Umsetzung eines Programms zur Sensibilisierung der auditiven Wahrnehmung

Dynamikbearbeitung

Bachelorarbeit aus Aufnahmetechnik 1, SE

Raphael Kapeller

Betreuung: Dr. Alois Sontacchi

Graz, 21. Juni 2010

UNIVERSITÄT FÜR MUSIK UND DARSTELLENDE KUNST GRAZ - AUSTRIA

institut für elektronische musik und akustik

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 21.Juni 2010

……..

Unterschrift

„One apple may be different from another apple. In this case, we compare one piece with another piece.“

Zwicker/Fastl

Abstract

Compressors play an important role in audio technology. Their broad field of application, as well as the wide range of properties and adjustment possibilities require trained ears and experience in dealing with dynamic processing. In order to help creating these conditions, a software programme, which specializes in loudness and compression in audio signals, has been developed within this thesis. The programme consists of three different exercises and allows an easy first step into dynamic processing that is of higher quality and more efficient. The theoretical basis for this is provided in the chapters about loudness and compressors. As another major part the present work contains the technical implementation of the software.

Zusammenfassung

Kompressoren spielen im Bereich der Audiotechnik eine bedeutende Rolle. Ihr vielseitiges Einsatzgebiet, sowie die große Palette an Eigenschaften und Einstellmöglichkeiten der Parameter, verlangen ein geschultes Gehör und Erfahrung im Umgang mit Dynamikbearbeitungen. Um diese Voraussetzungen zu schaffen, entstand im Zuge dieser Arbeit ein Softwareprogramm, welches sich speziell auf Lautheit und Kompression in Audiosignalen spezialisiert. Das Programm besteht aus drei verschiedenen Übungen und soll einen Einstieg in eine qualitativ hochwertigere und effizientere Dynamikbearbeitung ermöglichen. Die theoretische Basis dazu liefern die Kapitel über Lautheit und Kompressoren. Ein weiterer Bestandteil der Arbeit ist die technische Umsetzung der Software.

Inhaltsverzeichnis:

1 Einleitung

Ein Apfel kann sich von einem anderen unterscheiden. Aus diesem Grund vergleichen wir einen Gegenstand mit dem Nächsten. Erfahrung und die Möglichkeit, Dinge zu unterscheiden, entstehen durch eine Vielzahl von Vergleichen und Tests, sowie der Erweiterung des vorhandenen Wissens. Die Fertigkeit, Situationen in kürzester Zeit analysieren und bewerten zu können, befähigt einen Menschen, sofort zu reagieren und Maßnahmen zu treffen.

Diese Überlegungen und der Wunsch, die eigenen Fähigkeiten zu verbessern, schneller und effizienter zu agieren, bildeten die Inspiration für diese Arbeit. In vielen Büchern wird das Thema als *technical ear training* bezeichnet und soll dem / der Übenden eine Möglichkeit bieten, das Gehör speziell für die Aufgabenbereiche eines Toningenieurs / einer Toningenieurin oder eines Tontechnikers / einer Tontechnikerin auszubilden, um den täglichen Herausforderungen gewachsen zu sein. Mit Hilfe eines Übungsprogramms kann das Training gezielt durchgeführt werden und dient als Vorbereitung zur professionellen Anwendung in *live-, studio-* oder *rehearsal-* Situationen. Dabei werden oft drei Themengebiete als Grundpfeiler der auditiven Wahrnehmung genannt: Klangfarbenbeeinflussung, Dynamikbearbeitung und räumliches Hören. Da jedes Thema für sich schon sehr umfangreich ist, wurde zur Umsetzung der Software, sowie zur Erarbeitung der Theorie eine Aufteilung auf wiederum drei Personen gefunden.

Klangfarbenbeeinflussung:	Martin Czuka
Dynamikbearbeitung:	Raphael Kapeller
Räumliches Hören:	Josef Kulmer

In dieser Arbeit wird das Themengebiet der Dynamikbearbeitung mit Fokus auf Kompressoren und Lautheit behandelt. Die Dynamikbearbeitung ist ein wichtiger Bestandteil der Umsetzung musikalischer Ideen. Sie unterstützt den Ausdruck

einer Darbietung, fügt dramatische Momente hinzu und ist Sprachrohr der musikalischen Botschaft. [16] Der praktische Teil besteht aus der Entwicklung eines durch MATLAB® von MathWorks® gesteuerten Programms, das drei grundlegende Übungen zur Verbesserung der Wahrnehmung von Dynamikbearbeitungen beinhaltet. Im Vordergrund steht dabei die Sensibilisierung auf die unterschiedlichen Klangeigenschaften und Effekte durch veränderte Parametereinstellungen eines Kompressors. Vorbereitend dazu ist eine Übung zur Unterscheidung von Schalldruckpegeln implementiert. Außerdem wird auf die Realisierung in beiden Entwicklungsebenen, MATLAB® als Steuerung und *Pure Data* (*PD*) als in Echtzeit rechnendes Ausführungsprogramm, eingegangen. Im Abschnitt davor geben die beiden theoretischen Kapitel über Lautheit und Kompressoren einen Einblick in die Materie und vermitteln so ein grundlegendes Verständnis.

Kapitel 2 beschäftigt sich mit dem psychoakustischen Begriff der Lautheit und der damit verbundenen Wahrnehmung unterschiedlicher Audiosignale durch den Menschen. Grundlage dafür ist das Lautheitsmodell nach Zwicker.

In Kapitel 3 werden Kompressoren und ihre Anwendungsbereiche, sowie Parameter vorgestellt. Das Hauptaugenmerk liegt dabei auf der digitalen Umsetzung von Kompressoren.

Kapitel 4 enthält Erklärungen und Beschreibungen zu den realisierten Übungen, sowie deren Umsetzung in *Pure Data* und MATLAB®.

In Kapitel 5 erfolgt ein Ausblick auf mögliche Erweiterungen und fasst die Arbeit zusammen.

2 Psychoakustik

Die Psychoakustik ist ein interdisziplinäres Forschungsfeld der Psychologie, der Physik, der Medizin (Audiologie), der Biologie und der Ingenieurswissenschaften. Sie beschäftigt sich mit der Beziehung zwischen akustischer Anregung und der Hörempfindung des menschlichen Gehörs. Begriffe wie Lautheit, Tonhöhe, Rauheit, Schärfe, Schwankungsstärke, Rhythmus, Maskierung oder binaurales Hören werden geprägt und erklärt.

Als Grundlage der Forschung dienen immer Versuche mit Testpersonen, anhand deren Ergebnisse und Mittelungen Aussagen getroffen werden. Als Methoden kommen unter anderem die *Methode of Adjustment* (Änderungsmethode), *Methode of Tracking* (Verfolgungsmethode), *Magnitude Estimation* (Größenbewertung), *Yes-No Procedure* (Ja-Nein-Verfahren), *Two Interval Forced Choice* (Zwei-Intervall Methode), *Adaptive Procedures* (Adaptiv-Verfahren) und *Comparison of Stimulus Pairs* (Vergleich von zwei Anregungspaaren) zum Einsatz.

2.1 Hörbereich

Der Hörbereich oder auch die Hörfläche gibt jenes Feld an, in dem sich ein Schallereignis befinden muss, um von einem normal hörenden Menschen wahrgenommen zu werden. Der Hörbereich ist abhängig von der Frequenz, sowie vom Schalldruckpegel der auf das Ohr treffenden Schallwelle.

Abb. 2.1 verdeutlicht den Hörbereich – die Ordinate zeigt den Schalldruckpegel und die Abszisse die Frequenz im logarithmischen Maßstab.

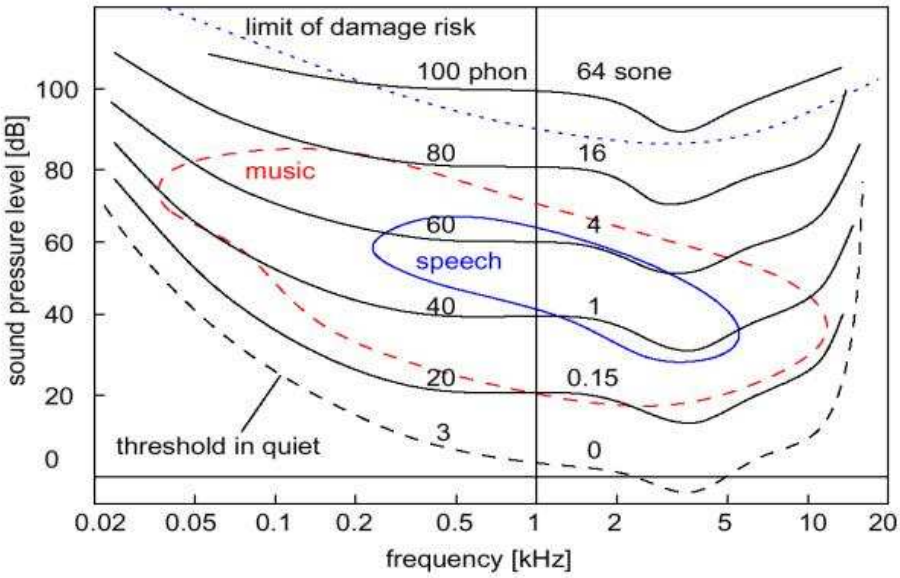

Abb. 2.1: Hörbereich eines normal hörenden Menschen in Abhängigkeit der Frequenz sowie des Schalldruckpegels. Schwarz strichliert: Hörschwelle; punktiert: Schmerzgrenze; durchgezogen: Kurven gleicher Lautheit bezogen auf einen 1kHz Testton.

Im Allgemeinen wird ein Frequenzbereich von 20Hz – 20kHz als wahrnehmbar für einen etwa 20-jährigen Menschen definiert. Mit zunehmendem Alter wird die Hörfähigkeit zu hohen Frequenzen hin eingeschränkt.

Ein Schalldruck von $20 \cdot 10^{-6}$Pa ist bei einer Frequenz von 1kHz gerade noch wahrnehmbar. Dieser Wert p_0 ist der Bezugswert für die Umrechnung des Schalldruckes in den Schalldruckpegel.

$$L = 20 \cdot \log\left(\frac{\tilde{p}}{\tilde{p}_0}\right)[dB]$$
$$L = 20 \cdot \log\left(\frac{\tilde{p}_0}{\tilde{p}_0}\right) = 20 \cdot \log(1) = 0[dB]@1kHz \quad (1)$$

Die schwarz strichlierte Linie in Abb. 2.1 zeigt die Hörschwelle, ab welcher eine Frequenz gehört werden kann. Wie aus der Grafik hervorgeht, ist das menschliche Gehör in einem Bereich zwischen 2kHz und 5kHz am empfindlichsten gegenüber dem Schalldruckpegel. Die Hörschwelle nimmt aber für tiefe Frequenzen stark zu, das heißt, man nimmt Frequenzen, die kleiner als

200Hz sind, erst bei höheren Schalldruckpegeln wahr. Die Hörempfindlichkeit im Frequenzbereich über 5kHz nimmt ebenfalls ab und unterliegt hier auch dem Alter der Testperson. *„[...] Im Alter von 60 Jahren ist die Hörschwelle bei 10kHz etwa um 20 dB, bei 5kHz etwa um 15dB angehoben, unter 2kHz gibt es keine Veränderungen; [...]“* (M. Dickreiter, „Handbuch der Tonstudiotechnik Band 1“ 6. Auflage, Sauer Verlag KG, München, 1997, pp. 110)

Die obere Grenze des Hörbereiches in Hinblick auf den Schalldruckpegel ist die so genannte Schmerzgrenze (*threshold of pain*). Sie liegt ebenfalls unterschiedlich hoch für verschiedene Frequenzen, kann aber mit einem Schalldruckpegel von 130dB als Mittelwert angenommen werden. Die punktierte Linie in der Grafik zeigt die Grenze zu Schalldruckpegeln, die einen bleibenden Schaden am Gehör verursachen können. In tiefen Frequenzen ist das Gehör auch gegen sehr hohe Schalldrücke nicht sehr empfindlich. Dabei ist allerdings die Zeit als die dritte Dimension in die Überlegungen miteinzubeziehen. Die Linie gilt als Richtwert für eine Beschallung von 8h an 5 Tagen pro Woche. Im Normfall wird eine Erhöhung des Schalldruckpegels auf 90dB oder 100dB in einem Zeitraum von 50 bzw. 5 Minuten keine Auswirkungen nach sich ziehen. Dies sind allerdings nur Richtwerte, an die man sich anlehnen kann und es muss klar sein, dass jede Person individuell reagiert. Eine Überstrapazierung des Gehörs führt zu temporären oder auch bleibenden Verschiebungen der Hörschwelle oder sogar zum Gehörverlust, der nicht behoben werden kann.

2.2 Lautheit

Lautheit ist einer der wichtigsten Begriffe in der Psychoakustik. Sie gehört zu den Intensitätsempfindungen und beschäftigt sich mit der Lautstärkeempfindung. Als einer der wichtigsten Forscher ist Eberhart Zwicker zu nennen, der die theoretischen Hintergründe zu Lautheit, unter Einbezugnahme von Maskierungen, spektralen und zeitlichen Effekten erklärte.

Wie oben erwähnt, werden (zum Beispiel) Sinustöne verschiedener Frequenzen unterschiedlich laut wahrgenommen. Eine anschauliche Darstellung dieser Tatsache ist durch die Kurven gleicher Lautheit gegeben, bei denen ein 1kHz-Sinuston als Bezugs- oder Richtwert dient. Ausgangspunkt der Untersuchung an Testpersonen ist also ein 1kHz-Sinuston, der mit einem gewissen Schalldruckpegel angeboten wird. Dieser wird dann mit einem zweiten Ton anderer Frequenz verglichen (*Loudness comparison*), bzw. wird der Zweite so eingestellt, dass er gleich laut erklingt wie der erste (*Methode of Adjustment*). Aus dieser Vergleichsprozedur erhält man dann die in Abb. 2.1 dargestellten Kurven gleicher Lautheit in phon für verschiedene Lautstärkepegel (L_N) im freien Feld unter frontal einfallendem Schall. Die Kurve für 3phon (entspricht 0sone) gibt die Hörschwelle an. Für den in der Praxis viel häufigeren Fall eines diffusen Schallfeldes müssen die Kurven gleicher Lautheit mit einer Dämpfung a_D angepasst werden. Die Messungen der Dämpfung werden nicht mit Sinustönen, sondern mit schmalbandigem Rauschen vorgenommen. Für tiefe Frequenzen ist die Dämpfung 0dB, da das menschliche Gehör im Bereich unter 200Hz eine omnidirektionale Höreigenschaft aufweist. Abb. 2.2 zeigt bei 1kHz eine Dämpfung von -3dB und ein Maximum von ca. +2dB bei 2,5 kHz.

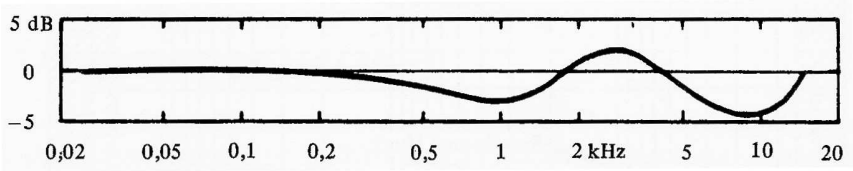

Abb. 2.2: Dämpfung a_D der Kurven gleicher Lautheit für diffuse Schallfelder in Abhängigkeit von der Frequenz

Mit dieser Darstellung können Schallereignisse gleicher Lautheit verglichen werden, jedoch kann nur schwer eine Aussage über das Verhältnis zweier ungleich großer Ereignisse gemacht werden. Dazu wurde das Sone-Maß eingeführt, wobei 1sone als der Pegel eines 1kHz Tones bei 40dB SPL[1] definiert wird. Zur Ermittlung der Verhältnisse werden Verdoppelungen und Halbierungen der wahrgenommenen Lautheit an einem Testton in Bezug auf einen

[1] Sound Pressure Level

Referenzton durchgeführt. Hier entsprechen 2sone der doppelten Lautheit von 1sone. Im Bereich über dem Schalldruckpegel von 40dB entspricht die Anhebung des SPL um 10dB einer Verdoppelung. Darunter steigt die Kurve logarithmisch stark an. Der geradlinige Teil der Kurve ist durch folgende mathematische Gleichung gegeben:

$$N = 2^{(L-40dB)/(10dB)}[sone] \quad (2)$$

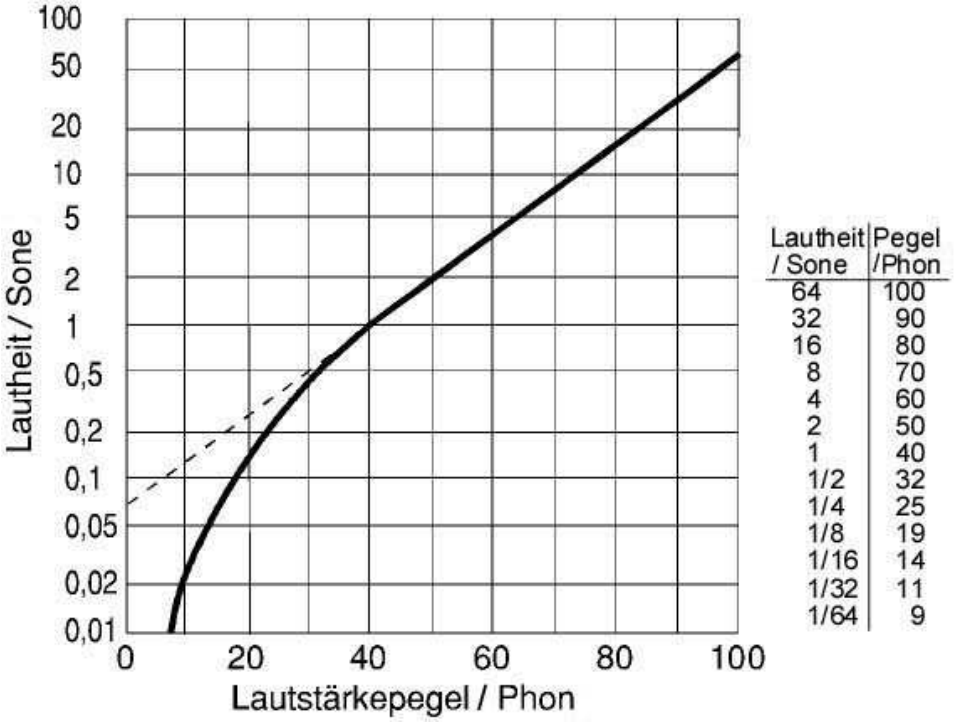

Abb. 2.3: Lautheitsfunktion eines 1kHz-Tones und eines UEN in Abhängigkeit der Frequenz sowie des Schalldruckpegels

Die Lautheitsfunktion kann berechnet werden und ist in Abb. 2.3 ersichtlich. Man erkennt, dass unter 40phon der Faktor der Lautheit stark abfällt. Einer Lautheit von 20phone entsprechen 0,15sone. Im oberen Bereich gilt die Regel, dass bei

einer Verdoppelung der Lautheit der Schalldruckpegel um 10dB steigen muss. Das heißt, 60phone ergeben 4sone, 80phone ergeben 16sone. Aus den Datensätzen der Lautheitsfunktion eines 1kHz-Sinustones verglichen mit der Lautheit von breitbandigem Rauschen (UEN – *uniform exciting noise*) kann die Lautheitsfunktion für breitbandiges Rauschen ermittelt werden. Die Lautheit von UEN wird wiederum mit der Anpassungsmethode ermittelt (Methode of Adjustment). Dabei liegt die Lautheitsfunktion von UEN über jener von einem 1kHz-Sinuston. Das bedeutet der Anstieg ist steiler für den Bereich von 3dB-60dB und geht dann wiederum in eine lineare Funktion über.

Bei dem zweiteiligen Experiment werden Testpersonen dazu aufgefordert, zuerst das so genannte variable Objektrauschen (UEN) so einzustellen, dass es gleich laut wie der Standardton (1kHz) ist. Im zweiten Teil des Versuches muss der Standardton gleich laut eingestellt werden wie das Objektrauschen. Dabei ist festzustellen, dass es bei den Einstellungen eine systematische Diskrepanz zwischen den Resultaten gibt. Man neigt dazu, die Variable lauter einzustellen als den Vergleichsklang, egal ob der 1kHz-Ton oder das UEN verändert wird. Ebenfalls geht aus diesem Experiment hervor, dass Rauschen mit dem selben Schalldruckpegel (gültig ab etwa 20dB SPL) viel lauter wahrgenommen wird als ein einzelner Sinuston.

2.2.1 Lautheitsmodell nach Zwicker

Um das Lautheitsmodell nach Zwicker zu verstehen, müssen das oben erwähnte frequenzselektive und von der Anregungslautstärke abhängige Hörverhalten des Menschen und der Begriff „kritische Bandbreite“ bekannt sein. Die Abhängigkeit der Lautheit von der kritischen Bandbreite kann auf drei verschiedene Arten dargestellt werden.

Die erste Form der Definition der kritischen Bandbreite ist auf die Herleitung der Lautheitsfunktion zurückzuführen. Bei diesem Versuch (siehe oben) werden ein

Sinuston und ein Objektrauschen mit gleich hohem Schalldruckpegel miteinander verglichen. Der große Unterschied der beiden Testsignale ist die Bandbreite, wobei die Bandbreite des UEN variiert werden kann und die des Sinustones gleich Null ist. Lässt man die Bandbreite des UEN von Null an steigen und vergleicht das Rauschen mit dem Sinussignal bezüglich der Lautheit, so ist zu erkennen, dass bis zu einer gewissen Breite Δf_G kein Unterschied wahrnehmbar ist und erst dann die Lautheit des Rauschens größer wird. Δf_G ist die kritische Bandbreite. Sie weist bei einem Sinuston von 1kHz die Breite von 160Hz auf.

Der zweite Weg, um die Bedeutung der kritischen Bandbreite zu zeigen, ist ein Versuch, bei dem erneut ein einzelner Sinuston (1kHz) als Referenzsignal mit einem Sinustonpaar verglichen wird. Alle Töne weisen einen Schalldruckpegel von 60dB auf. Das Referenzsignal ist äquidistant in der Mitte der beiden Sinustöne angeordnet. Das heißt, dass es die zentrale Frequenz zwischen Ton A und Ton B ist, wobei der Abstand der Töne A und B als Bandbreite Δf bezeichnet wird. Die Auswertung des Experimentes zeigt, dass für eine Bandbreite bis etwa 10Hz das Sinustonpaar einem 1kHz-Sinuston mit 66phon gleichgestellt wird. Der Schalldruck der beiden Töne wird addiert. Im Bereich der Bandbreite von 20 – 160Hz ergibt sich eine Lautheit von etwa 63phon, was einer Addition der Schallintensität entspricht. Ab 160Hz bis 2kHz (obere Grenze, da 1kHz als Mittenfrequenz gewählt: f_U=1kHz-1kHz=0Hz) steigt die Lautheit an und strebt gegen 70phon, was einer Erhöhung um 10phon und somit einer Addition der Lautheit entspricht.[2] Auch hier ist die kritische Bandbreite Δf_G = 160Hz, ab welcher die Lautheit steigt, für einen Testton von 1kHz erkennbar,

Zwicker zeigt die Abhängigkeit der Lautheit von der Bandbreite auf eine weitere Art, indem ein frequenzvariabler 60dB-Objektton simultan mit einem 65dB-

---

[2] $L = 20 \cdot \log\left(\frac{p_1}{p_0}\right)[dB] \Rightarrow Verdoppelung \rightarrow +6dB$

$L_I = 10 \cdot \log\left(\frac{I}{I_0}\right)[dB] \Rightarrow Verdoppelung \rightarrow +3dB$

Hochpassrauschen bei 1kHz abgespielt wird, um dann mit einem 1kHz-Referenzton verglichen zu werden. Der Abstand zwischen dem Objektton und dem Hochpassrauschen wird als Bandbreite Δf bezeichnet. Obwohl der variable 60dB-Ton nicht im Frequenzbereich des Hochpassrauschens liegt, wird er maskiert, das heißt, seine Lautheit wird verringert. Für große Bandbreiten ist die Maskierung nicht vorhanden, doch schon für Δf=100Hz wird die Lautheit des Objekttones halbiert.

Aus der Tatsache heraus, dass sich zwei Töne gegenseitig beeinflussen und so die Lautheit formen, müssen die Einflüsse der Hörkurven, sowie die Einflüsse der kritischen Bandbreite berücksichtigt werden. Daher wird das Integral der neu eingeführten spezifischen Lautheit N' über die Tonheit z zwischen den Grenzen von 0 bis 24Bark definiert. Bark ist die Einheit der Tonheit und gibt aufgrund von Hörversuchen der Tonhöhe ein Bezugsmaß, ähnlich wie die Lautheit, an. Die Tonheit beurteilt das Tonhöhenverhältnis zur Frequenz in psychoakustischen Maßstäben.

Das Modell nach Zwicker besagt:

$$N = \int_{0}^{24\,Bark} N' \, dz \qquad (3)$$

wobei N die Lautheit und N' die spezifische Lautheit ist.

2.2.2 Lautheitsmodell nach Moore und Glasberg als Weiterentwicklung des Modells nach Zwicker

Das überarbeitete Modell von Moore und Glasberg ist dem von Zwicker sehr ähnlich, weist aber einige Neuerungen und leicht veränderte Ergebnisse auf.

Die Umrechnung der Lautheit von sone in phon erfolgt ebenfalls durch die empirisch ermittelten Daten eines Hörversuches mit einem 1kHz-Sinuston. Die Kurven werden jedoch anhand einer *cubic spline*[3]-Interpolation ermittelt. Ein Kritikpunkt am Modell von Zwicker ist, dass die Berechnungen der Kurven gleicher Lautheit nicht mit den neuerlich empirisch ermittelten festgehaltenen Werten der Kurve für den Frequenzbereich unter 1kHz übereinstimmen. Dabei ist die mittlere absolute Abweichung bei Zwicker 1,6dB und bei dem überarbeiteten Modell 0,6dB. Außerdem beinhaltet das Modell von Zwicker aufgrund des Rechenvorganges Unstetigkeiten, die in den empirischen Daten nicht vorhanden sind und somit einen größeren Fehler beinhalten.

Des Weiteren wurde im Modell nach Moore und Glasberg die Übertragungsfunktion des Außen- und des Mittelohrs überarbeitet, sowie die Abhängigkeit der kritischen Bandbreite näher beleuchtet.

[3] Kurve 3.Ordnung

3 Kompressoren

Der Kompressor gehört zur Gruppe der Regelverstärker und wird zur Optimierung der Übertragung sowie zur Klanggestaltung eingesetzt. Bei einem Regelverstärker ist der Ausgangspegel vom Grad der Verstärkung bzw. der Verringerung des Eingangspegels abhängig. Anhand einer statischen Kennlinie, bei der sich der relative Eingangspegel an der Abszisse und der relative Ausgangspegel an der Ordinate gegenüberstehen, wird die so genannte *gain reduction* definiert. Unter diesem Begriff versteht man das Maß in dB, um welches das Eingangssignal verringert wird. Ohne Eingriff eines Regelverstärkers ist hier die statische Kennlinie eine 45°-Gerade (Eingang = Ausgang). In dieser Arbeit sollen nur „*downward*"-Kompressoren besprochen werden. Diese zeichnen sich dadurch aus, dass laute Stellen eines Signals abgeschwächt werden, um so mit den leiseren Passagen homogener zusammenzuklingen.

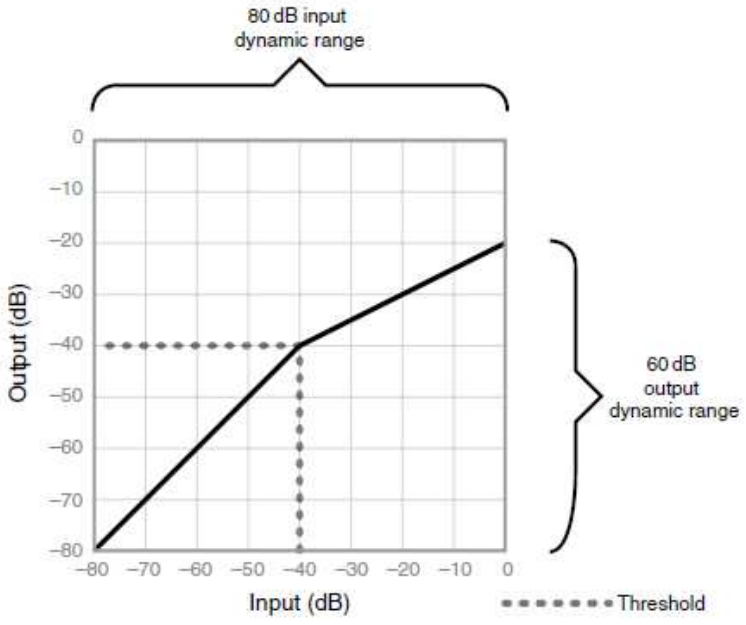

Abb. 3.1: Statische Kennlinie eines Kompressors. 45° Gerade bis zum Threshold. Danach Bedämpfung mit einer *ratio* von 2:1.

Mit Hilfe der Parameter *threshold, ratio, attack time, release time, hold* und *make-up gain* werden die Einstellungen des Kompressors verändert. Die statische Kennlinie übernimmt die Eigenschaften und illustriert das Verhältnis zwischen Aus- und Eingangspegel.

Anfänglich kamen Kompressoren in der Radioübertragung zum Einsatz. Sie wurden zum Schutz gegen zu hohe Ausgangspegel bei Live-Shows (z.B. Sportmoderationen) verwendet, da jeder Kanal eine definierte obere Grenze („*peak limit*“) aufweist, die nicht überschritten werden darf, da es ansonsten zu Verzerrungen und *clippings* kommt. Eine weitere wichtige Aufgabe ist die Einengung der Dynamik. Durch die Verdichtung des Signals wird bei Sendungen im Lang-, Mittel- oder Kurzwellenbereich die Sendereichweite vergrößert, da leise Passagen nicht mehr durch den Rauschanteil in der Übertragungskette maskiert werden. Ein Signal wird durch die Komprimierung verständlicher und in den Vordergrund gerückt – das Signal bekommt mehr Fülle. Analoge Geräte färben außerdem das Klangbild eines Signals, wodurch der Kompressor als Werkzeug zur Klanggestaltung eine bedeutende Rolle innehat. Dieser Effekt wird vor allem in der U-Musik häufig verwendet.

Dabei spielt die technische Realisierung eine große Rolle. Analoge Geräte weisen eine Vielzahl an unterschiedlichen Möglichkeiten zur Umsetzung mit verschiedensten, daraus resultierenden Eigenschaften auf. Die ersten Kompressoren wurden mit Hilfe der Röhrentechnologie (*variable-mu tubes/valves*) gefertigt. Weitere Entwicklungen wie *field effect transistors* (FET), *optical compressor controls* (Opto) und *voltage-controlled amplifiers* (VCA) folgten und sind heute noch beliebte Kompressoren. In der weiterführenden Arbeit, speziell in Kapitel 3.2, werden ausschließlich digitale Kompressoren behandelt.

Im Folgenden sollen die Parameter zur Einstellung eines Kompressors behandelt werden.

3.1 Parameter

3.1.1 Threshold

Der *threshold* (dt. Schwellenwert) gibt an, ab welchem dB-Wert die Kompression einsetzen soll. Je mehr das Signal den *threshold* übersteigt, desto stärker wird verdichtet (dies setzt eine *ratio* ungleich 1:1 voraus). In der Praxis gibt es zwei Arten von Kompressoren, und zwar die Variante mit variablem *threshold*, bei der der Regler bestimmt, wann die Kompression beginnt, und eine Ausführung mit fixem *threshold*, bei der das Inputsignal verstärkt oder abgesenkt wird, um so in die Kompression „hineinzufahren".

3.1.2 Ratio

Die *ratio* (dt. Größenverhältnis) gibt jenen Wert an, um welches Verhältnis das den *threshold* übersteigende Signal abgeschwächt werden soll. Einstellungen wie 2:1, 6:1, 8:1, ∞:1 werden häufig verwendet. Übersteigt ein Signal den *threshold* um 10dB, so wird bei einer *ratio* von 2:1 das Outputsignal nur noch 5dB über dem *threshold* liegen. Die *ratio* gibt die Steigung der statischen Kennlinie ab dem *threshold* an. Es gilt folgende Formel:

$$R = \frac{\Delta P_I}{\Delta P_O} \tag{4}$$

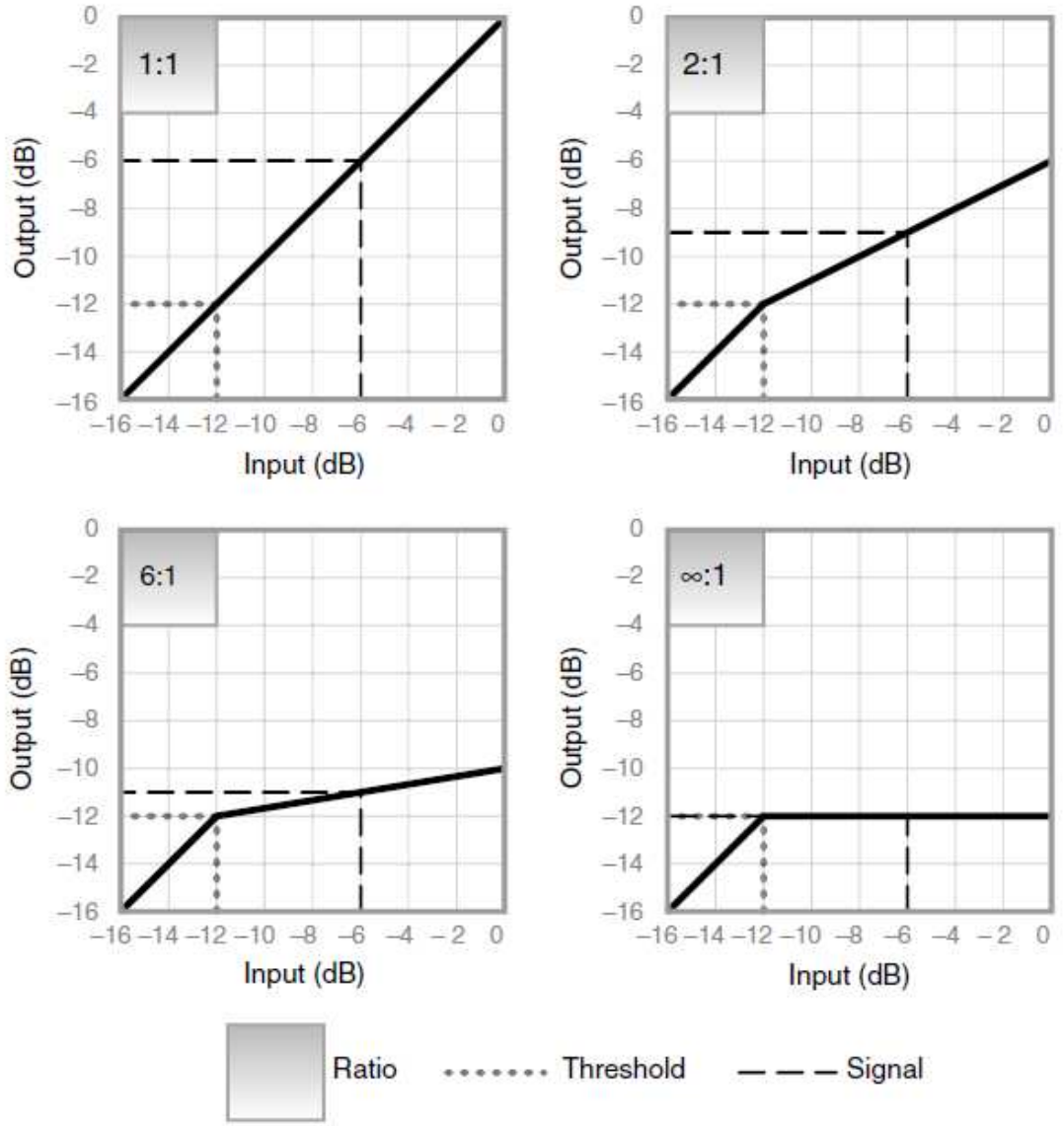

Abb. 3.2: Statische Kennlinie mit verschiedenen *ratio settings.*

3.1.3 Attack time und release time

Als *attack time* oder auch Einschwingzeit bezeichnet man jene Dauer, die benötigt wird, um die Kompression einsetzen zu lassen. Die *release time* beschreibt dasselbe für die Ausschwingzeit nach Unterschreitung des *threshold*. Werden beispielsweise beide Zeitkonstanten auf 1ms gesetzt, so ist eine sofort steigende bzw. fallende *gain reduction* zu erwarten. Die Zeitkonstanten werden üblicherweise in Millisekunden (ms) eingegeben. Typische Werte für die *attack*

time sind 0,5ms – 250ms und 5 – 3000ms für die *release time*. Die beiden Zeitkonstanten nehmen großen Einfluss auf die Klanggestaltung des Signals. Mit einer etwas längeren *attack time* bleibt zum Beispiel der Einschwingvorgang eines Instrumentes unverändert und nur *sustain* und Ausschwingvorgang werden komprimiert. Dies wird etwa bei Bassdrums eingesetzt, um den Kick zu betonen und dem Gesamtklang trotzdem Fülle und „Bauch“ zu geben. Bei zu kurzen *attack times* besteht die Gefahr, dass das Signal unnatürlich klingt. Auch bei der *release time* besteht diese Gefahr. Wenn die Zeit zu kurz gewählt ist, kommt es zu sprunghaften Änderungen der wahrgenommenen Lautstärke. Bei einer zu langen Zeitkonstante wiederum wird ein leises Signal, das unmittelbar auf ein lautes folgt, noch immer komprimiert.

Von vielen Herstellern wird allerdings eine Hilfestellung angeboten, die ein schnelles Arbeiten ohne Nachjustierung erlaubt und für viele Bereiche gut einsetzbar ist. Mit dem *auto attack button* und dem *auto release button* werden die Zeitkonstanten variabel zur jeweiligen Zeit automatisch eingestellt. Dabei verlieren der *attack*- und *release-Regler* ihre Wirkung. Ein weiterer wichtiger Parameter, der allerdings nicht in allen Kompressoren vorhanden ist, ist das *timing law* (dt. Zeitgesetz). Es besagt, dass das Einsetzen der *gain reduction* linear oder exponentiell erfolgen soll. Hier gilt ein natürlicher Klang bei exponentiellem Einsetzen, während ein lineares Einsetzen den Klang färbt.

3.1.4 Hold

Eine dritte Zeitkonstante ist die so genannte *hold time*. Sie steht zwischen der *attack*- und *release time* und besagt, wie lange die *gain reduction* aufrecht erhalten bleiben soll, wenn der *threshold* unterschritten wird, bevor die *release time* einsetzt.

3.1.5 Make-up gain

Die *gain reduction* macht das Signal im ersten Schritt leiser. Um jedoch wieder zum Arbeitspegel zu gelangen, bzw. um das Bypass-Signal mit dem komprimierten Signal besser vergleichen zu können, stellt man mit dem *make-up gain* die Stärke des Signals ein. Diese Funktion beeinflusst sowohl alle Passagen, die nicht komprimiert werden, als auch jene, die komprimiert werden.

3.1.6 Hard and soft knee

Das *knee* (dt. Knie) bezeichnet jene Stelle der statischen Kennlinie, an der die *gain reduction* einsetzt. Bisher wurde in dieser Arbeit nur das *hard knee* behandelt. Bei diesem befindet sich der Übergang unmittelbar nach dem Über- bzw. Unterschreiten des *threshold*. Das so genannte *soft knee* ermöglicht einen subtileren Übergang, bei dem je nach Einstellung zum Beispiel 4dB unter dem *threshold* schon die Kompression beginnt, die *ratio* aber kontinuierlich gesteigert wird, bis sie 4dB über dem *threshold* die eingestellte *ratio* erreicht (Bandbreite=8dB). Aus diesem Grund wird das *soft knee* auch *soft-ratio* oder *over-easy* genannt.

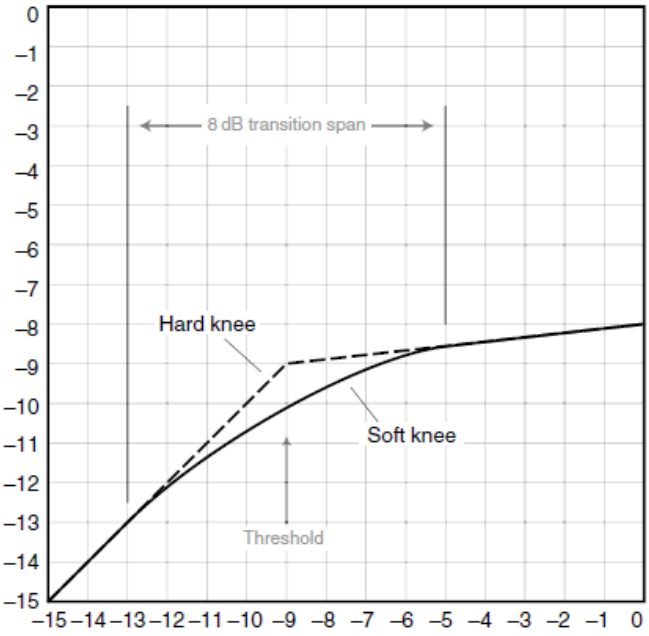

Abb. 3.3: *Hard knee* und *soft knee* mit einer 8dB-Breite im Vergleich

3.1.7 Side chain

Der *side chain*-Eingang ermöglicht, ein Signal in Bezug auf eine andere Quelle zu komprimieren. So kann ein Musikstück mit dem Signal eines Sprechers komprimiert werden, sodass, sobald der Sprecher ein Signal in den *side chain* schickt, das Musikstück leiser wird (*Ducking* – wird im Radio oder bei DJ-Sets angewendet). Eine andere Möglichkeit ist, eine Bassdrum mit dem Signal eines Basses, oder umgekehrt, zu komprimieren.

3.1.8 Peak- und RMS-Level Measurement

Grundsätzlich kann der Eingangspegel in einem Kompressor auf zwei Arten gemessen werden. *Peak-level measurement* dient vorwiegend für Signale mit perkussivem Charakter oder zur Sicherung gegen ein zu hohes Outputsignal (Limiterfunktion). Die Kompression wird abhängig von den Signalspitzen angewendet, daher ist eine sehr schnelle Reaktion der *gain reduction* zu erwarten, die deutlich hörbar ist. Die RMS[4]-Methode ist eine „dezentere" Anwendung, mit deren Hilfe man zum Beispiel Vocals transparenter, durchsetzungsfähiger und lauter gestalten kann.

3.2 Digitale Kompressoren

Mit den heute zur Verfügung stehenden Rechenleistungen digitaler Systeme werden Kompressoren, wie auch fast alle anderen analogen audiosignalbearbeitenden Geräte, digital realisiert. Im folgenden Teil soll die grundsätzliche Vorgehensweise zur Implementierung digitaler nichtlinearer Prozesse beleuchtet werden.

Abb. 3.4 zeigt die wichtigsten Bausteine des Systems. Dabei stellt x(n) den Input und y(n) den Output dar. Das *level measurement* dient zum Messen des Eingangspegels und kann in *peak* oder RMS erfolgen, je nachdem welche

[4] Root Mean Square

Funktion der Kompressor ausführen soll. Im Block *static curve* werden alle Einstellungen bezüglich *threshold*, *ratio* und *make-up gain* vorgenommen. Die Parameter *attack*- und *release time* werden separat vorgenommen. Wie man in Abb. 3.4 sehen kann, wird das Eingangssignal x(n) oder $X_{dB}(n)$ (Pegel) abgegriffen um eine Gewichtungsfunktion g(n) oder $G_{dB}(n)$ zu ermitteln, die mit dem tatsächlichen Signal multipliziert wird. x(n) muss verzögert werden, um synchron mit der Gewichtungsfunktion zu sein (Delay). Daher gilt folgende Formel:

$$y(n) = g(n) \cdot x(n - D) \qquad \text{bzw. für Pegel in dB}$$
$$Y_{dB}(n) = X_{dB}(n) + G_{dB}(n) \qquad (5)$$

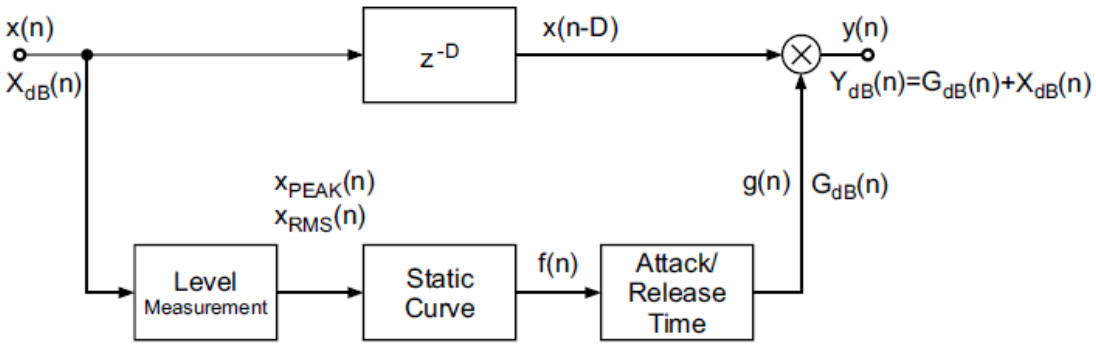

Abb. 3.4: Blockschaltbild eines digital realisierten Dynamik-Prozessors. Das Eingangssignal x(n) wird im unteren Zweig gemessen (Pegel); der Wert mit der statischen Kennlinie gewichtet und anhand der Zeitkonstanten *attack*- und *release time* bearbeitet. Im oberen Zweig wird das Eingangssignal verzögert. Anschließend wird das verzögerte Signal mit der Gewichtungsfunktion $G_{dB}(n)$ multipliziert (=Output $Y_{dB}(n)$).

3.2.1 Level Measurement

Wie oben erwähnt, gibt es zwei Möglichkeiten, um das Eingangssignal zu messen. Die Charakteristika und Einsatzzwecke wurden in Kapitel 3.1.8 für *peak*- und RMS-*measurement* besprochen. Hier soll der mathematische Hintergrund beleuchtet werden.

Beim *peak*-Verfahren wird der Absolutbetrag des Eingangs mit dem Spitzenbetrag (*peak value*) verglichen, und je nachdem, ob $x_{peak}(n)$ größer bzw.

kleiner ist, mit der *attack time* (AT) bzw. *release time* (RT) multipliziert. Für die beiden Fälle $|x(n)|>x_{peak}(n-1)$ bzw. $|x(n)|\leq x_{peak}(n-1)$ gelten folgende Herleitungen.

Attack time:

$$x_{peak}(n) = (1-AT)\cdot x_{peak}(n-1) + AT\cdot|x(n)|$$
$$H(z) = \frac{AT}{1-(1-AT)\cdot z^{-1}} \qquad (6)$$

Release time:

$$x_{peak}(n) = (1-RT)\cdot x_{peak}(n-1)$$
$$H(z) = \frac{1}{1-(1-RT)\cdot z^{-1}} \qquad (7)$$

Die zweite Möglichkeit des *level measurement* ist die RMS-Methode, bei welcher der Betrag mit Hilfe der Wurzel der Summe der Quadrate der n-Teile des Signals x errechnet wird.

$$x_{RMS}(n) = \sqrt{\frac{1}{N}\sum x^2(n-i)} \qquad (8)$$

Die Mittelung wird mit einem Tiefpassfilter erster Ordnung mit dem Koeffizienten TAV[5] realisiert, wobei t_M aus den Berechnungen der Zeitkonstante stammt.

$$TAV = 1-\exp\left(\frac{-2{,}2\cdot T_A}{t_M/1000}\right) \qquad (9)$$

Das Prinzip des Blockschaltbildes ist wieder dasselbe wie bei der *peak*-Messung, daher sind auch die anzuwendenden Formeln von ähnlicher Gestalt.

[5] TAV – time averaging coefficient

$$x_{RMS}^2(n) = (1-TAV)\cdot x_{RMS}^2(n-1) + TAV \cdot x^2(n)$$
$$H(z) = \frac{TAV}{1-(1-TAV)\cdot z^{-1}} \tag{10}$$

3.2.2 Zeitkonstanten

Die Zeitkonstanten werden aus der Sprungantwort des vorhandenen Systems berechnet und steuern das Verhalten eines Kompressors. Als Beispiel soll eine Sprungantwort eines zeitinvarianten Systems mit

$$g(t) = 1-\exp\left(\frac{-t}{\tau}\right) \Rightarrow g(nT_S) = 1-\exp\left(\frac{-nT_S}{\tau}\right) = 1-z_\infty^n \tag{11}$$

angenommen werden (nT_S ist die Darstellung der diskreten Sprungantwort – Sampling). Mit Hilfe der Z-Transformation wird die diskrete Sprungantwort in den Bildbereich (Spektrum) gebracht,

$$G(z) = \frac{z}{z-1} - \frac{1}{1-z_\infty \cdot z^{-1}} = \frac{1-z_\infty}{(z-1)(1-z_\infty \cdot z^{-1})} \tag{12}$$

und durch die Übertragungsfunktion

$$H(z) = \frac{z-1}{z} G(z) = \frac{(1-z_\infty)z^{-1}}{1-z_\infty z^{-1}} \tag{13}$$

wird die Polstelle z_∞ errechnet.

$$z_\infty = \exp\left(\frac{-2{,}2\cdot T_S}{t_a}\right) \tag{14}$$

z_∞ stellt die *attack*- bzw. *release time* ein.

3.3 Anwendungsbereiche von Kompressoren

Im vorangegangenen Kapitel wurde der Kompressor definiert, und erklärt, welche Parameter verändert werden können. Des Weiteren wurde ein kurzer Einblick in analoge und digitale Umsetzungen gegeben. Zum Abschluss sollen noch Beispiele für die verschiedenen Anwendungsmöglichkeiten der Dynamik-Bearbeitung gegeben werden. In Roey Izhakis Buch „Mixing Audio“ [5] findet man folgende Zusammenfassung:

“The following list is a summary of compressor applications:

- *Making sounds bigger, denser, fatter*
- *Accentuating the inner details of sounds*
- *Adding warmth*
- *Containing levels*
- *Balancing levels*
- *Loudening*
- *Softening or emphasizing dynamics*
- *Reshaping dynamic envelopes:*
- *Adding punch*
- *Accenting the natural attack*
- *Accenting transients*
- *Reviving transients*
- *Reconstructing lost dynamics*
- *Emphasizing or de-emphasizing decays*
- *Bringing instruments forward or backward in the mix*
- *Making programmed music sound more natural*
- *Applying dynamic movement*
- *Ducking*
- *De-essing*
- *Applying dynamic equalization“*

3.3.1 Loudening

Ein wichtiges Einsatzgebiet ist das Erhöhen der Lautheit. Dies wird zumeist erst im Masteringprozess angewendet, um dem Stereomix mehr Kraft zu verleihen. Zum Beispiel kann dadurch ein Instrument mit wenig Durchsetzungsvermögen durch geringe Kompression in den Vordergrund gebracht werden. Die Wahl sollte auf ein *RMS-level measurement* fallen, da diese Anwendung ein weniger aufdringliches Verhalten erfordert. Außerdem ist ein sehr breites *soft knee* von Vorteil, um den Klang so natürlich wie möglich zu belassen. Der *threshold* sollte so gewählt werden, dass er mit der halben *soft knee*-Breite unter 0dB zum Liegen kommt. Das bedeutet bei Einstellen einer *ratio* von ∞:1 (Limiting-Funktion), dass bei 0dB das Signal vollständig begrenzt wird, und darunter das *soft knee* die variable Kompression übernimmt. Dabei müssen das *make-up gain* und der *threshold* den gleichen Betrag mit unterschiedlichen Vorzeichen aufweisen, um die *gain reduction* auszugleichen. Die Einstellung der *attack*- und *release time* ist abhängig davon, ob man einen perkussiveren Klang (längere Zeiten) für das Mastering anwenden möchte, oder nicht. Mit einer geringeren *ratio* bleibt die Dynamik besser erhalten, doch auch hier ist es eine Frage des Geschmacks und der Erfahrung, wie die Einstellungen miteinander zum gewünschten Ergebnis führen.

3.3.2 Bearbeitung des Dynamikumfanges – Hüllkurve

Wie oben erwähnt, kann man mit der richtigen Kompressoreinstellung den Einschwingvorgang (perkussiv, „punch“) verstärken oder abschwächen. Mit Hilfe einer langen *attack time* bleibt das natürliche Einschwingen bestehen und die *gain reduction* setzt erst später ein. Dieser Effekt kann durch eine lange r*elease time* noch weiter hervorgehoben werden, da der Ausschwingvorgang ebenfalls komprimiert und dadurch leiser gemacht wird. Der umgekehrte Effekt ist mit kurzen *attack*- und *release*-Zeiten zu erreichen. Dadurch wird der „punch“ weggenommen, der *sustain* des Signals aber erhöht.

3.3.3 De-essing

Speziell bei Gesangaufnahmen kommt es oft vor, dass S-Laute oder so genannte Frikativlaute, die zumeist in einem Frequenzbereich zwischen 4kHz und 8kHz zu finden sind, überzeichnet dargestellt werden. Um dem entgegen zu wirken, kann man einen De-Esser zum Einsatz bringen, indem man das Signal aufteilt und zusätzlich in den *side-chain*-Eingang einspeist. Zuvor wird das Signal jedoch mit Hilfe eines Equalizers so bearbeitet, dass die S-Laute verstärkt werden. Der Kompressor wird nun durch das *side-chain*-Signal auf die Frikativlaute „getriggert“ und glättet somit deren Auftreten.

4 Realisierte Übungen

In den beiden vorangegangen Kapiteln („Psychoakustik“ und „Kompressoren“), wurde der theoretische Hintergrund zu den praktischen Anwendungen der Gehörbildungssoftware beleuchtet. In diesem Abschnitt sollen nun die implementierten Aufgaben vorgestellt werden.

Folgende drei Grundthemen werden dabei aufgegriffen und in Übungsaufgaben umgesetzt: *Loudness, Attributes* und *Adjustment*. Dabei besteht jede Übung aus zehn Fragen. Die Antworten können direkt nach jeder Frage eingesehen werden und werden zusätzlich am Ende der Übung als SCORE ausgegeben.

Die Realisierung wird einerseits mit dem Programm MATLAB® von MathWorks® und andererseits mit dem Programm *Pure Data*, im Weiteren PD genannt, vorgenommen. Dabei wird mit MATLAB® die Steuerung von PD übernommen und die graphische Benutzeroberfläche (GUI – graphical user interface) erstellt. Über ein TCP[6] Protokoll werden dabei Parameter von MATLAB® an PD geschickt, die in Echtzeit umgesetzt werden und das wiedergegebene Signal bearbeiten.

4.1 Allgemeines

In diesem Kapitel sollen jene Teile des Programms erklärt werden, auf deren Basis alle drei Übungen aufbauen.

[6] Transmission Control Protocol

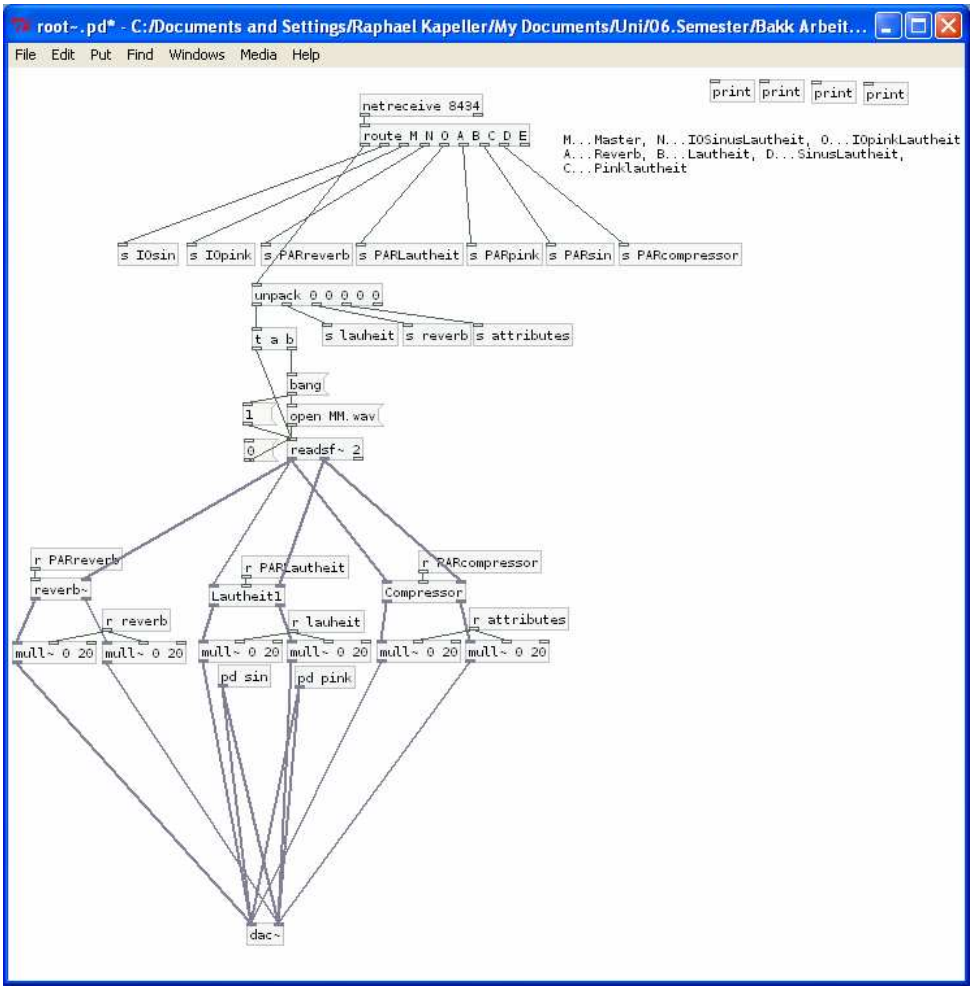

Abb. 4.1: *root~-patch* in PD als Grundlage zu allen weiteren Bearbeitungen und Zusammenführung aller *subpatches*.

4.1.1 Allgemeine technische Umsetzung in PD

Abb. 4.1 zeigt den Hauptpatch *root~.pd*, der alle *subpatches* zusammenführt, die Parameter entpackt und Musikaufnahmen öffnet. Den Eingang bildet dabei der Befehl *netreceive 8434,* der es erlaubt, über ein TCP-Protokoll Daten von anderen Programmen zu empfangen. Hier werden die Parameter von MATLAB über die *portnumber* 8434 gesendet. Der darauf folgende Befehl *route* ist der eigentliche Schnittpunkt zwischen den beiden Programmen, da hier alle Parameter übergeben und sortiert werden. Die übertragenen Daten sind zu einem *string* gepackt, dessen erstes Zeichen immer einen Buchstaben beinhaltet, der die Sortierung und somit das Ansprechen von verschiedenen

Anwendungen ermöglicht. Nachstehende Tabelle gibt Auskunft über die verwendeten Nomenklaturen.

Name	**Beschreibung**
M	Master – *.wav-Files* werden geöffnet, abgespielt, gestoppt
N	IOSinusLautheit – Ein- und Ausschalten des Sinusoszillators
O	IOPinkLautheit – Ein- und Ausschalten des generierten Rosa Rauschen
A	Parameter Reverb
B	Parameter Lautheit1
C	Parameter Lautheit1 bei abgespieltem Rosa Rauschen
D	Parameter Lautheit1 bei abgespielten Sinustönen
E	Parameter Compressor

Tab. 1: Nomenklaturen der verwendeten Indizes zur Unterscheidung und Steuerung der Übungen in PD

Nach der Aufteilung der eingegangenen Signale werden diese mit *send*-Befehlen intern in PD an die jeweiligen *return*-Befehle weitergesendet, um dort die Zahlenwerte der Variablen wie *soundstatus, mull, deziBell, threshold, ratio, attack time, release time, make-up gain*,[7] usw. zu übergeben.

Der als *master* bezeichnete s*tring* beinhaltet an erster Stelle die Variable *soundstatus,* die das Starten bzw. Stoppen des Befehles *readsf~* und das Öffnen des Soundbeispieles steuert. Die folgenden Variablen mit dem Überbegriff *mull* dienen dazu, den Verstärkungsfaktor von 1 beim Starten nicht diskret auszuführen, sondern einen kontinuierlich ansteigenden Verlauf des Verstärkungsfaktors innerhalb von 20ms zu implementieren. Dadurch werden Störgeräusche weitestgehend umgangen. Dabei ist zu beachten, dass nur jenes Zeichen im String eine 1 ist, das den Befehl *mull~* [8] des zu diesem Zeitpunkt

[7] Die Aufgaben der genannten Variablen werden in den Erklärungen zu den technischen Umsetzungen der Übungen erläutert.

[8] mull~ © Thomas Musil 2000-2005, musil@iem.at, IEM KUG Graz, Austria

verwendeten *subpatch* in PD steuert. Die restlichen Zeichen müssen auf Null gesetzt sein, da diese alle Signale aus anderen *patches* in PD abschalten.

Im unteren Drittel des *root.pd-patch* stehen die *subpatches* zur Dynamikbearbeitung (*Lautheit1.pd* und *Compressor.pd*) sowie jene zur räumlichen Abbildung (*reverb~.pd*) von Josef Kulmer und zur Klangfarbenbeeinflussung (*EQ~.pd)* von Martin Czuka, die alle über den oben erwähnte *mull*~-Befehl an den Ausgang gelegt werden.

4.1.2 Allgemeine technische Umsetzung in MATLAB®

Die gesamte Signalverarbeitung des Übungsprogramms wurde in PD gelöst, doch um eine tatsächliche Übung programmieren zu können, ist man auf eine andere Arbeitumgebung angewiesen. Wie einleitend beschrieben, wurden das *grafical user interface* (GUI), der Übungsablauf, die Auswertung, das Berechnen der Parameter, sowie die ausführende Datei (.exe) in MATLAB® erstellt. Hier soll nur ein kurzer Überblick über die verwendeten Funktionen gegeben werden.

berechneNeu() berechnet mithilfe von Zufallszahlen die Werte der Parameter entweder direkt, oder erstellt einen *pointer,* der die Auswahl der Parameter generiert.
auswertung() vergleicht, ob die vom Benutzer eingegebenen Werte oder Antworten mit den Lösungen übereinstimmen und gibt am Ende den erreichten SCORE bei 10 Versuchen an.

pd_aktualisieren() sendet die benötigten Daten an PD.

Weitere Funktionen, die durch Betätigen der *push-buttons* oder *radio-buttons* in der Bedienoberfläche aktiviert werden, sind grundsätzlich in drei Gruppen einzuteilen:

Playing Back Audio:	Stimulus (A,B), Referenz, Adjustment, Attributes, Stop
Answers:	alle *radiobuttons*
Exercise Controlls:	weiter, Antwort zeigen, abbrechen

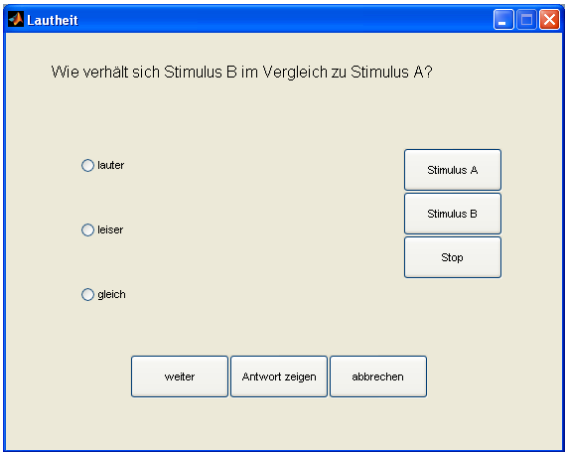

Abb. 4.2: Generiertes GUI der Übung Lautheit

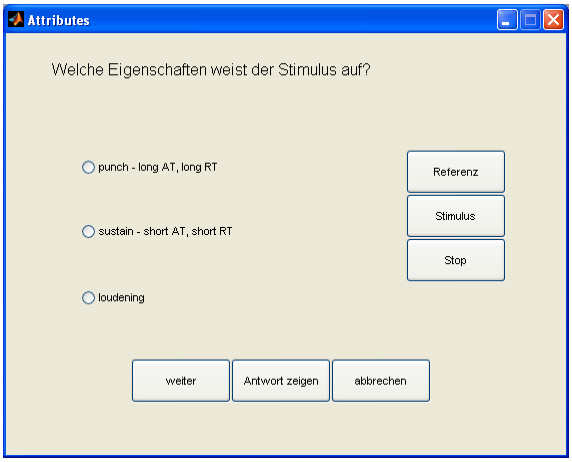

Abb. 4.3: Generiertes GUI der Übung

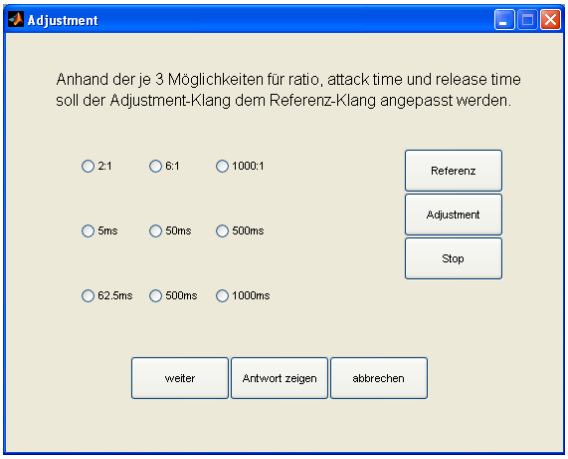

Abb. 4.4: Generiertes GUI der Übung Adjustment

4.2 Loudness

Die erste und zugleich grundlegende Übung beschäftigt sich mit der Unterscheidung von Lautstärken basierend auf der Vorlage von „The Practice and Study of Ear Training on Discrimination of Sound Attributes“ [10]. Auch in Dave Moultans „Golden Ears“ [11] ist eine ähnliche Übung enthalten, der eine große Bedeutung beigemessen wird:

„The ability to hear a signal as being louder or softer seems obvious, but given that the louder sound usually seems to sound better, it is essential to know when loudness is the only difference between two signals. That way, you neither fool yourself nor get fooled ...“ [11]

Die Übung besteht aus zwei Stimuli, die verglichen werden sollen. Dabei ist Stimulus A das Referenzsignal, zu dem lauter, leiser oder gleich laut abgespielten Signal Stimulus B. Die Änderung zwischen Stimulus A und Stimulus B beträgt von 0,5dB bis maximal 6dB, wobei dieser Wertebereich in diskrete 0,5dB-Stufen aufgeteilt wird. Ein Teilziel der Übung ist es, Erfahrungen bezüglich der Frequenz- und Lautstärkeabhängigkeit der Lautheit zu sammeln. Als Signalquelle dienen Sinustöne im Bereich von 80Hz – 6kHz, *pink noise* (Rosa Rauschen) und Musikaufnahmen.

Diese Übung ist außerdem eine gute Vorbereitung zur Erkennung von Lautheit, wie sie in „Timbre Solfege: A Course in technical Listening for Sound Engineers“ von A. Miśkiewicz [12] unter dem Punkt „3.2 Second-Year Program“ beschrieben wird.

„...At the beginning, students are presented with auditory demonstrations dealing with the fundamentals of loudness perception, that is, loudness scaling, loudness discrimination, memory for loudness, and loudness summation. ...“ [12]

Dies kann eine Anregung für weitere Arbeiten sein, wobei Miśkiewiczs Übungen integriert werden könnten, um dann in Übungen für Klangfarben bei verschiedenen Lautheiten überzugehen, und somit eine Querverbindung der *equalization*-Übungen und der *dynamic*-Übungen zu finden.

4.2.1 Technische Umsetzung von Loudness in PD

Die technische Umsetzung für die Lautstärkenänderung wird in PD mit drei *subpatches* für die drei möglichen Signalquellen Musik, *pink noise* und Sinustöne vorgenommen. Abb. 4.6 zeigt den Aufbau des *patch Lautheit1.pd*. In den *inlet* wird der Zahlenwert zur Absenkung des Wiedergabepegels durch einen Routing-Befehl im Hauptpatch mit dem Index B geschickt. An den beiden Eingängen *inlet~* liegt das Audiosignal an, das ebenfalls im Hauptpatch mit dem Index M ein- bzw. ausgeschalten wird. Die einfache Addition von 100 (Signal mit *unity gain*) und der Abschwächung von maximal -6dB bis minimal 0dB speist einen Fader (*Vslider*). Der eingefügte *bang* stellt dabei sicher, dass immer von *unity gain* als Ausgangspunkt gerechnet wird. Am Befehl *gainvu~*[9] werden die Daten- und Audiosignale zusammengeführt und an den Ausgang und somit zurück zum Hauptpatch geschickt.

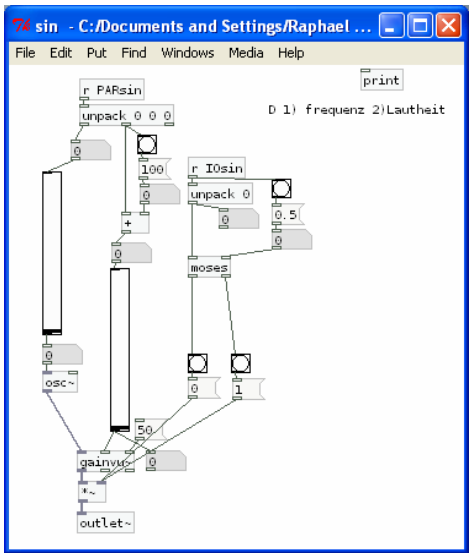

Abb. 4.5: *sin.pd-patch* zur Erzeugung von Sinustönen sowie zur Änderung des wiedergegebenen Schallpegels.

[9] gainvu~ © Thomas Musil 2000-2005, musil@iem.at, IEM KUG Graz, Austria

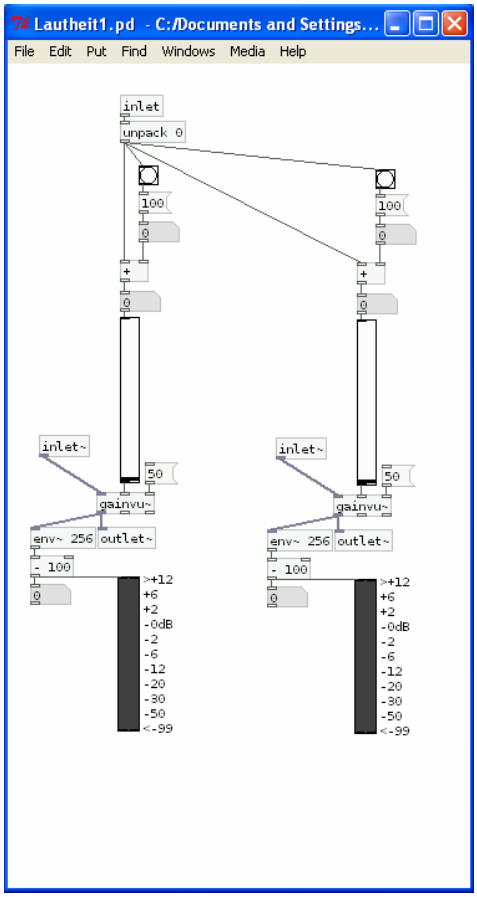

Abb. 4.6: *Lautheit1.pd-patch* zur Änderung des wiedergegebenen Schallpegels von Musikaufnahmen.

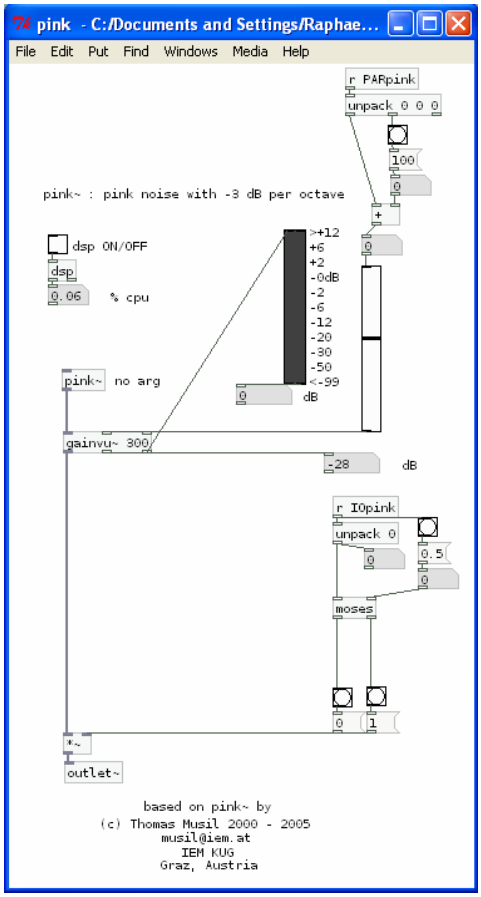

Abb. 4.7: *pink.pd-patch* zur Erzeugung von Rosa Rauschen (*pink noise*) sowie zur Änderung des wiedergegebenen Schallpegels

Abb. 4.5 und Abb. 4.7 zeigen die *patches* für *pink noise* und Sinustöne. Die Umsetzungen sind identisch zu Lautheit1.pd, bis auf das Quellsignal, das direkt im jeweiligen Fenster erzeugt wird. Dazu dienen ein einfacher Oszillator für Sinustöne von 80Hz bis 6kHz sowie pink~ zur Erzeugung von Rosa Rauschen.

Ein- bzw. ausgeschalten werden der Oszillator und das Rosa Rauschen durch das empfangene Signal *IOsin* bzw. *IOpink* durch die Funktion *moses*, die je nach Eingangssignal von kleiner oder größer 0,5 den Verstärkungsfaktor auf 0 bzw. 1 setzt.

4.2.2 Technische Umsetzung Loudness in MATLAB®

Wie oben erwähnt, ist der gesamte Übungsablauf und die Benutzeroberfläche in MATLAB® implementiert. Die Berechnung für die Abschwächung des Signals erfolgt durch zwei zufallsgenerierte Zahlen *deziBell1* und *deziBell2* die die Zahlenwerte für Stimulus A bzw. Stimulus B darstellen. Umgesetzt wurde die Rechnung durch den Random-Befehl *rand(1),* der einen Zahlenwert zwischen 0 und 1 liefert. Die Multiplikation mit 12 und das anschließende Runden auf ganze Zahlen ergibt eine Zahl aus dem Zahlenbereich von 0 bis 12. Mit der Division durch 2 erhält man eine Zahl von 0 bis 6 in diskreten 0,5 Schritten. Diese beiden Variablen werden, je nachdem welcher Stimulus abgespielt wird, in die dritte Variable *deziBell* gespeichert und durch die *function pd_aktualisieren_Lautheit()* als *string* mithilfe folgender Code-Zeile an PD geschickt:

Musik:
```
fwrite(pd,strcat(sprintf('B %f;', deziBell)));
```
Pink Noise:
```
fwrite(pd,strcat(sprintf('C %f;', deziBell)));
```
Sinustöne:
```
fwrite(pd,strcat(sprintf('D %f
%f;',frequ_sin(pointer_frequ_sin),deziBell)))
```

Die Buchstaben B, C und D geben das *routing* an, zu welchen Parametern und *patches* in PD die Daten übertragen werden. Für Sinustöne müssen zusätzlich der Vektor *frequ_sin* sowie die Zufallszahl *pointer_frequ_sin* eingeführt werden. Die Zufallszahl wird prinzipiell identisch zu *deziBell1* errechnet, nur die Zahlenwerte ergeben 1 bis 8. Durch diese Zufallszahl wird ein Element aus dem Vektor und somit die generierte Frequenz ausgewählt.

$$frequ\_sin = [80 \quad 160 \quad 250 \quad 500 \quad 1000 \quad 2000 \quad 4000 \quad 6000] \tag{15}$$

Ein weiterer *String* wird unter der *function pd_aktualisieren_Lautheit()* übertragen, mit der Aufgabe, die Stimuli abzuspielen bzw. anzuhalten. Dabei

übernimmt die Variable *soundstatus* das Starten bzw. Stoppen des Befehles *readsf~* mit dem das Soundbeispiel geladen wird, bzw. die Ausgabe des Sinustones oder des Rosa Rauschens. Die zweite Variable ist *mull* und dient wie oben erwähnt dazu Störgeräusche weitestgehend zu umgehen und die den Subpatches *Lautheit1.pd* zu steuern. Die angehängten Nullen im *String* schalten alle Signale aus anderen Patches in PD ab.

```
fwrite(pd,strcat(sprintf('M %f %f 0 0 0;', soundstatus, mull)));
```

Auch hier gibt es drei Varianten für die drei Signalquellen, die sich nur durch den Index-Buchstaben unterscheiden. (M…Musik, N…Sinustöne, O…Pink Noise)

4.3 Attributes

In Kapitel 3.3 wurde erklärt, anhand welcher Parametereinstellungen eines Kompressors man unterschiedliche Effekte bzw. Anwendungen erreichen kann. In Übung 2 ist das Ziel, die Eigenschaften eines Kompressors zu erkennen und zu betiteln. Folgende Einstellungsarten sind dabei vorgesehen: *punch, sustain* und *loudening*.

Dabei beziehen sich die beiden ersten Auswahlmöglichkeiten auf die *Bearbeitung des Dynamikumfanges – Hüllkurve* (Kapitel 3.3.2), und die dritte Möglichkeit auf *loudening* (Kapitel 3.3.1).

Inspiriert wurde diese Übung von Dave Moltons „Golden Ears“ [11]. In diesem Trainingsprogramm müssen ebenfalls kurze und lange *release times* erkannt werden. Diese Grundidee wurde insofern adaptiert, als dass es zusätzlich möglich, ist das unbearbeitete Signal als Referenz anzuhören (Kompressor im *bypass*).

4.3.1 Technische Umsetzung Attributes in PD

Ein Hauptteil der praktischen Ausführung dieser Arbeit liegt in der Implementierung eines Kompressors in PD. Die Herausforderung bestand darin, einen einfachen *hard knee*-Kompressor zu programmieren, der alle gängigen Parameter beinhaltet und einfach steuerbar ist. Als Basis diente dabei der Kompressor von J. Kreidler, der im Online-PD-Tutorial unter Kapitel 3.9.1.2 zu finden ist [14].

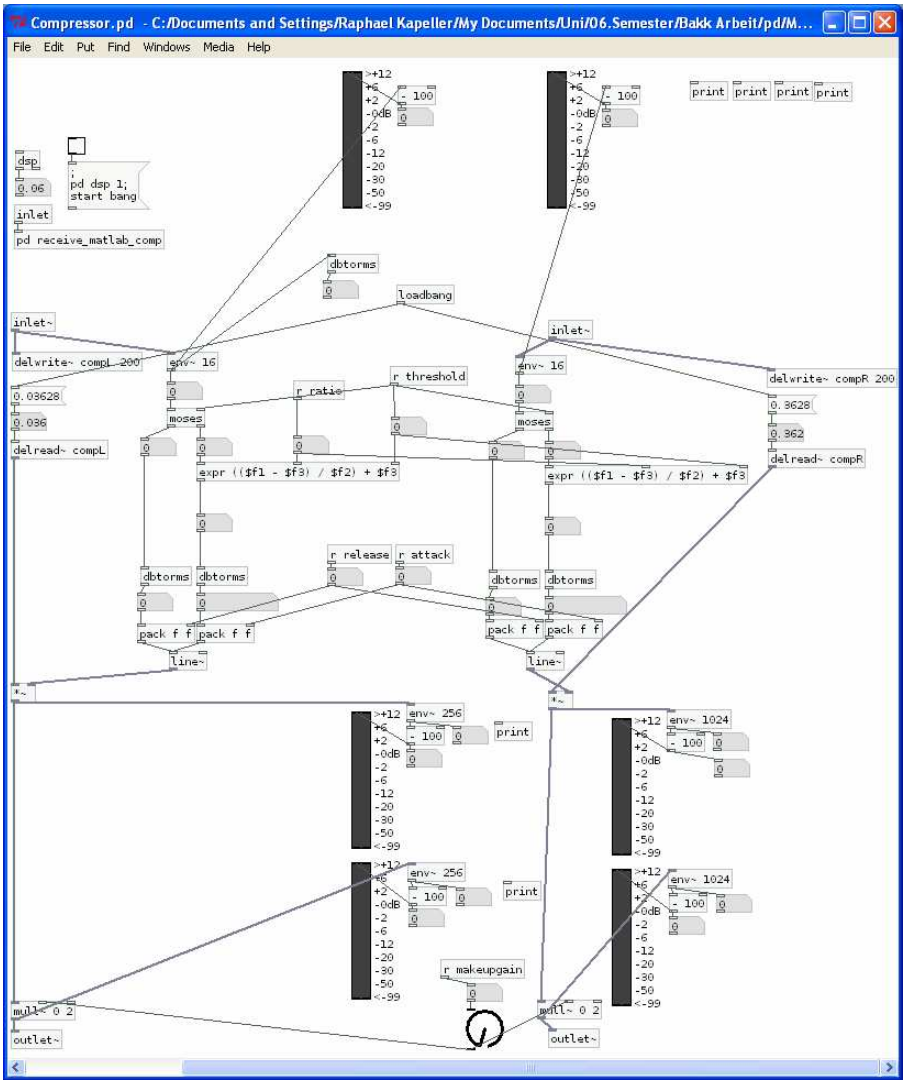

Abb. 4.8: *Compressor.pd-patch.* Stereo-Ausführung mit zwei Ein- bzw. Ausgängen Parameter werden über eine Quelle gesteuert. expr(($f1 - $f3) / $f2) + $f3 entspricht Formel (16).

Abb. 4.8 zeigt den *Compressor.pd-patch*. Der strukturelle Aufbau ist identisch mit dem Blockschaltbild in Abb. 3.4 aus Kapitel 3.2 An den beiden Eingängen *inlet~* liegt das Audiosignal in stereo an, das auf zwei Zweige aufgeteilt wird. Mit *delwrite~* wird es im ersten Zweig um 0,03628ms[10] verzögert (vergleiche Blockschaltbild – *delay*). Mithilfe eines *envelope followers* (*env~*) wird alle 16samples der RMS-Wert des Audiosignals gemessen und in dB ausgegeben. Anschließend teilt der Befehl *moses* die Datenleitung auf, wobei am linken Ausgang, dem *release*-Zweig, alle Zahlenwerte unter dem eingestellten *threshold,* und am rechten Ausgang, dem *attack* Zweig, alle Zahlenwerte über dem *threshold* ausgegeben werden. Im *release*-Zweig wird der Zahlenwert in dB zurückgerechnet und im nächsten Schritt zu einem *string,* bestehend aus dem RMS-Wert und der *release time* gepackt. Der Befehl *line~* generiert lineare Rampenfunktionen, die mit einer Zeitkonstanten in Millisekunden einen kontinuierlichen Übergang eines gemessenen RMS-Werts zum Nächsten implementieren. Diese Eigenschaft entspricht der Zeit, die ein Kompressor benötigt, um Wirkung zu zeigen, also der *attack*- bzw. *release time*. Ohne *line~* wäre die schnelle Änderung der gemessenen RMS-Werte als „Knackser" wahrnehmbar. Im *attack* Zweig wird die *gain reduction* (gr) für Pegel über dem *threshold* durch folgende Gleichung vorgenommen und danach gleich weiterbearbeitet wie der *release* Zweig.

$$gr = \left(\frac{gemessenerWert - threshold}{ratio} \right) + threshold \qquad (16)$$

Im nächsten Schritt wird das verzögerte Audiosignal mit der Gewichtsfunktion (*line~*) multipliziert und anschließend mit einer weiteren Multiplikation mit dem *make-up gain* auf Arbeitspegel gebracht.

[10] Zeit begründet sich aus 16samples/44,1kHz=0,03628ms

4.3.2 Technische Umsetzung Attributes in MATLAB®

Die Aufgaben des *Attributes.m*[11]-File sind die selben wie jene der ersten Übung *Loudness*, und unterscheiden sich nur durch die übertragenen Variablen. Der PD-Patch *Compressor.pd* erhält folgende 5 Variablen, deren Aufgaben aus den gewählten Namen eindeutig hervorgehen:

threshold, ratio, attack, release und *make-up*

Unter neuerlicher Anwendung einer zufallsgenerierten Zahl vom Wert 1, 2 oder 3, wird die Auswahl der Parametereinstellung getroffen. Dabei sind für jede *Art* (= Name der Variable für 1, 2, 3) fixe Zahlenwerte der fünf Kompressorparameter gespeichert, die darauf folgend an PD mit Hilfe der *function pd_aktualisieren()* sowie der *function pd_aktualisieren_Attributes()* übergeben werden. Für die Möglichkeit, das Signal unbearbeitet wiederzugeben, wurde der Kompressor in PD komplett umgangen und an seiner statt der in Übung *loudness* verwendete Patch *Lautheit1.pd* verwendet. Der Grund dafür besteht in der Notwendigkeit der Unabhängigkeit von der *attack*- und *release time*, sowie der Abtastung in 256samples. Ansonsten wäre eine *ratio* von 1:1 denkbar. Umgesetzt wurde dies mit den Variablen *mulllautheit* und *mullattributes*, die, je nachdem, ob die Referenz oder das bearbeitete Signal abgespielt werden, das Signal aus *Lautheit1.pd* oder aus *Compressor.pd* an den Ausgang schalten.

4.4 Adjustment

Ein oft angewendetes Testverfahren im Bereich der Psychoakustik ist die *method of adjustment,* bei der die Testperson ein Signal einem zweiten anpassen muss. Diese Methode wird in der dritten und zugleich schwierigsten Übung umgesetzt. Man hört ein Referenzsignal mit einer bereits angewendeten Dynamikbearbeitung und muss das Signal *Adjustment* so einstellen, dass es

[11] Dateiformat MATLAB® (name.m)

exakt gleich klingt wie die Referenz. Dabei können folgende Einstellungen getroffen werden:

Ratio:
2:1, 6:1, 1000:1
Attack time:
5ms, 50ms, 500ms
Release time:
62.5ms, 500ms, 1000ms

Der *threshold,* sowie das *make-up gain* werden vorgegeben und können nicht verändert werden. Die oben angeführten Werte sind übliche Parametereinstellungen, wie sie in den meisten Kompressoren vorhanden sind. Die *release time* von 62,5ms bezieht sich auf den Text „Subjective Evaluation of Dynamic Compression in Music“ [13], in welchem dieser Wert als „zu bevorzugend“ bezeichnet wird. Außerdem wird eine *attack time* von 5ms bzw. 10ms vorgeschlagen. In [13] werden jene Einstellungen von Kompressoren behandelt, um die Qualität von Audiosignalen nach der Bearbeitung zu optimieren. Dabei wurde allerdings nicht auf das so genannte „Pumpen“, das durch zu kurze *release times* entsteht, eingegangen. Daher kann der Wert von 62,5ms nicht als Optimum angenommen werden.

4.4.1 Technische Umsetzung Adjustment PD

Auch in dieser Übung wurde auf den PD-Patch *Compressor.pd* zurückgegriffen. Erklärungen dazu sind im Kapitel 4.3.1 zu finden.

4.4.2 Technische Umsetzung Adjustment MATLAB®

Die Übung *Adjustment* ist von den vorgestellten Übungen die umfangreichste Ausführung in MATLAB®. Es muss einerseits eine Referenz, die bereits in ihrer Dynamik bearbeitet ist, und andererseits ein variables und in Echtzeit

gerechnetes Soundbeispiel *Adjustment* generiert werden. Dazu ist es nötig, für jeden veränderbaren Parameter (*ratio, attack time, release time*) je einen Vektor zu generieren, dessen Elemente die Zahlenwerte der Parameter beherbergen. Das bedeutet, man erstellt einen *pool* an möglichen Werten (siehe oben), die je einen fixen Platz im Vektor besitzen und so durch einen *pointer* ausgewählt werden können. Die Datenstruktur geht dabei von zwei *pointers* aus, wobei der erste (*sollratio, sollAT, sollRT*) der vom Programm zufallsgenerierte Referenzwert ist, und der zweite (*istratio, istAT, istRT*) die vom Benutzer eingestellte Antwort. Durch das Betätigen des Referenz-Buttons bzw. des Adjustment-Buttons werden die Werte der *soll-pointer* bzw. die Werte der *ist-pointer* in die drei Variablen *ratiopointer*, *attackpointer* und *releasepointer* geschrieben. Diese neuen Variablen zeigen nun auf jenes Element im *pool,* das gefragt bzw. ausgewählt wurde. Die Antworten werden durch je drei *radiobuttons* ausgelesen und gewichtet.

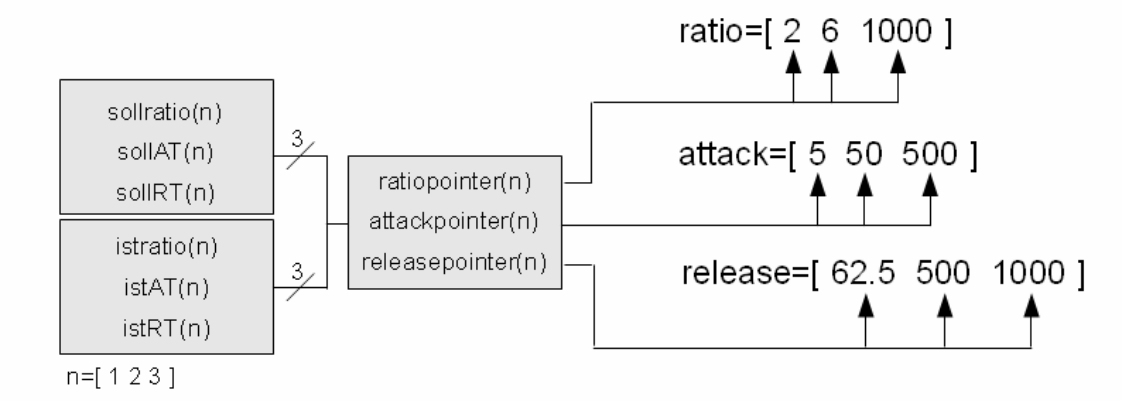

Abb. 4.9: Blockschaltbild der Variablen zur Auswahl der an PD geschickten Parameter für den Compressor.pd-Patch. Soll- bzw. ist-Block wird in die Variable (xxx)pointer geschrieben. Dieser verweist auf das Element im Pool.

Die Übertragung an PD erfolgt identisch zu den beiden anderen Übungen und soll hier der Vollständigkeit wegen noch einmal als Code angegeben werden:

Parameter:

```
fwrite(pd,strcat(sprintf('E %f %f %f %f %f;'
,threshold,ratio(ratiopointer(crun(1))),attack(attackpointer(crun(1))),r
elease(releasepointer(crun(1))),makeup)));
```

IO:

```
fwrite(pd,strcat(sprintf('M %f 0 0 %f ;', soundstatus, mull)));
```

5 Zusammenfassung und Ausblick

In den vorangegangenen Kapiteln wurde versucht, einen Einblick in den theoretischen Hintergrund der Dynamikbearbeitung, ihrer psychoakustischen Wurzeln und der theoretischen Beschreibung von Kompressoren zu geben. Anschließend wurde die Umsetzung des Übungsprogramms in groben Zügen beschrieben. Über die Software ist zusammenfassend zu sagen, dass sie einen guten Grundstein für ein Trainingsprogramm zur Sensibilisierung der auditiven Wahrnehmung in Bezug auf Dynamikbearbeitung gelegt hat. Es müsste jedoch noch viel Arbeit, Know-how und vor allem Zeit investiert werden, um eine qualitativ hochwertige Nutzung zu erreichen.

Einer der zu überarbeitenden Punkte wäre unter anderem der implementierte und zurzeit sehr rudimentär gehaltene Kompressor. Wichtig dabei wäre eine Erweiterung, die unterschiedliche *soft knee*-Versionen beinhaltet, ein Umschalten zwischen *RMS*- und *peak-measurement* ermöglicht, oder auch eine völlig andere statische Kennlinie mit Eigenschaften eines analogen Gerätes simuliert.

Die Übungen müssen ebenfalls umfangreicher gestaltet werden, so wäre ein System mit verschiedenen Levels von großem Vorteil, das einerseits zum Üben motiviert und andererseits das gelernte festigt, und Schritt für Schritt das „Können“ forciert. Ein wichtiger Schritt dabei wäre vor allem das Überarbeiten der Parametereinstellungen, sodass ein linearer Verlauf des Schwierigkeitsgrades gegeben ist. Dies ist vor allem bei Aufgabe 2 – Attributes ein wesentlicher Punkt. Hier gilt es zu klären, welche Parametereinstellungen eindeutig und schon nach kurzer Zeit erkennbar sind und andererseits, ab wann die Grenzen von einer Eigenschaft des Klangbildes zur nächsten verfließen. Außerdem ist eine Erweiterung der Begrifflichkeiten denkbar und auch zielführend, da unterschiedliche Eigenschaften, wenn einmal erkannt und evaluiert, ebenfalls der Öffentlichkeit zugänglich gemacht werden sollen, um somit neues Publikum zum Üben anzuregen. Auch die Übung „Adjustment“ kann in unterschiedlichen

Schwierigkeitsgraden ausgeführt werden. Die Unterschiede der Einstellung sind sofort abspielbar und können daher um vieles verfeinert werden, bis die Grenze des nicht mehr Wahrnehmbaren erreicht ist.

Außerdem sollte es ein Ziel sein, eine gesamte Soundsample-Library anzulegen, auf die jederzeit und vom Benutzer definiert zurückgegriffen werden kann. Problematisch dabei ist die Einstellung des *threshold* sowie des *make-up gain*, die für jedes Signal unterschiedlich sein müssen. Eine Möglichkeit, dies zu lösen wäre, wie in J. Coreys Buch „Audio Production and Critical Listening – Technical Ear Training" [15] beschrieben, beide Parameter als *fader* in das Übungs-GUI einzubauen und so dem Benutzer die Möglichkeit zur Regelung zu überlassen.

Zum Abschluss sei nochmals erwähnt, dass die Dynamikbearbeitung nur eines der drei Teilgebiete ist, die das Gesamtprogramm beinhaltet. Nur die Kombination im Trainingsablauf mit den Übungen aus Klangfarbenbeeinflussung von Martin Czuka und den Übungen aus räumlichem Hören von Josef Kulmer ist sinnvoll. Durch diese drei Teilgebiete ist ein Großteil der erforderlichen Fähigkeiten im Bereich der auditiven Wahrnehmung abgedeckt und sollte zu einer Sensibilisierung im gesamten Aufgabenfeld eines/einer Tontechniker/in, Tonmeister/in oder Toningenieures/Toningnieuerin führen.

Quellennachweis:

Kapitel 1: Einleitung

[16] W. Moylan, „Understanding and Crafting the Mix", 2. Auflage, Focal Press, USA, 2007, pp.138

Kapitel 2: Psychoakustik

[1] E. Zwicker, H. Fastl, „Psychoacoustics – Facts and Models", Springer Verlag Berlin Heidelberg, pp. 15-19, pp. 181-200, 1990.

[2] B. C. J. Moore, B. R. Glasberg, "A Revision of Zwicker's Loudness Model", ACUSTICA acta acustica, Vol.82, pp. 335-345, May 1995

[3] E. Terhart, "Akustische Kommunikation", Springer Verlag, pp. 271-283, 1998

[4] M. Dickreiter, „Handbuch der Tonstudiotechnik – Band 1", 6. Auflage, Sauer Verlag KG, München, 1997, pp. 110-114

Kapitel 3: Kompressoren

[5] R. Izhaki, „Mixing Audio", Focal Press, Great Britain, pp.270 – 334, 2008

[6] U. Zölzer, "DAFX – Digital Audio Effects", J. Wiley & Sons Ltd, Hamburg – Germany, 2008, pp. 225-239

[7] G. W. McNally, „Dynamic Range Control of Digital Audio Signals", J. Audio Eng. Soc., Vol. 32, No. 5, pp. 316 – 327, May 1984

[8] T. Sandmann, „Effekte & Dynamics", 5. Auflage, PPV-Medien GmbH, Bergkirchen, 2006, pp.25 – 36

[9] M. Dickreiter, „Handbuch der Tonstudiotechnik – Band 1", 6. Auflage, Sauer Verlag KG, München, 1997, pp. 399 – 415

Kapitel 4: Realisierte Übungen

[10] Z. Liu, F. Wu, Q.Yang, „The Practice and Study of Ear Training on Discrimination of Sound Attributes" J. Audio Eng. Soc., Convention Paper 7114, Vienna Austria, May 2007

[11] D. Moultan, A. Hodge, P. Alhadeff, „Golden Ears" KIQ Productions Inc. USA, 1995, pp.30

[12] A. Miśkiewicz „Timbre Solfege: A Course in technical Listening for Sound Engineers" ", J. Audio Eng. Soc., Vol. 40, No. 7/8, July/August 1992, pp.623 – 625

[13] W. M. Wagenaars, A. J. M. Houtsma, R.A.J.M. Van Lieshout, „Subjective Evaluation of Dynamic Compression in Music" J. Audio Eng. Soc., Vol.34, No.1/2, January/February 1986, pp.16-17

[14] J. Kreidler [Online] available: http://www.pd-tutorial.com/english/ch03s09.html, Kapitel 3.9.1.2

[15] J. Corey, "Audio Production and Critical Listening – Technical Ear Training" 1.Auflage, Focal Press, USA, 2010, pp. 100-103

Abbildungsverzeichnis:

[Abb. 2.1] T. Hermann [Online] available: http://www.techfak.uni-bielefeld.de/ags/ami/datason/l/web_psychoak/html/slide_3.html

[Abb. 2.2] M. Dickreiter, „Handbuch der Tonstudiotechnik – Band 1", 6. Auflage, Sauer Verlag KG, München, 1997, pp. 112

[Abb. 2.3] Freies Bild [Online] available: http://de.wikipedia.org/wiki/Lautheit

[Abb. 3.1] R. Izhaki, „Mixing Audio", Focal Press, Great Britain, 2008, pp. 266

[Abb. 3.2] R. Izhaki, „Mixing Audio", Focal Press, Great Britain, 2008, pp. 279

[Abb. 3.3] R. Izhaki, „Mixing Audio", Focal Press, Great Britain, 2008, pp. 289

[Abb. 3.4] U. Zölzer, "DAFX – Digital Audio Effects", J. Wiley & Sons Ltd, Hamburg – Germany, 2008, pp. 226

[Abb. 4.1] R. Kapeller, screenshot, *root~*.pd

[Abb. 4.2] R. Kapeller, screenshot, *Lautheit GUI*

[Abb. 4.3] R. Kapeller, screenshot, *Attributes GUI*

[Abb. 4.4] R. Kapeller, screenshot, *Adjustment GUI*

[Abb. 4.5] R. Kapeller, screenshot, *sin*.pd

[Abb. 4.6] R. Kapeller, screenshot, *Lautheit1*.pd

Tabellenverzeichnis: