

# Qualitätsanalyse von Audiosignalen Auswirkung der lautheits- und True Peak-orientierten Aussteuerung nach EBU R 128 auf Datenreduktion

Bachelorarbeit aus Aufnahmetechnik 1, SE

Christopher Alexander Kopel – Matr.-Nr.: 0730613

in Zusammenarbeit mit dem Westdeutschen Rundfunk, Köln (WDR)

Betreuung seitens des IEM: Dr. Alois Sontacchi

Betreuung seitens des WDR: Carsten Berndt

Graz, Februar 2014



**UNIVERSITÄT**  
FÜR MUSIK UND  
DARSTELLENDEN KUNST  
GRAZ - AUSTRIA

institut für elektronische musik und akustik



## **Zusammenfassung**

In dieser Bachelorarbeit wird die Auswirkung einer lautheits- und „True Peak“-orientierten Aussteuerung gemäß ITU-R BS.1770 und EBU R 128 auf die Tonqualität bei der im Rundfunk eingesetzten Datenreduktion mit MPEG-1/2 Layer II (MP2) und Layer III (MP3) untersucht. Dazu werden zunächst theoretische Grundlagen der Themenbereiche „Lautheit“, „Aussteuerung im Rundfunk“ und „Perzeptive Audiodatenreduktion“ erläutert. Anschließend werden Anordnung, Durchführung und Auswertung der praktischen Untersuchung beschrieben. Bei der Untersuchung wurden Referenz-Testbeispiele der EBU sowie exemplarische Beiträge aus dem Hörfunk des WDR verwendet. Alle Beispiele wurden einerseits in Originalaussteuerung und andererseits ausgesteuert nach EBU R 128 encodiert. Mittels der Software PQevalAudio (einer freien Implementierung von PEAQ gemäß ITU-R BS.1387) wurde die durch die Encodierung entstandene Qualitätsverschlechterung bestimmt. Ein Vergleich der Messwerte ergab, dass die Aussteuerung gemäß EBU R 128 keinen negativen Einfluss auf die Tonqualität der Datenreduktion hat und überdies mit hoher Wahrscheinlichkeit auch nicht von einem positiven Einfluss ausgegangen werden kann.

## **Abstract**

This Bachelor thesis examines the impact of levelling with regard to loudness and “True Peak Levels” in accordance with ITU-R BS.1770 and EBU R 128 on the audio data compression by MPEG-1/2 Layer II (MP2) and Layer III (MP3), applied in broadcasting. The introductory theoretical part therefore contains a critical summary of the topics “loudness”, “levelling in broadcasting”, and “perceptual audio compression”. In the succeeding parts of this paper, the design, procedure, and analysis of the practical experiment are described. For the experiment, reference test samples provided by the EBU as well as exemplary contributions to the radio programme of the WDR (public west German broadcasting corporation) were used. All samples were encoded on the one hand in original levelling, on the other hand levelled in accordance with EBU R 128. The sound quality impairment caused by the data compression was determined by means of PQevalAudio, a free implementation of PEAQ according to ITU-R BS.1387. A comparison of the values measured showed that levelling in accordance with EBU R 128 has no negative impact on the sound quality of the audio compression and that, furthermore, with high probability, any positive impact cannot be assumed either.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretischer Hintergrund</b>	<b>3</b>
2.1	Lautheit allgemein . . . . .	3
2.1.1	Definition . . . . .	3
2.1.2	Ermittlung der Lautheit . . . . .	4
2.2	Lautheit im Rundfunk . . . . .	7
2.2.1	Bisherige Aussteuerungspraxis . . . . .	8
2.2.2	Problematik . . . . .	9
2.2.3	Ansätze zur Lautheitsmessung im Rundfunk . . . . .	12
2.2.3.1	Psychoakustische Verfahren . . . . .	12
2.2.3.2	Verfahren basierend auf zeitlicher Mittelung . . . . .	13
2.2.4	Die Standardisierung eines Lautheitsmessverfahrens . . . . .	14
2.2.5	Lautheitsmessung nach ITU-R BS.1770-2 . . . . .	15
2.2.6	Die Empfehlung EBU R 128 . . . . .	18
2.2.6.1	Normalisierung nach integriertem Lautheitspegel (Programmlautheit) . . . . .	19
2.2.6.2	Momentan- und Kurzzeitlautheitspegel . . . . .	19
2.2.6.3	Lautheitsbereich („Loudness Range“) . . . . .	19
2.2.6.4	Höchster echter Spitzenpegel („Maximum True Peak Level“) . . . . .	20

<i>Ch. Kopel: Auswirkung der Aussteuerung nach EBU R 128 auf Datenreduktion</i>	IV
2.3 Perzeptive Audiodatenreduktion . . . . .	21
2.3.1 Definition . . . . .	21
2.3.2 Prinzipielle Funktionsweise verlustbehafteter Verfahren . . . . .	22
2.3.3 Qualitätsverlust – Artefakte . . . . .	24
2.3.4 Besonderheiten gängiger Datenreduktionsverfahren . . . . .	25
2.3.4.1 ISO-MPEG1/2 Layer I und II (MP1 und MP2) . . . . .	25
2.3.4.2 ISO-MPEG1/2 Layer 3 (MP3) . . . . .	26
<b>3 Praktische Untersuchung</b>	<b>28</b>
3.1 Fragestellung, Einordnung und Hypothesenbildung . . . . .	28
3.2 Versuchsanordnung . . . . .	30
3.2.1 Hard- und Software . . . . .	30
3.2.1.1 Octave . . . . .	31
3.2.1.2 Datenreduzierte Codierung . . . . .	31
3.2.1.3 Psychoakustische Qualitätsanalyse mittels „PQevalAudio“	32
3.2.2 Testsignale . . . . .	34
3.2.2.1 Referenz-Testsignale der EBU . . . . .	34
3.2.2.2 Exemplarische Testsignale aus dem Rundfunk . . . . .	35
3.2.3 Pegelanalyse und -änderung mittels Octave . . . . .	37
3.2.3.1 Ermittlung der Programmlautheit . . . . .	37
3.2.3.2 Ermittlung des Maximum True Peak Levels . . . . .	38
3.2.3.3 Ermittlung des Spitzenpegels . . . . .	40
3.2.3.4 Pegeländerung . . . . .	41
3.3 Versuchsdurchführung . . . . .	41
3.4 Auswertung . . . . .	44
3.4.1 Pegelanalyse der Sequenz . . . . .	46
3.4.2 Deskriptive Statistik der Qualitätsanalyse . . . . .	46
3.4.3 Hypothesenprüfung . . . . .	48
<b>4 Schlussbetrachtungen</b>	<b>51</b>
<b>Quellenverzeichnis</b>	<b>53</b>
<b>Tabellenverzeichnis</b>	<b>57</b>

<i>Ch. Kopel: Auswirkung der Aussteuerung nach EBU R 128 auf Datenreduktion</i>	V
<b>Abbildungsverzeichnis</b>	<b>58</b>
<b>A Octave-Quellcode</b>	<b>59</b>
A.1 Funktion zur Ermittlung der Programmlautheit (Datei loudness.m) . . . . .	59
A.2 Funktion zur Ermittlung des Maximum True Peak Levels (Datei truepeak.m)	62
A.3 Funktion zur Ermittlung des maximalen Spitzenpegels (Datei ppm.m) . . . . .	63
A.4 Funktion zur Berechnung und Änderung des logarithmischen Pegels (Datei level.m) . . . . .	65
A.5 Analyse der MP2-codierten Sequenzen (Datei PQevalAudio_blockwise_MP2.m) . . . . .	67
<b>B Interne Übersichten der ARD zu verwendeten Verfahren   und Datenraten bei der Quellcodierung</b>	<b>69</b>
B.1 Digitales Fernsehen (DVB) . . . . .	69
B.2 Hörfunk und Fernsehen im Internet . . . . .	70
<b>C Online-Quellen</b>	<b>71</b>
C.1 Loudness Alliance – Loudness Normalization . . . . .	72
C.2 FFmpeg documentation . . . . .	76
C.3 LAME – Detailed command line switches . . . . .	81
C.4 Kurzbeschreibung PQevalAudio . . . . .	88
C.5 Kurzbeschreibung ResampAudio . . . . .	90

# Kapitel 1

## Einleitung

Im Mittelpunkt dieser Bachelorarbeit steht die Einführung eines lautheits- und True Peak-orientierten Aussteuerungsverfahrens in Hörfunk und Fernsehen sowie dessen mögliche Auswirkungen auf die Tonqualität perzeptiver Audiodatenreduktion.

Schon seit Jahrzehnten ist die Wissenschaft der Akustik bestrebt, die Lautheitsempfindung des menschlichen Gehörs möglichst originalgetreu und gleichzeitig universell erfassen zu können. Dieses Thema hat in den letzten Jahren erheblich an Bedeutung gewonnen, da Hörer und Zuschauer wie auch Toningenieure immer häufiger Lautheitsunterschiede zwischen verschiedenen Rundfunksendern und Programmblöcken beklagen. Als wichtige Gründe hierfür können erstens der Trend zur verstärkten Lautheitsmaximierung mittels Dynamikkompression angesehen werden, zweitens die zunehmende Automatisierung der Aussteuerung im Sendebetrieb. Aus diesem Grund standardisierten die International Telecommunication Union (ITU) und die European Broadcasting Union (EBU) in den Empfehlungen ITU-R BS.1770 und EBU R 128 ein Verfahren zur Messung der Lautheit des Sendematerials und empfehlen die Aussteuerung auf einen einheitlichen Lautheitspegel sowie die Einhaltung einer Obergrenze durch den „True Peak Level“. Während diese Art der Aussteuerung von mehreren Fernsehsendern bereits eingeführt worden ist, zögern die meisten Hörfunksender noch mit der Umsetzung dieser Empfehlungen. Da auch mir persönlich sowie Freunden und Verwandten die Lautheitsunterschiede sehr störend aufgefallen sind, wurde mein Interesse für diese Thematik geweckt.

Vom 2. Mai bis zum 31. Juli 2013 erhielt ich die Gelegenheit, zur Erstellung meiner Bachelorarbeit ein Praktikum in der Abteilung „Technologie und Innovationsmanagement“ des Westdeutschen Rundfunks, Köln (WDR) zu absolvieren. Der WDR äußerte großes Interesse an einer Untersuchung möglicher Auswirkungen der Aussteuerung gemäß der Empfehlung EBU R 128 auf die Tonqualität bei der Datenreduktion, die zur digitalen Programmverbreitung in Hörfunk und Fernsehen eingesetzt wird. Von besonderer Bedeutung ist dabei die Frage, ob durch die neue Art der Aussteuerung im Zusammenwirken mit Datenreduktion ein Qualitätsverlust bzw. eine höhere erforderliche Datenrate und damit ein größerer finanzieller Aufwand verbunden sein kann. Deshalb wird dieses Thema in der vorliegenden Arbeit untersucht.

Diese Bachelorarbeit gliedert sich in vier Hauptkapitel: Kapitel 2 setzt sich ausführlich mit diversen Verfahren der Pegelmessung und Aussteuerung im Hinblick auf Lautheit und Tonqualität auseinander. In diesem Zusammenhang werden die neuen Empfehlungen zur Lautheitsmessung und Aussteuerung beschrieben. Außerdem werden das Funktionsprinzip perzeptiver Audiodatenreduktion und die damit einhergehenden Qualitätseinbußen in den wichtigsten Grundzügen erläutert. Die praktische Untersuchung wird in Kapitel 3 beschrieben, wobei auf verwendetes Material, einige von mir für die Untersuchung programmierte Funktionen in Octave und die genaue Versuchsdurchführung eingegangen wird. Schließlich werden die erhaltenen Messergebnisse dargestellt und analysiert. In Kapitel 4 werden Schlussfolgerungen aus den gefundenen Ergebnissen gezogen und Empfehlungen ausgesprochen.

## Kapitel 2

# Theoretischer Hintergrund

### 2.1 Lautheit allgemein

Vor der Behandlung des ersten Themenschwerpunkts dieser Arbeit soll zunächst auf den allgemeinen Lautheitsbegriff und die verschiedenen Ansätze zur Bestimmung der Lautheit eingegangen werden. Diese Einführung wird bewusst kurz gehalten, da die Thematik bereits in vielen wissenschaftlichen Arbeiten ausführlich beleuchtet wird (vgl. [1], [2], [3], [4]). Sofern nicht anders angegeben, stützen sich die Ausführungen in diesem Kapitel auf [1, S. 17 – 25].

#### 2.1.1 Definition

Der Begriff „Lautheit“ beschreibt eine Größe aus dem Gebiet der Psychoakustik, also der Wissenschaft der subjektiven Wahrnehmung von Schall. Die Lautheit bezeichnet die empfundene Intensität eines Schallereignisses. Es handelt sich dabei um eine Übersetzung des zunächst im Englischen von Fletcher und Munson eingeführten Begriffs „Loudness“:

„LOUDNESS is a psychological term used to describe the magnitude of an auditory Sensation.“ [5, S. 82]

Wie jedes psychoakustische Maß lässt sich auch die Lautheit nicht unmittelbar aus physikalischen Größen bestimmen. Neben dem messbaren Schalldruck bzw. der Schallintensität am Ort des Hörers spielen physiologische Eigenschaften wie die Beschaffenheit des Kopfes, die Übertragungseigenschaften des Außen- und Mittelohres sowie die im Innenohr stattfindende Signalanalyse und die kognitive Verarbeitung der vom Ohr gesendeten neuronalen Impulse eine entscheidende Rolle. Die quantitative Bestimmung der Lautheit eines beliebigen Schallreizes stellt daher ein komplexes Problem dar, welches bis heute Gegenstand der Forschung ist. Hinzu kommt, dass die genannten physiologischen Parameter von Mensch zu Mensch verschieden sind und darüber hinaus auch psychologische Faktoren in die Wahrnehmung einfließen. Daher ist es kaum möglich, die individuelle Lautheitsempfindung eines bestimmten Menschen zu ermitteln, sondern nur, diese durch

ein allgemeines Modell anzunähern. Ziel dabei ist es, einen Zusammenhang zwischen Parametern des Schallreizes und der durchschnittlich empfundenen Lautheit herzuleiten.

### 2.1.2 Ermittlung der Lautheit

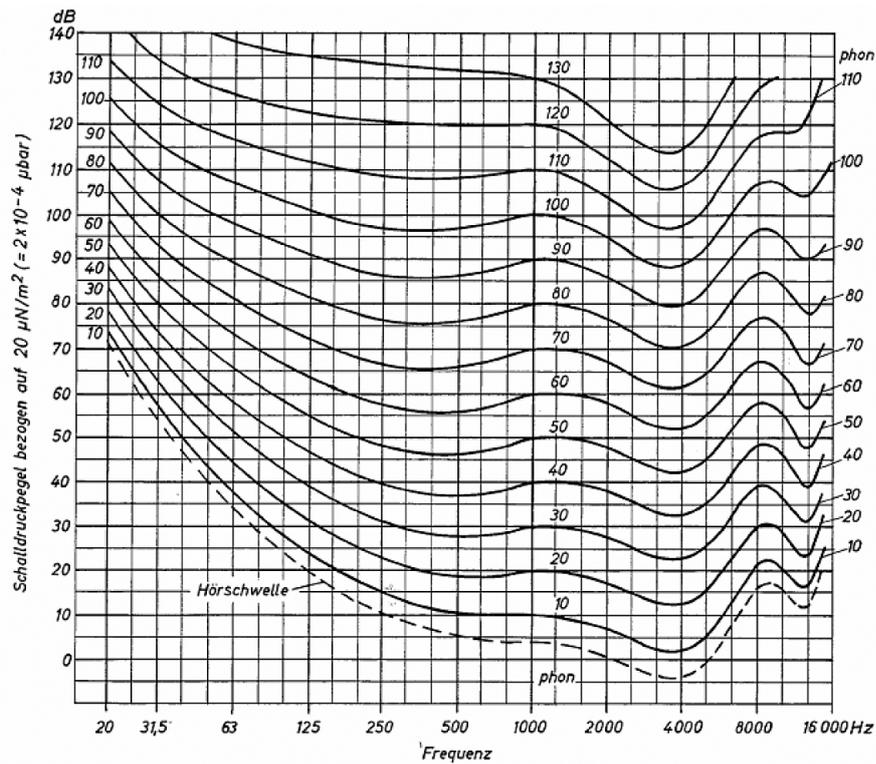


Abbildung 2.1: Kurven gleicher Lautstärke (Isophone, aus [6], übernommen aus [3, S. 11])

Barkhausen konnte mittels Hörversuchen 1926/27 zeigen, dass die empfundene Schallintensität nicht nur vom gemessenen Schalldruckpegel abhängt, sondern auch von der Frequenz des Schalls, und führte den Lautstärkepegel<sup>1</sup>  $L_N$  mit Maßeinheit „phon“ ein [2, S. 8]. 1 phon entspricht genau der empfundenen Intensität eines Sinustons mit Schalldruckpegel  $L_p = 1$  dBSPL<sup>2</sup> bei 1000 Hz. Eine graphische Darstellung der Frequenzabhängigkeit des Gehörs lieferten erstmals Fletcher und Munson 1933 mit ihren „Kurven gleicher Lautstärke“: Die Kurven zeigen den Schalldruckpegel in dBSPL über der Frequenz, den ein Sinuston aufweisen muss, um zu einer gleich hohen Lautstärkeempfindung in phon zu führen, weshalb sie auch als „Phonkurven“ oder „Isophone“ bezeichnet werden. Abb. 2.1 zeigt die in ISO 226 genormte Version der Kurven (nach Robinson und Dadson 1956). Man erkennt aus den Kurven, dass das Gehör bei Frequenzen zwischen 2000 und 4000 Hz die höchste Sensibilität besitzt und zu höheren sowie besonders zu niedrigeren

1. Im Unterschied zum Begriff „Lautheit“ wird mit dem Begriff „Lautstärke“ oder „Lautstärkepegel“ in der Akustik meistens die frequenzabhängige empfundene Intensität in phon bezeichnet.

2. dBSPL = decibel Sound Pressure Level

Frequenzen hin unempfindlicher wird, wobei diese Frequenzabhängigkeit wiederum bei höheren Schalldruckpegeln weniger stark ausgeprägt ist als bei niedrigeren. Anzumerken ist hierbei, dass die Kurven gleicher Lautstärke in diversen Hörversuchen neu bestimmt wurden und sich dabei teilweise erhebliche Abweichungen der einzelnen Messergebnisse zueinander zeigten (siehe Abb. 2.2).

Stevens ermittelte 1956, dass schon der Zusammenhang zwischen Schalldruck oder Schalldruckpegel und Lautheit nicht linear ist [2, S. 8 – 9]. So führt z. B. die Zunahme des logarithmischen Pegels eines Sinustons bei 1000 Hz von 40 auf 50 dB SPL (bzw. die Zunahme der linearen Schallintensität um den Faktor 10) nur zu einer Verdoppelung des Lautheitseindrucks. Eine Zunahme des Pegels um 20 dB (bzw. ein Anstieg der Intensität um den Faktor 100) führt entsprechend zu einer Vervierfachung der Lautheit. Für Pegel unter 40 dB SPL ergibt sich eine Erhöhung des Lautstärkeindrucks schon bei geringeren Pegelanstiegen. Um die Lautheitsverhältnisse zweier Schallereignisse quantitativ ausdrücken zu können, führte Stevens die Einheit „sone“ ein, welche auch heute in der Regel als Lautheitsmaß verwendet wird. Die Verdoppelung der empfundenen Schallintensität wird damit durch eine Verdoppelung des Lautheitswertes in sone ausgedrückt, eine Halbierung der empfundenen Intensität durch eine Halbierung des sone-Wertes. 1 sone wurde dabei die Lautheit  $N$  eines Sinustons bei 1000 Hz mit Schalldruckpegel  $L_p = 40$  dB SPL, also Lautstärkepegel  $L_N = 40$  phon, zugeordnet, 2 sone folglich ein Lautstärkepegel von 50 phon und 4 sone ein Pegel von 60 phon.

Bei Laborversuchen im Jahre 1967 stellten Zwicker und Feldtkeller fest, dass sich auch die Dauer eines Reizes auf die Lautheit auswirkt: Ein Ton von 100 ms Länge wird doppelt so laut empfunden wie ein Ton gleicher Frequenz und Schallintensität mit nur 10 ms Dauer. Das Ohr führt also eine Art zeitliche Integration der Schallintensität durch. Dieser Einfluss lässt sich nur bis zu einer Dauer von 200 ms beobachten; bei länger anhaltenden Schallreizen wirkt sich die Dauer nicht mehr auf die Lautheit aus.

Die dargestellten Ergebnisse genügen nur zur Berechnung der Lautheit reiner Sinustöne und allenfalls schmalbandigen Rauschens innerhalb einer Frequenzgruppe des menschlichen Gehörs. Im Falle von breitbandigeren Klängen oder Geräuschen müssen weitere Faktoren wie die Frequenzgruppenauflösung und die Verdeckung durch das Innenohr (siehe Kap. 2.3.2, S. 22) berücksichtigt werden. So ermittelten Zwicker und Feldtkeller auch, dass eine Verbreiterung des Frequenzspektrums nur innerhalb der Grenzen einer Frequenzgruppe des Gehörs bei gleichzeitig konstant gehaltenem Schalldruckpegel nicht zu einem Lautheitsanstieg führt, wohl aber eine Verbreiterung über die Grenzen der Frequenzgruppe hinaus.

- ▽ Kirk(1983)PF      △ Möller(1984)PF      ● Betke(1989)      ○ Suzuki(1989)  
 ■ Fastl(1990)      ● Watanabe(1990)      ▲ Poulsen(1994)      ◇ Suzuki(1996)  
 ▼ Lydolf(1997)      ◀ Lydolf(1997)PF      □ Takeshima(1997)      ▷ Bellmann(1999)  
 ◁ Bellmann(1999)PF      ⊗ Takeshima(2001)      + Robinson-Dadson      × Brinkmann(1973)HT  
 \* Vorländer(1991)HT      \* Poulsen(2000)HT      (1956)HT
- PF: Gemessen im Druckfeld      HT: Nur Hörschwelle

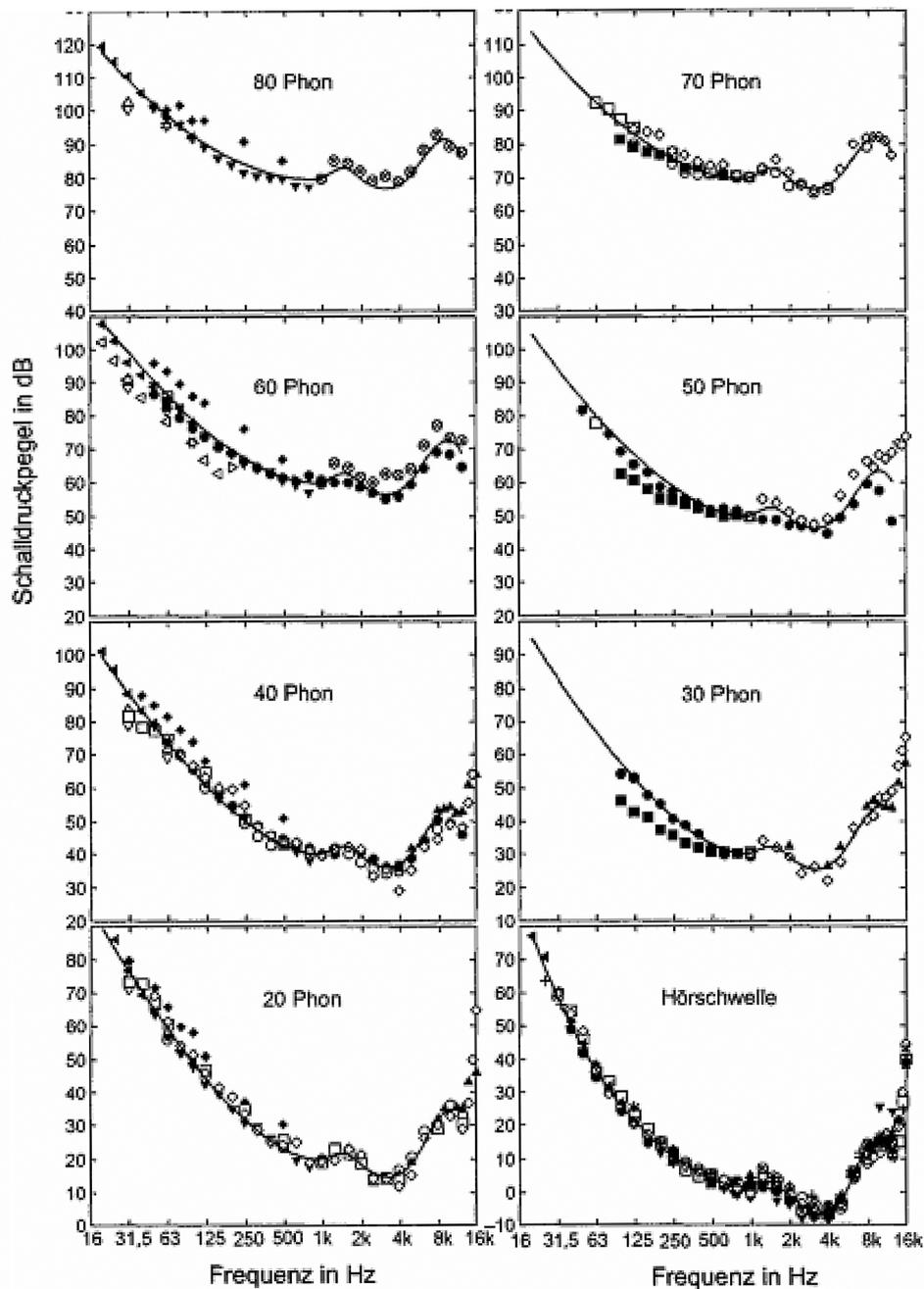


Abbildung 2.2: Kurven gleicher Lautstärke verschiedener Versuchsreihen (aus [6], übernommen aus [3, S. 12])

Ein von dieser Erkenntnis ausgehendes und bis heute verwendetes Modell der Lautheitswahrnehmung geht auf Zwicker (1967) zurück. Es basiert auf der Bestimmung eines so. g. Erregungsmusters. Dabei wird das Frequenzspektrum des Schallsignals zunächst mit der Übertragungsfunktion des Außen- und Mittelohres gewichtet und in Frequenzbänder unterteilt, die näherungsweise den Frequenzgruppen des Innenohres entsprechen. Aus den einzelnen Schallintensitätspegeln innerhalb jeder Frequenzgruppe wird nun ein „Erregungspegel-Tonheitsmuster“ berechnet.<sup>3</sup> Die so erhaltenen frequenzgruppenabhängigen Gesamterregungspegel werden anschließend mittels einer Potenzfunktion in eine Verteilung spezifischer Lautheiten  $N'(z)$  umgerechnet. Die Integration der spezifischen Lautheiten über alle Frequenzgruppen  $z$  führt schließlich zur Gesamtlautheit  $N$  in sone. Diese als Zwicker-Verfahren bekannte Methode wurde in den Normen DIN 45631 und ISO R 532 zur Lautheitsberechnung standardisiert. Erweiterungen bzw. Verbesserungen wurden seither vorgeschlagen (vgl. [7]).

Alle bisher vorgestellten Verfahren zur Lautheitsberechnung lassen sich nur auf stationäre Klänge oder Geräusche anwenden. Zwicker und Fastl definierten später die Lautheit  $N_5$  eines instationären Schallereignisses als das 95 %-Quantil der Verteilung der gemessenen Lautheitswerte, also als denjenigen Lautheitswert, der während der Dauer des Schallereignisses in 5 % der Zeit überschritten wird [8, S. 9]. Glasberg und Moore stellten 2002 ein Modell zur Berechnung einer „Short-Term Loudness“ und einer „Long-Term Loudness“ für zeitvariante Signale vor [9]. Ihre Methode sieht die Bestimmung von Momentan-Lautheitswerten („Instantaneous Loudness“) gemäß ihres Modells in [7] vor, die blockweise im Abstand von 1 ms erfolgt. Diese Momentan-Lautheitswerte werden zur „Short-Term Loudness“ aufintegriert und gemittelt, wobei der jeweils aktuelle Momentanwert mit dem aktuellen „Short-Term Loudness“-Wert verglichen wird. Auf die gleiche Weise erfolgt wiederum die Berechnung der „Long-Term Loudness“ aus den Werten der „Short-Term Loudness“. Anzumerken ist, dass beide Modelle nur für synthetische Klänge wie amplitudenmodulierte Sinus- bzw. Rauschsignale oder Impulsfolgen validiert wurden. Die Aussagekraft dieser Modelle für natürliche Signale wie Sprache, Musik oder ein ganzes Rundfunkprogramm ist umstritten [10, S. 2]. Hier stellen Hörversuche nach wie vor die zuverlässigste Methode dar [3, S. 10].

## 2.2 Lautheit im Rundfunk

In diesem Kapitel werden die bisher im Rundfunk üblichen Aussteuerungsverfahren und eine neue, lautheitsorientierte Art der Aussteuerung einander gegenübergestellt.

Die Umstellung ganzer Rundfunkanstalten auf eine völlig neue Aussteuerungsmethode ist zweifellos mit einem hohen Aufwand und nicht zu unterschätzenden Kosten verbunden: Sämtliche Aussteuerungsmessgeräte müssen ausgewechselt, evtl. eingesetzte automatische Aussteuerungssysteme entsprechend in ihrem Regelverhalten angepasst werden. Techniker und eigenständig produzierende Journalisten müssen sich mit der neuen Art

---

3. Die Gesamterregung jeder Frequenzgruppe wird durch Intensitätsaddition einer so g. Kernerregung und einer Flankenerregung bestimmt. Da dieses Verfahren für die Betrachtungen dieser Arbeit keine zentrale Bedeutung besitzt, wird hier nicht näher darauf eingegangen.

der Aussteuerung vertraut machen. Darüber hinaus herrscht bisweilen Unsicherheit über die Konsequenzen einer geänderten Aussteuerung nicht nur auf Datenreduktionsverfahren, sondern z. B. auf das von einem Rundfunksender entworfene „Sound Design“. Angesichts dieser Überlegungen stellt sich die Frage, weshalb überhaupt eine Änderung der Aussteuerungspraxis notwendig ist. Deshalb werden die Nachteile des derzeitigen Umgangs mit Pegel und Lautheit im Rundfunk untersucht. Ein Überblick der Lösungsansätze zu dieser Problematik führt zu einer Vorstellung der entsprechenden Empfehlungen durch die EBU und die ITU. Dabei wird der Entwicklungsprozess beschrieben, der zu dem von der ITU empfohlenen Messalgorithmus geführt hat. Eine Bewertung dieser Messmethode im Vergleich zu anderen Lautheitsmessverfahren ist nicht Gegenstand dieser Arbeit. Die Empfehlung ITU-R BS.1770-2 [11] wird als De-facto-Standard anerkannt und dient als Basis für alle praktischen Untersuchungen.

### 2.2.1 Bisherige Aussteuerungspraxis

Aussteuerung in der Audiotechnik definiert Weinzierl [12, S. 552] als „eine im Hinblick auf den Übertragungskanal optimierte Einstellung des Signalpegels.“.

Die Aussteuerung erfolgt mit einem Aussteuerungsmesser, der allgemein den logarithmischen Pegel in Dezibel (dB) des Tonsignals ausgibt. Im Falle analoger Technik ergibt sich der Pegel  $L$  aus dem logarithmierten Verhältnis der Signalspannung  $U$  zu einer Bezugsspannung  $U_0$  gemäß

$$L = 20 \log_{10} \frac{U}{U_0}. \quad (2.1)$$

Der Wert für die Bezugsspannung, der damit auch dem Pegel  $L = 0$  dB entspricht, wird dabei je nach Anwendungsfall und verwendeter Elektronik unterschiedlich festgelegt, häufig z. B. auf  $U_0 = 0,775$  V (Spannungspegel in dBU),  $U_0 = 1$  V (Spannungspegel in dBV) oder  $U_0 = 1,55$  V (Funkhauspegel in dB der Rundfunkanstalten in Deutschland) [12, S. 30].

Bei digitalen Systemen hingegen ist eine eindeutige obere Aussteuerungsgrenze gegeben. Sie entspricht der bei einer gewählten Datenwortlänge  $k$  höchsten darstellbaren Zahl [12, S. 553], d. h. der höchstmöglichen Quantisierungsstufe  $Q$ . Diese berechnet sich – ausgehend von einer Ganzzahldarstellung ohne Vorzeichen – zu

$$Q = 2^k - 1 \quad (2.2)$$

Ein Überschreiten dieser Grenze führt zu hörbaren nichtlinearen Verzerrungen (Clipping). Der Pegel wird hier aus dem logarithmierten Verhältnis der wertdiskreten Signalamplitude  $x_d$  zu der höchsten Quantisierungsstufe  $Q$  gebildet:

$$L = 20 \log_{10} \frac{x_d}{Q} \quad (2.3)$$

Der Bezug auf den Maximalwert  $Q$  wird durch die Angabe „decibel relative to Full Scale“ (dBFS) angezeigt.

Sowohl in der analogen als auch in der digitalen Domäne finden sich verschiedene Typen von Aussteuerungsmessern, die im Wesentlichen durch ihr zeitliches Verhalten charakterisiert werden: Es gibt so g. Spitzenspannungsmesser (analoger Fall) bzw. Sample Peak Level Meters (digitaler Fall), die den (betragsmäßigen) unmittelbaren Momentanwert der Signalamplitude anzeigen, und solche, die über ein bestimmtes Zeitintervall den betragsmäßigen oder quadratischen Mittelwert bilden [12, S. 555 – 564]. Die erste Variante mag zunächst als das Naheliegendste erscheinen, denn nur auf diese Weise kann sicher überprüft werden, ob die festgelegte Aussteuerungsgrenze auch kurzzeitig nicht überschritten wird. Deshalb ist gerade bei digitaler Aufnahmetechnik ein sample-genauer Aussteuerungsmesser weit verbreitet. Im europäischen Rundfunk ist ein Messinstrument mit dieser Eigenschaft als „Peak Programme Meter“ (PPM) in DIN IEC 60268-18 standardisiert.

Ein Nachteil des PPM liegt darin, dass es nur sehr bedingt die Lautheit des Tonsignals abbildet, da kurzzeitige Signalspitzen – wie v. a. bei Sprache häufig anzutreffen – deutlich weniger zur Lautheit beitragen als länger anhaltende Signale (siehe Kap. 2.1.2, S. 5). Deshalb werden im Rundfunkbereich meist Aussteuerungsmesser mit einer kurzen Mittelungs- bzw. Ansprechzeit (Integrationszeit) zwischen 5 und 10 ms bevorzugt. Die Rundfunkanstalten in Deutschland verwenden das in DIN IEC 60268-18 und [13] spezifizierte „Quasi Peak Programme Meter“ (QPPM). Es lässt sich zwischen einer sample-genauen Spitzenwertanzeige und einer Ansprechzeit von 10 ms umschalten. Da jedoch auch diese Mittelung allein einerseits nicht allen für die Lautheit relevanten Faktoren Rechnung trägt, andererseits aber kurzzeitige Übersteuerungen von weniger als 10 ms Dauer nicht zuverlässig angezeigt werden, kann eine solche Mittelung als „Kompromiss“ [12, S. 555] zwischen beiden Forderungen betrachtet werden.

## 2.2.2 Problematik

Die Tatsache, dass keines der bisher vorgestellten Aussteuerungsinstrumente zuverlässig Auskunft über die Lautheit des Programmmaterials gibt, macht sich schon bei verschiedenen Programmkategorien wie Sprache, Popular- oder Kunstmusik bemerkbar, da diese von sich aus unterschiedliche Signalverläufe aufweisen [12, S. 559]. Werden alle Inhalte mittels eines PPM oder QPPM auf einen gleichen Zielwert angesteuert, ist daher ein Ungleichgewicht der Lautheiten zwischen verschiedenen Programmbeiträgen zu erwarten. Aus diesem Grund wird in diversen Richtlinien empfohlen, Sprache 6 dB niedriger als Kunstmusik auszusteuern sowie Populärmusik 6 dB niedriger als Sprache.

Hinzu kommt jedoch, dass durch eine Nachbearbeitung mittels Regelverstärkern wie Begrenzern und v. a. Kompressoren die Dynamik einzelner Beiträge oder ganzer Rundfunkprogramme verringert wird, was – nach anschließender Verstärkung auf den ursprünglichen Maximalpegel (durch entsprechende Anhebung des „Output Gain“) – zu einer Erhöhung der Lautheit führt [12, S. 732], [14, S. 1]. Dieser Effekt kommt zustande, da Signalanteile mit geringerem Pegel gegenüber dem Maximalpegel eines Signals verstärkt

werden und sich so eine höhere mittlere Signalleistung ergibt. Die zeitliche Integration des Gehörs (siehe Kap. 2.1.2, S. 5) ergibt somit einen höheren Wert.

Der Einsatz eines Kompressors dient traditionell der Anpassung der Dynamik natürlicher Schallquellen wie Sprache, Gesang und akustischer Musikinstrumente an die begrenzte Dynamik eines Mediums oder einer Abhörsituation [12, S. 730, 738]. So ist es z. B. im Rundfunk üblich, wichtige Programmelemente wie Nachrichten und Verkehrshinweise stärker zu komprimieren, um eine gute Sprachverständlichkeit auch bei schlechtem Empfang und an Orten mit hohem Grundgeräuschpegel (z. B. im Straßenverkehr) zu gewährleisten [2, S. 15 – 20], [12, S. 780 – 781]. Auch als Mittel der Klanggestaltung kommen schon seit Jahrzehnten Kompressoren zum Einsatz. Seit einigen Jahren wird Komprimierung jedoch zunehmend auch gezielt dazu eingesetzt, um bei Einhaltung der vorgegebenen Aussteuerungsgrenze die Lautheit zu erhöhen. Dies hat erstens zum Ziel, größere Aufmerksamkeit beim potentiellen Hörer zu wecken, zweitens verspricht man sich von einer höheren Lautheit auch eine subjektiv empfundene Klangverbesserung [12, S. 739, 780 – 781].<sup>4</sup> Diese Praxis war und ist insbes. in den Bereichen Werbung und populäre Musik zu beobachten. Doch auch zur Erhöhung der Lautheit eines ganzen Rundfunksenders wird im Rahmen des „Sound Designs“ heutzutage oftmals eine Dynamikeinengung vorgenommen, um sich gegen andere Programme „durchzusetzen“ und einem potentiellen Hörer bei der Sendersuche am stärksten aufzufallen. Diverse Quellen wie [12, S. 781], [15, S. 10 – 14] und [16, S. 1] sprechen in diesem Zusammenhang von einem „Lautheitswettbewerb“ oder „Lautheitskrieg“, der sich mittlerweile unter Produzenten bzw. Rundfunksendern entwickelt habe.

Die beschriebene Entwicklung führt wiederum dazu, dass es sowohl zwischen Rundfunksendern als auch innerhalb eines Rundfunkprogramms zu erheblichen Lautheitsdifferenzen kommen kann, da verschiedene Produktionsstudios und Rundfunksender jeweils unterschiedliche Komprimierungsparameter verwenden können und ältere Produktionen tendenziell einen geringeren Grad an Komprimierung aufweisen [16, S. 1 – 2]. Hinzu kommt, dass die Programmblöcke im Zuge von Aufnahme, Mischung, Mastering und Ausstrahlung meistens unterschiedlich oft komprimiert werden. Das Problem wird dadurch weiter verschärft, dass der Sendebetrieb heutzutage oftmals ohne Techniker erfolgt und die Aussteuerung von automatischen Aussteuerungsreglern (so g. „Automatic Gain Controls“ = AGCs) durchgeführt wird [12, S. 559]. Hat im Falle einer manuellen Aussteuerung der Techniker noch die Möglichkeit, nach Gehör oder Beobachtung des Verlaufs der QPPM-Werte zumindest eine begrenzte Lautheitskorrektur vorzunehmen, heben nach QPPM arbeitende AGCs ohne Kenntnis der Lautheit des Signals den Pegel auf den eingestellten Zielwert an.

Tatsächlich häufen sich Beschwerden von Zuschauern und Zuhörern über störende Lautheitsunterschiede [17, S. 3 – 4]. Bei Messungen deutscher Rundfunkprogramme wurden Lautheitssprünge um mehr als den Faktor 2 beobachtet [18, S. 361]. Aus diesem Grund plädieren schon seit einigen Jahren diverse Autoren für die Entwicklung und Einführung eines lautheitsorientierten Aussteuerungsverfahrens [14], [16], [18], [19], [20]. So argu-

---

4. Ein solcher Effekt konnte allerdings bisher noch nicht wissenschaftlich nachgewiesen werden. Vielmehr fanden Birkenhake und Struck [2], dass eine Dynamikkompression oftmals nicht als Klangverbesserung empfunden bzw. von vielen Hörern gar nicht wahrgenommen wird.

mentieren beispielsweise die Mitglieder der „Loudness Alliance“ in [16, S. 1 – 3] nicht nur mit der Vermeidung der besagten Lautheitsunterschiede, sondern auch damit, dass mit der aktuellen Fokussierung auf maximale Lautheit der Verlust des Gestaltungsmittels Dynamik und eine Verschlechterung der Klangqualität einhergehe. Dieser Verlust an Dynamik lasse das Gehör ermüden.

Ein weiterer Nachteil der QPPM-Messung ergibt sich dadurch, dass Übersteuerungen unerkannt bleiben können: Aufgrund der zeitlichen Mittelung werden kurzzeitige Spitzen des Sample-Pegels nicht angezeigt (siehe Kap. 2.2.1, S. 9). Aus diesem Grund wird im europäischen Rundfunkbereich üblicherweise auf einen Zielwert von -9 dBFS ausgesteuert [18, S. 361]. Ob diese Übersteuerungsreserve („Headroom“) jedoch auch im Einzelfall ausreicht, ist nicht gesichert. So soll für einen 0,4 ms langen Sinuston bei 5000 Hz und Sample Peak Level 0 dBFS gemäß Spezifikation [13, S. 7] der auf dem QPPM angezeigte Pegel nur -15 dBFS betragen.

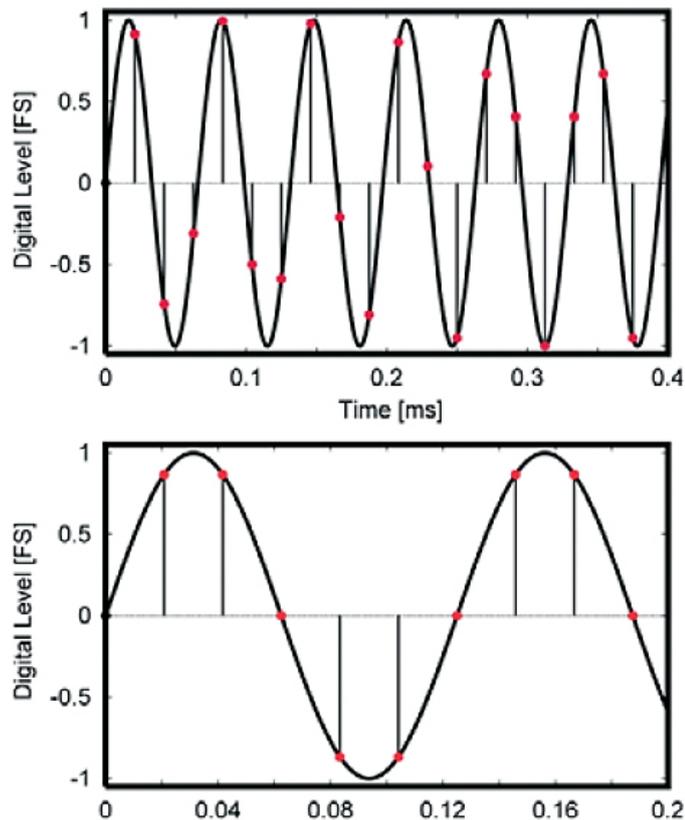


Abbildung 2.3: Sinussignal vor (Linie) und nach der Digitalisierung (Punkte) mit einem True Peak Level oberhalb des Sample Peak Levels (aus [14, S. 2])

Das Übersteuerungsrisiko erhöht sich bei digitaler Audiotechnik dadurch, dass die diskreten Abtastzeitpunkte (Samples) eines digitalen Systems im Allgemeinen nicht mit den Zeitpunkten übereinstimmen müssen, zu denen das entsprechende (originale oder rekonstruierte) zeitkontinuierliche Tonsignal seine Maximalwerte erreicht [11, S. 21 – 22]. Der

Spitzenpegel eines zeitkontinuierlichen Signals – auch bezeichnet als „True Peak Level“ – kann damit höher liegen als der messbare Sample Peak Level des digitalisierten Signals. Zur Verdeutlichung ist in Abb. 2.3 ein Sinussignal vor und nach der Digitalisierung dargestellt.

Dies hat zur Folge, dass auch bei einem dauerhaft unter 0 dBFS gehaltenen Sample Peak Level der True Peak Level die Grenze von 0 dBFS überschreiten kann und z. B. bei einer Digital-Analog-Umsetzung Übersteuerungen des Systems auftreten können. Doch auch bei rein digitaler Weiterverarbeitung eines Signals wie etwa Filterung, Abtastratenwandlung oder Datenreduktion kann nach Untersuchungen von Lund [14, S. 3 – 5] ein True Peak Level über 0 dBFS zu hörbaren Verzerrungen führen. Lund empfiehlt daher einen Zielwert von höchstens -3 dBFS schon für den Sample Peak Level.

### 2.2.3 Ansätze zur Lautheitsmessung im Rundfunk

In den letzten Jahren wurden verschiedene Messverfahren zur Lautheitsmessung typischen Rundfunkmaterials vorgeschlagen, die sich grob in zwei Kategorien einteilen lassen: jene, die auf den komplexeren psychoakustischen Modellen nach Zwicker, Glasberg und Moore beruhen, und solche, die im Wesentlichen nur eine Filterung und zeitliche Mittelung durchführen. Anders formuliert unterscheiden sich beide Ansätze nach [8, S. 5] darin, ob eine möglichst präzise Modellierung des menschlichen Gehörs angestrebt wird oder nur die Erfassung eines Eingangs-Ausgangs-Verhaltens. Die erste Variante führt zwangsläufig zu einem Multiband-Verfahren, d. h. einer Signalanalyse, welche der Frequenzgruppeneinteilung durch das Gehör nachempfunden ist, während für die zweite Variante meist einfachere Verfahren ohne Frequenzanalyse zum Einsatz kommen.

#### 2.2.3.1 Psychoakustische Verfahren

Eine Anwendung der auf Zwicker basierenden psychoakustischen Modelle für die Aussteuerung im Rundfunk bringt zwei Probleme mit sich: Erstens wurden sie nicht für beliebige natürliche Signale wie Sprache und Musik entwickelt (siehe Kap. 2.1.2, S. 7), zweitens sind sie sehr komplex, d. h. schwieriger zu implementieren und rechenintensiv. Dennoch stellte schon im Jahre 1996 die Firma DK-Audio eine rechnergestützte Implementierung des Zwicker-Verfahrens vor, die für Aussteuerungsanwendungen geeignet sein, also Echtzeitanforderungen genügen sollte [19, S. 6]. Eine ebenfalls auf Frequenzanalyse beruhende Methode zur lautheitssensitiven Aussteuerung für Rundfunk und Mastering entwickelte Ks Waves Ltd. 1998 [21]. Das Modell bestand hier aus einer Filterbank in rekursiver Binärbaumstruktur, wodurch auf relativ recheneffiziente Weise die Frequenzauflösung des Gehörs nachgebildet werden sollte. Für die Berechnung eines Langzeit-Lautheitswerts schlugen die Autoren die Integration über die höchsten 20 % der im Verlauf ermittelten Lautheitswerte vor.

In einer Gegenüberstellung und Evaluierung diverser Lautheitsmessverfahren testeten Skovenborg und Nielsen [8, S. 9 – 10] zwei weitere psychoakustische Varianten: Bei der ersten wurde das Verfahren nach Zwicker und Fastl zur Ermittlung der Lautheit  $N_5$

(siehe Kap. 2.1.2, S. 7) in der Audio-Software SI++ implementiert. Die zweite Version sah eine Frequenzerlegung in 1/3-Oktavbänder und eine anschließende Berechnung des quadratischen Mittelwerts für jedes Band vor. Erst die so erhaltenen frequenzabhängigen zeitlichen Mittelwerte wurden nach dem Zwicker-Verfahren ausgewertet.

### 2.2.3.2 Verfahren basierend auf zeitlicher Mittelung

Lautheitsmessverfahren, die keine Frequenzanalyse durchführen und nur auf zeitlicher Mittelung beruhen, sind schon seit längerem weiter verbreitet. Sie bieten den Vorteil einer einfachen Realisierung und schnellen Berechnung.

Als ein traditionelles Standard-Aussteuerungsinstrument, das sich eher an der Lautheit eines Tonsignals orientiert als die anderen im Rundfunk üblichen, kann das „Volume Units Meter“ (VU-Meter) betrachtet werden [14, S. 5], [22, S. 5]. Es bildet einen Mittelwert über 300 ms [12, S. 558], wodurch die zeitliche Intensitätsintegration des Gehörs von 200 ms (siehe Kap. 2.1.2, S. 5) zumindest angenähert wird. Allerdings berücksichtigt es andere für die Lautheitsempfindung relevante Faktoren wie die Frequenzabhängigkeit des Gehörs (siehe Kap. 2.1.2, S. 4) nicht. Das VU-Meter ist v. a. in den USA verbreitet und wurde auch in DIN IEC 60268-17 standardisiert; im europäischen Rundfunk hat es sich jedoch nicht durchgesetzt.

Der bewertete energieäquivalente Pegel  $L_{eq(H)}$  wird ermittelt, indem das Signal gemäß einer so g. Bewertungskurve gefiltert und anschließend der quadratische Mittelwert (auch als Effektivwert oder „Root Mean Square“ = RMS bezeichnet) gebildet wird [8, S. 7 – 8], [12, S. 560]:

$$L_{eq(H)} = 20 \log_{10} \left( \sqrt{\int_0^T x_H(t)^2 dt} \right) \quad (2.4)$$

$H$  steht dabei für die angewandte Bewertungskurve,  $x_H(t)$  für das  $H$ -bewertete Zeitsignal und  $T$  für die Mittelungszeit. Mit  $L_{eq}$  ohne Angabe einer Kurve wird im Allgemeinen eine ungefilterte Leistungsmessung bezeichnet. Für bewertete Messungen wird als Kurve ein angenäherter Frequenzgang des Außen- und Mittelohres für einen bestimmten Schalldruckpegel gewählt; meistens die aus der Geräusch- und Lärmesstechnik bekannte A-, B-, C- oder D-Kurve. Letztere entsprechen den (vereinfachten) Inversen der Kurven gleicher Lautstärke [3, S. 16]. Für die Lautheitsmessung von Kinoton kommt oftmals die so g. M-Kurve zum Einsatz. Die Kurven sind in Abb. 2.4 dargestellt.

Vickers [23, S. 2 – 3] schlug vor, das zu messende Signal mit einer der bekannten Bewertungskurven zu filtern, daraus in kurzen Blöcken RMS-Werte zu berechnen und anschließend einen arithmetischen Mittelwert der einzelnen RMS-Werte zu bilden. Hierbei konnte optional eine höhere Gewichtung der energiereicheren Blöcke vorgenommen werden, ausgehend von einer an [21, S. 11 – 12] angelehnten Hypothese, dass höhere Kurzzeit-Lautheitswerte stärker in die Langzeit-Lautheitswahrnehmung eingingen. Als Bewertungskurve schlug er die B-Kurve (entspricht Schalldruckpegel von 60 dB SPL) vor, da diese am ehesten für eine moderate Abhörlautstärke geeignet sei.

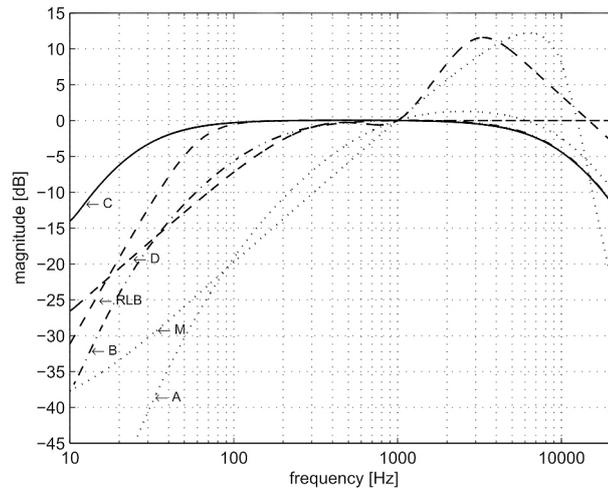


Abbildung 2.4: Bewertungskurven: A (für Schalldruckpegel von 40 dB SPL), B (für Schalldruckpegel von 60 dB SPL), C (für Schalldruckpegel von 80 dB SPL), D (für Fluglärm), M (für Kinoton) und RLB (siehe Kap. 2.2.4); aus [8, S. 7]

Einen Spezialfall stellen Messungen auf Grundlage eines modifizierten QPPM (siehe Kap. 2.2.1, S. 9) dar. So besaßen einige Aussteuerungsmesser des Herstellers RTW eine Lautheitsanzeige, die sich aus einer QPPM-Messung mit verlängerter Integrationszeit und einer Frequenzbewertung mit einer abgewandelten C-Kurve ergab [12, S. 561]. Das Institut für Rundfunktechnik, München (IRT) entwickelte einen Lautheitsmesser, der den Median der Pegelwerte eines gewöhnlichen QPPM in einem Zeitintervall von 3 s ausgab [8, S. 8 – 9], [18, S. 362].

## 2.2.4 Die Standardisierung eines Lautheitsmessverfahrens

Im Jahr 2002 begann eine speziell eingesetzte Gruppe der ITU mit der Erarbeitung eines Standards zur Lautheitsmessung im Rundfunk [18, S. 362 – 364]. Durch Hörversuche an weltweit fünf Forschungslabors wurden mittlere subjektive Lautheitswerte für 48 mono-phonie Testbeispiele aus gesendetem Rundfunkmaterial ermittelt. Anschließend wurden insgesamt zwölf Lautheitsmessverfahren am Communications Research Centre, Ottawa (CRC) unter der Leitung von Soulodre gegen die subjektiven Werte getestet. Zu den evaluierten Lautheitsmessern gehörten u. a. jener des IRT (siehe Kap. 2.2.3.2), diverse auf Grundlage psychoakustischer Modelle sowie vier auf energieäquivalenter Pegelmessung beruhende Verfahren. Dabei zeigten sich für letztere insgesamt ähnlich gute Übereinstimmungen wie für die komplexeren psychoakustischen Verfahren. Die besten Ergebnisse in fast allen Auswertungsparametern (z. B. die höchste Korrelation mit den subjektiven Lautheitswerten,  $r = 0,982$ ) lieferte die energieäquivalente Pegelmessung mit einer von Soulodre selbst vorgeschlagenen modifizierten B-Bewertungskurve für den tieffrequenten Bereich, bezeichnet als „Revised Low Frequency B Curve“ (RLB). Die Filterkurve ist in Abb. 2.4 dargestellt. Dieses mit  $L_{eq(RLB)}$  bezeichnete Verfahren wurde aufgrund seiner Stabilität bei typischem Rundfunkmaterial und seiner gleichzeitig geringen Komplexität

im Abschlussbericht der Studie als Standard für die Lautheitsmessung monophoner Audiosignale vorgeschlagen. Eine bestimmte Integrationszeit wurde nicht angegeben; aus der Beschreibung des Evaluierungsverfahrens geht hervor, dass stets die durchschnittliche Lautheit eines gesamten Testbeispiels gefragt war. Es folgte eine Erweiterung des Messverfahrens auf bis zu fünf Tonkanäle und 2006 schließlich die erstmalige Standardisierung in der Empfehlung ITU-R BS.1770.

In der 2011 veröffentlichten und derzeit gültigen Version wurde der Messalgorithmus um eine optionale zweistufige Gate-Funktionalität erweitert [11, S. 7 – 8]. Das Gating wurde auf Anregung der EBU in das Messverfahren aufgenommen, da nach Untersuchungen der EBU längere Abschnitte innerhalb eines Signals, die Stille oder nur Hintergrundgeräusch enthalten, sonst einen zu niedrig gemessenen Lautheitswert zur Folge hätten [24, S. 10 – 11].

### 2.2.5 Lautheitsmessung nach ITU-R BS.1770-2

Im Folgenden wird das derzeit gültige, in ITU-R BS.1770-2 [11, S. 4 – 9] empfohlene Lautheitsmessverfahren erläutert. Dabei werden weitestgehend die Formeln und Symbole aus dem Text der Empfehlung übernommen. Das zugehörige Blockschaltbild ist in Abb. 2.5 dargestellt.

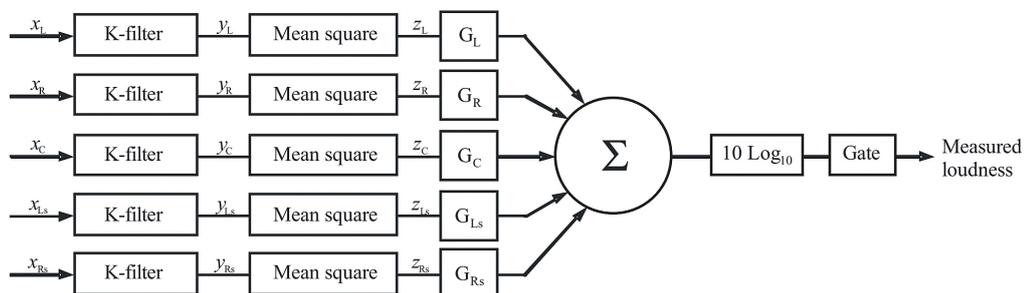


Abbildung 2.5: Blockschaltbild des von der ITU empfohlenen Lautheitsmessverfahrens (aus [11, S. 5])

Zunächst erfolgt eine zweistufige Filterung, die zusammenfassend als  $K$ -Gewichtung bezeichnet wird. Bei der ersten Filterstufe handelt es sich um einen Shelving-Filter, der die Übertragungseigenschaften des menschlichen Kopfes modellieren soll. Der Kopf wird dabei als schallharte Kugel („rigid sphere“) betrachtet. Der Frequenzgang ist in Abb. 2.6 zu sehen. Die zweite Filterstufe entspricht der RLB-Kurve nach Souloudre (siehe Kap. 2.2.4, S. 14 und Abb. 2.4). Die RLB-Filterung soll den Frequenzgang des Außen- und Mittelohres nachbilden. Die Koeffizienten für eine Realisierung beider Filter in Infinite Impulse Response-Struktur 2. Ordnung können der Tab. 2.1 entnommen werden.

Soll kein Gating (siehe Kap. 2.2.4) eingesetzt werden, wird direkt im Anschluss für jeden Tonkanal getrennt die Leistungsberechnung des ganzen Signals durchgeführt:

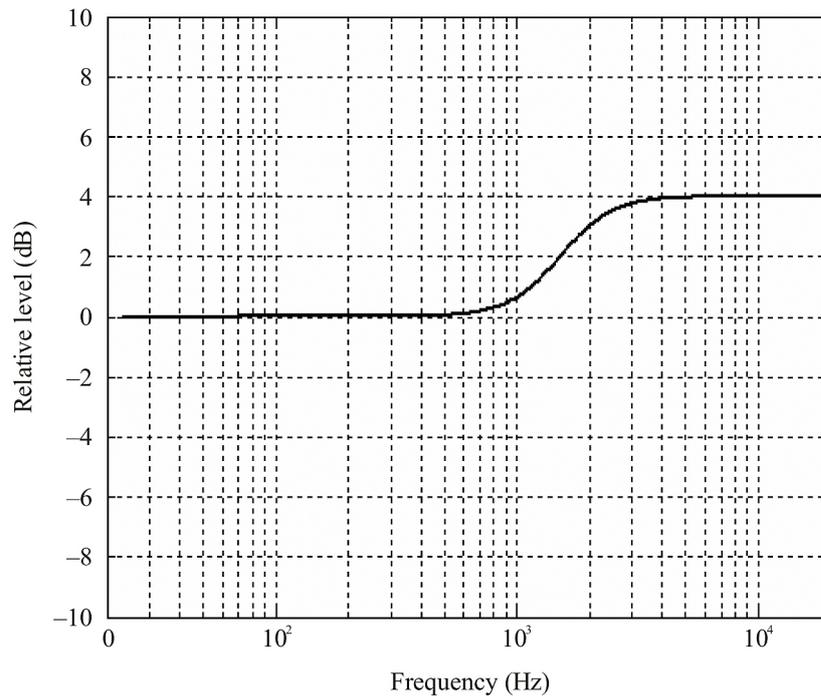


Abbildung 2.6: Frequenzgang der 1. Filterstufe zur Modellierung des Kopfes für das von der ITU empfohlene Lautheitsmessverfahren (aus [11, S. 5])

$$z_i = 1/T \int_0^T y_i^2(t) dt \quad (2.5)$$

Mit  $i$  wird der Tonkanal bezeichnet, mit  $y_i(t)$  das  $K$ -gefilterte Zeitsignal des Tonkanals  $i$ , mit  $z_i$  die zugehörige mittlere Leistung und mit  $T$  die Mittelungszeit. Die errechneten Leistungswerte aller Kanäle werden mit je einem kanalspezifischen Gewichtungsfaktor  $G_i$  multipliziert und aufsummiert, wobei ein eventuell vorhandener Basskanal („Low Frequency Effect“) nicht berücksichtigt wird. Die Gewichtungsfaktoren sind in Tab. 2.2 ersichtlich. Sie stützen sich auf die Annahme der ITU, dass rückwärtig einfallender Schall

	1. Stufe (Shelving)	2. Stufe (RLB)
$b_0$	1,53512485958697	1,0
$b_1$	-2,69169618940638	-2,0
$b_2$	1,19839281085285	1,0
$a_0$	1,0	1,0
$a_1$	-1,69065929318241	-1,99004745483398
$a_2$	0,73248077421585	0,99007225036621

Tabelle 2.1: Koeffizienten der  $K$ -Filter für das von der ITU empfohlene Lautheitsmessverfahren in IIR-Struktur 2. Ordnung bei Abtastfrequenz 48000 Hz (nach [11, S. 6 – 7])

Symbol	Kanal	Gewichtungsfaktor	Verstärkung [dB]
$G_L$	front links	1,0	0,0
$G_R$	front rechts	1,0	0,0
$G_C$	front centre	1,0	0,0
$G_{Ls}$	surround links	1,41	+1,5
$G_{Rs}$	surround rechts	1,41	+1,5

Tabelle 2.2: Gewichtungsfaktoren der Kanalsummutation bei 5.1-Surround für das von der ITU empfohlene Lautheitsmessverfahren (nach [11, S. 9])

lauter wahrgenommen werde als frontal eintreffender Schall [11, S. 15]. Die mit  $L_K$  symbolisierte Lautheit in „Loudness, K weighted, relative to nominal full scale“ (LKFS) entspricht dem energieäquivalenten Pegel, abzüglich einer Korrekturkonstanten:

$$L_K = -0,691 + 10 \log_{10} \left( \sum_i G_i z_i \right) \quad (2.6)$$

Soll hingegen die Gate-Funktion in die Lautheitsberechnung miteinbezogen werden, wird die mittlere Leistung jedes Kanals  $i$  zunächst in einzelnen Blöcken  $j$  der Länge  $T_g$  berechnet. Die Blöcke überlappen sich zu einem mit  $o$  (= „overlap“) bezeichneten Anteil, mit anderen Worten, die Anfangszeitpunkte der Blöcke liegen einen Anteil  $s = 1 - o$  ( $s$  = „step“) der Blocklänge auseinander:

$$z_{ij} = 1/T_g \int_{T_g js}^{T_g(j s+1)} y_i^2(t) dt \quad (2.7)$$

mit

$$j \in \{0, 1, 2, \dots, \lfloor \frac{T - T_g}{s T_g} \rfloor\}$$

Die Blocklänge wurde auf  $T_g = 400$  ms und der Überlappungsfaktor auf  $o = 0,75$  festgelegt, die Blöcke starten also jeweils in einem Zeitintervall von  $s T_g = 100$  ms.

Nach dem gleichen Prinzip wie in Gl. 2.6 wird nun die Teillautheit  $l_j$  jedes Blocks berechnet:

$$l_j = -0,691 + 10 \log_{10} \left( \sum_i G_i z_{ij} \right) \quad (2.8)$$

In der ersten Gating-Stufe werden nun all jene Blöcke identifiziert, deren Teillautheit kleiner gleich einem so g. absoluten Schwellwert von  $\Gamma_a = -70$  LKFS ist. Diese Blöcke werden als „Stille“ betrachtet und von der Messung ausgeschlossen, alle übrigen bilden die Menge  $J_g$  der für die Messung verwendeten Blöcke gemäß

$$J_g = \{j | l_j > \Gamma_a\}. \quad (2.9)$$

Durch arithmetische Mittelung der Leistungen aller in die Messung einbezogenen Blöcke sowie gewichtete Kanalsummation, Logarithmierung und die Subtraktion der Korrekturkonstanten gemäß Gl. 2.6 erhält man eine „vorläufige“ Gesamtlautheit:

$$L_{KG} = -0,691 + 10 \log_{10} \left( \sum_i G_i \left( \frac{1}{|J_g|} \sum_{J_g} z_{ij} \right) \right) \quad (2.10)$$

Der so g. relative Schwellwert  $\Gamma_r$  der zweiten Gating-Stufe wird nun auf 10 dB unterhalb der eben berechneten Gesamtlautheit gesetzt:

$$\Gamma_r = L_{KG} - 10 \text{ dB} \quad (2.11)$$

Alle Blöcke, deren Teillautheit kleiner gleich dem relativen Schwellwert ist, werden nun ebenfalls von der Messung ausgeschlossen. Damit wird die Menge der in die Messung einzubeziehenden Blöcke weiter verringert auf

$$J_g = \{j | l_j > \Gamma_r\}. \quad (2.12)$$

Die finale Gesamtlautheit, bezeichnet mit „Loudness, K weighted, gated“ ( $L_{KG}$ ) in LKFS wird nun auf Grundlage der neuen Menge  $J_g$  in genau gleicher Weise wie nach der ersten Gating-Stufe (Gl. 2.10) berechnet.

Das Verfahren besitzt die Eigenschaft, dass eine Änderung des Signalspitzenpegels in dB eine ebenso große Änderung des gemessenen Lautheitswerts zur Folge hat. Die Korrekturkonstante wurde so gewählt, dass der gemessene Lautheitswert eines zweikanaligen Sinussignals von 1000 Hz mit gleichem Pegel und gleicher Phasenlage auf beiden Kanälen gerade dessen Sample Peak Level in dBFS entspricht.

## 2.2.6 Die Empfehlung EBU R 128

Basierend auf der im vorigen Unterkapitel vorgestellten Empfehlung zur Lautheitsmessung durch die ITU veröffentlichte die EBU im Jahr 2011 die Empfehlung EBU R 128 [17]. Sie befasst sich mit der Messung eines Momentan- und eines Kurzzeitlautheitspegels, eines daraus abgeleiteten Lautheitsbereichs („Loudness Range“), der Einhaltung eines bestimmten True Peak Levels und der Einhaltung eines bestimmten durchschnittlichen Lautheitspegels für jedes Programm. Ein „Programm“ wird dabei folgendermaßen definiert:

„An individual, self-contained audio-visual or audio-only item to be presented in Radio, Television or other electronic media. An advertisement (commercial), trailer, promotional item ('promo'), interstitial or similar item shall be considered to be a programme in this context“ [17, S. 5]

Für die Lautheitsmessung wird stets das in Kap. 2.2.5 vorgestellte Verfahren gemäß ITU-R BS.1770-2 mit unterschiedlichen Integrationszeiten sowie je nach Anwendungsfall

mit oder ohne Gating empfohlen. Die Eigenschaften eines mit der EBU-Empfehlung konformen Aussteuerungsmessers werden in [25] genauer spezifiziert. Abweichend von ITU-R BS.1770-2 [11] wird die Einheit für relative Lautheitswerte mit „Loudness Units“ (LU) und die Einheit für absolute Lautheitswerte mit „Loudness Units, referenced to digital Full Scale“ (LUFS) bezeichnet. 1 LU entspricht dabei einem Pegelunterschied von 1 dB. 0 LUFS wird als genau der Wert definiert, der für ein Sinussignal mit einer Frequenz von 1000 Hz, gleichphasig anliegend auf zwei Frontkanälen mit einem Sample Peak Level von 0 dBFS, angezeigt wird.

#### 2.2.6.1 Normalisierung nach integriertem Lautheitspegel (Programmlautheit)

Es wird empfohlen, dass für jedes Programm gemäß der Definition auf S. 18 ein so g. integrierter Lautheitspegel über die Dauer des ganzen Programms gemessen wird [17, S. 4]. Bei der Messung soll die Gate-Funktion angewandt werden [25, S. 6 – 7]. Der so gemessene Lautheitspegel soll für jedes Programm -23,0 LUFS betragen und maximal  $\pm 0,1$  LU von diesem Zielwert abweichen.

Auf diese Weise sollen Lautheitsunterschiede zwischen einzelnen Programmen wie auch zwischen Rundfunksendern vermieden werden. Der Zielwert von -23,0 LUFS ergab sich aus Messungen der EBU, wonach die zur Zeit durchschnittliche Programmlautheit bei -20,0 LUFS liege. Dieser Wert erlaube jedoch keine ausreichende Übersteuerungsreserve für Signale mit größerer Dynamik [24, S. 11 – 12].

#### 2.2.6.2 Momentan- und Kurzzeitlautheitspegel

In Ergänzung zur Messung des integrierten Lautheitspegels (Programmlautheit) wird die Anzeige der beim Gating ermittelten Lautheitswerte der einzelnen 400 ms-Blöcke (bezeichnet als „Momentary Loudness“) und die Messung eines Kurzzeit-Lautheitspegels („Short-Term Loudness“) mit Mittelungszeit  $T = 3$  s empfohlen. Für die Messung der Kurzzeit-Lautheit soll kein Gating angewandt werden [25, S. 6].

Die Anzeige von Momentan- und Kurzzeitlautheitspegel dient nur der Beobachtung des Lautheitsverlaufs innerhalb eines Programms und zur Einhaltung eventuell seitens eines Senders selbst gesetzter Zielwerte. Seitens der EBU werden für beide Pegel keine Werte vorgegeben.

#### 2.2.6.3 Lautheitsbereich („Loudness Range“)

Zur besseren Beobachtung der Lautheitsunterschiede innerhalb eines Programms wird empfohlen, zusätzlich zur Programmlautheit auch die Verteilung der Kurzzeit-Lautheitswerte, bezeichnet als „Loudness Range“, des Programms zu messen [17, S. 4 – 5]. Die Messung soll nach dem in [26] spezifizierten Algorithmus erfolgen. Darin lautet die Definition der Loudness Range wie folgt:

„Loudness Range (abbreviated 'LRA') quantifies the variation in a time-varying loudness measurement.

...

Loudness Range measures the variation of loudness on a macroscopic time-scale, in units of LU (Loudness Units).

...

Loudness Range should not be confused with other measures of dynamic range or crest factor, etc.“ [26, S. 5]

Der Algorithmus basiert auf der Messung der Kurzzeit-Lautheitswerte gemäß Kap. 2.2.6.2 eines Programms in Blöcken zu 3 s, die sich zu mindestens 2 s überlappen sollen. Ähnlich dem für die Programmlautheit empfohlenen Verfahren wird anschließend ein zweistufiges Gating angewandt, damit sich Passagen mit Stille, Hintergrundgeräusch oder einer Blende nicht auf die Messung auswirken. Die Loudness Range ist schließlich die Differenz des 95 %-Quantils und des 10 %-Quantils der Lautheitswerteverteilung aller Blöcke, die die Gates passiert haben. Auf diese Weise soll vermieden werden, dass selten auftretende Werte besonders geringer oder hoher Lautheit zu einer höheren Loudness Range-Anzeige führen.

Auch für die Loudness Range werden keine Zielwerte vorgegeben. Es wird ausdrücklich darauf hingewiesen, dass – abhängig von Programm-Genre und Übertragungskanal – durchaus höhere Lautheitsunterschiede innerhalb eines Programms im Sinne einer ästhetischen Klanggestaltung zulässig oder sogar erwünscht seien.

„Again, this does NOT mean that within a programme the loudness level has to be constant, on the contrary!

...

Loudness variation is an artistic tool, and the concept of loudness normalisation according to R 128 actually encourages more dynamic mixing! It is the average, integrated loudness of the whole programme that is normalised.“ [24, S. 16]

#### 2.2.6.4 Höchster echter Spitzenpegel („Maximum True Peak Level“)

Es wird empfohlen, neben den Lautheitswerten auch den „Maximum True Peak Level“ (siehe Kap. 2.2.2, S. 11) zu messen [17, S. 4]. „Maximum True Peak Level“ wird dabei definiert als „The maximum value of the audio signal waveform of a programme in the continuous time domain“ [17, S. 5].

Zur Messung soll der in [11, S. 19 – 20] spezifizierte Algorithmus verwendet werden. Dieser besteht im Wesentlichen aus einer Überabtastung (Oversampling) des digitalen Signals auf mindestens 192000 Hz, um so den Verlauf des zeitkontinuierlichen Signals möglichst gut anzunähern. Der erlaubte Maximum True Peak Level soll -1 dBTP (= dB, measured with True Peak meter) betragen.

## 2.3 Perzeptive Audiodatenreduktion

Dieses Kapitel gibt eine Einführung in den zweiten Themenschwerpunkt dieser Arbeit, die wahrnehmungsbezogene Audiodatenreduktion. Hat diese Technologie im Konsumentenbereich schon seit über zehn Jahren wichtige Bedeutung durch die Speicherung und Downloads von Musik und Videos besessen, spielt sie auch im Rundfunk eine entscheidende Rolle: Schon in der Produktion und Archivierung kommen teilweise Datenreduktionsverfahren zum Einsatz, vor allem jedoch als Quellcodierung für jegliche digitale Programmverbreitung, sei es terrestrisch, via Kabel und Satellit oder über das Internet.

Bei den folgenden Betrachtungen soll die Irrelevanzreduktion im Mittelpunkt stehen, also jener Teil der Technologie, der zu einem mehr oder weniger hohen Datenverlust und damit zu einer irreversiblen Veränderung des Audiosignals führt. Verfahren der statistischen Redundanzreduktion, die in verlustbehafteten wie verlustlosen Datenreduktionsverfahren zum Einsatz kommen, werden in dieser Arbeit nicht behandelt, da davon ausgegangen wird, dass sie keinen Einfluss auf die Tonqualität haben und sie sich deshalb unproblematisch gegenüber einer lautheitsgerechten Aussteuerung, wie im vorigen Kapitel beschrieben, verhalten.

### 2.3.1 Definition

Audiodatenreduktion, auch bekannt als Audiokomprimierung (Audio Compression) oder Audiocodierung (Audio Coding), kann wie folgt definiert werden:

„Audio coding or audio compression algorithms are used to obtain compact digital representations of high-fidelity (wideband) audio signals for the purpose of efficient transmission or storage. The central objective in audio coding is to represent the signal with a minimum number of bits while achieving transparent signal reproduction, i.e., generating output audio that cannot be distinguished from the original input, even by a sensitive listener (“golden ears“).“ [27, S. 451]

Inwieweit die im Zitat genannte Forderung der Transparenz erfüllt werden kann, hängt offensichtlich davon ab, wie hoch der Anteil der einzusparenden Datenmenge zu sein hat, also letztlich von der zur Verfügung stehenden Speicher- bzw. Kanalkapazität [12, S. 866 – 868]. Es kommen weitere Faktoren zum Tragen wie der verwendete Algorithmus und dessen konkrete Implementierung, die Abhörbedingungen und die Struktur des Tonsignals. Um letzterem Aspekt Rechnung zu tragen, bieten viele Codierverfahren die Möglichkeit, anstatt einer konstanten Datenrate eine gewünschte Qualitätsstufe vorzugeben, zu deren Erreichen die Datenrate zeitvariant dem Signalverlauf angepasst wird (oft bezeichnet als „variable Bitrate“- = VBR-Modus).

Die Verfahren lassen sich in zwei Hauptkategorien einteilen: verlustlose und verlustbehaftete [12, S. 850 – 852], [27, S. 452]. Bei verlustlosen Verfahren kommen im Wesentlichen nur Techniken zur Redundanzreduktion im informationstheoretischen Sinn wie der Huffman-Code, der Lempel-Ziv-Welch-Code, die lineare Prädiktion und die Differential

Pulse Code Modulation zum Einsatz. Außerdem werden in mehreren Tonkanälen evtl. gleiche Signalanteile (ähnlich der aus der Mitte-Seite-Stereophonie bekannten Matrizierung) nur einmal gespeichert, was hier als „Joint Stereo“ bezeichnet wird. Verlustlose Verfahren bieten für Tonsignale eine Komprimierung auf etwa zwei bis ein Drittel der Datenmenge und eine originalgetreue Rekonstruktion bei der Decodierung. Verlustbehaftete Verfahren hingegen nutzen – neben den genannten Redundanzreduktionstechniken – psychoakustische Effekte aus, um Anteile des Signals einzusparen, die durch das menschliche Gehör in der Regel nicht wahrgenommen werden. Man spricht hier von Irrelevanzreduktion. Die Grenze, bis zu der Signalanteile für das Gehör irrelevant sind, ist jedoch individuell verschieden, weshalb die Qualität eines Audiocodierverfahrens in der Regel mittels psychoakustischer Versuche überprüft wird [12, S. 866 – 867].

### 2.3.2 Prinzipielle Funktionsweise verlustbehafteter Verfahren

Perzeptive Audiodatenreduktionsverfahren basieren auf der Tatsache, dass das menschliche Gehör nicht alle Anteile eines Tonsignals wahrnimmt [27, S. 454 – 461]. So lässt sich beispielsweise beobachten, dass nicht verstärkte Sprache unverständlich oder sogar unhörbar wird, wenn gleichzeitig Musik oder Geräusche hoher Intensität vorhanden sind. Man spricht in diesem Zusammenhang von Verdeckung oder Maskierung [1, S. 19 – 20]. Dieser Effekt ist auf die Funktionsweise des Innenohres zurückzuführen und lässt sich (vereinfacht) folgendermaßen beschreiben: Das Innenohr führt eine Frequenzanalyse des eintreffenden Schalls durch. Das Spektrum wird dabei in Frequenzbänder unterteilt, die als Frequenzgruppen (Critical Bands) bezeichnet werden und eine von der Tonhöhe abhängige Frequenzgruppenbreite (Critical Bandwidth) besitzen. Unterhalb von 500 Hz beträgt jede Frequenzgruppenbreite konstant 100 Hz, darüber steigt sie mit der Frequenz an; die obere und untere Grenzfrequenz einer Gruppe besitzen hier ungefähr den Abstand einer reinen großen Terz (Frequenzverhältnis 5:4) [3, S. 18 – 19]. Zur Angabe der Frequenzgruppen auf einer linear ansteigenden Skala wurde der Begriff der Tonheit  $z$  mit der Einheit bark eingeführt. 1 bark entspricht dabei einer Frequenzgruppenbreite. Erfolgt nun ein Schallreiz innerhalb einer bestimmten Frequenzgruppe, werden Schallereignisse wesentlich geringerer Intensität innerhalb des gleichen Frequenzbereichs verdeckt. Der Schalldruckpegel, unterhalb dessen die Verdeckung stattfindet, wird als Mithörschwelle bezeichnet. Die Mithörschwelle steigt mit dem Schalldruckpegel des verdeckenden Signals. Sie ist in der Frequenzgruppe des verdeckenden Schallreizes am höchsten und fällt zu benachbarten Frequenzgruppen hin ab. Dieser Abfall ist zu den niedrigeren Frequenzgruppen steiler als zu den höheren. Außerdem hängt die Mithörschwelle von der Art des verdeckenden Signals ab: Im Falle eines reinen Tons liegt die Schwelle höher als bei breitbandigeren Signalen wie Geräuschen.

Neben dieser so g. Simultanverdeckung gibt es außerdem eine zeitliche Verdeckung [1, S. 20], [27, S. 460 – 461]: Folgt auf einen Schallreiz ein weiteres Schallereignis geringerer Intensität und Dauer, wird das zweite Schallereignis nicht wahrgenommen; man spricht von Nachverdeckung. Das Zeitintervall zwischen beiden Schallereignissen, in welchem sich dieser Effekt beobachten lässt, bewegt sich zwischen 10 und 300 ms und hängt von der Dauer des ersten Schallreizes ab. Ein ähnlicher Effekt ist auch für kurze

Schallereignisse vor einem längeren Schallreiz als Vorverdeckung bekannt. Die Vorverdeckung geschieht allerdings nur in einem Abstand von bis zu 2 ms zwischen beiden Schallereignissen.

Bei der perceptiven Datenreduktion wird versucht, das Verdeckungsverhalten des menschlichen Innenohres in Form eines so. g. psychoakustischen Modells nachzubilden [27, S. 452, 461 – 467]. Dies geschieht meistens durch eine diskrete Fourier-Transformation in Blöcken zu mindestens 512 Abtastwerten, was z. B. bei einer Abtastrate von 48000 Hz eine Frequenzauflösung von  $48000 \text{ Hz} / 512 \text{ Samples} = 93,75 \text{ Hz/Sample}$  ergibt und damit ausreicht, um die Frequenzgruppenbreiten des Innenohres anzunähern. Ausgehend von den Beträgen der einzelnen Frequenzen errechnet das psychoakustische Modell die Mithörschwelle für jeden Frequenzbereich, wobei abgeschätzt wird, ob es sich bei den einzelnen verdeckenden Signalen um Töne oder Geräusche handelt. Abhängig von der Komplexität des Algorithmus wird auch die Nachverdeckung berücksichtigt. Anschließend werden alle Daten mit verdeckten Signalanteilen eliminiert [27, S. 467 – 469]: Frequenzbänder, die keine wahrnehmbaren Signalanteile enthalten, werden ganz gelöscht. Alle übrigen Frequenzbänder werden mit einer geringeren Auflösung neu quantisiert. Für diese so g. Requantisierung (auch als Perceptual noise Shaping bezeichnet) wird im Idealfall die größte Auflösung gewählt, bei welcher der Quantisierungsfehler – d. h. das entstehende Quantisierungsrauschen – noch gerade unterhalb der Mithörschwelle liegt. Diese Requantisierung erfolgt ebenfalls mittels einer Frequenzerlegung des Signals, für die es verschiedene Ansätze gibt: Sie ist entweder mit der des psychoakustischen Modells identisch oder wird in einem separaten Modul durchgeführt. In letzterem Fall kann eine Transformation (Fourier- oder Cosinus-Transformation), eine Parallelschaltung mehrerer Bandpassfilter gleicher oder verschiedener Bandbreite (Filterbank) oder auch eine Kombination mehrerer Techniken zum Einsatz kommen. Jeder Ansatz weist je nach Verwendungszweck Vor- und Nachteile auf. Wesentliche Kriterien sind möglichst geringe Komplexität, d. h. geringer Rechenaufwand bzw. geringe Zeitverzögerung des Algorithmus, eine gute Annäherung der Frequenzgruppenbreiten des Innenohres und eine feine zeitliche Auflösung. Gerade die beiden letztgenannten Forderungen stellen ein Problem dar, weil sich Zeit- und Frequenzauflösung gegensätzlich zueinander verhalten und die nötigen Grade der Auflösung außerdem stark vom zu codierenden Signal abhängen [12, S. 860], [27, S. 468 – 469]. So ist bei sehr tonhaltigen Signalen wie z. B. einer Flöte eine feine Frequenzauflösung vorteilhaft, während zur Übertragung kurzer Impulse wie etwa bei einer Snare Drum oder einer Kastagnette ein hohes zeitliches Auflösungsvermögen erforderlich ist. Deshalb besitzen anspruchsvollere Codieralgorithmen die Funktion, abhängig von den aktuellen Auswertungsergebnissen des psychoakustischen Modells zwischen Analysefenstern unterschiedlicher Länge umzuschalten (so g. Window Switching). Die requantisierten Spektren werden schließlich mit Redundanzreduktionsverfahren (z. B. Huffman, Lempel-Ziv-Welch oder Differential Pulse Code Modulation) codiert [27, S. 452].

### 2.3.3 Qualitätsverlust – Artefakte

Die Datenmenge, die ein durchschnittliches Gehör wahrnehmen kann, wurde von Johnston als „Perceptual Entropy“ bezeichnet [27, S. 461]. Demnach beträgt die maximale Komprimierbarkeit für eine transparente Codierung eines Tonkanals typischer CD-Produktionen 2,1 bit/Sample, das entspricht bei einer Abtastrate von 48000 Hz und einem monophonen Signal einer Datenrate von

$$48000 \text{ Hz} \cdot 2,1 \text{ bit/Sample} = 100800 \text{ bit/s} \approx 98,44 \text{ Kbit/s.}$$

Bei einer geringeren Datenrate pro Kanal muss folglich von einem hörbaren Qualitätsverlust ausgegangen werden, da der Pegel des durch die Requantisierung entstandenen Quantisierungsrauschens dann oberhalb der Mithörschwelle liegt. Die hörbaren Störgeräusche werden als Artefakte bezeichnet.

Manche Datenreduktionsverfahren nehmen zusätzlich eine konstante Tiefpassfilterung vor, sparen also durchgängig hochfrequente Spektralanteile ein, da diese bei typischen Musik- und Sprachsignalen weniger Energie besitzen und das Gehör zu Frequenzen oberhalb von 10000 Hz abnehmende Empfindlichkeit zeigt [12, S. 868 – 869]. Je nach Genauigkeit des Gehörs macht sich das Fehlen dieser Höhen als ein allgemeiner Verlust an Brillanz im Vergleich zum Original bemerkbar, was aber meistens nicht als sehr störend empfunden wird [28, S. 1].

Darüber hinaus können durch die Signalmanipulation weitere Artefakte entstehen. Kritisch ist z. B. die nur zeitweilige Löschung von Spektralanteilen. Diese auch als Spectral Valleys bezeichneten signalabhängigen Einbrüche führen zu einer schnellen laufenden Änderung der Klangfarbe, was als „Klirren“ oder „Zwitschern“ wahrgenommen werden kann; Liu et al. [29, S. 681 – 682] sprechen von „Birdie artifacts“.

Ein weiteres Problem stellen transiente Vorgänge dar, also impulsartige Signale wie z. B. bei Aufnahmen von Schlaginstrumenten oder Sprache: Unterschreitet die Dauer des Einschwingvorgangs die Länge des für die Signalanalyse verwendeten Zeitfensters und ist der Signalpegel in den betreffenden Frequenzbändern vorher deutlich niedriger als während des impulsartigen Signals, so setzt mit Beginn des Fensters ein Quantisierungsrauschen ein, dessen Pegel vor dem Impuls oberhalb der Mithörschwelle liegt [27, S. 471 – 473]. Da die Vorverdeckung des Gehörs nur maximal 2 ms umfasst (siehe Kap. 2.3.2, S. 23), wird dieses Quantisierungsrauschen von einem geschulten Ohr als Zischen oder als eine Aufweichung des Impulses wahrgenommen; man spricht von Pre-Echoes [27, S. 460 – 461]. Auf die gleiche Weise kann auch am Ende eines Impulses ein Quantisierungsrauschen entstehen, das länger anhält als der Impuls selbst, dies ist jedoch für das Gehör aufgrund der Nachverdeckung von mindestens 10 ms in der Regel nicht wahrzunehmen.

### 2.3.4 Besonderheiten gängiger Datenreduktionsverfahren

Bei den Versionen 1 und 2 der „International Standards Organization / Moving Picture Coding Experts Group“ (ISO/MPEG 1 / 2) handelt es sich um Verfahren zur datenreduzierten Codierung von Video- und Audiosignalen [12, S. 701, 703]. Die Codierung der Audiodaten kann in den Standards MPEG 1 und 2 je nach Einsatzgebiet und Anforderungen mit einem von drei perzeptiven Audiodatenreduktionsverfahren durchgeführt werden. Die Verfahren werden als Schichten (Layers) bezeichnet, da sie aufeinander aufbauen und der Algorithmus der jeweils nächst höheren Schicht eine höhere Komplexität besitzt [27, S. 491]. Allen Schichten ist gemeinsam, dass die Signalanalyse durch das psychoakustische Modell und die Signalcodierung blockweise in zwei parallelen Zweigen stattfinden. Während die Art der Codierung vorgegeben ist, handelt es sich bei den beschriebenen psychoakustischen Modellen nur um Empfehlungen, so dass sie im Zuge fortgeschrittener Erkenntnisse beliebig erweitert und verbessert werden können [27, S. 462]. Die encodierten Audiodaten werden in einen aus Blöcken (Frames) bestehenden Datenstrom zusammengefasst.

#### 2.3.4.1 ISO-MPEG1/2 Layer I und II (MP1 und MP2)

Die Schichten I und II basieren auf dem vom Institut für Rundfunktechnik, München (IRT) und der Firma Phillips entwickelten Verfahren „Masking Pattern Adapted Universal Subband Integrated Coding and Multiplexing“ (MUSICAM) [27, S. 479 – 480]. Es zeichnet sich durch eine niedrige Komplexität aus, so dass es sich schon in den 1990er Jahren trotz der damals noch geringeren Rechenleistung für Echtzeitanwendungen eignete.

Beide Schichten verwenden das in MPEG 1 standardisierte „psychoakustische Modell 1“ [27, S. 461 – 467]. Es analysiert das Audiosignal in einer Fast Fourier-Transformation (FFT), deren Blocklänge in Schicht I 512 Abtastwerte, in Schicht II 1024 Abtastwerte beträgt [27, S. 491]. Die Abtastwerte des Audiosignals werden vor der Transformation derart normalisiert, dass der höchste Sample-Pegel 0 dBFS entspricht. Zur Nachempfindung einer realistischen Abhörsituation werden nach der Transformation die in dBFS vorliegenden Spektralbeträge auf eine entsprechende Darstellung des Schalldruckpegels in dB SPL abgebildet, wobei der niedrigsten Quantisierungsstufe die niedrigste Ruheschwelle, d. h. jene bei 4000 Hz, entsprechend ein Schalldruckpegel von etwa -6 dB SPL, zugeordnet wird. Um die geringere Verdeckung durch reine Töne im Vergleich zu jener durch breitbandige Signale (siehe Kap. 2.3.2, S. 22) zu berücksichtigen, wird das Spektrum auf tonale Anteile untersucht. Dabei gilt jedes relative Spektralmaximum mit einer Pegeldifferenz von mindestens 7 dB zu den benachbarten Frequenzen als tonal. Die Leistung eines solchen Spektralmaximums wird mit den Leistungen der beiden angrenzenden Spektralkomponenten zur Gesamtleistung des Tons summiert. Alle danach verbliebenen Spektralkomponenten werden als nichttonal angesehen und durch Leistungssummation zu je einem geräuschartigen Signal pro Frequenzgruppe zusammengefasst.

Anschließend erfolgt eine Dezimation dieses Spektrums auf diejenigen Komponenten, die als wahrnehmbar angesehen werden. Dabei werden alle Frequenzen gelöscht, deren

Betrag entweder in der Schalldruckpegeldarstellung unterhalb der jeweiligen Ruheschwelle liegt oder deren Verdeckungseigenschaft schwächer als die der benachbarten Frequenzen (innerhalb eines Abstands von 0,5 bark) ist [27, S. 465 – 467]. Auf Grundlage dieses dezimierten Spektrums wird danach der Verdeckungseinfluss der verbliebenen Spektralkomponenten aufeinander berechnet: Es wird davon ausgegangen, dass jede der Spektralkomponenten sowohl in ihrem eigenen Frequenzband als auch in den benachbarten Frequenzbändern zu einer so g. individuellen Mithörschwelle führt. Die individuellen Mithörschwellen ergeben sich dabei durch eine stückweise lineare Funktion des Schalldruckpegels und des Frequenzabstands der verdeckenden Spektralkomponente, wobei für tonale und nichttonale verdeckende Spektralkomponenten unterschiedliche Funktionsgleichungen verwendet werden. Ferner werden die Verdeckungseinflüsse zwischen Frequenzbändern in einem Abstand von mehr als 10 bark vernachlässigt. Die Gesamtmithörschwelle jeder Spektralkomponente ergibt sich schließlich durch Leistungsaddition aller individuellen Mithörschwellen und der ebenfalls frequenzspezifischen Ruheschwelle. Die so ermittelten Gesamtmithörschwellen stellen die Richtwerte für die Requantisierung dar.

Parallel zur Signalanalyse im psychoakustischen Modell findet die eigentliche Codierung statt [27, S. 491 – 493]. Dazu wird das Signal in Blöcke zu je 384 Abtastwerten unterteilt. Die Frequenzerlegung erfolgt hier nicht durch eine Transformation, sondern durch eine Filterbank, bestehend aus 32 Kanälen konstanter Bandbreite. Ein solches Verfahren wird auch als Subband-Coding bezeichnet, die einzelnen Ausgangskanäle der Filterbank als Subbänder. Da das Ausgangssignal jedes Subbands nur noch eine auf 1/32 reduzierte Bandbreite besitzt, kann nach der Filterung die Abtastrate aller Subbänder um den Faktor 32 reduziert werden (Man spricht von kritischer Abtastung). Zur effizienteren Codierung wird jeder Block für jedes Subband derart normalisiert, dass alle höchsten Abtastwerte jedes Blocks und Subbands Maximalpegel besitzen. Der dafür notwendige Skalierungsfaktor wird mit je 6 bit codiert. In Layer II kann die Normalisierung, falls erforderlich, bis zu dreimal pro Block durchgeführt werden. Anschließend wird das Signal jedes Subbands gemäß der durch das psychoakustische Modell berechneten Mithörschwelle einerseits und entsprechend der gewünschten Datenrate andererseits requantisiert. Dabei wird in Layer II nicht nur die Simultan-, sondern teilweise auch die Nachverdeckung berücksichtigt, indem drei Blöcke gemeinsam verarbeitet werden.

Die Filterbank bietet eine recheneffiziente, jedoch nur eine konstante und sehr grobe Frequenzerlegung (z. B. bei einer Abtastrate von 48000 Hz eine Auflösung von  $\frac{24000 \text{ Hz}}{32 \text{ Subbänder}} = 750 \text{ Hz/Subband}$ ), die keinesfalls die Frequenzgruppenbreiten des menschlichen Gehörs nachbildet: Unterhalb von etwa 2800 Hz sind die Subbänder breiter als die Frequenzgruppen des Gehörs, oberhalb sind sie niedriger.

#### 2.3.4.2 ISO-MPEG1/2 Layer 3 (MP3)

Die Schicht III basiert auf dem am Fraunhofer-Institut Erlangen, den AT&T Labs und dem CNET entwickelten Verfahren „Adaptive Spectral Entropy Coding of High Quality Music Signals“ (ASPEC) [27, S. 475 – 477]. Es zeichnet sich durch ein im Vergleich zu MUSICAM aufwendigeres psychoakustisches Modell, eine besser an das Gehör angepasste

Frequenzerlegung und einen anspruchsvolleren Requantisierungsalgorithmus aus.

Das hier ursprünglich verwendete psychoakustische Modell basiert auf der Bestimmung der Perceptual Entropy nach Johnston und wurde in MPEG 1 und 2 als Modell 2 standardisiert [27, S. 461 – 462]. Es führt eine FFT mit 2048 Abtastwerten und ähnlich wie das psychoakustische Modell 1 eine Abbildung der Spektren auf eine normierte Schalldruckpegeldarstellung durch. Die individuellen Mithörschwellen (siehe Kap. 2.3.2, S. 22 und Kap. 2.3.4.1, S. 26) für jedes Frequenzband werden hier durch eine Faltung der Frequenzgruppenpegel mit einer Verdeckungsfunktion (Spreading Function) berechnet. Tonale Signalanteile werden durch die zeitliche Analyse des Verlaufs von Betrag und Phase jedes Spektralwerts bestimmt [27, S. 476]: Weist ein Wert über mehrere Blöcke ein quasistationäres Verhalten auf (bezeichnet als „High Predictability“), wird von einem Ton ausgegangen.

Für die Frequenzerlegung zur Signalcodierung wird eine so g. Hybrid-Filterbank verwendet [27, S. 492]. Diese besteht aus einer Serienschaltung der aus Schicht I und II bekannten Filterbank mit 32 Subbändern konstanter Bandbreite (siehe Kap. 2.3.4.1, S. 26) und einer modifizierten diskreten Cosinus-Transformation (MDCT). Die Filterbank mit den 32 gleich breiten Subbändern wurde übernommen, um Rückwärtskompatibilität der Layer-III-Decodierer zu den niedrigeren Schichten zu gewährleisten. Die nachgeschaltete MDCT mit 18 Abtastwerten pro Subband bietet eine feinere Frequenzauflösung und damit eine wesentlich genauere Nachbildung der Frequenzgruppenbreiten des Gehörs. Zusätzlich besteht die Möglichkeit, die Länge des Analysefensters bei impulshaften Signalen von 18 auf 6 Abtastwerte zu verkürzen (Window Switching, siehe Kap. 2.3.2, S. 23); dies entspricht bei einer Abtastrate von 48000 Hz einer Verkürzung der Zeitdauer von  $32 \cdot 18 \text{ Samples/Block}/48000 \text{ Hz} = 12 \text{ ms/Block}$  auf 4 ms.

Die Requantisierung jedes Frequenzbands erfolgt in zwei verschachtelten Schleifen [27, S. 475 – 477]: In der inneren Schleife wird die für die Quantisierung verwendete Datenmenge schrittweise reduziert und gleich im Anschluss eine redundanzreduzierende Huffman-Codierung durchgeführt. Die Requantisierung ist nichtlinear, so dass niedrige Signalpegel mit einer feineren Auflösung dargestellt werden als hohe. Ähnlich wie in den Schichten I und II wird nicht jeder Spektralwert absolut gespeichert, sondern meistens nur die Differenz zu einem Skalierungsfaktor bzw. zu den Werten aus den vorangegangenen Blöcken. Der Codierungsvorgang wird wiederholt, bis eine bestimmte vorgegebene Datenmenge nicht mehr überschritten wird. In der äußeren Schleife wird eine Decodierung des eben encodierten Spektrums durchgeführt, um zu überprüfen, ob das entstandene Quantisierungsrauschen in allen Frequenzbändern unterhalb der durch das psychoakustische Modell berechneten Mithörschwelle liegt. Dieser Algorithmus wird auch als „Analysis by Synthesis“-Verfahren bezeichnet. Für diejenigen Frequenzbänder, in denen das Quantisierungsrauschen oberhalb der Mithörschwelle liegt, wird die innere Schleife zur Requantisierung mit einer höheren erlaubten Datenmenge erneut ausgeführt. Die äußere Schleife wird wiederholt, bis in allen Frequenzbändern das Quantisierungsrauschen unterhalb der Mithörschwelle liegt. Lässt sich diese Forderung mit der vom Anwender vorgegebenen Gesamtdatenrate nicht erfüllen, wird die äußere Schleife jedoch abgebrochen.

## Kapitel 3

# Praktische Untersuchung

### 3.1 Fragestellung, Einordnung und Hypothesenbildung

Ausgangspunkt dieser Arbeit ist die Frage, ob sich die neue, in Kap. 2.2.6.1 und 2.2.6.4 beschriebene Aussteuerung gemäß der Empfehlung EBU R 128 [17] auf die Tonqualität bei den im Rundfunk verwendeten perceptiven Audiodatenreduktionsverfahren auswirkt. Wenngleich diese Frage offenbar bisher noch nicht gezielt wissenschaftlich untersucht worden ist, lassen sich aus diversen Aspekten Hypothesen herleiten. Daher handelt es sich hier nach [30, S. 1 – 12] um eine Hypothesen prüfende Untersuchung. Dabei kann die Aussteuerung als unabhängige Variable mit zwei Stufen aufgefasst werden: erstens „bisher im Rundfunk übliche Aussteuerung“ bzw. „Originalaussteuerung“, d. h. auf Quasispitzenpegel -9 dBFS (siehe Kap. 2.2.1, S. 9), und zweitens „neue Aussteuerung“ bzw. „lautheits- und True Peak-orientierte Aussteuerung“, d. h. auf Programmlautheitspegel -23 LUFS und Maximum True Peak Level -1 dBTP. Die durch die Datenreduktion hervorgerufene Tonqualitätsverschlechterung stellt die abhängige Variable der Untersuchung dar, die zunächst als kontinuierlich aufzufassen ist. Den einzigen weiteren beobachteten Parameter mit Relevanz für die Untersuchung (Kontrollvariable) bildet das jeweilige Datenreduktionsverfahren. Aus zeitlichen Gründen ist eine Durchführung mit allen im Rundfunk gängigen Verfahren nicht möglich, so dass sich die Untersuchung auf die zwei weit verbreiteten Verfahren MPEG-1/2 Layer II und Layer III beschränkt. Auf die Einführung weiterer Kontrollvariablen wird verzichtet. Vielmehr wird darauf geachtet, andere potentiell für die Untersuchung relevante Parameter konstant zu halten. So wird für alle Untersuchungen stets dasselbe Testmaterial mit gleicher Dynamik und zeitlicher Anordnung sowie (für jedes Datenreduktionsverfahren) dieselbe Software in gleicher Version und mit gleichen Parametern verwendet. Da ansonsten keine Komponenten zum Einsatz kommen, von denen eine signalabhängige, zeitvariante oder nichtlineare Beeinflussung des Signals zu erwarten ist (wie bei Dynamikprozessoren, A/D- oder D/A-Umsetzern der Fall), kann das Auftreten eventueller Störvariablen vernachlässigt werden. Es handelt sich somit um ein Experiment mit Laborcharakter.

Die untersuchte Fragestellung lässt sich von zwei Seiten her annähern: Einerseits bedeutet die Aussteuerung auf einen einheitlichen Lautheitspegel, dass die Datenreduktionsverfahren mit sehr verschiedenen Signalpegeln konfrontiert werden [15, S. 15], [16, S. 7]. Dies ist die direkte Folge aus der Tatsache, dass verschiedene Programmkomponenten, Musikstile etc. bei Aussteuerung auf einen gleichen Spitzenpegel oftmals sehr unterschiedliche Lautheiten aufweisen (siehe Kap. 2.2.2, S. 9). Somit lässt sich eine einheitliche Durchschnittslautheit sowohl durch Inhalte mit hoher Dynamik als auch durch stark komprimierte Signale erreichen, wobei jeder Programmblock umso niedriger angesteuert werden muss, je höher seine Lautheit ist. Dies bedeutet, dass der maximal mögliche Signalpegel i. A. deutlich unterschritten wird und die Verdeckungseffekte des Gehörs (siehe Kap. 2.3.2, S. 22) erst bei einer niedrigeren Mithörschwelle einsetzen, weshalb auch das tolerierbare Quantisierungsrauschen schwächer sein muss. Dies könnte zur Folge haben, dass entweder – für die Aufrechterhaltung eines gewünschten Tonqualitätsniveaus – die Requantisierung bei der perceptiven Datenreduktion mit einer höheren Datenwortlänge zu erfolgen hat oder aber dass – bei Konstanthalten der vorgegebenen Datenrate – mit hörbarem Quantisierungsrauschen, also einem höheren Qualitätsverlust, zu rechnen ist. Gegen diese Vermutung spricht jedoch zumindest im Falle der hier untersuchten Datenreduktionsverfahren, dass beide sowohl für die psychoakustische Analyse als auch für die Requantisierung eine interne Normalisierung vornehmen, so dass ein niedrigerer Signalpegel durch die Änderung eines Skalierungsfaktors ausgedrückt wird und nur die Differenzen der Spektralbeträge zu diesem Skalierungsfaktor quantisiert werden müssen (siehe Kap. 2.3.4.1, S. 25 und S. 26 sowie Kap. 2.3.4.2, S. 27). Die Requantisierung wäre damit nicht abhängig vom Gesamtpegel, sondern nur von den Verhältnissen der einzelnen Spektralbeträge zueinander.

Andererseits wird in [14, S. 1 – 2, 4 – 5] und [16, S. 2] berichtet, dass die aktuell übliche spitzenpegelorientierte Aussteuerung Verzerrungen bei der Datenreduktion begünstige. Demnach führe die oftmals zu beobachtende Praxis der Lautheitsmaximierung und das damit einhergehende häufige Erreichen hoher Signalpegel zu einer gelegentlichen Überschreitung der Aussteuerungsgrenze durch den True Peak Level (siehe Kap. 2.2.2, S. 11), was bei der perceptiven Datenreduktion zu deutlich stärkeren Artefakten führe. Aus diesem Grund könnte die neue lautheits- und True Peak-orientierte Aussteuerung auch positive Auswirkungen auf die Tonqualität bei der Datenreduktion haben.

Zusammenfassend kann eine Qualitätsverbesserung in Folge der neuen, lautheitsorientierten Aussteuerung gemäß EBU R 128 vermutet werden, nicht zu erwarten ist hingegen eine Qualitätsverschlechterung. Da beide Aspekte – sowohl der Ausschluss einer Qualitätsverschlechterung als auch die Bestätigung einer Qualitätsverbesserung – von Interesse sein können, erscheint es sinnvoll, sie als zwei separate Alternativhypothesen zu formulieren, die jeweils für beide untersuchten Datenreduktionsverfahren aufzustellen sind. Die untersuchten vier Alternativhypothesen lauten damit:

$H_{1,n,II}$ : „Die lautheitsorientierte Aussteuerung wirkt sich im Vergleich zur spitzenpegelorientierten Aussteuerung bei der Datenreduktion mit MPEG-1/2 Layer II *nicht negativ* auf die Tonqualität aus.“

$H_{1,p,II}$ : „Die lautheitsorientierte Aussteuerung wirkt sich im Vergleich zur spitzenpegelorientierten Aussteuerung bei der Datenreduktion mit MPEG-1/2 Layer II *positiv* auf die Tonqualität aus.“

$H_{1,n,III}$ : „Die lautheitsorientierte Aussteuerung wirkt sich im Vergleich zur spitzenpegelorientierten Aussteuerung bei der Datenreduktion mit MPEG-1/2 Layer III *nicht negativ* auf die Tonqualität aus.“

$H_{1,p,III}$ : „Die lautheitsorientierte Aussteuerung wirkt sich im Vergleich zur spitzenpegelorientierten Aussteuerung bei der Datenreduktion mit MPEG-1/2 Layer III *positiv* auf die Tonqualität aus.“

Wie die abhängige Variable „Tonqualitätsverschlechterung“ für diese Untersuchung operationalisiert und wie eine signifikant negative bzw. positive Auswirkung auf diese Variable definiert wird, wird im Rahmen der Auswertung (siehe Kap. 3.4) erläutert.

## 3.2 Versuchsanordnung

In diesem Kapitel werden die Hard- und Software sowie die Testsequenzen beschrieben, auf die bei der praktischen Untersuchung zurückgegriffen wird. Darüber hinaus werden die im Rahmen dieser Arbeit geschriebenen Funktionen zur Audiosignalanalyse näher erläutert.

### 3.2.1 Hard- und Software

Für diese Untersuchung werden im Einzelnen folgende Komponenten verwendet:

- Personal Computer mit Prozessor Intel Dual Core T2300, Taktfrequenz 1,66 GHz und Arbeitsspeicher 2,49 GB
- Microsoft Windows XP Professional Version 2002, Service Pack 3
- GNU Octave Version 3.2.4, kompiliert mit MinGW (GCC) 4.4.0, Bezugsquelle [31]
- Octave-Zusatz-Packages, Bezugsquelle [31]
  - AUDIO Version 1.1.4
  - SIGNAL Version 1.0.11
- GPL Ghostscript Version 9.01 (Artifex Software Inc.)
- Magix Samplitude Pro X Version 12.4.0.242
- FFmpeg Version N-54835-g835eee8, kompiliert mit GCC 4.7.3 / MinGW-w64 und Programmbibliothek LibTwoLAME Version 0.3.13, Bezugsquelle [32]
- LAME Version 3.99.5, kompiliert mit Intel Compiler 12.1, Bezugsquelle [33]
- Audio File Programs and Routines package (AFsp) Version 9r0, McGill University, kompiliert mit Microsoft Visual C, Bezugsquelle [34]

### 3.2.1.1 Octave

Bei der Software Octave handelt es sich um eine vom GNU-Projekt<sup>1</sup> entwickelte Umgebung für die Programmierung numerischer Berechnungen [35]. Sie ähnelt sehr der Software MATLAB der Firma Mathworks, ist aber frei erhältlich, in C++ geschrieben und bisher nur als Kommandozeilen-Anwendung ohne graphische Oberfläche verfügbar. Bis auf wenige Ausnahmen sind Befehlssatz und Syntax in MATLAB und Octave identisch. Alle folgenden Ausführungen bezüglich Octave gelten ebenso für MATLAB. Außerdem wurde bei dem in dieser Arbeit aufgeführten Quellcode auf dessen höchstmögliche Kompatibilität zu MATLAB geachtet.

Octave dient hier zur Pegelberechnung und -beeinflussung aller Testbeispiele sowie zur graphischen und statistischen Analyse der encodierten Signale.

### 3.2.1.2 Datenreduzierte Codierung

MPEG-1/2 Layer II (MP2) stellt nach wie vor ein im Rundfunk wichtiges Datenreduktionsverfahren dar. Aufgrund seiner geringen Komplexität und Verzögerungszeit bei der Encodierung wird es als Quellcodierverfahren für die erste Version des terrestrischen digitalen Hörfunks „Digital Audio Broadcasting“ (DAB) [37, S. 170] und des digitalen Fernsehens „Digital Video Broadcasting“ (DVB), insbes. auch hier bei terrestrischer Ausstrahlung, eingesetzt [38, S. 103 – 105]. Darüber hinaus speichern teilweise auch die Rundfunksender selbst Musik und Wortbeiträge datenreduziert und setzen hierfür MP2 ein, da sich Dateien in diesem Format aufgrund der geringeren Blocklänge von nur 384 Abtastwerten (siehe Kap. 2.3.4.1, S. 26) genauer schneiden lassen als MP3-codierte Dateien [12, S. 701, 870 – 871].

Für die Encodierung in MP2 wird bei dieser Untersuchung das freie Software-Paket FFmpeg [39] verwendet. Dabei handelt es sich um einen Audio- und Video-Codierer mit verschiedenen Datenreduktionsmodulen aus der MPEG-Familie und anderen Technologien. Für die Layer-II-Encodierung wird die Bibliothek LibTwoLAME genutzt.

MPEG-1/2 Layer III (MP3) kann als eines der bekanntesten und am weitesten verbreiteten Audiodatenreduktionsverfahren angesehen werden [12, S. 870 – 871]. Es besitzt im Rundfunk große Bedeutung für die Programmverbreitung im Internet. Besonders im Hörfunk wird es sowohl für Live-Streams als auch für Podcasts und Audio on Demand häufig eingesetzt [40].

Die Encodierung in MP3 wird bei dieser Untersuchung mit der freien Software LAME [41] durchgeführt.

Für die Decodierung sowohl der MP2- als auch der MP3-Dateien wird ebenfalls LAME verwendet. Dies ist möglich, da Layer-III-Decodierer per Spezifikation rückwärtskompatibel zu den niedrigeren Schichten sind (siehe Kap. 2.3.4.2, S. 27).

---

1. GNU = GNU's Not Unix. Das GNU-Projekt entwickelt Software unter freier Lizenz. Homepage: <http://www.gnu.org>

ODG	Bedeutung
0	Imperceptible
-1	Perceptible but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Tabelle 3.1: Beschreibung der fünf Beeinträchtigungsstufen des Objective Difference Grades (aus [44, S. 53])

### 3.2.1.3 Psychoakustische Qualitätsanalyse mittels „PQevalAudio“

Bei perzeptiven Audiodatenreduktionsverfahren handelt es sich um i. A. sehr komplexe, nichtlineare und zeitvariante Systeme, deren Qualität oder Qualitätsbeeinträchtigung sich nicht durch traditionelle Parameter der Elektro- und Audiotechnik wie Frequenzgang, Klirrfaktor und Signal-Rauschabstand bestimmen lässt [42, S. 7]. Daher standardisierte die ITU in ihrer Empfehlung ITU-R BS.1387 [42] ein Verfahren zur objektiven Messung der wahrgenommenen Tonqualität. Es ermöglicht den Vergleich eines datenreduzierten Testsignals mit einem Referenzsignal, das normalerweise unkomprimiert vorliegt und ansonsten mit dem Testsignal übereinstimmt. Als Ergebnis wird der Grad der Qualitätsverschlechterung des Testsignals im Vergleich zur Referenz auf einer fünfstufigen Skala im Intervall  $[0, 0; -4, 0]$  angegeben. Dieser als „Objective Difference Grade“ (ODG) bezeichnete Wert ist als eine Prädiktion des für Hörversuche zur Bestimmung geringfügiger Qualitätsverschlechterungen definierten „Subjective Difference Grade“ (SDG) zu verstehen [42, S. 9, 16], [43, S. 165]. Letzterer errechnet sich aus der Differenz der subjektiven Bewertung des Testsignals zur subjektiven Bewertung des Referenzsignals, jeweils auch auf einer fünfstufigen Skala. Die Bewertungsskala für den ODG bzw. SDG ist in Tab. 3.1 ersichtlich.

Ein Blockschaltbild des Messverfahrens ist in Abb. 3.1 dargestellt. Der Algorithmus besteht im Wesentlichen aus einem psychoakustischen Modell, welches die Signalanalyse des Innenohres nachbildet [42, S. 17 – 19]. Dazu werden das Test- und das Referenzsignal blockweise in eine Frequenzbereichsdarstellung entsprechend den Frequenzgruppen des Gehörs (siehe Kap. 2.3.2, S. 22) transformiert. Beide Signale werden dann im Hinblick auf diverse Eigenschaften verglichen. Die Ergebnisse dieser Vergleiche werden in den so g. „Model Output Variables“ (MOVs, vgl. [42, S. 68 – 70]) ausgedrückt, die am Ende der Analyse über alle Blöcke zeitlich gemittelt werden. Ein nachgebildetes neuronales Netzwerk verknüpft die so erhaltenen durchschnittlichen MOVs schließlich zum ODG.

Der Algorithmus wurde in zwei Varianten unterschiedlicher Genauigkeit und Komplexität spezifiziert: Die „Basic Version“ führt die Frequenzanalyse nur durch eine Fast Fourier-Transformation (FFT) durch und bildet elf MOVs, während die „Advanced Version“ die FFT und eine Filterbank in Kombination verwendet und fünf MOVs ausgibt. Die Basic Version ist laut [42, S. 8] für Echtzeitmessungen gedacht. Für alle anderen Anwendungen wird die Advanced Version empfohlen, welche ungefähr die vierfache Komplexität der Basic Version besitzen, aber wesentlich genauere Ergebnisse liefern soll. Dagegen fanden

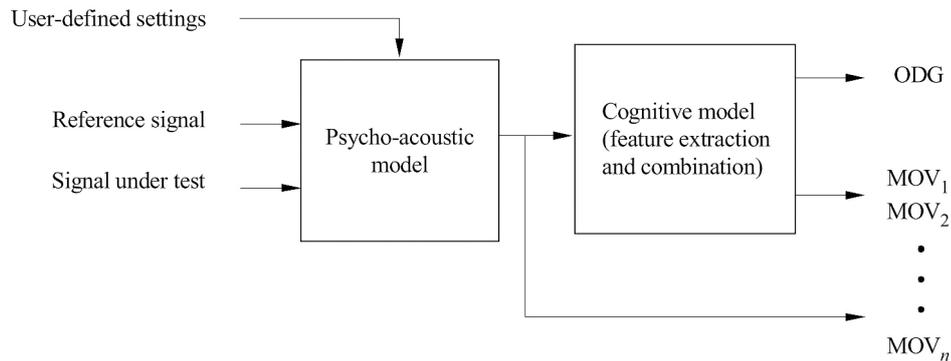


Abbildung 3.1: Blockschaltbild des Algorithmus zur objektiven Messung der wahrgenommenen Tonqualität (aus [42, S. 18])

Treurniet und Soulodre [43, S. 166], dass die Advanced Version nur geringfügig bessere Ergebnisse als die Basic Version zeige. Magadum [28, S. 5] stellte für die Advanced Version sogar größere Abweichungen von Ergebnissen eigener Hörversuche fest als für die Basic Version.

Das Messverfahren wird als Werkzeug zur Qualitätsbewertung für diese Untersuchung gewählt, da es v. a. für die Beurteilung der Qualitätsverschlechterung durch verlustbehaftete Audiodatenreduktion empfohlen wird [42, S. 7 – 8]. Gegen die (alleinige) Verwendung spricht, dass es laut Empfehlung nicht als genereller Ersatz für subjektive Hörversuche betrachtet werden könne. Magadum [28, S. 4 – 5] kritisiert zudem, dass der Algorithmus einige Artefakte grundsätzlich nicht erkenne und andere zu wenig in die Qualitätsbewertung miteinbeziehe. Dies betreffe insbes. zeitlich veränderliche Verzerrungen, die oft als sehr störend empfunden würden. Die zeitliche Mittelung führe oftmals dazu, dass die Einflüsse dieser veränderlichen Artefakte auf die MOVs am Ende zu wenig oder gar nicht zum Tragen kämen. In einer Gegenüberstellung diverser Qualitätsmessverfahren [45, S. 446 – 447] sehen die Verfasser ebenfalls Unsicherheiten bei der Erkennung gewisser Artefakte. Außerdem würden einige Faktoren der Qualitätsverschlechterung wie z. B. eine beeinträchtigte Reproduktion des Stereo-Klangbildes nicht berücksichtigt, während einige der MOVs kaum zur Einschätzung der Qualität beitragen. Die Autoren stellen jedoch insgesamt gute Übereinstimmungen mit subjektiven Versuchsreihen fest. Zu einem ähnlichen Ergebnis kommen auch Treurniet und Soulodre [43, S. 166 – 168], wobei sich nach Mittelung über alle Testbeispiele eine deutlich höhere Korrelation zwischen den ODGs und den aus Hörversuchen gewonnenen SDGs zeigte als zwischen den Graden für einzelne Testbeispiele. Die durchschnittlichen objektiven und subjektiven Grade wichen demnach nicht statistisch signifikant voneinander ab. Magadum [28, S. 4 – 5] empfiehlt den Algorithmus nicht als alleiniges Werkzeug zur Bewertung eines Datenreduktionsverfahrens, durchaus aber für Vergleiche zwischen verschiedenen Algorithmen, Implementierungen oder Parametern bei der Datenreduktion. Letztgenannter Anwendungsfall trifft auf diese Untersuchung zu: Nicht die Datenreduktionsverfahren selbst sollen bewertet werden, sondern nur der Einfluss einer Variation des Parameters „Aussteuerung“ auf die Qualitätsverschlechterung.

Der Algorithmus wird unter dem Namen „Perceptual Evaluation of Audio Quality“ (PEAQ) von der Firma Opticom als kommerzielle Software entwickelt. Für diese Untersuchung wird das Kommandozeilenprogramm PQevalAudio verwendet, eine freie Implementierung der Basic Version durch das Telecommunications & Signal Processing Laboratory der McGill University (vgl. [44], [46]). Dieses Programm ist Teil des „Audio File Programs and Routines package“ (AFsp), welches diverse Tools zur Audiosignalanalyse und -bearbeitung enthält. Für diese Untersuchung wird aus dem AFsp auch das Programm ResampAudio [47] zum Wandeln der Abtastrate von Wave-Dateien genutzt. Von denselben Autoren existiert ebenfalls eine MATLAB-Implementierung des Programms PQevalAudio; diese zeigt sich in Tests aber als in der Ausführung deutlich zu langsam, um die beschriebene Menge an Testbeispielen in akzeptabler Zeit zu verarbeiten.

Als ein weiteres objektives Messverfahren zur Bewertung der Tonqualität kommt zunächst auch die ebenfalls von Opticom entwickelte Software „Perceptual Evaluation of Speech Quality“ (PESQ) in Betracht. Dieses Verfahren ist jedoch für reine Sprachübertragung bei niedrigen Datenraten wie Telefonieanwendungen gedacht und erwartet als Referenzsignale reine, phonetisch ausgewogene Sprach-Samples [45, S. 443 – 446]. Es bewertet nicht die Qualität des Klangs, sondern die Sprachverständlichkeit. Daher erscheint es für vorliegenden Fall nicht geeignet.

### 3.2.2 Testsignale

Bei der Zusammenstellung des Testmaterials wird darauf Wert gelegt, für die Audiodatenreduktion möglichst anspruchsvolle Signalformen zu wählen; das bedeutet zum einen Tonsignale mit reichhaltigen Spektren, andererseits impulshaltige Signale. Auf diese Weise sollen die getesteten Datenreduktionsverfahren an ihre Grenzen geführt und die in Kap. 2.3.3 beschriebenen Artefakte hervorgerufen bzw. demonstriert werden. Es werden dazu sowohl Aufnahmen aus Referenzmaterial der EBU als auch Ausschnitte gesendeter Beiträge aus dem Hörfunkprogramm des WDR genutzt, um eine realistische Anwendung der Datenreduktionsverfahren im alltäglichen Betrieb zu simulieren.

Alle Testsequenzen liegen zweikanalig vor, davon die Sprachsignale monophon und alle übrigen stereophon.

#### 3.2.2.1 Referenz-Testsignale der EBU

Zur Überprüfung der Tonqualität eines Audiosystems bzw. der durch ein solches hervorgerufenen Qualitätsbeeinträchtigung bietet die EBU das „Sound Quality Assessment Material“ (SQAM) [48]. Dabei handelt es sich um eine Sammlung von verschiedenen Testbeispielen (ursprünglich als CD mit 70 Tracks), die zweikanalig mit Datenwortlänge 16 bit und Abtastrate 44100 Hz vorliegen. Die Sammlung umfasst synthetische Signale, akustische Musikinstrumente mit Einzeltönen, Akkorden und kurzen Melodien bzw. Rhythmen sowie Sprachbeispiele und Ausschnitte aus ganzen Musikstücken. Für diese Untersuchung werden ausschließlich natürliche Aufnahmen gewählt, da sich synthetische Testbeispiele nicht für die psychoakustische Qualitätsbewertung eignen [42, S. 14]. Die

35 aus dem SQAM gewählten Beispiele sind in Tab. 3.2 mit Begründung der Wahl aufgelistet. Die angegebenen Track-Nummern entsprechen der Numerierung der Tracks auf der Test-CD. Folgen auf diese Nummer ein Bindestrich und eine weitere Ziffer, so bezeichnet Letztere eines von mehreren Beispielen innerhalb des Tracks. Die Begründungen stützen sich oftmals auf eine Empfehlung der EBU in [48, S. 6 – 9] oder der ITU in [42, S. 14], was ggf. durch eine entsprechende Abkürzung oder Nummer in der Tabelle vermerkt ist.

Die Bedeutung der einzelnen in der Tabelle verwendeten Abkürzungen und Nummern lautet wie folgt:

**s.t.** single tone

**arp.** arpeggio

**m.p.** melodious phrase

**fem.** female voice

**m.** male voice

**(brr)** von EBU empfohlen für Anwendungsfall „bit rate reduction“

**(fr)** von EBU empfohlen für Anwendungsfall „frequency response“

**(dr)** von EBU empfohlen für Anwendungsfall „dynamic range“

**(ov)** von EBU empfohlen für Anwendungsfall „overload after processing“

**(1)** ITU-Empfehlung „Transients: pre-echo sensitive, smearing of noise in temporal domain.“

**(2)** ITU-Empfehlung „Tonal structure: noise sensitive, roughness.“

**(3)** ITU-Empfehlung „Natural speech (critical combination of tonal parts and attacks): distortion sensitive, smearing of attacks.“

**(4)** ITU-Empfehlung „Complex sound: stresses the Device Under Test.“

**(5)** ITU-Empfehlung „High bandwidth: stresses the Device Under Test, loss of high frequencies, programme-modulated high frequency noise.“

### 3.2.2.2 Exemplarische Testsignale aus dem Rundfunk

Die verwendeten Testbeispiele aus dem Rundfunk stammen aus einer Zusammenstellung für die ebenfalls beim WDR verfasste Bachelorarbeit [49]. Diese umfasst im Hörfunkprogramm des WDR gesendete Beiträge, Originaltöne und Musikaufnahmen, teilweise in Auszügen. Alle Beispiele liegen einzeln als zweikanalige Wave-Dateien mit Abtastfrequenz 48000 Hz und Datenwortlänge 16 bit vor. Hieraus werden für diese Untersuchung wegen des hohen Musikanteils beim SQAM (siehe Tab. 3.2) ausschließlich Testsequenzen mit Sprache verwendet.

Track	Dauer	Inhalt	Begründung/Empfehlung
22-1	0:17	Trombone arp.	obertonreich
22-2	0:23	Trombone m.p.	obertonreich
26-1	0:18	Claves s.t.	ITU (1)
26-2	0:21	Claves s.t., rhythm	ITU (1)
27	0:20	Castanets s.t., rhythm	ITU (1)
28-1	0:18	Side drum Without snares s.t.	Impulse
28-2	0:19	Side drum Without snares roll	Impulse
28-3	0:17	Side drum With snares s.t.	ITU (1)
28-4	0:22	Side drum With snares roll	ITU (1)
32-1	0:43	Triangles s.t.	EBU (brr), ITU (1, 2, 5)
32-2	0:31	Triangle roll	EBU (brr), ITU (1, 2, 5)
35-1	0:35	Glockenspiel arp.	EBU (brr), ITU (1, 2, 5)
35-2	0:24	Glockenspiel m.p.	EBU (brr), ITU (1, 2, 5)
36-1	0:26	Xylophone s.t.	ITU (1, 2)
36-2	0:24	Xylophone m.p.	ITU (1, 2)
38	0:21	Marimba m.p.	ITU (1)
39-1	0:20	Grand piano arp.	obertonreich
39-2	0:42	Grand piano s.t. (mezzoforte)	EBU (brr)
39-3	0:48	Grand piano s.t. (fortissimo)	EBU (brr)
39-4	0:27	Grand piano m.p.	EBU (brr)
40-1	0:23	Harpsichord arp.	EBU (brr), ITU (1, 2, 4)
40-2	0:30	Harpsichord m.p.	ITU (1, 2, 4)
49	0:23	Fem. speech English	ITU (3)
50	0:22	M. speech English	ITU (3)
53	0:21	Fem. speech German	Bsp. für deutschspr. Rundfunk
54	0:21	M. speech German	ITU (3)
55	0:32	Trumpet Haydn	ITU (2)
57	0:20	Organ Bach	komplexes Spektrum, Impulse
60	1:32	Piano Schubert	EBU (brr, dr), ITU (2)
61	2:59	Soprano Mozart	EBU (fr, ov), ITU (4)
63	0:57	Soloists Verdi	EBU (fr), komplexes Spektrum
64	0:31	Choir Orff	EBU (fr), komplexes Spektrum
65	1:52	Orchestra R. Strauss	EBU (brr, dr, fr)
69	0:33	ABBA	EBU (brr)
70	0:21	Eddie Rabbitt	EBU (ov)

Tabelle 3.2: Für die Untersuchung verwendete Testbeispiele aus EBU-SQAM; Sprache 2-kanalig monophon, alle übrigen stereophon; Abkürzungen siehe Legende auf S. 35

Bei der Wahl der 14 Beispiele wird sowohl auf möglichst große Vielfalt (Sprechertexte und Originaltöne, Aufnahmen mit und ohne Hintergrundgeräusch) wie auch auf eine gewisse Repräsentativität für den alltäglichen Rundfunkbetrieb geachtet. Letzteres bedeutet konkret, dass jedes Beispiel mindestens eine der folgenden Eigenschaften besitzt:

- Stimme eines oft im WDR zu hörenden Journalisten
- Stimme eines oft im WDR zu hörenden Originaltongebers (eines bekannten Politikers oder Sportlers)
- Geräusche, Atmosphären
- eine von Technikern des WDR als kritisch bewertete Tonqualität (Übersteuerungen bzw. nichtlineare Verzerrungen, starker Raumhall, unsaubere Schnitte, hoher Nebengeräuschpegel)

### 3.2.3 Pegelanalyse und -änderung mittels Octave

Im Rahmen meines Praktikums beim WDR erstellte ich in Octave Funktionen zur Berechnung der Programmlautheit und des maximalen True Peak Levels gemäß ITU-R BS.1770-2 [11] sowie des maximalen Spitzen- des durchschnittlichen unbewerteten RMS- und des durchschnittlichen VU-Pegels eines ganzen Audiosignals. Mit einer weiteren Funktion kann je nach Wahl einer der genannten Pegel auf einen gewünschten Zielwert geändert werden. Diese Funktionen werden im Folgenden beschrieben.

#### 3.2.3.1 Ermittlung der Programmlautheit

Der in ITU-R BS.1770-2 [11] empfohlene Algorithmus zur Berechnung der Lautheit mit Gating (siehe Kap. 2.2.5) ist in der Funktion `loudness` implementiert. Der zugehörige Quellcode findet sich in Anhang A.1. Die Funktion ermittelt die Programmlautheit gemäß EBU R 128 [17] (siehe Kap. 2.2.6.1) eines ganzen Signals.

Zunächst werden die in der aktuellen Empfehlung vorgegebenen Zahlenwerte (Blocklänge, Überlappungsfaktor der Blöcke, absoluter Schwellwert für die 1. Gating-Stufe, Gewichtungsfaktoren der Tonkanäle (Tab. 2.2) und Korrekturkonstante) festgelegt, um diese ggf. später leichter abändern zu können. Danach erfolgt – sofern erforderlich – eine Abtastratenwandlung des Signals auf 48000 Hz, da in der Empfehlung nur Filterkoeffizienten für diese Abtastrate angegeben sind.

Die Funktion `loudness` erwartet ein Tonsignal in der für das Octave-Paket „Audio“ üblichen Form, d. h. als Matrix, deren Zeilen den einzelnen Abtastwerten und deren Spalten den Tonkanälen entsprechen. Sollte die Darstellung der Abtastwerte und Tonkanäle vertauscht sein, wird dies von der Funktion durch einen Vergleich von Spalten- und Zeilenanzahl erkannt und mittels Transposition der Matrix geändert. Die Funktion kann gemäß dem Algorithmus der ITU Signale mit bis zu fünf Tonkanälen verarbeiten. Diese müssen in der Spaltenreihenfolge

```
[left, right, centre, left surround, right surround]
```

angeordnet sein.

Danach erfolgt die zweistufige Filterung des zu messenden Signals mittels der Octave-eigenen Funktion `filter` und der in der Empfehlung angeführten Koeffizienten (siehe Tab. 2.1). Der in Octave mittels der Funktion `freqz` (Package „Signal“) aus den Koeffizienten errechnete Betragsfrequenzgang ist für beide Filterstufen in Abb. 3.2 bzw. 3.3 graphisch dargestellt.

Im Anschluss wird in einer `for`-Schleife die mittlere Leistung jedes Blocks in jedem Tonkanal gemäß Gl. 2.7 berechnet. Dazu wird die Octave-Funktion `sumsq` zur Bestimmung der quadratischen Summe genutzt, die dann nur noch durch die Anzahl der Samples eines Blocks dividiert werden muss. Alle Leistungswerte werden in einer zuvor entsprechend der Signallänge und der Anzahl der Kanäle initialisierten Matrix gespeichert.

Die Teillautheiten der einzelnen Blöcke gemäß Gl. 2.8 werden durch Multiplikation der Matrix mit den Leistungswerten und des Vektors mit den Kanalgewichtungsfaktoren sowie anschließende Logarithmierung berechnet. Die Teillautheiten müssen jedoch nicht zwischengespeichert werden, sondern werden direkt mit dem absoluten Schwellwert  $\Gamma_a$  der ersten Gating-Stufe verglichen, was der Umsetzung von Gl. 2.9 entspricht. Das Ergebnis dieses Vergleichs liefert einen Vektor, der für den Index jedes Blocks mit Teillautheit oberhalb des absoluten Schwellwerts den Wert 1 und für alle anderen Indices den Wert 0 enthält. Dieser Vektor wird als Menge der in die Messung einzubeziehenden Blöcke in der Variablen `Jg` gespeichert. Zur Ermittlung der „vorläufigen“ Gesamtlautheit nach Gl. 2.10 wird zunächst die Multiplikation der Matrix mit den Blockleistungen und des Vektors mit den Kanalgewichtungsfaktoren durchgeführt. Ohne Zwischenspeicherung wird sofort aus dem Ergebnisvektor und dem Vektor `Jg` das Skalarprodukt gebildet und durch die Summe aller Werte von `Jg`, d. h. die Anzahl der in die Messung einzubeziehenden Blöcke, dividiert. Auf diese Weise erhält man die durchschnittliche Leistung aller Blöcke, deren Logarithmus der Gesamtlautheit entspricht. Durch Subtraktion von 10 dB von diesem Ergebnis erhält man den relativen Schwellwert nach Gl. 2.11, der in  $\Gamma_r$  gespeichert wird. In genau gleicher Weise wie bei der ersten Gating-Stufe wird nun auch die zweite Stufe gemäß Gl. 2.12 durchgeführt, mit dem einzigen Unterschied, dass die Teillautheiten der Blöcke nun mit  $\Gamma_r$  anstatt mit  $\Gamma_a$  verglichen werden. Das Ergebnis wird wieder in `Jg` gespeichert. Die Berechnung der finalen Gesamtlautheit entspricht vollständig der beschriebenen Berechnung der „vorläufigen“ Gesamtlautheit; lediglich verweist `Jg` nun nur noch auf diejenigen Blöcke, die die zweite Gating-Stufe passiert haben.

### 3.2.3.2 Ermittlung des Maximum True Peak Levels

Die Funktion `truepeak` berechnet den höchsten echten Spitzenpegel (siehe Kap. 2.2.6.4) für jeden Kanal eines Signals. Der Quellcode ist in Anhang A.2 zu finden. Da in ITU-R BS.1770-2 [11] für die Überabtastung kein konkreter Interpolationsfilter empfohlen wird, verwendet `truepeak` dazu die Funktion `resample` aus dem Octave-Package „Signal“.

Zunächst wird die Matrix des Eingangssignals wie in Kap. 3.2.3.1, S. 37 beschrieben auf die korrekte Anordnung der Abtastwerte und Kanäle überprüft und ggf. transponiert. Die Abtastratenwandlung auf  $f_{up} = 192000$  Hz erfolgt kanalweise sowie in Blöcken,

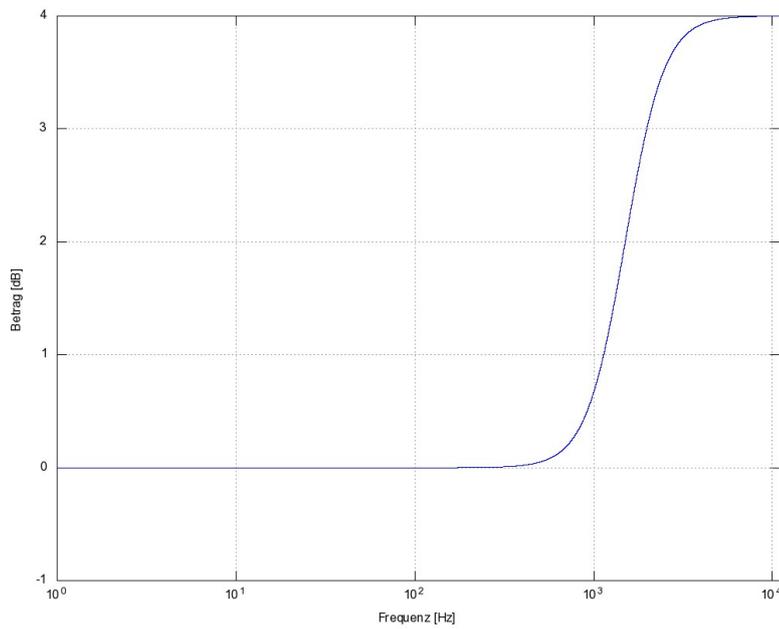


Abbildung 3.2: In Octave errechneter Betragsfrequenzgang der 1. Filterstufe (Shelving) für die Lautheitsmessung

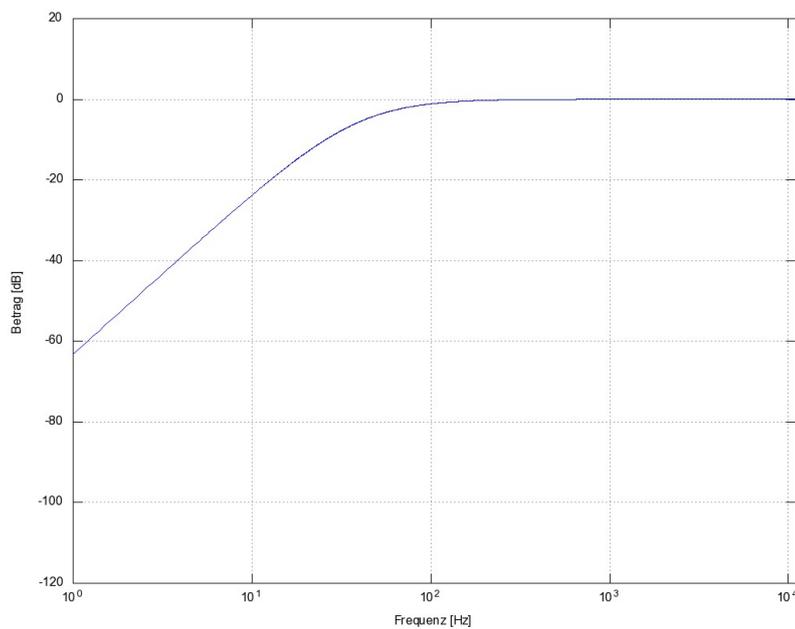


Abbildung 3.3: In Octave errechneter Betragsfrequenzgang der 2. Filterstufe (RLB) für die Lautheitsmessung

da sie zwangsläufig zu einer Vervielfachung der Datenmenge führt und eine zu hohe Beanspruchung des Arbeitsspeichers einen sofortigen Abbruch der Ausführung durch Octave zur Folge hat. Die Blockgröße wird dabei antiproportional zur neuen Abtastrate  $f_{up}$  berechnet. Als ein guter Kompromiss zwischen Stabilität und Rechenzeit zeigte sich bei Tests auf der verwendeten Hardware eine Berechnung der Blockgröße zu

$$\text{blocksize} = 480000000000 \text{ Samples} / f_{up}. \quad (3.1)$$

Zur besseren Ausnutzung des Arbeitsspeichers wird überdies das Eingangssignal gleich zu Beginn der Funktion in „Single Precision“, d. h. ein Fließkomma-Format mit nur 32 bit, umgerechnet; Octave rechnet standardmäßig mit 64-bit-Fließkommazahlen („Double Precision“). Auch die verwendete Funktion `resample` wurde für diese Untersuchung auf eine Berechnung und Ausgabe im Single-Format zu `resamplesg` modifiziert.

Die berechneten True Peak Levels für alle Kanäle werden im Vektor `L` gespeichert. Der Vektor wird vor der Berechnung für alle Kanäle mit  $-\infty$  initialisiert. In zwei verschachtelten `for`-Schleifen – die äußere für die Kanäle, die innere für die Blöcke – wird nun die Überabtastung jeweils für Block `n` aus Kanal `i` des Signals ausgeführt. Der Sample Peak Level des Blocks nach der Überabtastung, d. h. der True Peak Level des aktuell analysierten Blocks, wird in Element `i` des Vektors `L` gespeichert, sofern er höher ist als der zur Zeit in `L(i)` abgelegte Wert. Nach Ausführung der Schleifen enthält `L` den maximalen True Peak Level jedes Kanals und wird ausgegeben.

Die Berechnung des Sample Peak Levels erfolgt mit der ebenfalls für diese Arbeit erstellten Funktion `ppm` (siehe folgendes Kapitel).

### 3.2.3.3 Ermittlung des Spitzenpegels

Die Funktion `ppm` gibt den maximalen Sample Peak Level oder – je nach Wahl durch den Anwender – den quadratischen Mittelwert mit einer gewünschten Integrationszeit aus. Der für die Berechnung relevante Quellcode ist in Anhang A.3 aufgeführt. Geplant war zunächst auch die Berechnung des Quasispitzenpegels (QPPM-Betrieb, siehe Kap. 2.2.1, S. 9), hierfür ließ sich jedoch mangels entsprechender Angaben in der Literatur bislang keine Lösung realisieren, die den in [13] spezifizierten Anforderungen gerecht wird.

Die Berechnung des quadratischen Mittelwerts erfolgt in Blöcken, die zueinander um je einen Abtastwert verschoben sind und deren Länge durch die gewünschte Integrationszeit bestimmt wird. Die entsprechende Blocklänge in Abtastwerten wird in der Variablen `type` gespeichert; im Fall „samplepeak“ ist ihr Wert 1, da keine Mittelung erfolgt (PPM-Betrieb; siehe Kap. 2.2.1, S. 9). Die blockweise Mittelung wird für kürzere Blöcke durch eine Faltung des quadrierten Eingangssignals mit einem Einheitsvektor der Länge `type` durch die Octave-eigene Funktion `filter` realisiert, da dies im genannten Fall durch Octave schneller ausgeführt wird als die Berechnung in einer Schleife. Mittels der Octave-eigenen Funktionen `max` und `sqrt` sowie der für diese Arbeit erstellten Funktion `level` (siehe folgendes Kapitel) wird die Quadratwurzel des maximalen quadratischen Mittelwerts in einen logarithmischen digitalen Pegel umgerechnet.

### 3.2.3.4 Pegeländerung

Die Funktion `level` (Quellcode in Anhang A.4) hat zwei verschiedene Aufgaben, die durch den Anwender mittels entsprechender Parameter gewählt werden können: Einerseits rechnet sie eine Signalamplitude in den logarithmischen digitalen Pegel gemäß Gl. 2.3 um. Bezugswert 0 dBFS ist der Amplitudenwert  $Q = 1$ , was der Darstellung von Audiosignalen durch die Funktionen `wavread` und `wavwrite` des Octave-Packages „Audio“ entspricht. Mittels der Funktion `round` wird der errechnete Pegel auf eine Nachkommastelle gerundet.

Andererseits kann `level` dazu genutzt werden, den Pegel eines gesamten Signals um eine bestimmte Differenz  $L$  in dB oder auf einen bestimmten absoluten Zielwert  $L_{FS}$  zu ändern. Der absolute Pegel kann dabei ein Lautheitswert in LUFS, ein Spitzenpegel in dBFS oder ein anderer Pegeltyp sein. Dazu muss die Umkehrfunktion von Gl. 2.3 gebildet und das Ergebnis mit den Abtastwerten  $x_{in}$  des Eingangssignals multipliziert werden:

$$x_{out} = 10^{L/20} x_{in} \quad (3.2)$$

Im Falle einer relativen Pegeländerung, also einer Verstärkung oder Dämpfung, entspricht  $L$  der vom Anwender angegebenen Differenz in dB. Für die Änderung des Pegels auf einen absoluten Zielwert berechnet sich  $L$  aus der Differenz des vom Anwender bestimmten Zielwerts  $L_{FS}$  zum Pegel des Eingangssignals, je nach gewünschter Betriebsart gemessen mit `loudness` bzw. `ppm`.

In beiden Fällen führt `level` nur eine Verstärkung oder Dämpfung des *gesamten* Signals, also eine statische Pegeländerung durch. Die Dynamik des Signals wird nicht beeinflusst.

## 3.3 Versuchsdurchführung

Die Vorgehensweise bei dieser Untersuchung lässt sich folgendermaßen beschreiben:

Zunächst wird die Abtastrate der Beispiele aus dem SQAM mittels des Programms `ResampAudio` aus dem AFsp von 44100 Hz (original) auf 48000 Hz (Standard im Rundfunk) gewandelt. Außerdem werden die Tracks aus dem SQAM, die mehrere Beispiele enthalten, mit Hilfe des Tonbearbeitungsprogramms `Samplitude` in einzelne Dateien gespeichert, um diese separat messen bzw. aussteuern zu können. Mittels der Funktion `level` (siehe Kap. 3.2.3.4) sowie der Funktionen `wavread` und `wavwrite` aus dem Octave-Package „Audio“ wird nun zu allen Testsignalen aus dem SQAM (siehe Tab. 3.2) und aus dem Hörfunkprogramm des WDR (siehe Kap. 3.2.2.2) eine Kopie mit geänderter Pegel erstellt, so dass jedes Testbeispiel in zwei unkomprimierten Wave-Dateien mit verschiedenen Aussteuerungstypen vorliegt: Die Testbeispiele werden einerseits im Originalpegel und andererseits normalisiert auf einen Programmlautheitspegel von -23 LUFS verwendet. Einzig das Beispiel „Castanets s.t. rhythm“ aus dem SQAM wird nicht auf -23 LUFS ausgesteuert, da dies einen maximalen True Peak Level von +2,4 dBTP zur Folge hätte (siehe Tab. 3.4); hier wird deshalb eine Dämpfung vorgenommen, so dass der maximale True Peak Level gerade -1 dBTP beträgt. Außerdem werden alle Beispiele

mittels der Funktionen `ppm`, `loudness` und `truepeak` auf ihren maximalen Spitzen- und ihren Lautheitspegel sowie ihren maximalen True Peak Level hin untersucht.

Mit `Samplitude` wird schließlich aus den Testbeispielen in Originalaussteuerung eine Sequenz erstellt, um eine Rundfunksendung bestehend aus verschiedenen abwechselnd musikalischen und sprachlichen Elementen zu simulieren. Eine zweite „Pseudo-Rundfunksendung“ wird auf einer weiteren Spur – zeitlich parallel zur ersten Sequenz – aus den auf -23 LUFS normalisierten Testbeispielen zusammengesetzt. In `Samplitude` findet dabei keinerlei Pegelbeeinflussung statt. Schließlich werden noch auf beiden Spuren parallel je 3 s Stille am Ende aller SQAM-Tracks herausgeschnitten, um lange Pausen zu vermeiden. Beide der Pseudo-Rundfunksendungen besitzen damit im Hinblick auf die enthaltenen Testbeispiele und die zeitliche Anordnung einen identischen Aufbau und unterscheiden sich einzig durch die Art der Aussteuerung voneinander: Die eine besteht nur aus Elementen in originaler Aussteuerung, die andere nur aus Elementen mit einem Programmlautheitspegel von -23 LUFS und einem maximalen True Peak Level von -1 dBTP. Auf diese Weise soll die Reaktion der Datenreduktionsverfahren auf eine Abfolge unterschiedlicher Signale unter Verwendung der spitzenpegelorientierten Aussteuerung einerseits und der lautheitsorientierten Aussteuerung andererseits getestet werden. Die Sequenz ist in Tab. 3.3 durch Auflistung der Testbeispiele, geordnet nach Reihenfolge der Verwendung, mit Startposition, ggf. Track-Nummer auf der SQAM-CD und Kurzbezeichnung (Abkürzungen siehe Kap. 3.2.2.1, S. 35) dargestellt.

Beide Sequenzen werden in je eine Wave-Datei gespeichert. Diese Wave-Dateien werden folgendermaßen encodiert:

Die Datenreduktion in MP2 erfolgt mit einer Datenrate von 192 Kbit/s und einer Abtastfrequenz von 48000 Hz. Dies entspricht einer üblichen Konfiguration bei der Quellcodierung für DAB und DVB [37, S. 170], [38, S. 103 – 105], [50].

Für die Encodierung in MP3 wird eine konstante Datenrate von 128 Kbit/s, eine Abtastfrequenz von 44100 Hz und der „Regular Stereo“-Modus (kein Joint Stereo, sondern separate Behandlung beider Tonkanäle) gewählt, was einem gängigen Format für Hörfunk im Internet (Livestreams, Podcasts und Audio on Demand) entspricht [40]. Darüber hinaus bietet die verwendete Software LAME insgesamt zehn verschiedene Codieralgorithmen unterschiedlicher Qualität und Komplexität an. Bei Tests auf der verwendeten Hardware zeigt sich, dass der in der LAME-Dokumentation [41] empfohlene Algorithmus mit der dritthöchsten Komplexität (Qualitätsstufe 2) Echtzeitanforderungen (wie bei der Quellcodierung für Livestreams) genügen kann. Daher wird dieser Algorithmus für die MP3-Encodierung genutzt.

Die so entstandenen MP2- und MP3-Dateien werden ohne weitere Bearbeitung wieder in Wave mit Datenwortlänge 16 bit decodiert, da weder `Octave` noch `PQevalAudio` MPEG-Dateiformate analysieren können. Dabei wird die Abtastrate der aus der MP3-Decodierung hervorgegangenen Wave-Dateien mit dem Programm `ResampAudio` aus dem AFsp von 44100 Hz wieder auf 48000 Hz geändert, da das Analysewerkzeug `PQevalAudio` nur Signale mit 48000 Hz verarbeiten kann.

Der für die anschließende Analyse in `Octave` eingegebene Quellcode ist zur Verdeutlichung für den Fall der MP2-Dateien in Anhang A.5 dargestellt: Jede der decodierten Da-

Startpos. [mm : ss : msmsms]	Nr.	Testbeispiel
00:00:000	22-1	Trombone arp.
00:17:000	22-2	Trombone m.p.
00:37:000	WDR	Bericht von Ralph Sina
01:18:030	26-1	Claves s.t.
01:36:030	26-2	Claves s.t., rhythm
01:54:030	WDR	O-Ton Angela Merkel
02:06:793	27	Castanets s.t., rhythm
02:23:793	WDR	Bericht von Katrin Brand
03:06:492	28-1	Side drum Without snares s.t.
03:24:492	28-2	Side drum Without snares roll
03:43:492	28-3	Side drum With snares s.t.
04:00:492	28-4	Side drum With snares roll
04:19:492	WDR	O-Ton Renate Künast
04:27:636	32-1	Triangles s.t.
05:10:636	32-2	Triangle roll
05:38:636	WDR	O-Ton Joachim Gauck
05:47:252	35-1	Glockenspiel arp.
06:22:252	35-2	Glockenspiel m.p.
06:43:252	WDR	O-Ton Mario Götze
06:52:370	36-1	Xylophone s.t.
07:18:370	36-2	Xylophone m.p.
07:39:370	WDR	Bericht von Thomas Bormann
08:23:443	38	Marimba s.t.
08:41:443	WDR	O-Ton Hannelore Kraft
08:53:363	39-1	Grand piano arp.
09:13:363	39-2	Grand piano s.t. (mezzoforte)
09:55:363	39-3	Grand piano s.t. (fortissimo)
10:43:363	39-4	Grand piano m.p.
11:07:363	WDR	Bericht Lateinamerika
11:47:382	40-1	Harpsichord arp.
12:10:382	40-2	Harpsichord m.p.
12:37:382	49	Fem. speech English
12:57:382	55	Trumpet Haydn
13:26:382	50	M. speech English
13:45:382	57	Organ Bach
14:02:382	53	Fem. speech German
14:20:382	60	Piano Schubert
15:49:382	54	M. Speech German
16:07:382	61	Soprano Mozart
19:03:382	WDR	O-Töne Theater
20:48:401	63	Soloists Verdi
21:42:401	WDR	Straßenumfrage
22:23:070	64	Choir Orff
22:51:070	WDR	Bericht Montenegro
25:30:848	65	Orchestra R. Strauss
27:19:848	WDR	Beitrag mit O-Tönen
30:10:361	69	ABBA
30:40:361	WDR	ital. Beitrag mit O-Tönen
34:30:792	70	Eddie Rabbitt
34:48:792		Ende

Tabelle 3.3: Sequenz der Testbeispiele mit Startposition

teien wird aus Octave heraus mit der zugehörigen nicht datenreduzierten Wave-Datei mittels des Programms PQevalAudio verglichen, um die objektive Qualitätsverschlechterung durch die Datenreduktion zu messen. Der Aufruf des Kommandozeilen-Programms PQevalAudio.exe erfolgt dabei mittels der Octave-eigenen Funktion `system`, welche auch die Bildschirmausgabe eines externen Programms zurückgeben kann. Wegen der beschriebenen Schwächen des Algorithmus in der Erkennung zeitlich veränderlicher Artefakte (siehe Kap. 3.2.1.3, S. 33) wird die Messung jedoch nicht über eine gesamte Sequenz durchgeführt, sondern in Blöcken zu je 100 ms. Die Bildschirmausgabe von PQevalAudio wird in Octave zwischengespeichert; das interessierende Endergebnis jeder Messung – der ODG – kann durch Extraktion der letzten sieben Zeichen des Output-Strings gewonnen werden. Alle so erhaltenen einzelnen ODGs werden in je einem Vektor pro Datenreduktionsverfahren und Aussteuerungstyp gespeichert, so dass insgesamt vier Vektoren mit blockweise errechneten ODGs vorliegen. Danach werden die Differenzen derjenigen Vektoren zueinander gebildet, die die ODGs des gleichen Datenreduktionsverfahrens enthalten. Jeder der so entstandenen zwei Vektoren speichert damit die Differenzen zwischen den bei den beiden verschiedenen Aussteuerungstypen gemessenen ODGs. Erst diese Einzelergebnisse werden mit der Funktion `mean` aus dem Octave-Package „Statistics“ für jedes der beiden Datenreduktionsverfahren über alle Blöcke zu je einer durchschnittlichen ODG-Differenz gemittelt. Außerdem werden mit den Funktionen `median`, `mode` und `std` aus dem gleichen Paket der Median, der Modus und die Standardabweichung berechnet sowie mit den Octave-eigenen Funktionen `min` und `max` der negative und der positive Extremwert der ODG-Differenzen ausgegeben. Die Funktion `hist` aus dem Package „plot“ wird zur Erstellung eines Histogramms über die Häufigkeitsverteilung der ODG-Differenzen genutzt.

### 3.4 Auswertung

Im Folgenden werden die bei der Versuchsdurchführung gewonnenen Messergebnisse graphisch und tabellarisch dargestellt. Außerdem werden auf Grundlage dieser Ergebnisse mittels Signifikanztests die in Kap. 3.1, S. 30 aufgestellten vier Alternativhypothesen überprüft.

Die Auswertung beschränkt sich auf die Differenzen zwischen den für die lautheitsorientierte Aussteuerung einerseits und die Originalaussteuerung andererseits ermittelten ODGs des gleichen zeitlichen Blocks, wobei die Untersuchungen jeweils separat für jedes der beiden Datenreduktionsverfahren durchgeführt werden. In Frage käme zunächst auch die Betrachtung einzelner MOVs, da diese laut [42, S. 15, 19] zur Ermittlung der Ausprägung bestimmter Artefakte genutzt werden könnten. Dagegen spricht jedoch, dass für die meisten MOVs weder eine Skala noch eine Dimension oder ein Bezugswert angegeben wird, so dass eine objektive Bewertung der Zahlenwerte schwer möglich erscheint. In diesem Zusammenhang fand Magadum [28, S. 4 – 5], dass die MOVs nicht auf die Typen einzelner Artefakte schließen ließen. Darüber hinaus besteht nach [44, S. 59] Unsicherheit darüber, ob und wie die berechneten Zahlenwerte der MOVs skaliert oder normiert werden müssten. Deshalb wird hier auf die Auswertung der MOVs verzichtet.

Nr.	Testbeispiel	Sample Lautheit		True Peak		Sample Peak neu	
		Peak Orig. [dBFS]	[LUFS]	Peak [dBTP]		Peak [dBFS]	
22-1	Trombone arp.	-6,9; -6,9	-18,3	-6,9; -6,9		-11,6; -11,6	
22-2	Trombone m.p.	-9,3; -11,6	-21,6	-9,3; -11,6		-10,7; -13,1	
WDR	Bericht von Ralph Sina	-5,9; -5,9	-19,1	-5,9; -5,9		-9,8; -9,8	
26-1	Claves s.t.	-7,0; -7,4	-23,7	-6,9; -7,4		-6,3; -6,7	
26-2	Claves s.t., rhythm	-1,9; -1,9	-19,4	-1,9; -1,8		-5,5; -5,5	
WDR	O-Ton Angela Merkel	-5,3; -5,2	-18,2	-5,3; -5,2		-10,1; -10,0	
27	Castanets s.t., rhythm	0,0; 0,0	-24,2	0,0; +1,2		-2,2; -2,2	
WDR	Bericht von Katrin Brand	-5,2; -5,2	-21,2	-5,2; -5,2		-7,0; -7,0	
28-1	Side drum Without snares s.t.	-7,1; -6,7	-23,4	-7,1; -6,7		-6,7; -6,3	
28-2	Side drum Without snares roll	-1,9; -0,6	-16,1	-1,9; -0,6		-8,8; -7,5	
28-3	Side drum With snares s.t.	-7,8; -7,5	-25,2	-7,8; -7,5		-5,6; -5,2	
28-4	Side drum With snares roll	-11,4; -11,7	-22,7	-11,4; -11,7		-11,7; -12,0	
WDR	O-Ton Renate Künast	-4,6; -4,4	-16,9	-4,5; -4,4		-10,7; -10,5	
32-1	Triangles s.t.	-7,6; -5,8	-22,8	-7,6; -5,4		-7,7; -6,0	
32-2	Triangle roll	-6,9; -10,3	-22,7	-6,9; -8,8		-7,2; -10,6	
WDR	O-Ton Joachim Gauck	-0,1; -0,1	-16,0	-0,1; -0,1		-7,1; -7,1	
35-1	Glockenspiel arp.	-2,8; -3,4	-18,3	-2,8; -3,3		-7,5; -8,2	
35-2	Glockenspiel m.p.	-4,2; -2,9	-16,9	-3,9; -2,8		-10,4; -9,0	
WDR	O-Ton Mario Götze	-2,4; -2,4	-19,0	-2,4; -2,4		-6,4; -6,4	
36-1	Xylophone s.t.	-14,3; -14,6	-32,2	-14,2; -14,5		-5,1; -5,4	
36-2	Xylophone m.p.	-10,1; -9,4	-23,1	-10,1; -9,4		-10,0; -9,4	
WDR	Bericht von Thomas Bormann	-1,2; -1,2	-20,1	-1,2; -1,2		-4,1; -4,1	
38	Marimba s.t.	-20,7; -20,4	-31,3	-20,7; -20,4		-12,4; -12,0	
WDR	O-Ton Hannelore Kraft	-4,8; -4,7	-19,5	-4,8; -4,7		-8,3; -8,2	
39-1	Grand piano arp.	-8,8; -8,3	-19,5	-8,8; -8,3		-12,3; -11,8	
39-2	Grand piano s.t. (mezzoforte)	-8,7; -7,1	-21,8	-8,7; -7,1		-9,9; -8,3	
39-3	Grand piano s.t. (fortissimo)	-2,9; -1,6	-16,4	-2,9; -1,6		-9,5; -8,2	
39-4	Grand piano m.p.	-2,0; -0,9	-16,0	-2,0; -0,9		-9,1; -7,9	
WDR	Bericht Lateinamerika	-5,6; -5,6	-20,6	-5,6; -5,6		-8,0; -8,0	
40-1	Harpsichord arp.	-11,2; -11,3	-25,0	-11,2; -11,3		-10,1; -9,3	
40-2	Harpsichord m.p.	-14,7; -13,4	-29,0	-14,7; -13,2		-8,7; -7,4	
49	Fem. speech Engl.	-4,6; -4,7	-19,5	-4,6; -4,7		-8,1; -8,2	
55	Trumpet Haydn	-5,1; -4,8	-19,9	-5,1; -4,7		-8,2; -7,9	
50	M. speech Engl.	-3,6; -3,9	-18,6	-3,6; -3,9		-8,0; -8,4	
57	Organ Bach	-17,6; -19,9	-29,1	-17,6; -19,8		-11,5; -13,7	
53	Fem. speech Ger.	-3,1; -3,2	-20,1	-3,1; -3,2		-6,1; -6,1	
60	Piano Schubert	-3,0; -1,9	-21,8	-3,0; -1,9		-4,3; -3,1	
54	M. Speech Ger.	-3,7; -3,8	-20,0	-3,7; -3,7		-6,7; -6,8	
61	Soprano Mozart	-5,6; -2,2	-20,7	-5,5; -2,1		-7,9; -4,5	
WDR	O-Töne Theater	-3,5; -3,5	-20,8	-3,4; -3,4		-5,7; -5,7	
63	Soloists Verdi	-0,4; -0,2	-17,7	-0,3; -0,2		-5,7; -5,6	
WDR	Straßenumfrage	-6,0; -6,0	-21,2	-5,9; -5,9		-7,7; -7,7	
64	Choir Orff	-2,2; -3,0	-12,5	-2,2; -3,0		-12,7; -13,4	
WDR	Bericht Montenegro	-8,5; -8,5	-20,8	-8,3; -8,3		-10,7; -10,7	
65	Orchestra R. Strauss	-0,8; -1,1	-15,1	-0,7; -1,0		-8,8; -9,0	
WDR	Beitrag mit O-Tönen	-4,6; -4,6	-20,5	-4,6; -4,6		-7,1; -7,1	
69	ABBA	-4,4; -4,4	-22,9	-4,4; -4,4		-4,5; -4,5	
WDR	Ital. Beitrag mit O-Tönen	-4,5; -2,0	-21,9	-4,5; -2,0		-5,5; -3,0	
70	Eddie Rabbitt	-4,0; -4,2	-20,1	-4,0; -4,2		-6,9; -7,1	

Tabelle 3.4: Gemessene Pegelwerte der Testbeispiele

### 3.4.1 Pegelanalyse der Sequenz

Die mittels der Funktionen ppm, loudness und truepeak in Octave ermittelten Pegelwerte der in der Sequenz verwendeten Testbeispiele in Originalaussteuerung sind in Tab. 3.4 aufgelistet. Man erkennt, dass die Programmlautheitspegel der Testbeispiele sich um bis zu 19,7 LU unterscheiden. Neun Beispiele weisen eine Programmlautheit unterhalb von -23 LUFS auf; in allen anderen 40 Fällen liegt sie darüber. Zu beachten ist hierbei jedoch, dass sich auch die Spitzenpegel einiger SQAM-Beispiele um bis zu 20,7 dB unterscheiden. Aus diesem Grund sind zur besseren Vergleichbarkeit in der letzten Spalte der Tabelle die Sample Peak Levels aller auf -23 LUFS ausgesteuerten Beispiele angegeben. Auch hier zeigen sich beträchtliche Differenzen, in diesem Fall bis zu rund 10 dB, was bei einer Aussteuerung auf einen gemeinsamen Sample Peak Level Lautheitsunterschiede von bis zu rund 10 LU bedeuten würde. Ähnliche Differenzen lassen sich auch innerhalb der SQAM-Beispiele beobachten, während sich innerhalb der Aufnahmen aus dem WDR-Hörfunkprogramm nur Pegeldifferenzen von bis zu ca. 7 dB zeigen. Eine Besonderheit stellt das Beispiel „Castanets s.t. rhythm“ dar, welches in Originalaussteuerung einen Sample Peak Level an der Aussteuerungsgrenze und gleichzeitig eine Lautheit von nur -24,2 LUFS aufweist.

Der maximale True Peak Level ist meistens gleich dem maximalen Sample Peak Level oder liegt allenfalls 0,1 dB darüber, in nur zwei Fällen um 0,2 dB. Ausnahmen bilden die Beispiele „Glockenspiel m.p.“ mit einer Pegeldifferenz von 0,3 dB, „Triangles s.t.“ mit 0,4 dB, „Castanets s.t. rhythm“ mit einer Differenz von 1,2 dB und „Triangle roll“ mit 1,5 dB. Die in EBU R 128 [17] empfohlene Grenze von -1 dBTP wird bei nur fünf Beispielen in Originalaussteuerung überschritten, wobei in jedem dieser Fälle schon der Sample Peak Level über -1 dBFS liegt.

### 3.4.2 Deskriptive Statistik der Qualitätsanalyse

Die mit Octave erstellten Histogramme zeigen die prozentuale Häufigkeitsverteilung der Differenzen zwischen den zeitlich entsprechenden ODGs mit beiden Aussteuerungstypen für MP2 (Abb. 3.4) bzw. MP3 (Abb. 3.5). Dabei werden die von PQevalAudio eigentlich mit fünf Nachkommastellen ausgegebenen ODG-Werte in einer Rasterung von 0,1 zusammengefasst, da nach [42, S. 10] die Auflösung des ODG auf diesen Wert begrenzt ist.

Kennwert	MP2	MP3
arith. Mittel	$\Delta ODG_{II} = -0,0044746$	$\Delta ODG_{III} = -0,0015546$
Median	$\Delta ODG_{II} = 0,003$	$\Delta ODG_{III} = 0,002$
Modus	$\Delta ODG_{D,II} = 0,0$	$\Delta ODG_{D,III} = 0,0$
Minimum	$\min\{\Delta ODG_{II}\} = -2,3180$	$\min\{\Delta ODG_{III}\} = -1,9140$
Maximum	$\max\{\Delta ODG_{II}\} = 2,3390$	$\max\{\Delta ODG_{III}\} = 2,0040$
Std.-Abweichung	$\sigma_{\Delta ODG,II} = 0,28618$	$\sigma_{\Delta ODG,III} = 0,20844$

Tabelle 3.5: Statistische Kennwerte der gemessenen ODG-Differenzen

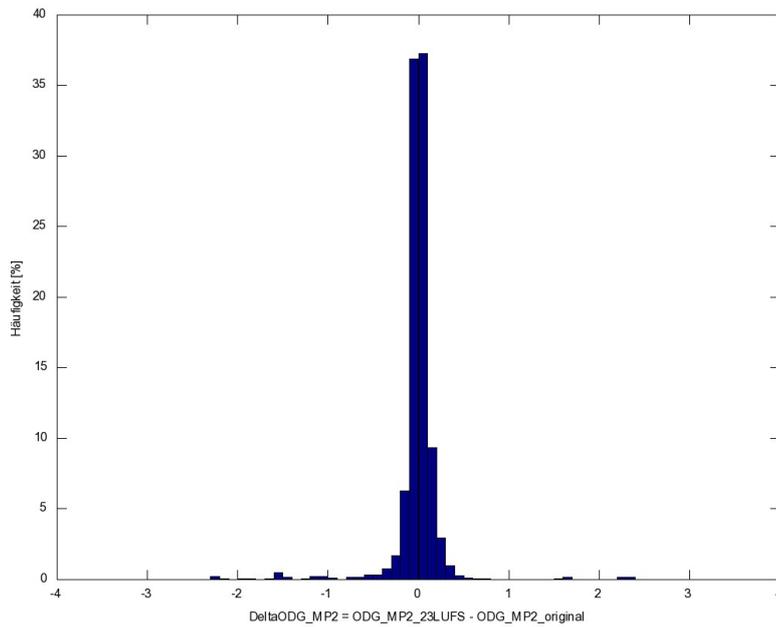


Abbildung 3.4: Histogramm über Häufigkeitsverteilung der gemessenen ODG-Differenzen für MP2

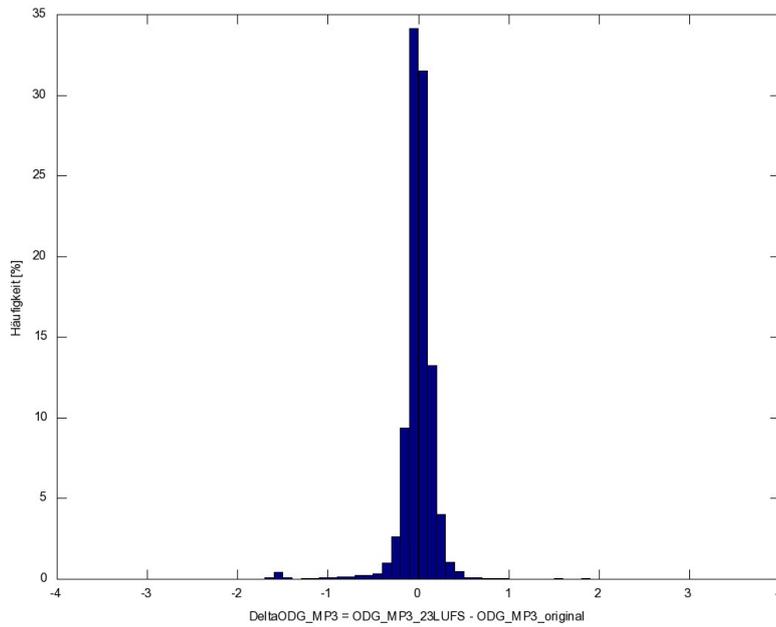


Abbildung 3.5: Histogramm über Häufigkeitsverteilung der gemessenen ODG-Differenzen für MP3

Man erkennt, dass für beide Datenreduktionsverfahren der Großteil der ODG-Differenzen im Intervall  $\pm 0,1$  liegt (bei MP2 knapp 74,2 %, bei MP3 über 65,6 %) sowie in beiden Fällen über 94 % im Intervall  $\pm 0,3$ , was vermuten lässt, dass die Art der Aussteuerung keinen erheblichen Einfluss auf das Ausmaß der durch die Datenreduktion hervorgerufenen Qualitätsverschlechterung hat. Den gleichen Eindruck vermitteln die statistischen Kennwerte in Tab. 3.5. Bemerkenswert ist überdies, dass der Anteil der positiven ODG-Differenzen bei beiden Datenreduktionsverfahren mit 3,4 bzw. 1,6 Prozentpunkten leicht höher ist als jener der negativen, was auf eine geringfügige Qualitätsverbesserung durch die lautheitsorientierte Aussteuerung hindeuten könnte.

### 3.4.3 Hypothesenprüfung

Zur Überprüfung der in Kap. 3.1, S. 30 aufgestellten vier Alternativhypothesen anhand der bei der Untersuchung gefundenen Messwerte ist ein Signifikanztest erforderlich. Für diese Untersuchung wird der  $t$ -Test zum Vergleich eines Stichprobenmittelwertes  $\bar{x}$  mit dem Mittelwert  $\mu_0$  einer Grundgesamtheit (Populationsmittelwert) [30, S. 136 – 139] angewandt. Dabei wird zunächst davon ausgegangen, dass die Gegenhypothese zur jeweiligen Alternativhypothese – die so g. Nullhypothese  $H_0$  – wahr sei und die Alternativhypothese daher verworfen werden müsse. Es wird nun die Wahrscheinlichkeit  $P$  des Falls bestimmt, dass unter dieser Bedingung („ $H_0$  wahr“) ein Mittelwert der Messergebnisse auftritt, der sich vom in der Nullhypothese angenommenen Mittelwert  $\mu_0$  der Grundgesamtheit mindestens um eine Differenz von  $\bar{x} - \mu_0$  unterscheidet. Liegt diese Irrtumswahrscheinlichkeit unterhalb eines bestimmten Signifikanzniveaus  $\alpha$ , kann die jeweilige Alternativhypothese angenommen werden.

Der Test stützt sich auf das zentrale Grenzwerttheorem, wonach die Mittelwerte  $\bar{x}$  für Stichproben hinreichenden Umfangs ( $n > 30$ ) – unabhängig von der Verteilung des gefragten Merkmals – näherungsweise normalverteilt um den Mittelwert der zugehörigen Grundgesamtheit mit einem Standardfehler  $\sigma_{\bar{x}}$  ist [30, S. 90 – 94]. Der Standardfehler berechnet sich aus dem Verhältnis der Merkmalsstreuung  $\sigma$  (Standardabweichung) der Grundgesamtheit zur Wurzel des Stichprobenumfangs:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \quad (3.3)$$

Ist die Merkmalsstreuung der Grundgesamtheit bekannt, kann die Irrtumswahrscheinlichkeit für den Stichprobenmittelwert  $\bar{x}$  direkt aus der Standardnormalverteilung abgelesen werden, indem die Differenz des Stichprobenmittelwertes zum Mittelwert der Grundgesamtheit auf den Standardfehler der Mittelwertverteilung bezogen wird:

$$\bar{z} = \frac{\bar{x} - \mu_0}{\sigma_{\bar{x}}} = \sqrt{n} \frac{\bar{x} - \mu_0}{\sigma} \quad (3.4)$$

Ist die Merkmalsstreuung der Grundgesamtheit hingegen nicht bekannt, wie im vorliegenden Fall, muss sie in letzterer Gleichung durch die geschätzte Varianz der Grundge-

samtheit ersetzt werden [30, S. 90 – 95]. Diese errechnet sich aus den Stichprobenwerten zu

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}, \quad (3.5)$$

so dass sich für die Mittelwertverteilung ein geschätzter Standardfehler ergibt:

$$S_{\bar{x}} = \sqrt{\frac{S^2}{n}} = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu_0)^2}{n(n - 1)}} \quad (3.6)$$

Der Quotient aus der Differenz der Mittelwerte und dem geschätzten Standardfehler ist aber nicht normal-, sondern  $t$ -verteilt. Die Irrtumswahrscheinlichkeit  $P$  muss nun also aus der  $t$ -Verteilung anhand der Prüfgröße  $t$  abgelesen werden:

$$t = \frac{\bar{x} - \mu_0}{S_{\bar{x}}} = \sqrt{n(n - 1)} \frac{\bar{x} - \mu_0}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \quad (3.7)$$

Zur Überprüfung der vier Alternativhypothesen wird die Funktion `t_test` des Octave-Packages „Statistics“ genutzt (siehe Quellcode in Anhang A.5, S. 68). Durch die Analyse der Testsequenzen von 34:48 min Länge in Blöcken zu 100 ms ist ein Stichprobenumfang von  $n \gg 30$  gegeben, so dass von einer Normalverteilung der Stichprobenmittelwerte  $\overline{\Delta ODG}$  um den Mittelwert  $\mu_{\Delta ODG}$  der Grundgesamtheit ausgegangen werden kann und der  $t$ -Test damit zulässig ist. Aufgrund der Formulierung der Auswirkungen „nicht negativ“ bzw. „positiv“ in den Hypothesen müssen die Tests rechtsseitig erfolgen ( $H_1 : \mu_{\Delta ODG} > \mu_0$ ). Es stellt sich lediglich die Frage, wie groß eine Differenz von ODGs sein muss, um als Signifikante Verbesserung oder Verschlechterung betrachtet zu werden. Hierzu finden sich in der Literatur nicht eindeutige Angaben: In der Empfehlung ITU-R BS.1387 [42, S. 10] wird die Auflösung des ODG auf eine Zehntelstelle festgelegt, weiter heißt es jedoch, dass eine Differenz von 0,1 nicht notwendigerweise als signifikant zu betrachten sei. Magadum [28, S. 4] und Preihs et al. [51, S. 5 – 7] werten ODG-Differenzen zwischen 0,1 und 0,2 als signifikant, während Thiede et al. [52, S. 21] die Messunsicherheit mit 0,25 im oberen Bereich und mit 0,5 im unteren Bereich der ODG-Skala angeben. Für diese Untersuchung wird mit einer mittleren ODG-Differenz von  $\mu_{0,n} = -0,1$  bzw.  $\mu_{0,p} = +0,1$  das strengste der genannten Entscheidungskriterien gewählt. Die mathematische Formulierung der vier Alternativhypothesen lautet damit

$$H_{1,n,II} : \mu_{\Delta ODG,II} > -0,1 \quad (3.8)$$

$$H_{1,p,II} : \mu_{\Delta ODG,II} > +0,1 \quad (3.9)$$

$$H_{1,n,III} : \mu_{\Delta ODG,III} > -0,1 \quad (3.10)$$

$$H_{1,p,III} : \mu_{\Delta ODG,III} > +0,1. \quad (3.11)$$

Die Funktion  $t\_test$  liefert folgende Irrtumswahrscheinlichkeiten.

$$H_{1,n,II} : P(\overline{\Delta ODG_{II}} > -0,1 | H_{0,n,II}) = 1,1413 \cdot 10^{-13} \quad (3.12)$$

$$H_{1,p,II} : P(\overline{\Delta ODG_{II}} > +0,1 | H_{0,p,II}) = 1 \quad (3.13)$$

$$H_{1,n,III} : P(\overline{\Delta ODG_{III}} > -0,1 | H_{0,n,III}) = 0 \quad (3.14)$$

$$H_{1,p,III} : P(\overline{\Delta ODG_{III}} > +0,1 | H_{0,p,III}) = 1 \quad (3.15)$$

Damit können  $H_{1,n,II}$  und  $H_{1,n,III}$  auf sehr hohem Signifikanzniveau ( $\alpha = 0,01$ ) [30, S. 114] angenommen werden: Die lautheitsorientierte Aussteuerung gemäß EBU R 128 wirkt sich demnach sowohl im Falle von MP2 als auch im Falle von MP3 *nicht negativ* auf die Tonqualität aus.

Dagegen können  $H_{1,p,II}$  und  $H_{1,p,III}$  nicht bestätigt werden: Eine signifikante Qualitätsverbesserung in Folge der lautheitsorientierten Aussteuerung gemäß EBU R 128 lässt sich sowohl im Falle von MP2 als auch im Falle von MP3 bei dieser Untersuchung nicht nachweisen.

## Kapitel 4

# Schlussbetrachtungen

Auf Grundlage der theoretischen Betrachtungen zu den für diese Arbeit zentralen Themen „Lautheit und Aussteuerung“ sowie „perzeptive Audiodatenreduktion“ wurde eine praktische Untersuchung zur Auswirkung der Aussteuerung auf die Qualitätsverschlechterung bei der im Rundfunk gängigen Quellcodierung durchgeführt. In der praktischen Untersuchung konnte gezeigt werden, dass die neue, in EBU R 128 empfohlene lautheits- und True Peak-orientierte Art der Aussteuerung mit sehr hoher Wahrscheinlichkeit die Tonqualität sowohl bei der Codierung mit MP2 als auch bei der Codierung mit MP3 nicht negativ beeinflusst. Die Aussteuerung gemäß EBU R 128 kann somit nach dem Ergebnis dieser Untersuchung im Hinblick auf die Quellcodierverfahren MP2 und MP3 ohne Bedenken eingesetzt werden. Hingegen konnte eine eventuelle Qualitätsverbesserung bezüglich der Audiodatenreduktion als Folge der neuen Art der Aussteuerung im Vergleich zur bisherigen Aussteuerungspraxis nicht bestätigt werden. Wenngleich ein negatives Ergebnis eines Signifikanztests theoretisch keine eindeutige Widerlegung der Alternativhypothese bedeutet, erscheint es angesichts der in Octave ausgegebenen Irrtumswahrscheinlichkeiten (Gln. 3.13 und 3.15) sehr unwahrscheinlich, dass eine Änderung der Aussteuerungspraxis einen positiven Effekt auf die Qualität der Audiodatenreduktion hat. Eine mögliche Erklärung für dieses Ergebnis könnte sein, dass nur bei fünf der 49 verwendeten Testbeispiele in Originalaussteuerung der True Peak Level tatsächlich die empfohlene Grenze von -1 dBTP überschreitet. Die Alternativhypothesen zur Qualitätsverbesserung wurden aber aus der Annahme hergeleitet, dass der maximal zulässige True Peak Level bei der bisherigen Aussteuerungspraxis oftmals überschritten würde.

Zusammenfassend wäre die baldige konsequente Umsetzung der Empfehlung EBU R 128 in Hörfunk und Fernsehen meines Erachtens sehr zu begrüßen. Dafür sprechen nicht nur die Untersuchungsergebnisse, sondern v. a. die in Kap. 2.2.2 beschriebenen Nachteile der spitzenpegelorientierten Aussteuerung und nicht zuletzt die in diesem Zusammenhang erwähnten Beschwerden durch die Hörer und Zuschauer. Inwieweit der in ITU-R BS.1770-2 empfohlene Messalgorithmus tatsächlich zu einer subjektiv empfundenen Ausgewogenheit in der Lautheit der Sender und Programmbeiträge führen kann, wird die Praxis zeigen. Sowohl die nach wie vor andauernde Forschung zu Lautheitsmessverfahren als auch die permanent steigende Rechenleistung der Mikroprozessoren geben Anlass zu

hoffen, dass sich in den nächsten Jahren ein allseits zufriedenstellendes Lautheitsmessverfahren etablieren wird.

Es wird angeregt, die Ergebnisse dieser Arbeit durch subjektive Hörversuche zu verifizieren. Außerdem erscheint es sinnvoll, diese Untersuchung auf andere im Rundfunk eingesetzte Audiodatenreduktionsverfahren wie Dolby AC3 und die Varianten von MPEG-2/4 Advanced Audio Coding (AAC) zu erweitern. Schließlich könnte auch der Einfluss einer automatisierten Lautheitsangleichung mittels einer entsprechend modifizierten Automatic Gain Control auf die Tonqualität bei der Datenreduktion untersucht werden.

In diesem Zusammenhang sei noch ein alternativer Ansatz zur Lösung des Lautheitsproblems genannt, der u. a. in [15] erwähnt wird: Anstatt schon bei der Aussteuerung im Sendestudio und in der Programmverbreitung eine einheitliche Lautheit der Programmbeiträge herzustellen, können alle Beiträge, Musikstücke etc. mit Zusatzdaten (Tags) ausgestrahlt werden, die deren Lautheitspegel angeben. Eine Pegelkorrektur nimmt in diesem Fall erst das Empfangsgerät vor. Dieses Verfahren bietet den Vorteil, dass der Hörer das Regelverhalten nach eigenem Geschmack beeinflussen kann, so dass z. B. Musiktitel oder Wortbeiträge bevorzugt werden. Da viele neuere Codierverfahren bereits die Definition von Lautheits-Tags beinhalten, ist die Einführung einer solchen Technik für den Rundfunkbetrieb in nächster Zeit durchaus möglich. Eine wichtige Voraussetzung bildet jedoch auch in diesem Fall ein zuverlässiges Lautheitsmessverfahren.

## Quellenverzeichnis

- [1] A. Sontacchi, „Entwicklung eines Modulkonzeptes für die psychoakustische Geräuschanalyse unter MatLab“, Diplomarbeit, Inst. Nachrichtentechnik u. Wellenausbreitung, TU Graz, 1998.
- [2] H. Birkenhake und Ch. Struck „Untersuchung zur Wahrnehmbarkeit von Dynamikeinengungen“, Diplomarbeit, Erich-Thienhaus-Institut, Detmold, 2006.
- [3] M. Fresner, „Untersuchung von Messmethoden zur Bestimmung eines einzelnen Lautheitswertes anhand realer Audiodaten“, Diplomarbeit, Inst. elektronische Musik u. Akustik, Univ. Musik u. darstell. Kunst Graz, 2004.
- [4] M. Pflüger, „Modelle des peripheren Gehörs am Beispiel der menschlichen Lautheitsempfindung“, Dissertation, Inst. Nachrichtentechnik u. Wellenausbreitung, TU Graz, 1997.
- [5] H. Fletcher und W. A. Munson, „Loudness, Its Definition, Measurement and Calculation“, J. Acoust. Soc. America, Vol. V, Issue 2, S. 82 – 108, 10/1933.
- [6] Normalkurven gleicher Lautstärke, ISO 226, 2001.
- [7] B. C. J. Moore et al., „A Model for the Prediction of Thresholds, Loudness, and Partial Loudness“, J. Audio Eng. Soc., Vol. 45, No. 4, S. 224 – 240, 04/1997.
- [8] E. Skovenborg und S. Nielsen, „Evaluation of Different Loudness Models with Music and Speech Material“, vorgestellt auf 117th Audio Eng. Soc. Conv., San Francisco, USA, 2004, Conv. Paper 6234.
- [9] B. R. Glasberg und B. C. J. Moore, „A Model of Loudness Applicable to Time-Varying Sounds“, J. Audio Eng. Soc., Vol. 50, No. 5, S. 331 – 342, 05/2002.
- [10] S. Ferguson et al., „Comparing continuous subjective loudness responses and computational models of loudness for temporally varying sounds“, vorgestellt auf 129th Audio Eng. Soc. Conv., San Francisco, USA, 2010, Conv. Paper 8177.
- [11] Algorithms to measure audio programme loudness and true-peak audio level, Recommendation ITU-R BS.1770-2, 2011.
- [12] S. Weinzierl (Hrsg.), Handbuch der Audiotechnik. Berlin, Heidelberg: Springer, 2008.
- [13] Aussteuerungsmesser, 2. Aufl., IRT Technische Richtlinie der öffentlich-rechtlichen Rundfunkanstalten in der Bundesrepublik Deutschland Nr. 3/6, 1998.
- [14] T. Lund, „Level and Distortion in digital broadcasting“, EBU Tech. Review 310, S. 1 – 12, 04/2007.

- [15] Practical guidelines for distribution systems in accordance with EBU R 128, EBU – TECH 3344, 2011.
- [16] Loudness Alliance. (27.11.2013). Loudness Normalization: The Future of File-Based Playback (Version 1.0), White Paper

*PDF*

- . Siehe Anhang C.1, S. 72, online: [http://music-loudness.com/PDFs/Loudness\\_Alliance\\_White\\_Paper\\_final\\_v1.pdf](http://music-loudness.com/PDFs/Loudness_Alliance_White_Paper_final_v1.pdf)
- [17] Loudness normalisation and permitted maximum level of audio signals, EBU – Recommendation R 128, 2011.
- [18] G. Spikofski, „Lautstärkemessung im Rundfunk“ (aktual. Vortrag auf 23. Tonmeistertagung Leipzig 2004), Inst. Rundfunktechnik München FKT, S. 361 – 367, 07/2005.
- [19] K. Hansen, „Objective Reading of Loudness of a Sound Programme“, vorgestellt auf 100th Audio Eng. Soc. Conv., Kopenhagen, Dänemark, 1996, Preprint 4165.
- [20] S. Klar und G. Spikofski, „On levelling and loudness problems at television and radio broadcast studios“, vorgestellt auf 112th Audio Eng. Soc. Conv., München, Deutschland, 2002, Conv. Paper 5538.
- [21] I. Neoran und M. Shashua, „A Perceptive Loudness-Sensitive Leveler for Audio Broadcasting and Mastering“, vorgestellt auf 105th Audio Eng. Soc. Conv., San Francisco, USA, 1998, Conv. Paper 4852.
- [22] J. Riedmiller et al., „Practical Program Loudness Measurement for Effective Loudness Control“, vorgestellt auf 118th Audio Eng. Soc. Conv., Barcelona, Spanien, 2005, Conv. Paper 6348.
- [23] E. Vickers, „Automatic Long-Term Loudness and Dynamics Matching“, vorgestellt auf 111th Audio Eng. Soc. Conv., New York, USA, 2001, Conv. Paper 5495.
- [24] Practical guidelines for Production and Implementation in accordance with EBU R 128, EBU – TECH 3343, 2011.
- [25] Loudness Metering: „EBU Mode“ metering to supplement loudness normalisation in accordance with EBU R 128, EBU – TECH 3341, 2011.
- [26] Loudness Range: A measure to supplement loudness normalisation in accordance with EBU R 128, EBU – TECH 3342, 2011.
- [27] T. Painter und A. Spanias, „Perceptual Coding of Digital Audio“, Proc. IEEE, Vol. 88, No. 4, S. 451 – 513, 04/2000.
- [28] A. Magadam, „Evaluation of PEAQ for the Quality measurement of Perceptual Audio Encoders“, vorgestellt auf Audio Eng. Soc. 29th Int. Conf., Seoul, Korea, 2006.
- [29] Ch. M. Liu et al., „Compression Artifacts in Perceptual Audio Coding“, IEEE Trans. Audio, Speech, Lang. Process., Vol. 16, No. 4, S. 681 – 695, 05/2008.
- [30] J. Bortz, Statistik für Human- und Sozialwissenschaftler, 6. Aufl. Berlin, Heidelberg: Springer Medizin Verlag, 2005.

- [31] SourceForge. (09.06.2013). Download-Verzeichnis „Octave Windows binaries“

*HTML*

. Online: <http://sourceforge.net/projects/octave/files/Octave%20Windows%20binaries/>

- [32] Zerano. (14.07.2013). Zerano FFmpeg Windows builds

*HTML*

. Online: <http://ffmpeg.zerano.com/builds/>

- [33] Rare Wares. (14.07.2013). Download-Seite LAME Bundles

*HTML*

. Online: <http://www.rarewares.org/mp3-lame-bundle.php>

- [34] Telecommun. & Signal Process. Laboratory, McGill Univ., Kanada. (09.07.2013). Download-Verzeichnis „Audio File Programs and Routines package“ (AFsp)

*HTML*

. Online: <http://www-mmsp.ece.mcgill.ca/documents/Downloads/AFsp/>

- [35] The GNU-Projekt. (04.07.2013). Octave-Homepage

*HTML*

. Online: <http://www.gnu.org/software/octave/>

- [36] GNU Octave – A high-level interactive language for numerical computations, Edition 3 for Octave version 3.2.4, The GNU Project, 2007.

- [37] Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers, European Standard (Telecommun. series) Final draft ETSI EN 300 401 V1.4.1, 2006.

- [38] Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream, Technical Specification ETSI TS 101 154 V1.11.1, 2012.

- [39] The FFmpeg-Project. (01.07.2013). FFmpeg documentation

*HTML*

. Siehe Anhang C.2, S. 76, online: <http://www.ffmpeg.org/ffmpeg.html>

- [40] ARD, „Signalquellen und AV-Formate Stand: 14.08.2012“, unveröffentlicht, siehe Anhang B.2.

- [41] The LAME Project. (01.07.2013). Detailed command line switches

*HTML*

. Siehe Anhang C.3, S. 81, online: <http://lame.cvs.sourceforge.net/viewvc/lame/lame/doc/html/detailed.html>

- [42] Method for objective measurements of perceived audio quality, RECOMMENDATION ITU-R BS.1387-1, 2001.
- [43] W. C. Treurniet und G. A. Soulodre, „Evaluation of the ITU-R Objective Audio Quality Measurement Method“, J. Audio Eng. Soc., Vol. 48, No. 3, S. 164 – 173, 03/2000.
- [44] P. Kabal, „An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality (Version 2)“, Dept. Elect. & Comput. Eng., McGill Univ., Sci. Rep., 2002.
- [45] AES Staff, „Measuring and Predicting Perceived Audio Quality“, J. Audio Eng. Soc., Vol. 53, No. 5, S. 443 – 448, 05/2005.
- [46] Telecommun. & Signal Process. Laboratory, McGill Univ., Kanada. (09.07.2013). Kurzbeschreibung PQevalAudio  
*HTML*  
. Siehe Anhang C.4, S. 88, online: <http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/PQevalAudio.html>
- [47] Telecommun. & Signal Process. Laboratory, McGill Univ., Kanada. (09.07.2013). Kurzbeschreibung ResampAudio  
*HTML*  
. Siehe Anhang C.5, S. 90, online: <http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/ResampAudio.html>
- [48] Sound Quality Assessment Material – recordings for subjective tests, Users' handbook for the EBU SQAM CD, EBU – TECH 3253, 2008.
- [49] G. Wagner, „Untersuchung zur automatisierten Lautheitsnormalisierung nach EBU R128 für Anwendungen im Hörfunk“, Bachelorarbeit, Inst. Medien- u. Phototechnik, FH Köln, 2013.
- [50] ARD / ZDF, „ARD DVB Audio-Setup ARD\_ZDF allgemein“, unveröffentlicht, siehe Anhang B.1.
- [51] S. Preihs et al., „Comparison of Filter Bank Design Algorithms for use in Low Delay Audio Coding“, vorgestellt auf 133rd Audio Eng. Soc. Conv., San Francisco, USA, 2012, Conv. Paper 8789.
- [52] T. Thiede et al., „PEAQ – The ITU Standard for Objective Measurement of Perceived Audio Quality“, J. Audio Eng. Soc., Vol. 48, No. 1/2, S. 3 – 29, 01-02/2000.

# Tabellenverzeichnis

2.1	Koeffizienten der $K$ -Filter für das von der ITU empfohlene Lautheitsmessverfahren in IIR-Struktur 2. Ordnung bei Abtastfrequenz 48000 Hz (nach [11, S. 6 – 7]) . . . . .	16
2.2	Gewichtungsfaktoren der Kanalsummation bei 5.1-Surround für das von der ITU empfohlene Lautheitsmessverfahren (nach [11, S. 9]) . . . . .	17
3.1	Beschreibung der fünf Beeinträchtigungsstufen des Objective Difference Grades (aus [44, S. 53]) . . . . .	32
3.2	Für die Untersuchung verwendete Testbeispiele aus EBU-SQAM; Sprache 2-kanalig monophon, alle übrigen stereophon; Abkürzungen siehe Legende auf S. 35 . . . . .	36
3.3	Sequenz der Testbeispiele mit Startposition . . . . .	43
3.4	Gemessene Pegelwerte der Testbeispiele . . . . .	45
3.5	Statistische Kennwerte der gemessenen ODG-Differenzen . . . . .	46

# Abbildungsverzeichnis

2.1	Kurven gleicher Lautstärke (Isophone, aus [6], übernommen aus [3, S. 11])	4
2.2	Kurven gleicher Lautstärke verschiedener Versuchsreihen (aus [6], übernommen aus [3, S. 12])	6
2.3	Sinussignal vor (Linie) und nach der Digitalisierung (Punkte) mit einem True Peak Level oberhalb des Sample Peak Levels (aus [14, S. 2])	11
2.4	Bewertungskurven: A (für Schalldruckpegel von 40 dB SPL), B (für Schalldruckpegel von 60 dB SPL), C (für Schalldruckpegel von 80 dB SPL), D (für Fluglärm), M (für Kinoton) und RLB (siehe Kap. 2.2.4); aus [8, S. 7]	14
2.5	Blockschaltbild des von der ITU empfohlenen Lautheitsmessverfahrens (aus [11, S. 5])	15
2.6	Frequenzgang der 1. Filterstufe zur Modellierung des Kopfes für das von der ITU empfohlene Lautheitsmessverfahren (aus [11, S. 5])	16
3.1	Blockschaltbild des Algorithmus zur objektiven Messung der wahrgenommenen Tonqualität (aus [42, S. 18])	33
3.2	In Octave errechneter Betragsfrequenzgang der 1. Filterstufe (Shelving) für die Lautheitsmessung	39
3.3	In Octave errechneter Betragsfrequenzgang der 2. Filterstufe (RLB) für die Lautheitsmessung	39
3.4	Histogramm über Häufigkeitsverteilung der gemessenen ODG-Differenzen für MP2	47
3.5	Histogramm über Häufigkeitsverteilung der gemessenen ODG-Differenzen für MP3	47

# Anhang A

## Octave-Quellcode

### A.1 Funktion zur Ermittlung der Programmlautheit (Datei loudness.m)

```
function LKG = loudness(x, fs)

% LKG = loudness(x [, fs])
% Calculates the loudness level of the signal x
% according to recommendation ITU-R.BS 1770-2.
% Up to 5 channels are supported,
% where the columns of the matrix x must have the order
% [left, right, centre, left surround, right surround].
% One channel audio is treated as if played back on only one speaker;
% for monophonic signals to be played on two speakers,
% (loudness + 3dB) x has to be passed as two-columns matrix [x, x].
% By fs a sampling frequency can be specified; default is 48000 Hz.
% Note that the recommendation only exists
% for a sampling frequency of 48000 Hz,
% so for other sampling frequencies
% a resampling will have to be done, taking some additional time.
%
% See also: ppm, truepeak, level
1; % Ende des Hilfetexts
```

```
% Konstanten

Tg = 0.4; % Blocklänge für Gating in [s]
overlap = 0.75; % Überlappungsfaktor der Blöcke
Gamma_a = -70; % absoluter Threshold für 1. Gate in [LUFS]
G = [1; 1; 1; 1.41; 1.41]; % Gewichtungsfaktoren für Kanäle
c = 0.691; % Korrekturkonstante

% Bei anderer Sampling-Frequenz als 48000 Hz, Resampling

if ((nargin > 1) && (fs ~= 48000))
    x = resample(x, 48000, fs);
end % if

% Länge und Kanäle des Signals

[samples, channels] = size(x);

% Falls Kanäle keine Spaltenvektoren, Transposition:
if (channels > samples)
    x = x.';
    [samples, channels] = size(x);
end % if

% K-Filterung

% 1. Filterstufe (Shelving) zur Modellierung eines runden Kopfes:
b0 = 1.53512485958697;
b1 = -2.69169618940638;
b2 = 1.19839281085285;
a1 = -1.69065929318241;
a2 = 0.73248077421585;
xK = filter( [b0, b1, b2], [1, a1, a2], x );

% 2. Filterstufe (RLB-Gewichtung; Hochpass):
b0 = 1;
b1 = -2;
b2 = 1;
a1 = -1.99004745483398;
a2 = 0.99007225036621;
y = filter( [b0, b1, b2], [1, a1, a2], xK);
```

```

% Blockweise Leistungsberechnung für jeden Kanal

% Konstanten:
blockSamples = round(Tg * 48000);
blockStep = round(blockSamples * (1 - overlap));
% Anzahl der Blöcke:
blocks = uint64(floor( (samples - blockSamples) / blockStep ) ) + 1;

% Initialisierung einer Matrix für die Leistungswerte
% aller Blöcke auf allen Kanälen:
z = zeros(blocks, channels, 'single');

% Schleife mit Indexvektor für Blöcke:
for j = 0 : (blocks - 1)
    % Leistung eines Blocks:
    z(j + 1,:) = 1 / blockSamples * sumsq( y(j * blockStep + 1
        : j * blockStep + blockSamples, :) );
end % for

% Gating

% 1. Gate mit absolutem Threshold Gamma_a:
Jg = ( -c + 10 * log10( z * G(1:channels) ) ) > Gamma_a;

% relativer Threshold des 2. Gates
% (10 LU unter Gesamtlautheit mit 1. Gate):
Gamma_r = -c + 10 * log10( 1 / sum(Jg) * Jg.' *
    ( z * G(1:channels) ) ) - 10;

% 2. Gate mit relativem Threshold Gamma_r:
Jg = ( -c + 10 * log10( z * G(1:channels) ) ) > Gamma_r;

% Gesamtlautheit

LKG = -c + 10 * log10( 1 / sum(Jg) * Jg.' * ( z * G(1:channels) ) );

end % function

```

## A.2 Funktion zur Ermittlung des Maximum True Peak Levels (Datei truepeak.m)

```
function out = truepeak(x, fs)

% L = truepeak(x [, fs])
%
% Calculates the true peak level(s) of signal x
% using an oversampling to 192000 Hz, where fs is the sampling
% frequency of the input signal (default 48000 Hz).
% Note that calculation can take a while for longer signals
% (about 35 s for 1 min. of stereo audio).
1; % Ende des Hilfetexts

% Standardwert für Sampling-Frequenz

if (nargin == 1) fs = 48000 end % [Hz]

% Konvertierung des Eingangssignals in single-format
% zur besseren Speichernutzung

x = single(x);

% Samples und Kanäle des Eingangssignals;
% wenn Kanäle zeilenförmig, Transposition

[N, channels] = size(x);
if (channels > N)
    x = x.';
    [N, channels] = size(x);
end % if

% Konstanten

fup = 192000; % [Hz]
blocksize = round(480000000000 / fup);
```

```

% Wegen Speicherplatzmangels, block- und kanalweises Oversampling

% Kanalweise True Peak-Berechnung:
L = ones(1, channels) * -inf;
for i = 1 : channels

    % Zerlegung längerer Signale in Blöcke:
    for n = 1 : ceil(N / blocksize)

        % zu single-Verarbeitung modifizierte Funktion resample:
        xup = resamplesg( x((n-1) * blocksize + 1 :
            min(N, n * blocksize), i), fup, fs);
        % Wenn Spitzenpegel des Blocks höher ist als die bisherigen,
        % neuen übernehmen:
        L(i) = max(L(i), ppm( xup, 'samplepeak', fup) );

    end % for

end % for

% Ausgabe

out = L;

end % function

```

### A.3 Funktion zur Ermittlung des maximalen Spitzenpegels (Datei ppm.m)

```

function out = ppm(x, type, fs)

% L = ppm(x [, type] [, fs])
%
% Programme Peak Meter for maximum level of signal x,
% where type specifies the integration time for the measurement
% and fs the sampling frequency (default 48000 Hz).
%

```

```
% Possible values for type are (upper or lower case letters):
% 'SAMPLEPEAK', 'SAMPLE',    maximum sample-peak level (not true peak!)
% 'PEAK', 'SP', 'S', 'P', 0
% 'VOLUMEUNIT', 'VU'        the VU-meter with integration time 300 ms
% 'RMS'                      root mean square value
%                            (integrating the whole signal)
% 'CREST', 'C'              crest factor
%                            (ratio of max. sample-peak level and RMS)
% a number 0 < type < 32000 integration time in ms
% Note that the VU-meter still works very slowly at the moment.
%
% See also: truepeak, loudness, level
1; % Ende des Hilfetexts

% Konvertierung von 64- in 32-bit-Samples

x = single(x);

% Länge und Kanäle des Signals

[samples, channels] = size(x);

% Falls Kanäle keine Spaltenvektoren, Transposition:
if (channels > samples)
    x = x.';
    [samples, channels] = size(x);
end % if

% Festlegen von Standardwerten

...ausgelassen...

% Berechnung der Leistungswerte mit gewünschter Integrationszeit

% Bei längeren Inntegrationszeiten Summation mit for-Schleife:
if (type > 0.05 * samples * channels)

    P = zeros(samples, channels, 'single');
    for j = type : samples
        P(j, :) = 1 / type * sumsq( x(j - type + 1 : j, :) );
    end % for
```

```
% Bei kürzeren Integrationszeiten, Faltungsoperation:
else

    P = 1 / type * filter( ones(type, 1, 'single'), 1, x.^2);

end % if

% Berechnung und Ausgabe des maximalen logarithmischen Pegels in dBFS
out = level( sqrt( max( [P(type : samples, :); zeros(1,channels)]))));

end % function
```

## A.4 Funktion zur Berechnung und Änderung des logarithmischen Pegels (Datei level.m)

```
function out = level(x, L, dBFS_type, fs)

% L = level(x)
%
% Returns the logarithmic level in dBFS for each element of x.
% This is equivalent to: L = 20 * log10(abs(x))
% Note that level(x) assumes fractional numbers as input,
% i. e., x = +/-1 = 0 dBFS.
% Moreover, level(x) expects time signal values,
% so if x contains power values, the result has to be divided by 2.
%
% y = level(x, L)
%
% Returns the linear value(s) of x, increased by L dB.
% This can be used to amplify/attenuate a signal by dB values.
%
% y = level(x, L, dBFS_type [, fs])
%
% Returns the linear value(s) of x, normalised to L dBFS,
% where the target level type of x is specified by dBFS_type
% and the sampling frequency of the input signal by fs
% (default is 48000 Hz).
```

```
%  
% Possible values for dBFS_type are (upper or lower case letters):  
% 'SAMPLEPEAK', 'SAMPLE',    maximum sample-peak level (not true peak!)  
% 'PEAK', 'SP', 'S', 'P', 0  
% 'VOLUMEUNIT', 'VU'        the VU-meter with integration time 300 ms  
% 'RMS'                      root mean square value  
%                             (integrating the whole signal)  
% 'LOUDNESS', 'L'           loudness level in LUFS  
%                             according to ITU-R.BS 1770-2  
% a number 0 < type < 32000 integration time in ms  
% Note that the VU-meter still works very slowly at the moment.  
%  
% See also: truepeak, loudness, ppm  
1; % Ende des Hilfetexts
```

```
switch nargin
```

```
    case {1} out = round( 200 * log10(abs(x)) ) / 10;
```

```
    otherwise
```

```
        if (nargin >= 3)  
            if (nargin < 4) fs = 48000 end % [Hz]  
  
            switch upper(dBFS_type)  
                case {'LOUDNESS' 'L'}  
                    L = L - loudness(x, fs);  
                otherwise L = L - max(ppm(x, dBFS_type, fs));  
            end % switch  
        end % if
```

```
        out = x * 10^(L / 20);
```

```
    end % switch
```

```
end % function
```

## A.5 Analyse der MP2-codierten Sequenzen (Datei PQevalAudio\_blockwise\_MP2.m)

Der folgende Quellcode zeigt die Analyse mittels PQevalAudio am Beispiel der MP2-Dateien. Die Analyse für MP3 erfolgte analog. Aus Platzgründen werden die Dateipfade abgekürzt.

```
% Initialisierungen

PQevalAudio = 'c:\...\PQevalAudio.exe --info=0'; % Befehlszeilenaufruf
blocklength = 4800; % [Samples] = 0,1 s
samples_channels = wavread('c:\...\sequenz_original.wav', 'size');
samples = samples_channels(1);
clear samples_channels;
blocks = ceil(samples / blocklength);
ODG_MP2_original = zeros(blocks, 1);
ODG_MP2_23LUFSS = zeros(blocks, 1);

% Analyse mittels PQevalAudio

for n = 1 : blocks

    % String zur Angabe der Blockgrenzen in Samples:
    limits = [ '--limits=', num2str((n - 1) * blocklength, 10), ':' ];
    if n == blocks
        limits = [ limits, num2str( samples - 1 , 10) ];
    else
        limits = [ limits, num2str( n * blocklength - 1 , 10) ];
    end % if

    % Aufruf von PQevalAudio.exe,
    % Speicherung der Bildschirmausgabe in output_original/23LUFSS:
    [status, output_original] = system([ PQevalAudio, limits,
        ' C:\...\sequenz_original.wav
        C:\...\sequenz_original.mp2.wav' ]);
    [status, output_23LUFSS] = system([ PQevalAudio, limits,
        ' C:\...\sequenz_23LUFSS.wav
        C:\...\sequenz_23LUFSS.mp2.wav' ]);

    % Extraktion der ODGs aus Bildschirmausgabe-Strings:
    ODG_MP2_original(n) = str2double( output_original(end-7 : end) );
    ODG_MP2_23LUFSS(n) = str2double( output_23LUFSS(end-7 : end) );

end % for
```

```
% Löschen der Hilfsvariablen

clear PQevalAudio blocklength samples limits n;

% Berechnung der Differenz jedes ODG der Sequenz -23 LUFS
% zum zeitlich entsprechenden ODG der Sequenz in orig. Aussteuerung

DeltaODG_MP2 = ODG_MP2_23LUFS - ODG_MP2_original;

% Bildung des arithmetischen Mittel-, des Median- und des Modalwertes
% sowie der Standardabweichung über alle ODG-Differenzen

DeltaODG_MP2_mean = mean(DeltaODG_MP2)
DeltaODG_MP2_median = median(DeltaODG_MP2)
DeltaODG_MP2_mode = mode(DeltaODG_MP2)
DeltaODG_MP2_sigma = std(DeltaODG_MP2)

% Erstellung eines Histogramms der ODG-Differenzen
% mit Rasterung 0,1 im Intervall [-4; +4]

hist(DeltaODG_MP2, [-3.95 : 0.1 : 3.95], 100);
xlabel('DeltaODG\_MP2 = ODG\_MP2\_23LUFS - ODG\_MP2\_original');
ylabel('Häufigkeit [%]');

% Speichern der Grafik:

print('c:\...\DeltaODG_Histogramm_MP2.jpg', '-djpg');

% Signifikanzprüfung mittels rechtsseitiger t-Tests

p_n_MP2 = t_test(DeltaODG_MP2, -0.1, '>')
p_p_MP2 = t_test(DeltaODG_MP2, 0.1, '>')

% Speichern der Ergebnisse

save c:\...\ODG_MP2.txt
```

# Anhang B

## Interne Übersichten der ARD zu verwendeten Verfahren und Datenraten bei der Quellcodierung

### B.1 Digitales Fernsehen (DVB)

ARD DVB Audio-Setup ARD\_ZDF allgemein

Transponder		Tp xxx				
SDT	Service type	1 (SDTV)				
	Service ID	28xxx				
	Name	Programm xxx				
PMT	Audio PID lfd. Nr.	1	2	3 (ZDF)	3 (ARD) / 4 (ZDF)	
	Stream type	3	3	3	6	
	ISO 639 Language descriptor	ISO 639 Language code	deu	mis	qaa	deu
		Audio type	0	0	0	0
	AC-3 descriptor	Comp.type flag	X			0
		BSID flag				0
		Main ID flag				0
		ASVC flag				0
	Supplementary audio descriptor (EN 300 468 Annex J)	mix type	X			1
		editorial classification				1
language present		1				
ISO 639 Language code		deu				
Stream identifier descriptor	Component tag	2	4	7	17 (ARD) / 5 (ZDF)	
EIT	Component descriptor (dynamisch) (u. a. EN 300 468 Annex J)	Content&type (dynamisch)	0x203	0x203 0x248	0x203 0x442 0x444	
		Language	deu	deu	QAA deu	
		Text (dynamisch)	Stereo	ohne Audiodeskription mit Audiodeskription	Originalton Dolby Digital 2.0 Dolby Digital 5.1	
		Text (dynamisch)				
ES Elementary stream	Audio Codec	MPEG-1 L2	MPEG-1 L2	MPEG-1 L2	AC-3	
	Dolby Digital	X			2/0 oder 3/2	
	Dialog Level / dB				-22 (ARD) / -27 (ZDF)	
	Dolby Surround Mode				not indicated	
	Preferred Stereo Downmix				Lo/Ro	
	PCM-Audio Mode	stereo	stereo	stereo		
	Bitrate in kBit/s	256/192	192	256	448/384/256 ...	
	Sample rate in kHz	48	48	48	48	
	Copyright	0	0	0		
	Original	1	1	1		
Stream ID	192 (192 ... 223 zulässig)				189	

## B.2 Hörfunk und Fernsehen im Internet

### Signalquellen

#### 1) On Demand - Fernsehen

LRA	RB	RBB	SR	SWR	WDR
<b>Quellenformat</b>	DV25, DVCPRO50	IMX50 D10 DVB-S über Avid MPG2 5 MBit MAZ IMX50 oder MPG2 50 WMV 400kBit	IMX50 D10 MPEG-2 TS mit 5 Mbit/s	MPEG-2 TS mit 5 Mbit/s	SDI, ASI, DigiBeta,IMX, MPEG2 8Mbit/s
<b>AV-Format nach ARD-Katalog</b>	1.10, 1.14, 1.20, 1.24, 1.27, 1.28, 1.54, 1.58, 1.65, 1.69, 2.10, 2.14, 2.22, 2.26, 90.4	ARD-Web-M MP4 termID FormatCS2010:1.24 ARD-Web-M MP4 termID FormatCS2010:1.54 ARD-Web-M MP4 termID FormatCS2010:90.3  ARD-Web-L MP4 termID FormatCS2010:1.20 ARD-Web-L MP4 termID FormatCS2010:1.58 ARD-Web-L MP4 termID FormatCS2010:90.4	Web S Web M	ARD-Web-M MP4 termID FormatCS2010:1.24 ARD-Web-M MP4 termID FormatCS2010:1.54  ARD-Web-L MP4 termID FormatCS2010:1.20 ARD-Web-L MP4 termID FormatCS2010:1.58  ARD-Web-M WMV termID FormatCS2010:1.10	Web S, termID 1.27 Web M, termID 1.24 Web L, termID 1.20
<b>Videocodec</b>	H.264, WMV	H.264, WMV	H.264	H.264, WMV	H.264
<b>Audiocodec</b>	AAC, WMA	AAC, AAC-HEV1	AAC	AAC, AAC-HEV1, WMA	AAC, AAC-HEV1
<b>Streamingverfahren</b>	RTMP, mms, iOS-HTTP	RTMP, MMS, HTTP	RTMP	RTMP, HTTP, MMS	rtmp

#### 2) On Demand - Hörfunk

LRA	RB	RBB	SR	SWR	WDR
<b>Quellenformat</b>	MUSICAM (MP2), WAV	Digas/ MPG2 256kBit	MUSICAM (MP2), MP3, WAV	Digas/ MPG2 256kBit (?)	AES/EBU
<b>AV-Format nach ARD-Katalog</b>	3.3, 3.4, 3.5, 3.6, 3.7	ARD-Web-M MP3 termID FormatCS2010:3.3 ARD-Web-M MP3 termID FormatCS2010:3.14 ARD-Web-M MP3 termID FormatCS2010:3.4 ARD-Web-S MP3 termID FormatCS2010:3.6	Web M	ARD-Web-M MP3 termID FormatCS2010:3.3	Web M, termID 3.14
<b>Audiocodec</b>	MP3, WMA	MP3	MP3		MP3
<b>Streamingverfahren</b>	RTMP, mms, iOS-HTTP	RTMP, HTTP	RTMP	RTMP, HTTP	trmp

#### 3) Live - Fernsehen

LRA	RB	RBB	SR	SWR	WDR
<b>Quellenformat</b>	SDI	DVB-S	SDI	XXX	SDI
<b>AV-Format nach ARD-Katalog</b>	1.19, 1.23	ARD-Web-M MP4 termID FormatCS2010:1.23 ARD-Web-L MP4 termID FormatCS2010:1.19	Web M und L	XXX	s. Anlage
<b>Videocodec</b>	H.264	MP4	H.264	XXX	H.264, WMV9
<b>Audiocodec</b>	AAC	AAC, AAC-HEV1	AAC	XXX	AAC, AAC-HEV1, WMA9
<b>Streamingverfahren</b>	RTMP	RTMP	RTMP	XXX	rtmp, mms, http (HDS)

#### 4) Live - Hörfunk

LRA	RB	RBB	SR	SWR	WDR
<b>Quellenformat</b>	AES/EBU	AES/EBU / MPG2 / 256 kBit	AES/EBU	AES/EBU	AES/ EBU
<b>AV-Format nach ARD-Katalog</b>	3.9, 3.10, 3.15	ARD-Web-S MP4 termID FormatCS2010:3.12 ARD-Web-M MP4 termID FormatCS2010:3.10	Web M MP3 64 kBit/s MP3 128 kBit/s	ARD-Web-S MP3 termID FormatCS2010:3.12 ARD-Web-M MP3 termID FormatCS2010:3.10 ARD-Web-S WMA termID FormatCS2010:3.11 ARD-Web-M WMA termID FormatCS2010:3.S 5.1 MP3-kompatibel	Web M, termID 3.13 Web M, ähnlich termID 3.9 aber 64 kbit/s
<b>Audiocodec</b>	MP3, WMA	MP3	MP3	MP3, WMA	MP3, WMA9
<b>Streamingverfahren</b>	HTTP, mms	HTTP	HTTP	HTTP, MMS	rtmp, mms

## Anhang C

### Online-Quellen

Im Folgenden werden die in dieser Arbeit referenzierten Dokumente auszugsweise eingefügt, die ausschließlich im Internet verfügbar sind.

## C.1 Loudness Alliance [16]

# Loudness Normalization: The Future of File-Based Playback

Music Loudness Alliance<sup>1</sup>  
<http://music-loudness.com>  
Version 1.0, July 2012

Today, playback on portable devices dominates how music is enjoyed.<sup>2 3</sup> A large portion of songs present on the average music player come from online music services, obtained either by per-song purchase or by direct streaming. The listener often enjoys this music in shuffle play mode, or via playlists.

Playing music this way poses some technical challenges. **First**, the sometimes tremendous differences in loudness between selections requires listeners to adjust volume. **Second**, the reduction in sound quality of music production over the years due to a loudness war. The art of dynamic contrast has almost been lost because of the weaknesses of current digital systems. **Third**, the potential damage to the ear caused by these loudness differences and a tendency towards higher playback levels in portable listening especially when using earbuds.

## The Three Challenges

### Loudness Differences

In digital audio, the maximum (peak) audio modulation has a hard ceiling that cannot be crossed. Digital audio tracks are routinely peak normalized. This results in tremendous loudness differences from track to track because the **peak level** of a signal is not representative of its subjective loudness. Rather, the listener perceives loudness according to the **average energy** of the signal. Because of the widespread practice of peak normalization, program producers apply severe compression, limiting and clipping techniques to the audio. This removes the original peaks and allows normalization to amplify the signal increasing its average energy. This has resulted in a loudness war with large loudness differences between newer and older material

---

<sup>1</sup> The Music Loudness Alliance is a group of technical and production experts led by Florian Camerer and consists of audio engineers Eelco Grimm, Kevin Gross, Bob Katz, Bob Ludwig and Thomas Lund. We can be contacted via Florian's email: [florian.camerer@orf.at](mailto:florian.camerer@orf.at).

<sup>2</sup> Steve Guttenberg (September 2009). *Poll: Where do you listen to music?* CNet.  
<http://www.webcitation.org/68yrk7e5h>

<sup>3</sup> Reineke Reitsma (July 2010). *The Data Digest: Which Music Devices Do People Use?* Forrester.  
<http://www.webcitation.org/68yrr5gO2>

## Loudness Alliance – Loudness Normalization [16]

and different genres. When older recordings are included in a playlist with new material, the listener experiences noticeable jumps in loudness from track to track requiring frequent adjustments in playback level. The differences can be as large as 20 dB. The same problem occurs when different musical genres share a playlist. Portable device listening is therefore not the comfortable experience it could be, and computer playback exhibits some of the same problems.

### Restoration of Sound Quality to our Recorded Legacy

In the practice commonly referred to as the “loudness war”, many artists, recording engineers and record labels strive to make their recordings sound louder so they will stand out compared to others. The aggressive dynamic range compression used to produce loud recordings reduces peak-to-average-energy ratio. The effect has been that the important artistic and narrative tool of dynamic contrast has almost totally disappeared in modern music production.

The result of this pressure to be louder is that the steps of the production process, recording, mixing and mastering, produce masters that incorporate several generations of digital processing which can cumulate clipping and alias products. This distortion is exacerbated when the product is finally encoded to a lossy medium like AAC.<sup>4</sup> Cumulative distortion also leads to further significant distortion being added during distribution or playback.<sup>5</sup> This is fatiguing to the ear, which turns off some listeners and may even be the cause of reduced sales of contemporary music.<sup>6 7</sup> This reduction in signal quality and dynamic range amounts to a removal of the very parts of the sound which make programs sound interesting.

By switching from peak normalization to loudness normalization as a default in playback media, producers who wish to mix and master programs with wide dynamic range and without distortion can do so without fear that their programs will not be heard as loudly as the ‘competition’. Loudness normalization also permits older, more dynamic material to live alongside the newer, which will allow listeners to appreciate the sound qualities of more dynamic recordings and permit them to mix genres and recording styles.

### Hearing Damage

High playback levels, whether achieved by accident, chosen per personal preference or to overcome ambient noise, are a potential source of hearing damage.<sup>8</sup> This is especially true for headphones and earbuds which, due to their close proximity to the eardrums, require relatively

---

<sup>4</sup> <http://www.apple.com/itunes/mastered-for-itunes/>

<sup>5</sup> Nielsen & Lund (May 2003), *Overload in Signal Conversion*. Audio Engineering Society.

<sup>6</sup> Greg Milner (2010). *Perfecting Sound Forever: An Aural History of Recorded Music*. Faber & Faber. ISBN 978-0865479388.

<sup>7</sup> Earl Vickers (May 2011). *The Loudness War: Do Louder, Hypercompressed Recordings Sell Better?* Audio Engineering Society. <http://www.aes.org/e-lib/browse.cfm?elib=15934>

<sup>8</sup> Scientific Committee on Emerging and Newly Identified Health Risks (2008), *Potential health risks of exposure to noise from personal music players and mobile phones including a music playing function*, European Commission. <http://www.webcitation.org/68yrYudQh>

## Loudness Alliance – Loudness Normalization [16]

little power to reach damaging levels. In the past some European countries have attempted to address hearing damage by legislating maximum peak output level for portable players. The net result is that it is difficult to enjoy old recordings or dynamic genres like classical music at sufficient loudness on these output-limited devices. Unfortunately this has increased pressure on mastering engineers to remove dynamic peaks from tracks in order to provide loud enough playback levels for the restricted peak level. Again, peak output level is not directly connected to perceived loudness. It is also not used as a predictor of hearing damage potential in international law. Instead the integrated level over a certain period of time should be used.

### An Integrated Solution

#### ITU Loudness Normalization

There is a solution for problems of inconsistent playback, the loudness war, hearing damage and the sound quality issues. This solution is founded in the massive adoption of file-based music consumption in all kinds of formats. All playback devices and music servers are effectively computers that may analyze the average perceptual energy of a file and adjust its playback level accordingly. For international broadcasting, the ITU-R<sup>9</sup> BS.1770-2 standard for loudness measurement has recently been developed.<sup>10</sup> It defines the equivalent loudness of an audio signal as its **LUFS** level, meaning *Loudness Units relative to Full Scale*.<sup>11</sup> BS.1770-2 does a very good job in predicting subjective loudness.<sup>12</sup> Loudness normalization based upon BS.1770-2 is being rolled out worldwide for television broadcast.<sup>13</sup> Apple has successfully implemented loudness normalization in its **Sound Check** algorithm for iTunes and supported portable players. A similar open system known as **ReplayGain**<sup>14</sup> is available for other players. The adoption of BS.1770-2 by these systems would be advantageous in the sense that music normalization would then be based on one international standard for loudness measurement.

#### ON by Default

Listener experience will generally improve when a loudness normalization algorithm is turned **ON by default**. This will also facilitate compliance with regulations to prevent hearing loss. **Loudness normalization ON by default** would also help to put an end to the “loudness war” in music production. In order for playback devices not to drop in loudness level compared to what listeners had been familiar with, we suggest a different form of system level control that we call **Normalized Level Control**.

<sup>9</sup> International Telecommunication Union, a United Nations agency, Radiocommunication Division

<sup>10</sup> ITU (March 2011) *Recommendation ITU-R BS.1770-2, Algorithms to measure audio programme loudness and true-peak audio level*. <http://www.webcitation.org/68YrRmvrC>

<sup>11</sup> A step of 1 LU (Loudness Unit) is equivalent to 1 dB.

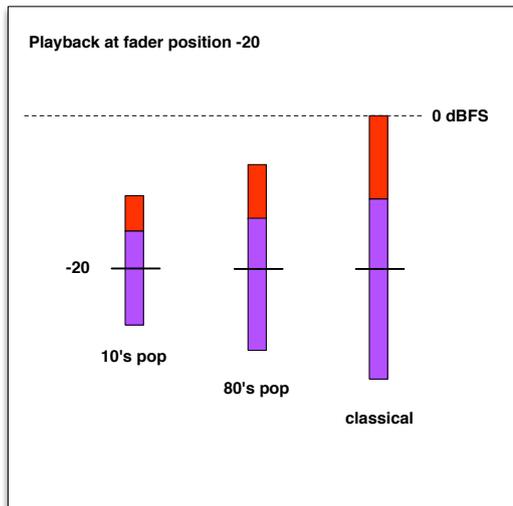
<sup>12</sup> Paul Nygren (2009). *Achieving equal loudness between audio files*. KTH Royal Institute of Technology. <http://www.webcitation.org/68ys6rwC0>

<sup>13</sup> EBU R 128 in Europe. CALM act and ATSC A/85 in the U.S.

<sup>14</sup> <http://www.replaygain.org>

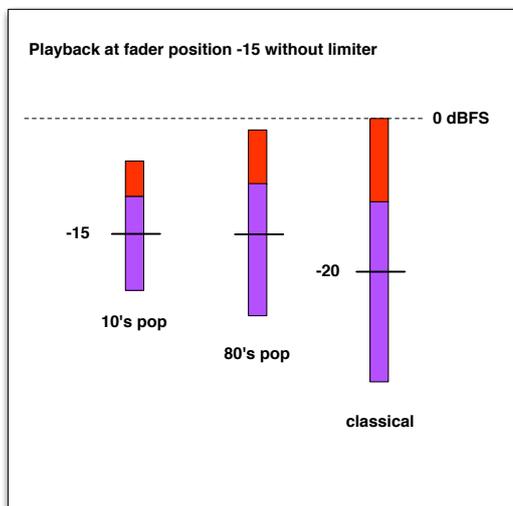
## Loudness Alliance – Loudness Normalization [16]

Now, we raise the level control to position -20:



Even when turning up NORM-L by 5 dB, the classical material is peaking to the maximum level but does not clip. The other two files still have ample headroom.

However, now the fader position is set to -15:



This setting would cause the classical music to clip, so NORM-L constrains its normalization to prevent this. In this case the file will be played back effectively at -20 although the fader is set to -15. The classical music plays back 5 dB quieter than the other two examples, but is not clipped.

## C.2 FFmpeg [39]

FFmpeg documentation :



News | About | Download | Documentation | Bug Reports | Contact | Donations | Consulting | Projects | Legal | Security | FATE

### ffmpeg Documentation

#### Table of Contents

- [1. Synopsis](#)
- [2. Description](#)
- [3. Detailed description](#)
  - [3.1 Filtering](#)
    - [3.1.1 Simple filtergraphs](#)
    - [3.1.2 Complex filtergraphs](#)
  - [3.2 Stream copy](#)
- [4. Stream selection](#)
- [5. Options](#)
  - [5.1 Stream specifiers](#)
  - [5.2 Generic options](#)
  - [5.3 AVOptions](#)
  - [5.4 Main options](#)
  - [5.5 Video Options](#)
  - [5.6 Advanced Video Options](#)
  - [5.7 Audio Options](#)
  - [5.8 Advanced Audio options:](#)
  - [5.9 Subtitle options:](#)
  - [5.10 Advanced Subtitle options:](#)
  - [5.11 Advanced options](#)
  - [5.12 Preset files](#)
- [6. Tips](#)
- [7. Examples](#)
  - [7.1 Preset files](#)
  - [7.2 Video and Audio grabbing](#)
  - [7.3 X11 grabbing](#)
  - [7.4 Video and Audio file format conversion](#)
- [8. See Also](#)
- [9. Authors](#)

#### 1. Synopsis

```
ffmpeg [global_options] {[input_file_options] -i 'input_file'} ... {[output_file_options]
'output_file'} ...
```

#### 2. Description

ffmpeg is a very fast video and audio converter that can also grab from a live audio/video source. It can also convert between arbitrary sample rates and resize video on the fly with a high quality polyphase filter.

ffmpeg reads from an arbitrary number of input "files" (which can be regular files, pipes, network streams, grabbing devices, etc.), specified by the `-i` option, and writes to an arbitrary number of output

<http://www.ffmpeg.org/ffmpeg.html>

## FFmpeg documentation [39]

FFmpeg documentation :

"files", which are specified by a plain output filename. Anything found on the command line which cannot be interpreted as an option is considered to be an output filename.

Each input or output file can, in principle, contain any number of streams of different types (video/audio/subtitle/attachment/data). The allowed number and/or types of streams may be limited by the container format. Selecting which streams from which inputs will go into which output is either done automatically or with the `-map` option (see the Stream selection chapter).

To refer to input files in options, you must use their indices (0-based). E.g. the first input file is 0, the second is 1, etc. Similarly, streams within a file are referred to by their indices. E.g. 2:3 refers to the fourth stream in the third input file. Also see the Stream specifiers chapter.

As a general rule, options are applied to the next specified file. Therefore, order is important, and you can have the same option on the command line multiple times. Each occurrence is then applied to the next input or output file. Exceptions from this rule are the global options (e.g. verbosity level), which should be specified first.

Do not mix input and output files – first specify all input files, then all output files. Also do not mix options which belong to different files. All options apply ONLY to the next input or output file and are reset between files.

- To set the video bitrate of the output file to 64 kbit/s:

```
ffmpeg -i input.avi -b:v 64k -bufsize 64k output.avi
```

- To force the frame rate of the output file to 24 fps:

```
ffmpeg -i input.avi -r 24 output.avi
```

- To force the frame rate of the input file (valid for raw formats only) to 1 fps and the frame rate of the output file to 24 fps:

```
ffmpeg -r 1 -i input.m2v -r 24 output.avi
```

The format option may be needed for raw input files.

### 3. Detailed description

The transcoding process in `ffmpeg` for each output can be described by the following diagram:



`ffmpeg` calls the libavformat library (containing demuxers) to read input files and get packets containing encoded data from them. When there are multiple input files, `ffmpeg` tries to keep them synchronized by tracking lowest timestamp on any active input stream.

Encoded packets are then passed to the decoder (unless streamcopy is selected for the stream, see further for a description). The decoder produces uncompressed frames (raw video/PCM audio/...) which can be processed further by filtering (see next section). After filtering, the frames are passed to the encoder, which encodes them and outputs encoded packets. Finally those are passed to the muxer, which writes the encoded packets to the output file.

## FFmpeg documentation [39]

FFmpeg documentation :

Complex filtergraphs are configured with the `-filter_complex` option. Note that this option is global, since a complex filtergraph, by its nature, cannot be unambiguously associated with a single stream or file.

The `-lavfi` option is equivalent to `-filter_complex`.

A trivial example of a complex filtergraph is the `overlay` filter, which has two video inputs and one video output, containing one video overlaid on top of the other. Its audio counterpart is the `amix` filter.

### 3.2 Stream copy

Stream copy is a mode selected by supplying the `copy` parameter to the `-codec` option. It makes `ffmpeg` omit the decoding and encoding step for the specified stream, so it does only demuxing and muxing. It is useful for changing the container format or modifying container-level metadata. The diagram above will, in this case, simplify to this:



Since there is no decoding or encoding, it is very fast and there is no quality loss. However, it might not work in some cases because of many factors. Applying filters is obviously also impossible, since filters work on uncompressed data.

### 4. Stream selection

By default, `ffmpeg` includes only one stream of each type (video, audio, subtitle) present in the input files and adds them to each output file. It picks the "best" of each based upon the following criteria: for video, it is the stream with the highest resolution, for audio, it is the stream with the most channels, for subtitles, it is the first subtitle stream. In the case where several streams of the same type rate equally, the stream with the lowest index is chosen.

You can disable some of those defaults by using the `-vn/-an/-sn` options. For full manual control, use the `-map` option, which disables the defaults just described.

### 5. Options

All the numerical options, if not specified otherwise, accept a string representing a number as input, which may be followed by one of the SI unit prefixes, for example: 'K', 'M', or 'G'.

If 'i' is appended to the SI unit prefix, the complete prefix will be interpreted as a unit prefix for binary multiplies, which are based on powers of 1024 instead of powers of 1000. Appending 'B' to the SI unit prefix multiplies the value by 8. This allows using, for example: 'KB', 'MiB', 'G' and 'B' as number suffixes.

Options which do not take arguments are boolean options, and set the corresponding value to true. They can be set to false by prefixing the option name with "no". For example using "nofoo" will set the boolean option with name "foo" to false.

#### 5.1 Stream specifiers

Some options are applied per-stream, e.g. bitrate or codec. Stream specifiers are used to precisely specify which stream(s) a given option belongs to.

## FFmpeg documentation [39]

FFmpeg documentation :

A stream specifier is a string generally appended to the option name and separated from it by a colon. E.g. `-codec:a:1 ac3` contains the `a:1` stream specifier, which matches the second audio stream. Therefore, it would select the `ac3` codec for the second audio stream.

A stream specifier can match several streams, so that the option is applied to all of them. E.g. the stream specifier in `-b:a 128k` matches all audio streams.

An empty stream specifier matches all streams. For example, `-codec copy` or `-codec: copy` would copy all the streams without reencoding.

Possible forms of stream specifiers are:

**'*stream\_index*'**

Matches the stream with this index. E.g. `-threads:1 4` would set the thread count for the second stream to 4.

**'*stream\_type*[:*stream\_index*]'**

*stream\_type* is one of following: 'v' for video, 'a' for audio, 's' for subtitle, 'd' for data, and 't' for attachments. If *stream\_index* is given, then it matches stream number *stream\_index* of this type. Otherwise, it matches all streams of this type.

**'*p*:*program\_id*[:*stream\_index*]'**

If *stream\_index* is given, then it matches the stream with number *stream\_index* in the program with the id *program\_id*. Otherwise, it matches all streams in the program.

**'#*stream\_id*'**

Matches the stream by a format-specific ID.

### 5.2 Generic options

These options are shared amongst the ff\* tools.

**'-L'**

Show license.

**'-h, -?, -help, --help [*arg*]'**

Show help. An optional parameter may be specified to print help about a specific item. If no argument is specified, only basic (non advanced) tool options are shown.

Possible values of *arg* are:

**'long'**

Print advanced tool options in addition to the basic tool options.

**'full'**

Print complete list of options, including shared and private options for encoders, decoders, demuxers, muxers, filters, etc.

**'decoder=*decoder\_name*'**

Print detailed information about the decoder named *decoder\_name*. Use the `'-decoders'` option to get a list of all decoders.

## FFmpeg documentation [39]

FFmpeg documentation :

Force input or output file format. The format is normally auto detected for input files and guessed from the file extension for output files, so this option is not needed in most cases.

**'-i filename (input)'**

input file name

**'-y (global)'**

Overwrite output files without asking.

**'-n (global)'**

Do not overwrite output files, and exit immediately if a specified output file already exists.

**'-c[:stream\_specifier] codec (input/output,per-stream)'**

**'-codec[:stream\_specifier] codec (input/output,per-stream)'**

Select an encoder (when used before an output file) or a decoder (when used before an input file) for one or more streams. *codec* is the name of a decoder/encoder or a special value `copy` (output only) to indicate that the stream is not to be re-encoded.

For example

```
ffmpeg -i INPUT -map 0 -c:v libx264 -c:a copy OUTPUT
```

encodes all video streams with libx264 and copies all audio streams.

For each stream, the last matching *c* option is applied, so

```
ffmpeg -i INPUT -map 0 -c copy -c:v:1 libx264 -c:a:137 libvorbis OUTPUT
```

will copy all the streams except the second video, which will be encoded with libx264, and the 138th audio, which will be encoded with libvorbis.

**'-t duration (output)'**

Stop writing the output after its duration reaches *duration*. *duration* may be a number in seconds, or in `hh:mm:ss[.xxx]` form.

`-to` and `-t` are mutually exclusive and `-t` has priority.

**'-to position (output)'**

Stop writing the output at *position*. *position* may be a number in seconds, or in `hh:mm:ss[.xxx]` form.

`-to` and `-t` are mutually exclusive and `-t` has priority.

**'-fs limit\_size (output)'**

Set the file size limit, expressed in bytes.

**'-ss position (input/output)'**

When used as an input option (before `-i`), seeks in this input file to *position*. Note the in most formats it is not possible to seek exactly, so `ffmpeg` will seek to the closest seek point before

<http://www.ffmpeg.org/ffmpeg.html>

## C.3 LAME [41]

Detailed command-line switches

<a href="#">Index page</a>
<a href="#">About LAME</a>
<a href="#">Intro to encoding</a>
<a href="#">Usage of LAME</a>
<a href="#">Common switches</a>
<a href="#">Detailed Settings</a>
<a href="#">Version history</a>
<a href="#">Coders of LAME</a>
<a href="#">Contact LAME</a>
<a href="#">LAME links</a>



### Detailed command line switches

#### Options ordered by feature

Setting	Brief description
<b>Input options</b>	
By default, LAME accepts a PCM audio sample inside a .WAV container as the input file, in 8, 16, 24 and 32 bits integer and in IEEE FLOAT. If it is compiled with <a href="#">libsndfile</a> , then it also supports the extra formats that the library supports. There is also support for RAW PCM data and piped input	
-	For piped/streamed Input, use "-" as the input and/or output file name.
--out-dir path	If no explicit output file is specified, a file will be written at given path. Ignored when using piped/streamed input
--mp1input <sup>1</sup>	Input file is an MPEG layer I file. Implicit if extension is .mp1
--mp2input <sup>1</sup>	Input file is an MPEG layer II file. Implicit if extension is .mp2
--mp3input <sup>1</sup>	Input file is an MPEG layer III file. Implicit if extension is .mp3
--scale number	Scale input (multiply PCM data) by number.
--scale-l number	Scale channel 0 (left) input (multiply PCM data ) by number
--scale-r number	Scale channel 1 (right) input (multiply PCM data) by number
--gain number	apply Gain adjustment in decibels, range -20.0 to +12.0.
--swap-channel	Swap Left and Right channels
--nogap file1 file2 ...	Gapless encoding for a set of contiguous files
--nogapout dir	Output dir for gapless encoding (must precede --nogap)
--nogaptags	Allow the use of VBR tags in gapless encoding
<b>Input options for RAW PCM</b>	
-r	Assume input file is raw PCM.
-s number	Input sampling frequency in kHz (Default is 44.1Khz)
--signed / --unsigned	Input is signed (default) or unsigned
--bitwidth w	Input bit width is w (default 16)
-x	Force byte-swapping of input stream

<http://lame.cvs.sourceforge.net/viewvc/lame/lame/doc/html/detailed.html>

# LAME – Detailed command line switches [41]

Detailed command-line switches

<b>Index page</b>
<b>About LAME</b>
<b>Intro to encoding</b>
<b>Usage of LAME</b>
<b>Common switches</b>
<b>Detailed Settings</b>
<b>Version history</b>
<b>Coders of LAME</b>
<b>Contact LAME</b>
<b>LAME links</b>

-X n[,m]	selects between different noise measurements n for long block, m for short. if m is omitted, m = n
--vbr-new	In LAME 3.98, the new VBR was made default, and this switch may be used for experimental tweaks.
<b>Aliases and removed settings</b>	
-f	Alias of <a href="#">-q 7</a> .
-h	Alias of <a href="#">-q 2</a> .
-v	Alias of <a href="#">-V 4</a>
-S	Alias of <a href="#">--silent</a>
--alt-preset	Alias of <a href="#">--preset</a>
--r3mix	Alias of <a href="#">-V 3</a>
--vbr-mtrh	Alias of LAME 3.98 default vbr mode
--pad-id3v2	Alias of <a href="#">--pad-id3v2-size 128</a>
-d	No longer supported
-k	No longer supported
--cwlimit	No longer supported
--ogginput	No longer supported. Compile with <a href="#">libsndfile</a> instead
--brief	Currently unused
--interch x	Currently unused
--npsytune	Currently unused

## Detailed description in alphabetical order

### **-a** Downmix

mix the stereo input file to mono and encode as mono.

This option is only needed in the case of raw PCM stereo input (because LAME cannot determine the number of channels in the input file). To encode a stereo PCM input file as mono, use "lame -m m -a"

For WAV and AIFF input files, using "-m m" will always produce a mono .mp3 file from both mono and stereo input.

### **--abr n** Average bitrate encoding (aka Safe VBR)

turns on encoding with a targeted average bitrate of n kbps, allowing to use frames of different sizes. The allowed range of n is 8...320 kbps, you can use any integer value within that range.

### **--add-id3v2** Force addition of version 2 tag

Tells LAME to add the tag information as id3v2. This implies adding both, a version 1 and a version 2 tag, if the values fit on a version 1 tag. See --id3v1-only and --id3v2-only if you want a more fine-grained control.

### **-b n** Bitrate

Codec	sample frequencies (kHz)	bitrates (kbps)
MPEG-1 layer III	32, 48, 44.1	32 40 48 56 64 80 96 112 128 160 192 224 256 320

# LAME – Detailed command line switches [41]

Detailed command-line switches

<p><b>Index page</b></p> <hr/> <p><b>About LAME</b></p> <hr/> <p><b>Intro to encoding</b></p> <hr/> <p><b>Usage of LAME</b></p> <hr/> <p><b>Common switches</b></p> <hr/> <p><b>Detailed Settings</b></p> <hr/> <p><b>Version history</b></p> <hr/> <p><b>Coders of LAME</b></p> <hr/> <p><b>Contact LAME</b></p> <hr/> <p><b>LAME links</b></p>	<p>MPEG-2 layer III 16, 24, 22.05 8 16 24 32 40 48 56 64 80 96 112 128 144 160</p>
	<p>MPEG-2.5 layer III 8, 12, 11.025 8 16 24 32 40 48 56 64</p>
	<p>The bitrate to be used. Default is 128kbps in MPEG1 (64 for mono), 64kbps in MPEG2 (32 for mono) and 32kbps in MPEG2.5 (16 for mono).</p>
	<p>When used with variable bitrate encodings (VBR), -b specifies the minimum bitrate to use. This is useful only if you need to circumvent a buggy hardware device with strange bitrate constrains. (You will need the -F setting too for digital silence). Default for ABR/VBR is the minimum allowed bitrate for the MPEG version. (i.e. no limit)</p>
	<p><b>-B n</b> Max bitrate</p> <hr/> <p>see also option "-b" for allowed bitrates.</p> <p>Maximum allowed bitrate when using VBR/ABR.</p> <p>Using -B is NOT RECOMMENDED. A 128 kbps CBR bitstream, because of the bit reservoir, can actually have frames which use as many bits as a 320 kbps frame. ABR/VBR modes minimize the use of the bit reservoir, and thus need to allow 320 kbps frames to get the same flexibility as CBR streams. This is useful only if you need to circumvent a buggy hardware device with strange bitrate constrains. Default value is the maximum allowed bitrate for the MPEG versio (i.e. no limit)</p>
	<p><b>--big-endian</b> Set the byte order to big-endian.</p> <hr/> <p>This switch tells LAME that the RAW pcm input is encoded in big-endian instead of little-endian.</p>
	<p><b>--bitwidth</b> Sets the bitwidth value</p> <hr/> <p>With RAW pcm input, this switch lets specify the bitwidth of the same (8 bits, 16 bits...)</p>
	<p><b>-c</b> copyright flag</p> <hr/> <p>mark the encoded file as copyrighted</p>
	<p><b>--cbr</b> Enforce constant bitrate encoding</p> <hr/> <p>This switch disables the VBR encoding. Examples:</p> <pre>lame -V 0 --cbr input.wav</pre> <p>Will encode at -b 128 (ignore completely -V and encode at default bitrate)</p> <pre>lame --preset 160 --cbr</pre> <p>Will encode at -b 160 (ignore ABR and use the specified bitrate)</p> <pre>lame --abr 200 --cbr</pre> <p>Will encode at -b 192 (ignore ABR and use the nearest bitrate to the one specified)</p>
	<p><b>--clipdetect</b> Clipping detection</p> <hr/> <p>Enable --replaygain-accurate and print a message whether clipping occurs and how far in dB the waveform is from full scale.</p>

<http://lame.cvs.sourceforge.net/viewvc/lame/lame/doc/html/detailed.html>

# LAME – Detailed command line switches [41]

## Detailed command-line switches

<a href="#">Index page</a>	<p>This option is not usable if the MP3 decoder was <b>explicitly</b> disabled in the build of LAME.</p>
<a href="#">About LAME</a>	<p>See also: <code>--replaygain-accurate</code></p>
<a href="#">Intro to encoding</a>	<p><code>--comp x</code> Indicate a compression ratio instead of a bitrate</p>
<a href="#">Usage of LAME</a>	<p>Use this in place of the <code>-b</code> setting if you prefer to indicate a compression ratio (integer value). The ratio is the value of <math>\frac{\text{original-filesize}}{\text{desired-filesize}}</math></p>
<a href="#">Common switches</a>	<p><code>--decode</code> MP3 decode capability</p>
<a href="#">Detailed Settings</a>	<p>This uses LAME's HIP<sup>2</sup> decoder to decode an MP3 (layers 1, 2 and 3) file to a wav file.</p> <p>If <code>-t</code> is used (disable wav header), LAME will output raw pcm in native endian format (use <code>-x</code> to swap bytes).</p> <p>This option is not usable if the MP3 decoder was <b>explicitly</b> disabled in the build of LAME.</p>
<a href="#">Version history</a>	<p><code>--decode-mp3delay x</code> Indicate a different encoder delay for decoding</p> <p>When decoding an mp3 file, LAME automatically corrects for the start delay that the encoder had to put into it. This setting lets you specify a different delay than LAME's own one, so that it is possible to compensate for the delay of mp3's generated with other encoders.</p>
<a href="#">Coders of LAME</a>	<p><code>--disptime secs</code> Update the display each secs</p>
<a href="#">Contact LAME</a>	<p>Use this setting to change the frequency that LAME updates the display when encoding a file. Allows decimal values (like 0.5)</p>
<a href="#">LAME links</a>	<p><code>-e n/5/c</code> Write the de-emphasis flag</p> <p>n = (none, default)  5 = 50/15 microseconds  c = ccitt j.17</p> <p>All this does is set a flag in the bitstream. If you have a PCM input file where one of the above types of (obsolete) emphasis has been applied, you can set this flag in LAME. Then the mp3 decoder should de-emphasize the output during playback, although most decoders ignore this flag.</p> <p>A better solution would be to apply the de-emphasis with a standalone utility before encoding, and then encode without <code>-e</code>.</p>
	<p><code>-F</code> Enforce the minimum bitrate</p> <p>Enforces the minimum bitrate. Without this option, passages of analog silence will be encoded at the minimum bitrate possible (32 or 8, depending on MPEG version).</p>
	<p><code>--flush</code> Flush the output stream as soon as possible</p> <p>This setting forces a flush of the data written as soon as the operation has finished. This is mostly useful in case of streams. With files, it may degrade performance, since the OS has to write to disk in smaller chunks.</p>
	<p><code>--freeformat</code> Encode to freeformat stream</p> <p>LAME will produce a fixed bitrate, free format bitstream. User must specify the desired bitrate in kbps, which can be any integer between 8 and 640.</p>

# LAME – Detailed command line switches [41]

Detailed command-line switches

<a href="#">Index page</a>
<a href="#">About LAME</a>
<a href="#">Intro to encoding</a>
<a href="#">Usage of LAME</a>
<a href="#">Common switches</a>
<a href="#">Detailed Settings</a>
<a href="#">Version history</a>
<a href="#">Coders of LAME</a>
<a href="#">Contact LAME</a>
<a href="#">LAME links</a>

## **-m x** Channel modes

<code>-m m</code>	mono
<code>-m l</code>	get only the left channel of a stereo signal for a mono output
<code>-m r</code>	get only the right channel of a stereo signal for a mono output
<code>-m s</code>	forced L/R stereo
<code>-m j / -m a</code>	automatic switch between L/R and M/S stereo
<code>-m f</code>	forced mid/side stereo
<code>-m d</code>	dual (independent) channels.

MONO is the default mode for mono input files. If "-m m" is specified for a stereo input file, the two channels will be averaged into a mono signal. (Note: See comments about the -a switch for RAW PCM streams)

FORCED L/R STEREO encodes the left and the right signals independently, and gives more or less bits to each, depending on the currently available.

JOINT STEREO is the default mode of encoding. jstereo means the encoder can use (on a frame by frame basis) either L/R stereo or mid/side stereo. In mid/side stereo, the mid (L+R) and side (L-R) channels are encoded, and more bits are allocated to the mid channel than the side channel. When there isn't too much stereo separation, this effectively increases the bandwidth, so having higher quality with the same amount of bits.

Using mid/side stereo inappropriately can result in audible compression artifacts. Too much switching between mid/side and regular stereo can also sound bad. To determine when to switch to mid/side stereo, LAME uses a much more sophisticated algorithm than the one described in the ISO documentation.

FORCED MID/SIDE STEREO forces all frames to be encoded with mid/side stereo. It should only be used if you are sure every frame of the input file has very little stereo separation.

DUAL CHANNEL mode is similar to encode the left and right as two mono signals. Its purpose was meant for Dual language streams where only one of them should be decoded. Most decoders just decode them as a stereo stream

## **--mp1input** **--mp2input** **--mp3input** MPEG layer I, II or III input file

Assume the input file is an MP1/2/3 file. LAME will decode the input file before re-encoding it. Since MP3 is a lossy format, this is not recommended in general. But it is useful for creating low bitrate mp3s from high bitrate mp3s. If the filename ends in ".mp3" LAME will assume it is an MP3. For stdin or MP3 files which don't end in .mp3 you need to use this switch.

## **--noasm value** Disables the specified assembler instructions

When the encoder is compiled with assembler optimizations, this setting allows to disable them at runtime (They are only enabled if the CPU supports them). The purpose of this setting is to detect problems in those optimizations, and/or check the speed difference.

## **--nogap file1 file2 [...]** Encodes multiple continuous files.

Encodes multiple files (ordered by position) which are meant to be played gaplessly.

## LAME – Detailed command line switches [41]

Detailed command-line switches

<b>Index page</b>	--preset phon+ / lw / mw-eu	-b 24 -m m
	--preset mw-us	-b 40 -m m
	--preset voice	-b 56 -m m
	--preset fm / radio	-b 112
	--preset hifi	-b 160
	--preset cd	-b 192
	--preset studio	-b 256
<b>About LAME</b>		
<b>Intro to encoding</b>		
<b>Usage of LAME</b>		
<b>Common switches</b>		
<b>Detailed Settings</b>		
<b>Version history</b>		
<b>Coders of LAME</b>		
<b>Contact LAME</b>		
<b>LAME links</b>		

**--priority value** Set process priority

(Windows and OS/2 only)

Sets the process priority for LAME while running under Windows and/or IBM OS/2. This can be very useful to avoid the system becoming slow and/or unresponsive. By setting LAME to run in a lower priority, you leave more time for the system to update basic processing (drawing windows, polling keyboard/mouse, etc). The impact in LAME's performance is minimal if you use priority 0 to 2.

The valid parameters are:

	Priority	Windows	OS/2
0	Low priority	IDLE_PRIORITY_CLASS	IDLE, delta = 0
1	Medium priority	IDLE_PRIORITY_CLASS	IDLE, delta = +31
2	Regular priority	NORMAL_PRIORITY_CLASS	REGULAR, delta = -31
3	High priority	HIGH_PRIORITY_CLASS	REGULAR, delta = 0
4	Maximum priority	HIGH_PRIORITY_CLASS	REGULAR, delta = +31

Note that if you call '--priority' without a parameter, then priority 0 will be assumed.

**-q n** Algorithm quality selection

Bitrate is of course the main influence on quality. The higher the bitrate, the higher the quality. But for a given bitrate, we have a choice of algorithms to determine the best scalefactors and Huffman coding (noise shaping).

For CBR, ABR and --vbr-old modes, the following table applies

-q 0	Use the best algorithms (Best Huffman coding search, full outer loop, and the highest precision of several parameters).
-q 1 to -q 4	Similar to -q 0 without the full outer loop and with decreasing precision of parameters the further from -q 0. -q 3 is the default
-q 5 and -q 6	Same as -q 7, but enables noise shaping and increases subblock gain
-q 7 to -q 9	Same as -f. Very fast, OK quality. Psychoacoustics are used for pre-echo and mid/side stereo, but no noise-shaping is done.

For the default VBR mode since LAME 3.98, the following table applies

-q 0 to -q 4	include all features of the other modes and additionally use the best search when applying Huffman coding.
-q 5 and -q 6	include all features of -q7, calculate and consider actual quantisation noise, and additionally enable subblock gain.

<http://lame.cvs.sourceforge.net/viewvc/lame/lame/doc/html/detailed.html>

## LAME – Detailed command line switches [41]

Detailed command-line switches

<b>Index page</b>	ReplayGain analysis does <code>_not_</code> affect the content of a compressed data stream itself, it is a value stored in the header of a sound file. Information on the purpose of ReplayGain and the algorithms used is available from <a href="http://www.replaygain.org/">http://www.replaygain.org/</a>
<b>About LAME</b>	Only the "Radio" ReplayGain value is computed. It is stored in the LAME tag. The analysis is performed with the reference volume equal to 89dB. Note: the reference volume has been changed from 83dB on transition from version 3.95 to 3.95.1.
<b>Intro to encoding</b>	This switch is enabled by default.
<b>Usage of LAME</b>	See also: <code>--replaygain-accurate</code> , <code>--noreplaygain</code>
<b>Common switches</b>	<code>--resample n</code> Output sampling frequency in kHz where n = 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 Output sampling frequency. Resample the input if necessary.
<b>Detailed Settings</b>	If not specified, LAME may sometimes resample automatically when faced with extreme compression conditions (like encoding a 44.1 kHz input file at 32 kbps). To disable this automatic resampling, you have to use <code>--resample</code> to set the output sample rate equal to the input sample rate. In that case, LAME will not perform any extra computations.
<b>Version history</b>	<code>-s n</code> Sampling frequency in kHz where n = sampling rate in kHz.
<b>Coders of LAME</b>	Required for raw PCM input files. Otherwise it will be determined from the header information in the input file.
<b>Contact LAME</b>	LAME will automatically resample the input file to one of the supported MP3 sample rates if necessary.
<b>LAME links</b>	<code>--scale arg</code> <code>--scale-l arg</code> <code>--scale-r arg</code> Scale the amplitude of the input file Scales input by arg. This just multiplies the PCM data (after it has been converted to floating point) by arg. arg > 1: increase volume arg = 1: no effect arg < 1: reduce volume Use with care, since most MP3 decoders will truncate data which decodes to values greater than 32768. The <i>l</i> and <i>r</i> variants apply the scaling only to left (channel 0) or right (channel 1) respectively. <code>--signed</code> Input RAW uses signed values Use with RAW pcm to indicate if the data is signed (values centered at zero) or unsigned (all positive values). This is the default value. <code>--silent</code> Don't print any output in the screen Disables the screen output. Useful for job processing or other times where screen output is undesirable. <code>--space-id3v1</code> Use spaces for padding. Use spaces instead of null values for padding data in an ID3v1 tag. <code>--strictly-enforce-ISO</code> Strict ISO compliance

<http://lame.cvs.sourceforge.net/viewvc/lame/lame/doc/html/detailed.html>

## C.4 Kurzbeschreibung PQevalAudio [46]

TSP (audio) - PQevalAudio

### PQevalAudio

#### Routine

PQevalAudio [options] AFileR AFileT

#### Purpose

Evaluate the perceptual quality degradation for audio files

#### Description

This program takes a reference audio file and a test audio file and measures the perceptual degradation of the test signal with respect to the reference signal. The measurement is based on the ITU-R BS.1387 (PEAQ) standard. The output combines a number of model output variables (MOV's) into a single measure, the Objective Difference Grade which is an impairment scale with the following meanings.

```
0 imperceptible
-1 perceptible but not annoying
-2 slightly annoying
-3 annoying
-4 very annoying
```

The measurement procedure has only been calibrated for a sampling rate of 48 kHz. The measurement procedure works for monaural or binaural (stereo) signals. Use ResampAudio to resample audio files with other sampling rates to 48 kHz. The measurement procedure also assumes that the files have been time and gain aligned. Use CompAudio to determine the required gain and delay compensation and CopyAudio to actually modify the gain and time-align the files.

This program implements the basic version of PEAQ. The measurement results on a standard database do not fall within the tight bounds specified in the standard (see the report cited below for a discussion of the difficulties in interpreting the standard). However, the results are close enough to be useful for quality impairment measurements.

#### References:

ITU-R Recommendation BS.1387, "Method for Objective Measurements of Perceived Audio Quality", Dec. 1998 (and subsequent corrections).

T. Thiede et al, "PEAQ - The ITU Standard for Objective Measurement of Perceived Audio Quality", J. Audio Eng. Soc., vol. 48, pp. 3-29, Jan.-Feb. 2000.

P. Kabal, "An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality", TSP Lab Technical Report, Dept. Electrical & Computer Engineering, McGill University, May 2002. (<http://www.TSP.ECE.McGill.CA/MMSP/Documents>)

#### Options

Input file(s): AFileR AFileT:

The environment variable AUDIOPATH specifies a list of directories to be searched for the input audio file(s). Specifying "-" as the input file indicates that input is from standard input.

-g GAIN, --gain=GAIN

A gain factor applied to the data from the input files. This gain applies to all channels in a file. The gain value can be given as a real number (e.g., "0.003") or as a ratio (e.g., "1/256"). This option may be given more than once. Each invocation applies to the input files that follow the option.

-l L:U, --limits=L:U

Sample limits for the input files (numbered from zero). Each invocation applies to the input files that follow the option. The specification ":" means the entire file; "L:" means from sample L to the end; ":U" means from sample

<http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/PQevalAudio.html>

## Kurzbeschreibung PQevalAudio [46]

### TSP (audio) - PQevalAudio

0 to sample U; "N" means from sample 0 to sample N-1.

**-L LEVEL --levelSPL=LEVEL**  
Listening level (in dB SPL) for a maximum amplitude sine (default 92 dB SPL)

**-i NI --info==NI**  
Print information for each NI'th PEAQ frame. The default is chosen to print PEAQ frame information for at most 50 frames. Set NI to zero to suppress the printout of intermediate information.

**-o OPTIONS --options=OPTIONS**  
Processing options.

"clip\_MOV" or "no\_clip\_MOV" - Clip MOV values (default "no\_clip\_MOV")

"PC\_init=V" - Initial value for the pattern correction factors (default 0)

"PD\_factor=V" - Forgetting factor for the maximum probability of detection calculation (default 1).

"overlap\_delay" or "no\_overlap\_delay" - Overlap warmup frames (frames before the data boundary) and the delay for calculating modulation difference and noise loudness values (default "overlap\_delay")

"data\_bounds" or "no\_data\_bounds" - Ignore frames with small data values at the beginning and end of the reference (default "data\_bounds")

"end\_min=N" - Stop processing at the frame that contains at least N samples (default value of N is 1024). The "-l" command line option can be used to shift the entire processing. For instance "-l -1024:" will add half a frame of zero padding before the start of data.

"EHS\_lag\_start=N" where N is 0 or 1. The default is 1.

**-t FTYPE, --type=FTYPE**  
Input audio file type. In the default automatic mode, the input file type is determined from the file header. For input from a non-random access file (e.g. a pipe), the input file type can be explicitly specified to avoid the lookahead used to read the file header. This option can be specified more than once. Each invocation applies to the input files that follow the option. See the description of the environment variable AF\_FILETYPE below for a list of file types.

**-P PARMS, --parameters=PARMS**  
Parameters to be used for headerless input files. This option may be given more than once. Each invocation applies to the input files that follow the option. See the description of the environment variable AF\_NOHEADER below for the format of the parameter specification.

**-h, --help**  
Print a list of options and exit.

**-v, --version**  
Print the version number and exit.

### Environment variables

#### AF\_FILETYPE:

This environment variable defines the input audio file type. In the default mode, the input file type is determined from the file header.

```
"auto"          - determine the input file type from the file header
"AU" or "au"    - AU audio file
"WAVE" or "wave" - WAVE file
"AIFF" or "aiff" - AIFF or AIFF-C sound file
"noheader"     - headerless (non-standard or no header) audio file
"SPHERE"      - NIST SPHERE audio file
"ESPS"        - ESPS sampled data feature file
"IRCAM"       - IRCAM soundfile
"SPPACK"      - SPPACK file
"INRS"        - INRS-Telecom audio file
"SPW"         - Comdisco SPW Signal file
"CSL" or "NSP" - CSL NSP file
"text"        - Text audio file
```

#### AF\_NOHEADER:

<http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/PQevalAudio.html>

## C.5 Kurzbeschreibung ResampAudio [47]

TSP (audio) - ResampAudio

### ResampAudio

#### Routine

ResampAudio [options] AFileI AFileO

#### Purpose

Resample data from an audio file

#### Description

This program resamples data from an audio file. This process involves interpolating between the samples in the original file to create a new sequence of samples with a new spacing (sampling rate). This program allows for an arbitrary ratio of output sampling rate to input rate. The default settings give a high quality sample rate conversion.

#### Theory of Operation

The process used for interpolation depends on the ratio of the output sampling rate to the input sampling rate.

- 1: If the output sampling rate over the input sampling rate is expressible as a ratio of small integers, the sample rate change process is done using a conventional interpolation filter designed for the interpolation factor (numerator of the sampling rate ratio) followed by subsampling by the subsampling factor (denominator of the sampling rate ratio).
- 2: For the general case, an interpolating filter is designed using an interpolation factor of 24. For each output sample, the interpolating filter is used to create two samples that bracket the desired sampling point. Linear interpolation is used between these values to generate the output value.

The default interpolation filter is a linear phase FIR filter designed by applying a Kaiser window to an ideal lowpass filter response. The filter is characterized by a cutoff frequency, a window shape parameter, and the number of coefficients. The window shape parameter (alpha) controls the passband ripple and the stopband attenuation. For a fixed number of coefficients, decreasing ripple and increasing attenuation (larger alpha) come at the expense of a wider transition width.

The cutoff of the default interpolation filter depends on the input and output sampling rates. Let  $f_{si}$  be the sampling rate of the input signal and  $f_{so}$  be the sampling rate of the output signal.

- 1:  $f_{so} > f_{si}$ . The cutoff of the interpolation filter is set to  $f_{si}/2$ .
- 2:  $f_{so} < f_{si}$ . The cutoff of the interpolation filter is set to  $f_{so}/2$ .

The default design aims for an 80 dB stopband attenuation and a transition width which is 15% of the cutoff frequency. The attenuation directly determines alpha. The value of alpha together with the transition width determines the number of filter coefficients.

The parameters of the interpolating filter can also be set by the user. The design parameters are the interpolation factor, the filter cutoff frequency, the Kaiser window parameter, and the number of filter coefficients. The following table shows the effect of changing the Kaiser window parameter alpha.

stopband attenuation	alpha	transition width D	passband ripple	
30 dB	2.210	1.536	+/- 0.270 dB	
40 dB	3.384	2.228	+/- 0.0864 dB	
50 dB	4.538	2.926	+/- 0.0274 dB	
60 dB	5.658	3.621	+/- 0.00868 dB	
70 dB	6.764	4.317	+/- 0.00275 dB	
80 dB	7.865	5.015	+/- 0.00089 dB	(default)
90 dB	8.960	5.712	+/- 0.00027 dB	
100 dB	10.056	6.408	+/- 0.00009 dB	

<http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/ResampAudio.html>

## Kurzbeschreibung ResampAudio [47]

TSP (audio) - ResampAudio

Input file name: AFileI:

The environment variable AUDIOPATH specifies a list of directories to be searched for the input audio file. Specifying "-" as the input file indicates that input is from standard input.

Output file name: AFileO:

The second file name is the output file. Specifying "-" as the output file name indicates that output is to be written to standard output. If the output file type is not explicitly given (-F option), the extension of the output file name is used to determine the file type.

```

".au"   - AU audio file
".wav"  - WAVE file
".aif"  - AIFF sound file
".afc"  - AIFF-C sound file
".raw"  - Headerless file (native byte order)
".txt"  - Headerless file (text data)

```

-s SFREQ, --srate=SFREQ

Sampling frequency for the output file.

-i SRATIO, --interpolate=SRATIO

Ratio of the output sampling rate to the input sampling rate. This argument is specified as a single number or as a ratio of the form N/D, where each of N and D can be floating point values. This option is an alternate means to specify the output sampling rate.

-a OFFS, --alignment=OFFS

Time offset of the first output sample relative to the input data. The units are samples of the input data. This value can be specified as a single number or a ratio.

-f FPARMS, --filter\_spec=FPARMS

Filter parameters. The filter parameters are given as keyword values. There are two cases: the filter coefficients are supplied in a file or the filter is calculated as a Kaiser windowed lowpass filter.

```

Filter file:
  file="file_name"  Input filter file name.  If specified, the filter
                    coefficients are read from the named file.
  ratio=Ir          Filter interpolation factor
  delay=Del        Filter delay in units of filter samples (default
                    for symmetrical filters is (N-1)/2, where N is the
                    number of coefficients).  The delay can be
                    specified as a single number or as a ratio.  The
                    filter delay must be supplied for non-symmetrical
                    filters.

Windowed lowpass:
  ratio=Ir          Filter interpolation factor.  The default depends
                    on the ratio of output sampling frequency to
                    input sampling frequency.  This parameter can be
                    specified as a single number or as a ratio.
  cutoff=Fc        Filter cutoff in normalized frequency relative to
                    the filter interpolation factor (0 to Ir/2).  This
                    value can be specified as a single number or as a
                    ratio.  The default cutoff frequency is determined
                    from the input and output sampling rates.  For an
                    increase in sampling rate, it is set to 0.5.  For
                    a decrease in sampling rate it is set to
                    0.5*fso/fsi.
  atten=A          Filter stopband attenuation in dB.  The attenuation
                    must be at least 21 dB.  The default is 80.  The
                    attenuation is an alternate way to specify the
                    Kaiser window parameter alpha.
  alpha=a          Kaiser window parameter.  Zero corresponds to a
                    rectangular window (stopband attenuation 21 dB).
                    The default is 7.865 corresponding to a stopband
                    attenuation of 80 dB.
  N=Ncof          Number of filter coefficients.  The default is
                    to choose the number of coefficients to give a
                    transition band which is 15% of the cutoff
                    frequency.
  span=Wspan       Window span.  This is the span of the non-zero part
                    of the window.  The default window span is equal
                    to the number of filter coefficients minus one.

```

<http://www-mmsp.ece.mcgill.ca/documents/Software/Packages/AFsp/ResampAudio.html>