen Aufnahmen recht sinnvoll sein).

Zwischen den Parametern „Loudness time window“, „Audio frames per block“ und „Window increment“ besteht folgender Zusammenhang: „Audio frames per block“ und „Window increment“, angegeben in Samples, werden intern auf ihre entsprechende Dauer in Millisekunden umgerechnet und mitgespeichert. Übersteigt dieser mitgespeicherte Wert im Zuge der Ausführung das „Loudness time window“, so wird die Lautheit für die seither vergangene Zeit berechnet und der mitgespeicherte Wert in Folge auf Null gesetzt. Insofern ist der Parameter „Loudness time window“ nicht völlig akkurat, der dabei begangene Fehler ist allerdings vernachlässigbar klein.

Plugin-Host

Im wesentlichen stellt der Plugin-Host, ein Command-Line-Tool zur Ausführung der Lautheits-Plugins, zweierlei Funktionalitäten bereit:

- Zum einen kann er die Lautstärke einer gegebenen Audiodatei (im WAVE-Format) gemäß dem verwendeten Filter (kann über Parameter gewählt werden) anpassen, sprich einen Clip etwa A-gewichtet abspeichern.
- Zum anderen kann er die empfohlene Lautheitsanpassung automatisch durchführen.



```
c:\ freeCommander - DOS
E:\STUDIUM\vorletztes Semester\Praktikum\Vamp\loudness host app\GainAdaption\bin\
Release>GainAdaption.exe

Loudness normalization tool.
Copyright: Ulrich Brandstaetter, 2008
Usage:
GainAdaption.exe inputfile.wav [outputfile.wav ]
GainAdaption.exe inputfile.wav outputfile.wav [LALIN]
GainAdaption.exe inputfile.wav outputfile.wav [LAA]
GainAdaption.exe inputfile.wav outputfile.wav [LAC]
GainAdaption.exe inputfile.wav outputfile.wav [LARLB]
GainAdaption.exe inputfile.wav outputfile.wav [LARG]
GainAdaption.exe inputfile.wav outputfile.wav [EQLIN]
GainAdaption.exe inputfile.wav outputfile.wav [EQA]
GainAdaption.exe inputfile.wav outputfile.wav [EQC]
GainAdaption.exe inputfile.wav outputfile.wav [EQRLB]
GainAdaption.exe inputfile.wav outputfile.wav [EQRG]

E:\STUDIUM\vorletztes Semester\Praktikum\Vamp\loudness host app\GainAdaption\bin\
Release>GainAdaption.exe "Anton Complete.wav" "EQRLB_Anton Complete.wav" EQRLB
Using Eq(RLB) equalization.
Assuming step size of 1024.
PluginChannelAdapter::initialise: accepting given number of channels (2)
Progress: 10%
Progress: 20%
Progress: 30%
Progress: 40%
Progress: 50%
Progress: 60%
Progress: 70%
Progress: 80%
Progress: 90%
Progress: 100%
Applying loudness equalization.

E:\STUDIUM\vorletztes Semester\Praktikum\Vamp\loudness host app\GainAdaption\bin\
Release>_
```

Abbildung 16: Ausgabe und Benutzungs-Beispiel des Plugin-Host

Der Plugin-Host verarbeitet bis zu drei Eingabe-Parameter:

- Wird kein Parameter angegeben gibt das Tool eine knappe Bedienungsanleitung aus (Abbildung 15).
- Wird nur ein Parameter angegeben, wird dieser als Pfad zur Eingabedatei interpretiert. Diese wird in dann ausgelesen und entsprechend der A-Bewertung analysiert. Schließlich wird eine Ausgabedatei mit der angewandten empfohlenen Lautstärkeanpassung (wieder A-bewertet) generiert (Benennungsschema: „EQA“ + Eingabedatei).
- Bei zwei gegebenen Parametern wird der erste als Pfad zur Eingabedatei interpretiert, der zweite als Pfad zur Ausgabedatei. Die Ausgabedatei wird entsprechend der empfohlenen Lautstärkeanpassung nach der A-Bewertung generiert.
- Wird darüberhinaus ein dritter Parameter angegeben (Abbildung 15), so kann dieser den

Modus bestimmen:

- Dieser Modus-Parameter setzt sich aus einem Befehlswort (Prefix) sowie einem Gewichtungsverfahren (Postfix) zusammen:
 - Prefix LA: Weist den Plugin-Host an, die empfohlene Lautstärkenanpassung auf die Ausgabedatei anzuwenden
 - Prefix EQ: Der Plugin-Host führt lediglich eine Frequenzgewichtung gemäß dem gewählten Verfahren (Postfix) durch.
 - Postfix LIN: Keine Gewichtung.
 - Postfix A: A-Gewichtung
 - Postfix C: C-Gewichtung
 - Postfix RLB: RLB-Gewichtung
 - Postfix RG: Gewichtung nach dem Replay-Gain Verfahren
- Beispiele wären etwa EQRLB (Abbildung 15), LAC oder EQA. Ebenfalls möglich ist EQLIN, dabei wird schlussendlich aber nur eine Kopie angefertigt.

Die Implementierung als Command-Line-Tool ermöglicht die Anwendung der Plugins im Stapelverarbeitungs-Modus, was es zu einem geeigneten Werkzeug für die Audio-Vorverarbeitung macht.

Der Plugin-Host unterstützt sämtliche Dateiformate, die auch von der Software-Bibliothek libsndfile [2] unterstützt werden.

Vergleich der Methoden

Für einen Vergleich der hier implementierter Verfahren zur Lautheits-Bestimmung wurden diese auf verschiedene Audio-Dateien angewandt. Um hierfür halbwegs repräsentative Daten zu erhalten wurden mehrere werbe- und unterbrechungsfreie Streaming-Radio Sender [19] unterschiedlicher Themen mitgeschnitten, jeweils mindestens eine Stunde lang:

„**Classic Guitar.cc**“ [20]: Sender, der ausschließlich Stücke mit klassischer Gitarre spielt

| Gewichtung | mean | peak | recommendation | 95% value |
|-----------------|--------|---------|----------------|-----------|
| A | -37.44 | -14.72 | -0.28 | -26.34 |
| C | -32.28 | -9.91 | -1.94 | -22.1 |
| RLB | -33.31 | -11.08 | + 0.1 | -23.08 |
| RG | -32.61 | -10.11 | -1.29 | -22.06 |
| - | -31.96 | -9.89 | + 0.56 | -22.02 |
| | | | | |
| diff (MIN, MAX) | 5.48 | 4.83 | 2.5 | 4.32 |
| mean | -33.52 | -11.142 | -1.17 | -23.12 |

„Classic Guitar.cc“ ist der leiseste, den Ergebnissen zufolge aber auch angenehmste (im Sinne der Lautstärke) hier vertretene Sender. Die Empfehlungen für die Lautstärke-Anpassung weichen geringfügig voneinander ab: Das ungewichtete Verfahren würde am meisten verstärken, RLB nur geringfügig, alle anderen Gewichtungen würden eine Reduktion der Lautstärke empfehlen.

„**WAMU 88,5 – American University Radio**“ [21]: Nachrichten Sender

| Gewichtung | mean | peak | recommendation | 95% value |
|-----------------|--------|--------|----------------|-----------|
| A | -28.75 | -10.1 | -9.78 | -16.89 |
| C | -25.15 | -8.26 | -9.85 | -14.18 |
| RLB | -25.15 | -8.74 | -8.32 | -14.66 |
| RG | -24.19 | -7.69 | -9.94 | -13.41 |
| - | -24.86 | -7.69 | -7.34 | -14.12 |
| | | | | |
| diff (MIN, MAX) | 3.89 | 2.41 | 2.6 | 3.48 |
| mean | -25.62 | -8.476 | -9.046 | -14.652 |

Dieser Nachrichtensender wurde ausgewählt, um auch Sprach-Samples zu reflektieren. Die Empfehlungen der A-, C- und ReplayGain-Gewichtung sind sich äußerst ähnlich, ungewichtet würde die Lautstärke am wenigsten verändert werden.

„Jazz At Hot Sauce Radio“ [22]: Contemporary Jazz Sender

| Gewichtung | mean | peak | recommendation | 95% value |
|-----------------|---------|--------|----------------|-----------|
| A | -25.67 | -8.66 | -10.07 | -16.6 |
| C | -19.38 | -5.5 | -12.06 | -11.98 |
| RLB | -20.32 | -5.16 | -9.88 | -13.1 |
| RG | -20.8 | -6.63 | -10.28 | -13.07 |
| - | -18.86 | -4.58 | -10.16 | -11.3 |
| | | | | |
| diff (MIN, MAX) | 6.81 | 4.08 | 2.18 | 5.3 |
| mean | -21.006 | -6.106 | -10.49 | -13.21 |

Interessanterweise würde bei diesem Jazz-Sender das RLB-Verfahren die Lautstärke weniger beeinflussen als das ungewichtete. Die C-Gewichtung empfiehlt hier eine deutlich höhere Reduktion der Lautstärke als alle anderen. Die A-, die ReplayGain-Gewichtung und die ungewichtete Empfehlung ähneln sich hier sehr stark.

„Hotspot Radio“ [23]: Drum n' Bass Sender

| Gewichtung | mean | peak | recommendation | 95% value |
|-----------------|---------|-------|----------------|-----------|
| A | -20.15 | -5.4 | -12.36 | -14.31 |
| C | -15.41 | -4.06 | -13.98 | -10.06 |
| RLB | -16.05 | -5.13 | -11.94 | -11.04 |
| RG | -16.64 | -4.14 | -12.26 | -11.09 |
| - | -14.58 | -3.97 | -12.13 | -9.33 |
| | | | | |
| diff (MIN, MAX) | 5.57 | 1.43 | 2.04 | 4.98 |
| mean | -16.566 | -4.54 | -12.534 | -11.166 |

Bei der Bewertung des HotSpot-Mitschnitts speziell auffällig ist die geringe Abweichung der Peaks. Relativ eng beieinander liegen auch die Empfehlungen der Gewichtungs-Verfahren. RLB empfiehlt die geringste Veränderung, C die höchste.

„Sonic Cathedral Sirens Radio“ [24]: Heavy Metal Sender

| Gewichtung | mean | peak | recommendation | 95% value |
|-----------------|---------|-------|----------------|-----------|
| A | -15.71 | -7.06 | -14.19 | -12.48 |
| C | -11.93 | -4.22 | -14.28 | -9.75 |
| RLB | -11.86 | -5.3 | -13.15 | -9.83 |
| RG | -11.5 | -3.82 | -14.65 | -8.7 |
| - | -11.39 | -3.95 | -12.41 | -9.05 |
| | | | | |
| diff (MIN, MAX) | 4.32 | 3.24 | 2.21 | 3.78 |
| mean | -12.478 | -4.87 | -13.736 | -9.962 |

Dieser Sender wird wie auch die meisten anderen als zu laut beurteilt. Ungewichtet beurteilt würde die Lautstärke am wenigsten reduziert werden, mit dem ReplayGain-Verfahren nach dem ungewichteten am meisten, wobei sich die A-, C- und ReplayGain-Empfehlungen stark ähneln.

Vergleich der Gewichtungsmethoden

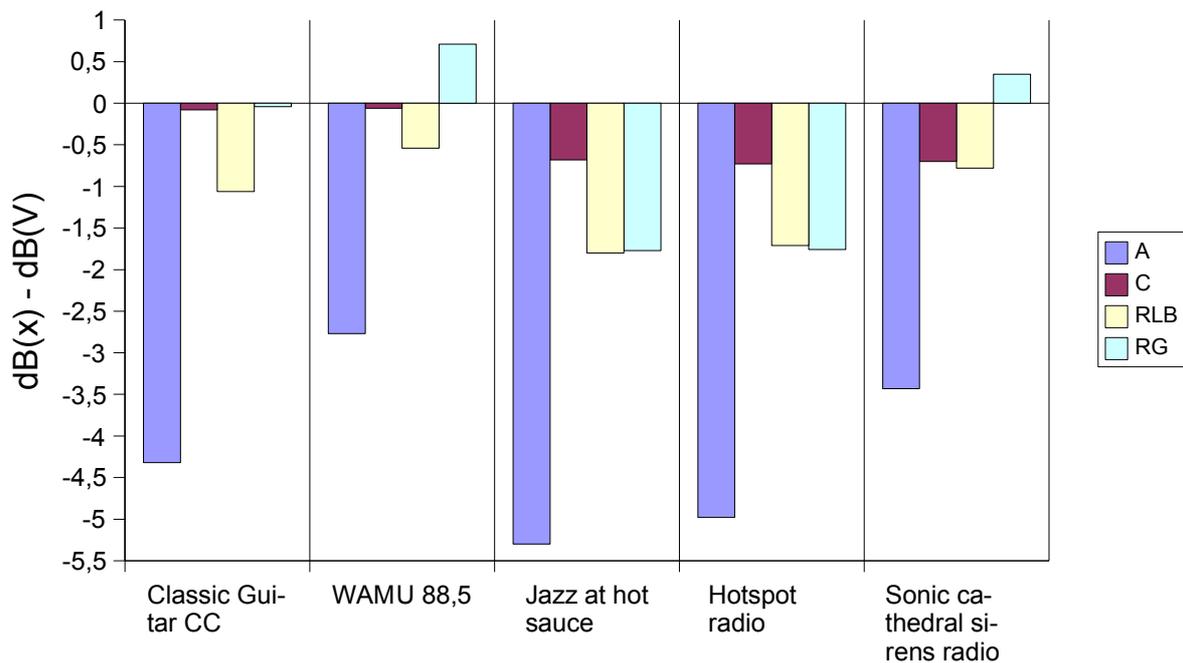


Abbildung 17: Anwendung der unterschiedlichen Gewichtungsmethoden auf die Mitschnitte (die ungewichtete Lautheit ist in dieser Darstellung von den gewichteten Lautheiten subtrahiert)

Allgemeine Schlussfolgerungen:

- Die 95%-Perzentil-Werte weichen in einem wesentlich geringerem Maß voneinander ab als die über die gesamte Analyse berechneten Mittelwerte.
- Unterschiedliche Musikrichtungen werden von den verschiedenen Verfahren auch unterschiedlich bewertet – die geringsten Lautstärke-Anpassung etwa werden bei diesen Mitschnitten von RLB und ungewichtet empfohlen, die höchsten von C und ReplayGain.
- Das RLB-Verfahren kann das Eingangssignal potentiell lauter bewerten als die ungewichtete Bewertung – dies liegt am darin enthaltenen Shelving-Filter, der die hohen Frequenzen verstärkt. Alle anderen Verfahren bewerten niemals lauter als das ungewichtete.
- Die A-Gewichtung verändert das Eingabesignal am stärksten, die C-Gewichtung am wenigsten.
- Eine Aussage, welches Verfahren den anderen überlegen ist, läßt sich nicht treffen.

Validierung

Der erste Teil der Validierung bestand in der Ausführung der Filter auf jene Datei, die zum Bestimmen der Referenzlautstärke für die unterschiedlichen Verfahren verwendet wurde. Das ideale (und auch erreichte) Ergebnis ist, wenn sämtliche Gewichtungen eine Empfehlung von Null Dezibel abgeben – in der nachfolgenden Grafik wird dieser Pegel von jedem Filter erreicht:

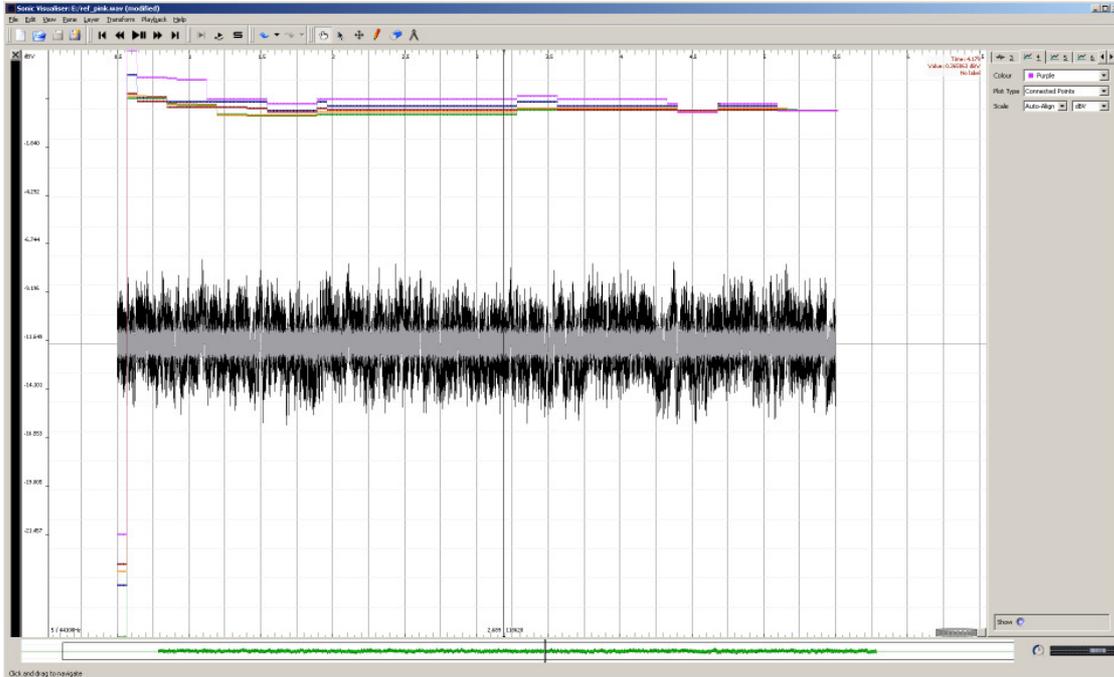


Abbildung 18: Sämtliche Verfahren liefern die empfohlene Lautheits-Anpassung von 0 dB.

Der zweite Teil sollte die Ergebnisse eines einzelnen Filters (A-Gewichtung) mit einer entsprechenden Octave-Referenz-Implementierung vergleichen:

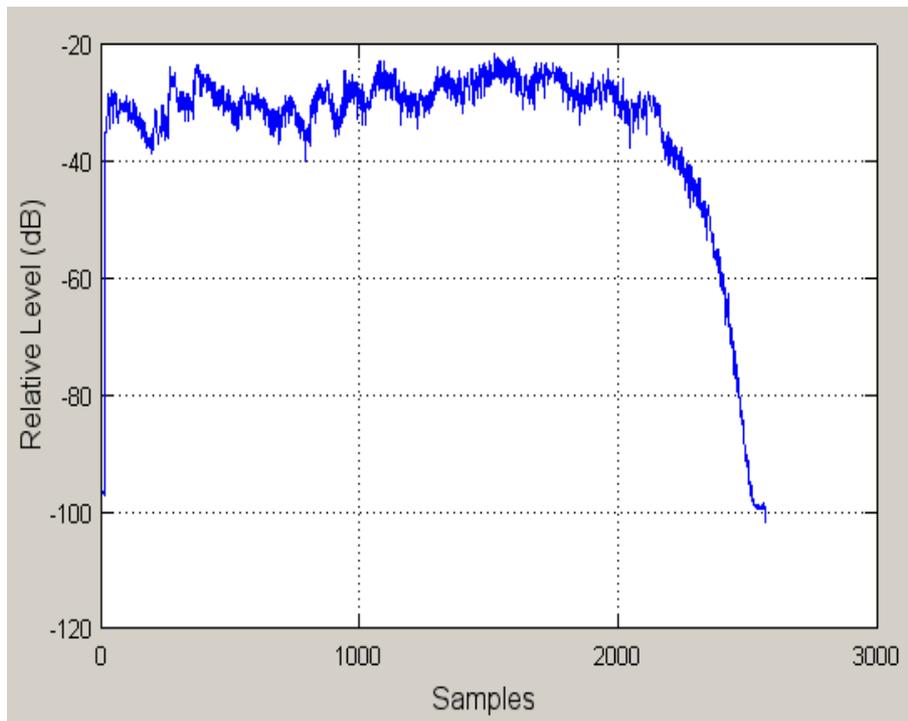


Abbildung 19: A-Lautheit eines Referenzclips, gerechnet mit Octave

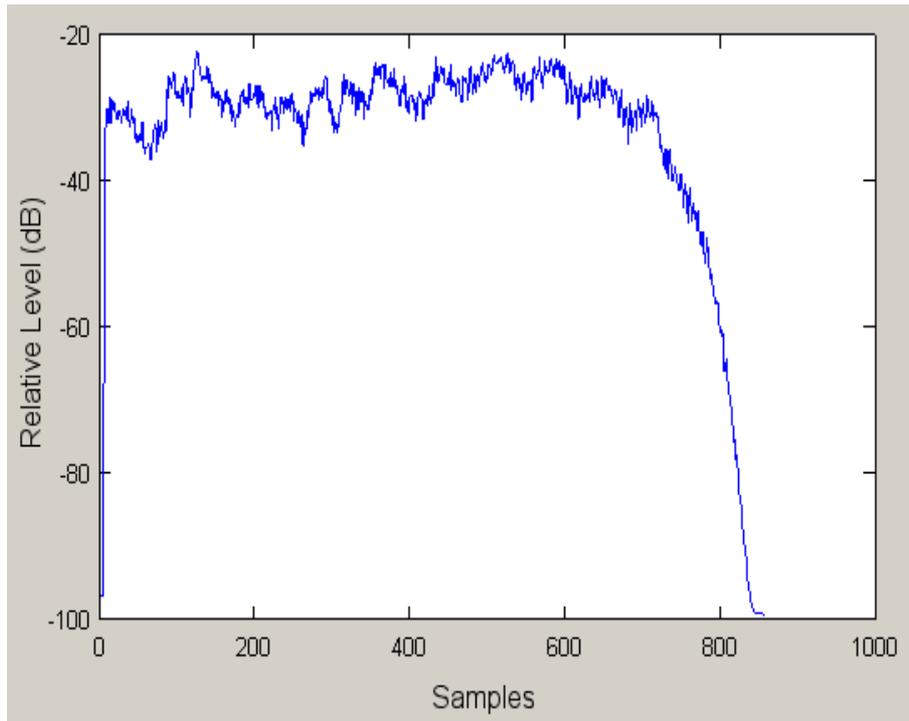


Abbildung 20: A-Lautheit eines Referenzclips, gerechnet mit VAMP-Plugin

Die Ergebnisse stimmen nahezu überein, der bemerkbare Unterschied der unterschiedlichen Anzahl an Datenpunkten stammt von der 50 Millisekunden-Fensterung (über die gemittelt wird), die im VAMP-Plugin eingestellt ist, während die Octave-Implementierung über keine zusätzliche Fensterung verfügt. Deshalb wirkt deren Ergebnis auch etwas kantiger.

Unglücklicherweise hat sich im Zuge dieses zweiten Tests herausgestellt, daß Single-Floating-Point-Precision für die Filterkoeffizienten eines IIR-Filters nicht ausreicht. Neben diesen mußte auch die IIR-Filterung selbst auf Double-Precision umgestellt werden, was sich natürlich negativ auf Speicherbedarf und Berechnungsgeschwindigkeit auswirkt. Double-Precision ist für diese Filter ausreichend, dennoch wurde mit der Einbindung der GMP-Bibliothek [4] (Gleitkommazahlen und -berechnungen einstellbarer Auflösung) in die Filter-Methode ausreichend Rücklage für IIR-Filter höherer Ordnung geschaffen (starke Erhöhung der Filter-Zeiten, daher deaktiviert).

Anhang

Filter Koeffizienten und Amplitudenantworten

A Gewichtung

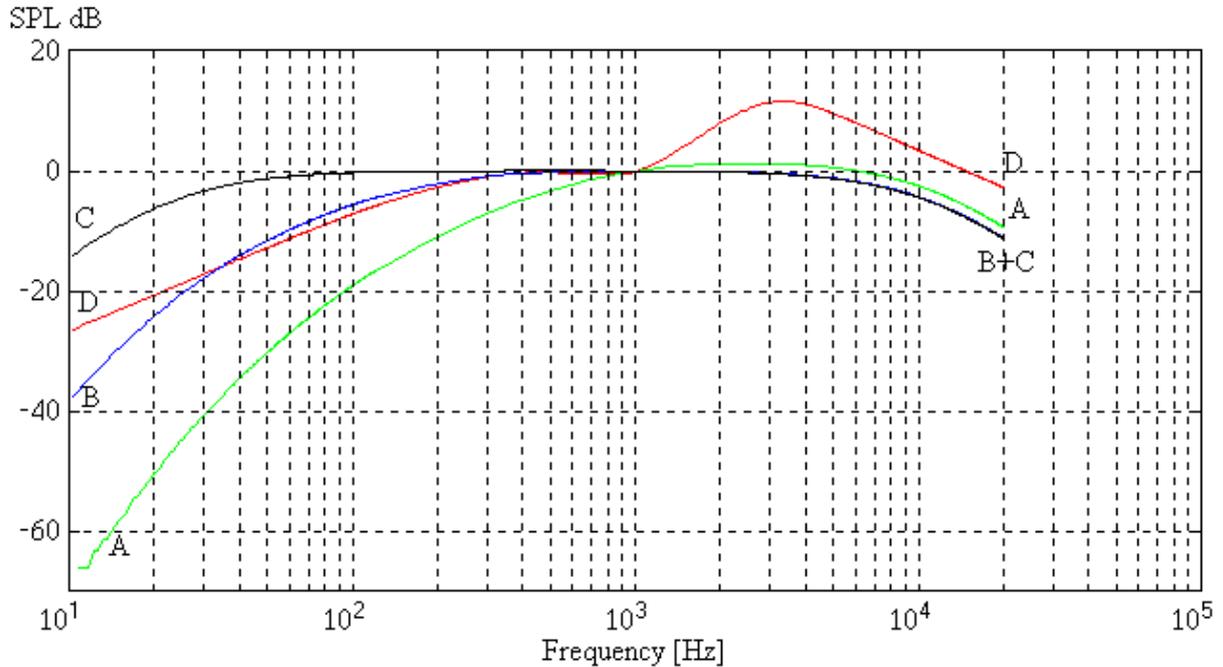


Abbildung 21: Gesuchte A und C Gewichtungs-Kurven

B-Koeffizienten der A-Gewichtung

| Frequenz | B(0) | B(1) | B(2) | B(3) | B(4) | B(5) | B(6) |
|-----------|--------------------------|---------------------------|---------------------------|--------------------------|---------------------------|---------------------------|--------------------------|
| 24000 Hz | 0.4256263892 89105570 | -0.8512527785 78211020 | -0.4256263892 89105680 | 1.7025055571 56421800 | -0.4256263892 89105680 | -0.8512527785 78211020 | 0.4256263892 89105570 |
| 32000 Hz | 0.3434583386 82430640 | -0.6869166773 64861390 | -0.3434583386 82430590 | 1.3738333547 29722800 | -0.3434583386 82430590 | -0.6869166773 64861390 | 0.3434583386 82430640 |
| 44100 Hz | 0.2557411252 04257570 | -0.5114822504 08515030 | -0.2557411252 04257620 | 1.0229645008 17030100 | -0.2557411252 04257620 | -0.5114822504 08515030 | 0.2557411252 04257570 |
| 48000 Hz | 0.2343017922 99513230 | -0.4686035845 99026510 | -0.2343017922 99513200 | 0.9372071691 98053030 | -0.2343017922 99513200 | -0.4686035845 99026510 | 0.2343017922 99513230 |
| 96000 Hz | 0.0995189897 59727478 | -0.1990379795 19454930 | -0.0995189897 59727505 | 0.3980759590 38909800 | -0.0995189897 59727505 | -0.1990379795 19454930 | 0.0995189897 59727478 |
| 192000 Hz | 0.0343321342 45487124 | -0.0686642684 90974262 | -0.0343321342 45487117 | 0.1373285369 81948520 | -0.0343321342 45487117 | -0.0686642684 90974262 | 0.0343321342 45487124 |

A-Koeffizienten der A-Gewichtung

| Frequenz | A(0) | A(1) | A(2) | A(3) | A(4) | A(5) | A(6) |
|-----------|------|---------------------------|---------------------------|----------------------------|---------------------------|---------------------------|--------------------------|
| 24000 Hz | 1.0 | -3.3259960042 41962300 | 3.6771610792 86317000 | -1.1064760768 28482500 | -0.4726706734 90872000 | 0.1861941760 23095900 | 0.0417877133 78295274 |
| 32000 Hz | 1.0 | -3.6564460432 33668500 | 4.8314684506 52581100 | -2.5575974965 81569500 | 0.2533680394 20530660 | 0.1224430322 45257100 | 0.0067640721 68341966 |
| 44100 Hz | 1.0 | -4.0195761811 15832400 | 6.1894064429 20693900 | -4.4531989035 44116400 | 1.4208429496 21876600 | -0.1418254738 30304060 | 0.0043511772 33494981 |
| 48000 Hz | 1.0 | -4.1130434087 75872000 | 6.5531217526 55050300 | -4.9908492941 63386000 | 1.7857373029 37575800 | -0.2461905953 19487390 | 0.0112242500 33231114 |
| 96000 Hz | 1.0 | -4.8022030442 25377600 | 9.4018072186 27228100 | -9.5661439435 69168000 | 5.3097759303 92620900 | -1.5173333604 52623100 | 0.1740971994 22891200 |
| 192000 Hz | 1.0 | -5.3059236896 74639100 | 11.659524374 661757000 | -13.575600927 005915000 | 8.8289069328 24192900 | -3.0394901209 88216600 | 0.4325834301 87038100 |

C Gewichtung

B-Koeffizienten der C-Gewichtung

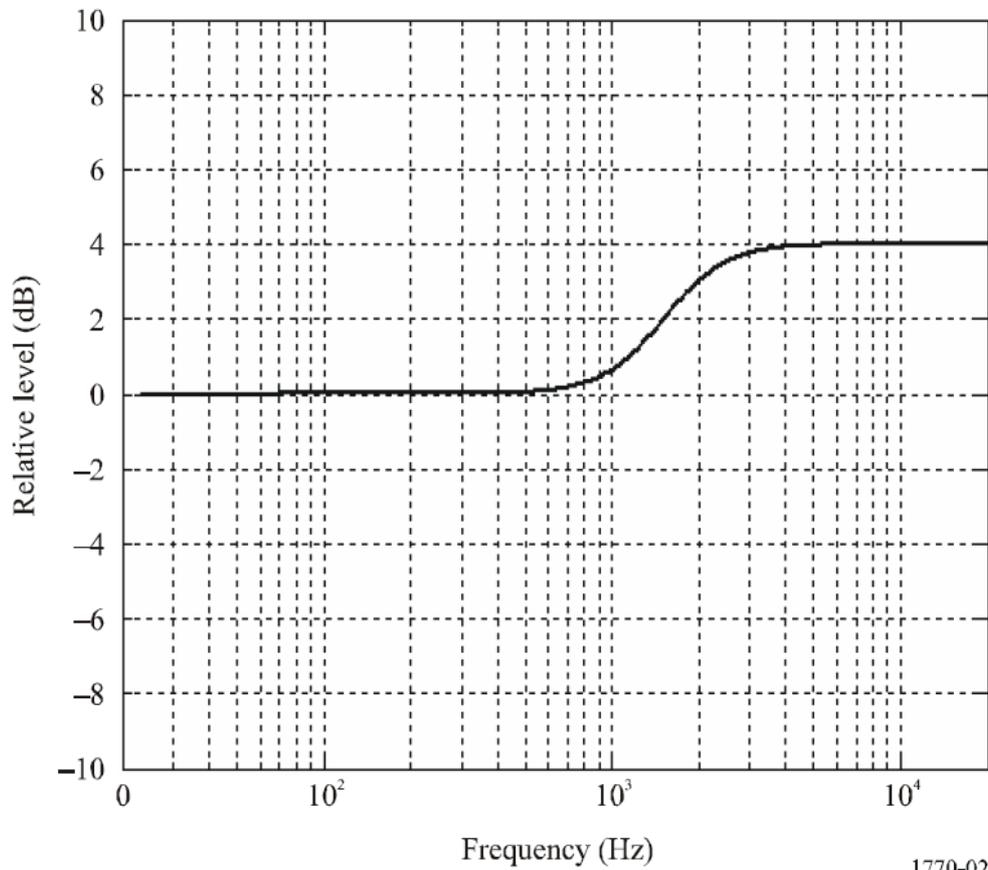
| Frequenz | B(0) | B(1) | B(2) | B(3) | B(4) |
|-----------|--------------------------|------|---------------------------|------|--------------------------|
| 24000 Hz | 0.3786678621 92496540 | 0.0 | -0.7573357243 84993080 | 0.0 | 0.3786678621 92496540 |
| 32000 Hz | 0.2977986488 23069220 | 0.0 | -0.5955972976 46138450 | 0.0 | 0.2977986488 23069220 |
| 44100 Hz | 0.2170085619 49218910 | 0.0 | -0.4340171238 98437830 | 0.0 | 0.2170085619 49218910 |
| 48000 Hz | 0.1978871200 26393250 | 0.0 | -0.3957742400 52786500 | 0.0 | 0.1978871200 26393250 |
| 96000 Hz | 0.0818286404 49797568 | 0.0 | -0.1636572808 99595140 | 0.0 | 0.0818286404 49797568 |
| 192000 Hz | 0.0278475546 85322792 | 0.0 | -0.0556951093 70645583 | 0.0 | 0.0278475546 85322792 |

A-Koeffizienten der C-Gewichtung

| Frequenz | A(0) | A(1) | A(2) | A(3) | A(4) |
|-----------|------|---------------------------|--------------------------|---------------------------|--------------------------|
| 24000 Hz | 1.0 | -1.5299453075 55420800 | 0.1283553182 11621050 | 0.3494608072 38572420 | 0.0521729194 93000944 |
| 32000 Hz | 1.0 | -1.8124553871 28178800 | 0.6425013281 15565820 | 0.1619857574 57858040 | 0.0079876497 13547733 |
| 44100 Hz | 1.0 | -2.1346749636 87040300 | 1.2793335332 36062200 | -0.1495598460 89395570 | 0.0049087001 74624705 |
| 48000 Hz | 1.0 | -2.2191729140 52801800 | 1.4551358789 47171600 | -0.2484960738 87783050 | 0.0125388231 47272494 |
| 96000 Hz | 1.0 | -2.8563785168 57566800 | 2.8976402375 59523600 | -1.2252659583 39703000 | 0.1840048283 55122580 |
| 192000 Hz | 1.0 | -3.3332988561 44166300 | 4.11146753624 0339400 | -2.2228890416 51291200 | 0.4447204118 12687840 |

RLB Gewichtung

Response of the pre-filter used to account for the acoustic effects of the head



1770-02

Abbildung 22: Gesuchte RLB-Vorfilter Gewichtungskurve

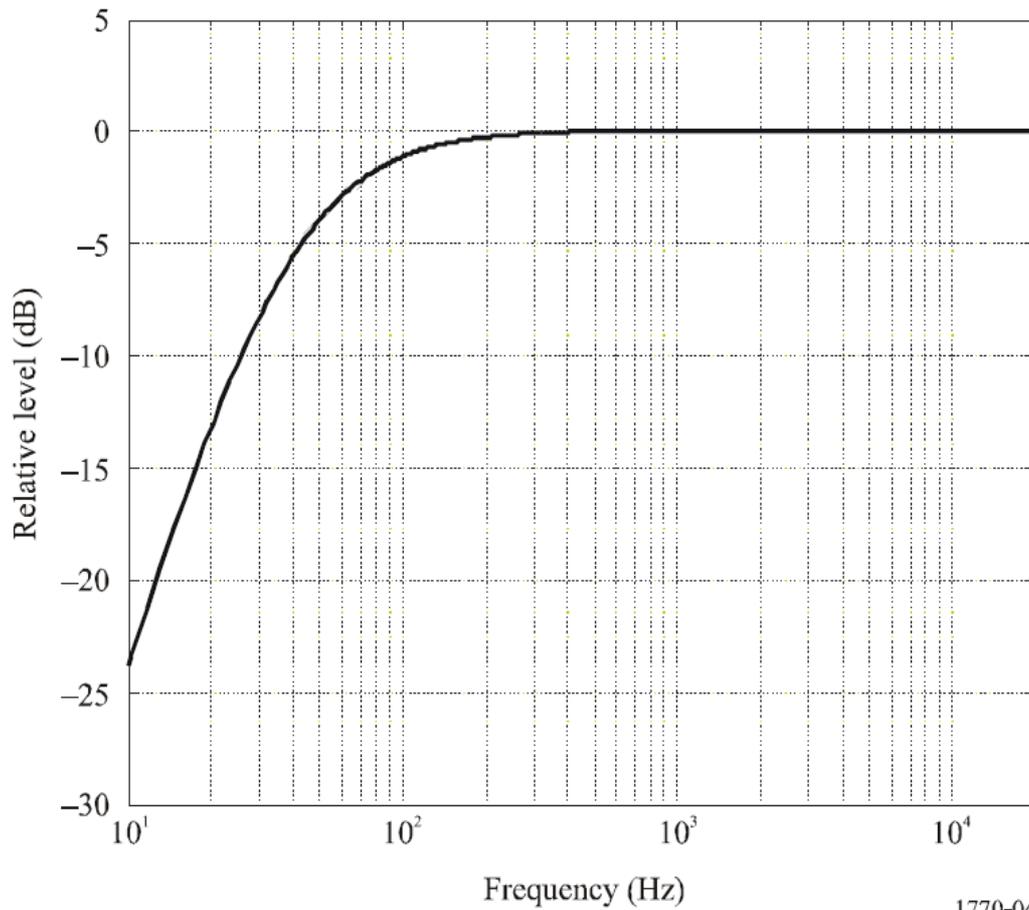
B-Koeffizienten des RLB-Vorfilter

| Frequenz | B(0) | B(1) | B(2) |
|----------|--------------------------|---------------------------|--------------------------|
| 32000 Hz | 1.5405504737 49923800 | -2.7432622428 95266200 | 1.2388206278 56079400 |
| 44100 Hz | 1.5405504737 49923800 | -2.7432622428 95266200 | 1.2388206278 56079400 |
| 48000 Hz | 1.5351248595 8697 | -2.6916961894 0638 | 1.1983928108 5285 |

A-Koeffizienten des RLB-Vorfilter

| Frequenz | A(0) | A(1) | A(2) |
|----------|------|---------------------------|--------------------------|
| 32000 Hz | 1.0 | -1.7242186054 36776100 | 0.7603274641 47512950 |
| 44100 Hz | 1.0 | -1.7242186054 36776100 | 0.7603274641 47512950 |
| 48000 Hz | 1.0 | -1.6906592931 8241 | 0.7324807742 1585 |

RLB weighting curve



1770-04

Abbildung 23: Gesuchte RLB-Hochpass Gewichtungskurve

B-Koeffizienten des RLB-Hochpaß

| Frequenz | B(0) | B(1) | B(2) |
|----------|--------------------------|---------------------------|------|
| 32000 Hz | 0.9955042330 58102650 | -0.9955042330 58102650 | 0.0 |
| 44100 Hz | 0.9967337469 21327550 | -0.9967337469 21327550 | 0.0 |
| 48000 Hz | 1.0 | -2.0 | 1.0 |

A-Koeffizienten des RLB-Hochpaß

| Frequenz | A(0) | A(1) | A(2) |
|----------|------|---------------------------|----------------------|
| 32000 Hz | 1.0 | -0.9910084661 16205190 | 0.0 |
| 44100 Hz | 1.0 | -0.9934674938 42655200 | 0.0 |
| 48000 Hz | 1.0 | -1.9900474548 3398 | 0.9900722503 6621 |

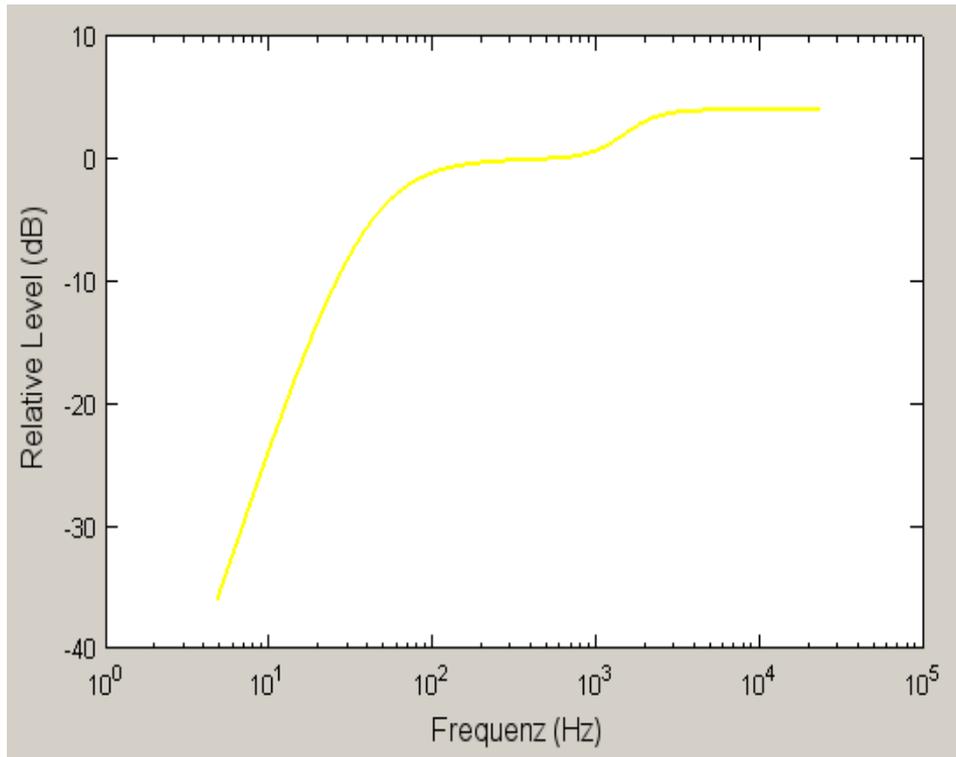


Abbildung 24: Amplitudenantwort der kaskadierten Vor- und Hochpaß-Filter für RLB

Replay Gain Gewichtung

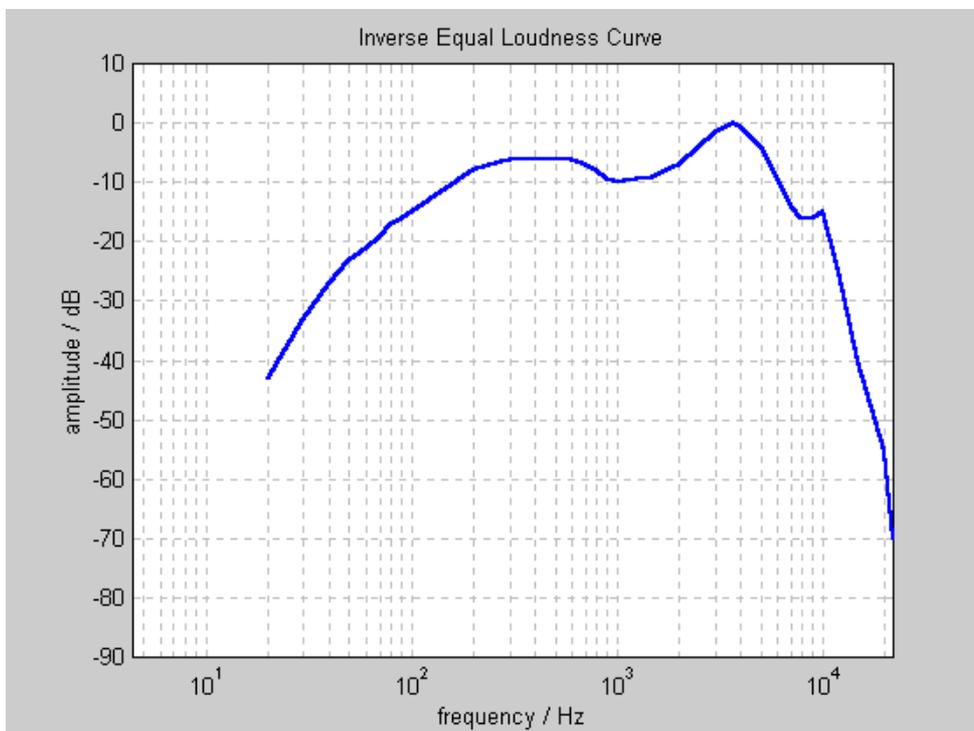


Abbildung 25: Gesuchte Replay-Gain Gewichtungskurve

Sämtliche Filterkoeffizienten der Replay Gain Gewichtung [8] wurden der Projekt-Homepage entnommen (http://replaygain.hydrogenaudio.org/equal_loud_coef.txt)

Code Auszug

Zur Umsetzung des Plugins wurde eine Lautheits-Basis-Klasse implementiert, welche die Erweiterung um zusätzliche Gewichtungen stark vereinfacht. Dies sei anhand der A-Gewichtung verdeutlicht:

Loudness_A.h

```
#ifndef LOUDNESS_A_H_INCLUDED
#define LOUDNESS_A_H_INCLUDED

#include "LoudnessBase.h"

class LoudnessAWeighted: public LoudnessBase {

    typedef LoudnessBase Base;

public:

    /** Constructor of a-weighting-loudness plugin.
     * @param inputSampleRate sample rate to use
     */
    LoudnessAWeighted(float inputSampleRate):
        Base(inputSampleRate,
            "a",
            "Leq(A) ",
            "A-weighted loudness value",
            "dBA",
            -2.4,
            0.75698,
            -26.6711) {}

    /// @see Vamp::Plugin::getIdentifier
    virtual std::string getIdentifier() const {

        return "a";
    }

    /// @see Vamp::Plugin::getName
    virtual std::string getName() const {

        return "Loudness A-weighted";
    }

    /// @see Vamp::Plugin::getDescription
    virtual std::string getDescription() const {

        return "Track A-weighted signal loudness";
    }

protected:

    /** Method to setup the frequency weighting filter.
     * This method is called once during initialization. */
    virtual void SetupWeightingFilter();

}; // class LoudnessAWeighted

#endif // LOUDNESS_A_H_INCLUDED
```

Loudness_A.cpp

```

#include <cmath>
#include "Loudness_A.h"

static const t_loudness_float c_b_coeffs[][7] = {
{0.425626389289105570, -0.851252778578211020, -0.425626389289105680, // 24000
 1.702505557156421800, -0.425626389289105680, -0.851252778578211020, // 24000
 0.425626389289105570}, // 24000
{0.343458338682430640, -0.686916677364861390, -0.343458338682430590, // 32000
 1.373833354729722800, -0.343458338682430590, -0.686916677364861390, // 32000
 0.343458338682430640}, // 32000
{0.255741125204257570, -0.511482250408515030, -0.255741125204257620, // 44100
 1.022964500817030100, -0.255741125204257620, -0.511482250408515030, // 44100
 0.255741125204257570}, // 44100
{0.234301792299513230, -0.468603584599026510, -0.234301792299513200, // 48000
 0.937207169198053030, -0.234301792299513200, -0.468603584599026510, // 48000
 0.234301792299513230}, // 48000
{0.099518989759727478, -0.199037979519454930, -0.099518989759727505, // 96000
 0.398075959038909800, -0.099518989759727505, -0.199037979519454930, // 96000
 0.099518989759727478}, // 96000
{0.034332134245487124, -0.068664268490974262, -0.034332134245487117, // 192000
 0.137328536981948520, -0.034332134245487117, -0.068664268490974262, // 192000
 0.034332134245487124} // 192000
};

static const t_loudness_float c_a_coeffs[][7] = {
{ 1.000000000000000000, -3.325996004241962300, 3.677161079286317000, // 24000
 -1.106476076828482500, -0.472670673490872000, 0.186194176023095900, // 24000
 0.041787713378295274}, // 24000
{ 1.000000000000000000, -3.656446043233668500, 4.831468450652581100, // 32000
 -2.557597496581569500, 0.253368039420530660, 0.122443032245257100, // 32000
 0.006764072168341966}, // 32000
{ 1.000000000000000000, -4.019576181115832400, 6.189406442920693900, // 44100
 -4.453198903544116400, 1.420842949621876600, -0.141825473830304060, // 44100
 0.004351177233494981}, // 44100
{ 1.0, -4.113043408775872000, 6.553121752655050300, // 48000
 -4.990849294163386000, 1.785737302937575800, -0.246190595319487390, // 48000
 0.011224250033231114}, // 48000
{ 1.000000000000000000, -4.802203044225377600, 9.401807218627228100, // 96000
 -9.566143943569168000, 5.309775930392620900, -1.517333360452623100, // 96000
 0.174097199422891200}, // 96000
{ 1.000000000000000000, -5.305923689674639100, 11.659524374661757000, // 192000
 -13.575600927005915000, 8.828906932824192900, -3.039490120988216600, // 192000
 0.432583430187038100} // 192000
};

/*virtual*/ void LoudnessAWeighted::SetupWeightingFilter() {

    // base is supposed to clear the filter if any
    Base::SetupWeightingFilter();
    m_filter_order = 6;
    size_t coeff_index;
    switch (static_cast<unsigned> (m_sample_rate)) {
        case 24000: coeff_index = 0; break;
        case 32000: coeff_index = 1; break;
        case 44100: coeff_index = 2; break;
        case 48000: coeff_index = 3; break;
        case 96000: coeff_index = 4; break;
        case 192000: coeff_index = 5; break;
        default:
            m_filter_order = 0; return;
    }
    for (size_t lv = 0; lv < m_filter_order + 1; lv++) {
        m_a_coeffs.push_back (c_a_coeffs[coeff_index][lv]);
        m_b_coeffs.push_back (c_b_coeffs[coeff_index][lv]);
    }
}

```

Referenzen

Verwendete Softwarebibliotheken

- [1] VAMP Plugins
<http://www.vamp-plugins.org/>
- [2] LibSndFile
<http://www.mega-nerd.com/libsndfile/>
- [3] Sonic Visualizer
<http://www.sonicvisualiser.org/>
- [4] GMP
<http://gmplib.org>

Lautheit

- [5] Skovenborg, E. and Nielsen, S. "Evaluation of Different Loudness Models with Music and Speech Material", *Proceedings of the AES 117th Convention*, San Francisco, CA, USA, 2004. http://www.tcelectronic.com/media/skovenborg_2004_loudness_m.pdf
- [6] Zwicker, E. (1960) "Ein Verfahren zur Berechnung der Lautstärke (A procedure for calculating loudness)", *Acustica*, vol.10, pp.304-308.

Filterentwicklung

- [7] Curtis Roads: the computer music tutorial, ISBN: 978-0-262-18158-7, ch. 10, p. 387ff
- [8] Replay Gain
<http://replaygain.hydrogenaudio.org/contents.html>
- [9] ITU RECOMMENDATION ITU-R BS.1770-1
- [10] Frequency Weighting Equations
<http://www.cross-spectrum.com/audio/weighting.html>
- [11] Introduction to Digital Filters
<http://www.dspguide.com/ch14.htm>
- [12] Online Tool for IIR and FIR Filter Design
<http://cnx.org/content/m13115/latest/>
- [13] IEC/CD 1672: Electroacoustics-Sound Level Meters, IEC, Geneva, Nov. 1996, for A- and C-weighting filters
- [14] Edward Zechmann: Continuous Sound and Vibration Analysis Matlab addon
<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=21384&objectType=file>

- [15] Octave
<http://www.gnu.org/software/octave/>
- [16] DAFX: Digital Audio Effects, Udo Zölzer e.a., ISBN 0471490784, 9780471490784 , p. 50f
- [17] IIR-Shelving-Filter Skript von Jeff Tackett nach Udo Zölzer
<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=16568&objectType=FILE>
- [18] Shelving-Filter Konstruktion
<http://circa.ucsd.edu/~msp/techniques/v0.08/book-html/node136.html>

Verwendete Audio-Streaming Sender

- [19] Live 365
<http://www.live365.com/index.live>
- [20] Classical Guitar cc
<http://www.live365.com/stations/radioman111>
- [21] American university radio
<http://wamu.org/>
- [22] Jazz@HotSauce
<http://www.live365.com/stations/fourgem>
- [23] Hotspot Radio
<http://www.hotspot.2ya.com/>
- [24] Sonic Cathedral Sirens Radio
<http://www.soniccathedral.com/>